Deep Adaptive Design: Amortizing Sequential Bayesian Experimental Design

Adam Foster [*1] **Desi R. Ivanova** [*1] **Ilyas Malik** [2] **Tom Rainforth** [1]

Abstract

We introduce *Deep Adaptive Design* (DAD), a method for amortizing the cost of adaptive Bayesian experimental design that allows experiments to be run in real-time. Traditional sequential Bayesian optimal experimental design approaches require substantial computation at *each* stage of the experiment. This makes them unsuitable for most real-world applications, where decisions must typically be made quickly. DAD addresses this restriction by learning an amortized *design network* upfront and then using this to rapidly run (multiple) adaptive experiments at deployment time. This network represents a design *policy* which takes as input the data from previous steps, and outputs the next design using a single forward pass; these design decisions can be made in milliseconds during the live experiment. To train the network, we introduce contrastive information bounds that are suitable objectives for the sequential setting, and propose a customized network architecture that exploits key symmetries. We demonstrate that DAD successfully amortizes the process of experimental design, outperforming alternative strategies on a number of problems.

1. Introduction

A key challenge across disciplines as diverse as psychology (Myung et al., 2013), bioinformatics (Vanlier et al., 2012), pharmacology (Lyu et al., 2019) and physics (Dushenko et al., 2020) is to design experiments so that the outcomes will be as informative as possible about the underlying process. Bayesian optimal experimental design (BOED) is a powerful mathematical framework for tackling this problem (Lindley, 1956; Chaloner & Verdinelli, 1995).

In the BOED framework, outcomes y are modeled in a Bayesian manner (Gelman et al., 2013; Kruschke, 2014) using a likelihood $p(y|\theta, \xi)$ and a prior $p(\theta)$, where ξ is our controllable design and θ is the set of parameters we wish to learn about. We then optimize ξ to maximize the *expected information gained* about θ (equivalently the mutual information between y and θ):

$$I(\xi) := \mathbb{E}_{p(\theta)p(y|\theta,\xi)} \left[\log p(y|\theta, \xi) - \log p(y|\xi)\right]. \quad (1)$$

The true power of BOED is realized when it is used to design a sequence of experiments $\xi_1, ..., \xi_T$, wherein it allows us to construct *adaptive* strategies which utilize information gathered from past data to tailor each successive design ξ_t during the progress of the experiment. The conventional, iterative, approach for selecting each ξ_t is to fit the posterior $p(\theta|\xi_{1:t-1}, y_{1:t-1})$ representing the updated beliefs about θ after $t-1$ iterations have been conducted, and then substitute this for the prior in (1) (Ryan et al., 2016; Rainforth, 2017; Kleinegesse et al., 2020). The design ξ_t is then chosen as the one which maximizes the resulting objective.

Unfortunately, this approach necessitates significant computational time to be expended *between each step of the experiment* in order to update the posterior and compute the next optimal design. In particular, $I(\xi)$ is doubly intractable (Rainforth et al., 2018; Zheng et al., 2018) and its optimization constitutes a significant computational bottleneck. This can be prohibitive to the practical application of sequential BOED as design decisions usually need to be made quickly for the approach to be useful (Evans & Mathur, 2005).

To give a concrete example, consider running an adaptive survey to understand political opinions (Pasek & Krosnick, 2010). A question ξ_t is put to a participant who gives their answer y_t and this data is used to update an underlying model with latent variables θ. Here sequential BOED is of immense value because previous answers can be used to guide future questions, ensuring that they are pertinent to the particular participant. However, it is not acceptable to have lengthy delays between questions to compute the next design, precluding existing approaches from being used.

To alleviate this problem, we propose *amortizing* the cost of sequential experimental design, performing upfront training before the start of the experiment to allow very fast design decisions at deployment, when time is at a premium. This amortization is particularly useful in the common scenario where the same adaptive experimental framework will be

[*] Equal contribution [1] Department of Statistics, University of Oxford, UK [2] Work undertaken whilst at the University of Oxford. Correspondence to: Adam Foster <adam.foster@stats.ox.ac.uk>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

deployed numerous times (e.g. having multiple participants in a survey). Here amortization not only removes the computational burden from the live experiment, it also allows for sharing computation across multiple experiments, analogous to inference amortization that allows one to deal with multiple datasets (Stuhlmüller et al., 2013).

Our approach, called **Deep Adaptive Design (DAD)**, constructs a single *design network* which takes as input the designs and observations from previous stages, and outputs the design to use for the next experiment. The network is learned by simulating hypothetical experimental trajectories and then using these to train the network to make near-optimal design decisions automatically. That is, it learns a *design policy* which makes decisions as a function of the past data, and we optimize the parameters of this policy rather than an individual design. Once learned, the network eliminates the computational bottleneck at each iteration of the experiment, enabling it to be run both adaptively and quickly; it can also be used repeatedly for different instantiations of the experiment (e.g. different human participants).

To allow for efficient, effective, and simple training, we show how DAD networks can be learned without any direct posterior or marginal likelihood estimation. This is achieved by reformulating the sequential BOED problem from its conventional iterative form, to a single holistic objective based on the *overall* expected information gained from the entire experiment when using a policy to make each design decision deterministically given previous design outcome pairs. We then derive contrastive bounds on this objective that allow for end-to-end training of the policy parameters with stochastic gradient ascent, thereby sidestepping both the need for inference and the double intractability of the EIG objective. This approach has the further substantial benefit of allowing non-myopic adaptive strategies to be learned, that is strategies which take account of their own future decisions, unlike conventional approaches.

We further demonstrate a key permutation symmetry property of the optimal design policy, and use this to propose a customized architecture for the experimental design network. This is critical to allowing effective amortization across time steps. The overall result of the theoretical formulation, novel contrastive bounds, and neural architecture is a methodology which enables us to bring the power of deep learning to bear on adaptive experimental design.

We apply DAD to a range of problems relevant to applications such as epidemiology, physics and psychology. We find that DAD is able to accurately amortize experiments, opening the door to running adaptive BOED in real time.

2. Background

Because experimentation is a potentially costly endeavour, it is essential to design experiments in manner that maximizes the amount of information garnered. The BOED framework, pioneered by Lindley (1956), provides a powerful means of doing this in a principled manner. Its key idea is to optimize the experimental design ξ to maximize the expected amount of *information* that will be gained about the latent variables of interest, θ, upon observing the experiment outcome y.

To implement this approach, we begin with the standard Bayesian modelling set-up consisting of an explicit likelihood model $p(y|\theta,\xi)$ for the experiment, and a prior $p(\theta)$ representing our initial beliefs about the unknown latent. After running a hypothetical experiment with design ξ and observing y, our updated beliefs are the posterior $p(\theta|\xi,y)$. The amount of information that has been gained about θ can be mathematically described by the reduction in entropy from the prior to the posterior

$$\text{IG}(\xi,y) = \text{H}\left[p(\theta)\right] - \text{H}\left[p(\theta|\xi,y)\right]. \tag{2}$$

The *expected information gain* (EIG) is formed by taking the expectation over possible outcomes y, using the model itself to simulate these. Namely we take an expectation with respect to $y \sim p(y|\xi) = \mathbb{E}_{p(\theta)}[p(y|\theta,\xi)]$, yielding

$$\begin{aligned} I(\xi) :&= \mathbb{E}_{p(y|\xi)}\left[\text{IG}(\xi,y)\right] \\ &= \mathbb{E}_{p(\theta)p(y|\theta,\xi)}\left[\log p(\theta|\xi,y) - \log p(\theta)\right] \\ &= \mathbb{E}_{p(\theta)p(y|\theta,\xi)}\left[\log p(y|\theta,\xi) - \log p(y|\xi)\right] \end{aligned}$$

which is the mutual information between y and θ under design ξ. The optimal design is defined as $\xi^* = \arg\max_{\xi\in\Xi} I(\xi)$, where Ξ is the space of feasible designs.

It is common in BOED settings to be able to run multiple experiment iterations with designs $\xi_1,...,\xi_T$, observing respective outcomes $y_1,...,y_T$. One simple strategy for this case is *static* design, also called fixed or batch design, which selects all $\xi_1,...,\xi_T$ before making any observation. The designs are optimized to maximize the EIG, with $y_{1:T}$ in place of y and $\xi_{1:T}$ in place of ξ, effectively treating the whole sequence of experiments as one experiment with enlarged observation and design spaces.

2.1. Conventional adaptive BOED

This static design approach is generally sub-optimal as it ignores the fact that information from previous iterations can substantially aid in the design decisions at future iterations. The power of the BOED framework can thus be significantly increased by using an *adaptive* design strategy that chooses each ξ_t dependent upon $\xi_{1:t-1}, y_{1:t-1}$. This enables us to use what has already been learned in previous experiments to design the next one optimally, resulting in a virtuous cycle of refining beliefs and using our updated beliefs to design good experiments for future iterations.

The conventional approach to computing designs adaptively is to fit the posterior distribution $p(\theta|\xi_{1:t-1}, y_{1:t-1})$ at each step, and then optimize the EIG objective that uses this posterior in place of the prior (Ryan et al., 2016)

$$I(\xi_t) = \mathbb{E}_{p(\theta|\xi_{1:t-1},y_{1:t-1})p(y_t|\theta,\xi_t)}\left[\log \frac{p(y_t|\theta,\xi_t)}{p(y_t|\xi_t)}\right] \quad (3)$$

where $p(y_t|\xi_t) = \mathbb{E}_{p(\theta|\xi_{1:t-1},y_{1:t-1})}[p(y_t|\theta,\xi_t)]$.

Despite the great potential of the adaptive BOED framework, this conventional approach is very computationally expensive. At each stage t of the experiment we must compute the posterior $p(\theta|\xi_{1:t-1}, y_{1:t-1})$, which is costly and cannot be done in advance as it depends on $y_{1:t-1}$. Furthermore, the posterior is then used to obtain ξ_t by maximizing the objective in (3), which is computationally even more demanding as it involves the optimization of a doubly intractable quantity (Rainforth et al., 2018; Foster et al., 2019). Both of these steps must be done during the experiment, meaning it is infeasible to run adaptive BOED in real time experiment settings unless the model is unusually simple.

2.2. Contrastive information bounds

In Foster et al. (2020), the authors noted that if $\xi \in \Xi$ is continuous, approximate optimization of the EIG at each stage of the experiment can be achieved in a single *unified* stochastic gradient procedure that both estimates and optimizes the EIG simultaneously. A key component of this approach is the derivation of several contrastive lower bounds on the EIG, inspired by work in representation learning (van den Oord et al., 2018; Poole et al., 2019). One such bound is the Prior Contrastive Estimation (PCE) bound, given by

$$I(\xi) \geq \mathbb{E}\left[\log \frac{p(y|\theta_0,\xi)}{\frac{1}{L+1}\sum_{\ell=0}^{L} p(y|\theta_\ell,\xi)}\right] \quad (4)$$

where $\theta_0 \sim p(\theta)$ is the sample used to generate $y \sim p(y|\theta,\xi)$ and $\theta_{1:L}$ are L contrastive samples drawn independently from $p(\theta)$; as $L \to \infty$ the bound becomes tight. The PCE bound can be maximized by stochastic gradient ascent (SGA) (Robbins & Monro, 1951) to approximate the optimal design ξ. As discussed previously, in a sequential setting this stochastic gradient optimization is repeated T times, with $p(\theta)$ replaced by $p(\theta|\xi_{1,t-1}, y_{1:t-1})$ at step t.

3. Rethinking Sequential BOED

To enable adaptive BOED to be deployed in settings where design decisions must be taken quickly, we first need to rethink the traditional iterative approach to produce a formulation which considers the entire design process holistically. To this end, we introduce the concept of a *design function*, or *policy*, π that maps from the set of all previous design–observation pairs to the next chosen design.

Let h_t denote the experimental *history* $(\xi_1, y_1), ..., (\xi_t, y_t)$. We can simulate histories for a given policy π, by sampling a $\theta \sim p(\theta)$, then, for each $t = 1, ..., T$, fixing $\xi_t = \pi(h_{t-1})$ (where $h_0 = \varnothing$) and sampling $y_t \sim p(y|\theta, \xi_t)$. The density of this generative process can be written as

$$p(\theta)p(h_T|\theta,\pi) = p(\theta)\prod_{t=1}^{T} p(y_t|\theta,\xi_t). \quad (5)$$

The standard sequential BOED approach described in § 2.1 now corresponds to a costly implicit policy π_s, that performs posterior estimation followed by EIG optimization to choose each design. By contrast, in DAD, we will learn a deterministic π that chooses designs directly.

Another way to think about π_s is that it is the policy which piecewise optimizes the following objective for $\xi_t|h_{t-1}$

$$I_{h_{t-1}}(\xi_t) := \mathbb{E}_{p(\theta|h_{t-1})p(y_t|\theta,\xi_t)}\left[\log \frac{p(y_t|\theta,\xi_t)}{p(y_t|h_{t-1},\xi_t)}\right] \quad (6)$$

where $p(y_t|h_{t-1},\xi_t) = \mathbb{E}_{p(\theta|h_{t-1})}[p(y_t|\theta,\xi_t)]$. It is thus the optimal *myopic* policy—that is a policy which fails to reason about its own future actions—for an objective given by the sum of EIGs from each experiment iteration. Note that this is not the optimal *overall* policy as it fails to account for future decision making: some designs may allow better future design decisions than others than others (González et al., 2016; Jiang et al., 2020).[1]

Trying to learn an efficient policy that directly mimics π_s would be very computationally challenging because of the difficulties of dealing with both inference and EIG estimation at each iteration of the training. Indeed, the natural way to do this involves running a full, very expensive, simulated sequential BOED process to generate each training example.

We instead propose a novel strategy that reformulates the sequential decision problem in a way that completely eliminates the need for calculating either posterior distributions or intermediate EIGs, while also allowing for non-myopic policies to be learned. This is done by exploiting an important property of the EIG: the total EIG of a sequential experiment is the sum of the (conditional) EIGs for each experiment iteration. This is formalized in the following result, which provides a single expression for the expected information gained from the entire sequence of T experiments.

Theorem 1. *The total expected information gain for policy π over a sequence of T experiments is*

$$\mathcal{I}_T(\pi) := \mathbb{E}_{p(\theta)p(h_T|\theta,\pi)}\left[\sum_{t=1}^{T} I_{h_{t-1}}(\xi_t)\right] \quad (7)$$

$$= \mathbb{E}_{p(\theta)p(h_T|\theta,\pi)}\left[\log p(h_T|\theta,\pi) - \log p(h_T|\pi)\right] \quad (8)$$

where $p(h_T|\pi) = \mathbb{E}_{p(\theta)}[p(h_T|\theta,\pi)]$.

[1] To give an intuitive example, consider the problem of placing two breakpoints on the line $[0, 1]$ to produce the most evenly sized segments. The optimal myopic policy places its first design at $1/2$ and its second at either $1/4$ or $3/4$. This is suboptimal since the best strategy is to place the two breakpoints at $1/3$ and $2/3$.

The proof is given in Appendix A. Intuitively, $\mathcal{I}_T(\pi)$ is the expected reduction in entropy from the prior $p(\theta)$ to the *final* posterior $p(\theta|h_T)$, without considering the intermediate posteriors at all. Note here a critical change from previous BOED formulations: $\mathcal{I}_T(\pi)$ is a function of the policy, not the designs themselves, with the latter now being random variables (due to their dependence on previous outcomes) that we take an expectation over. This is actually a strict generalization of conventional BOED frameworks: static design corresponds to policy that consists of T fixed designs with no adaptivity, for which (8) coincides with $I(\xi_{1:T})$, while conventional adaptive BOED approximates π_s.

By reformulating our objective in terms of a policy, we have constructed a single end-to-end objective for adaptive, non-myopic design and which requires negligible computation at deployment time: once π is learned, it can just be directly evaluated during the experiment itself.

4. Deep Adaptive Design

Theorem 1 showed that the optimal design function $\pi^* = \arg\max_\pi \mathcal{I}_T(\pi)$ is the one which maximizes the mutual information between the unknown latent θ and the full rollout of histories produced using that policy, h_T. DAD looks to approximate π^* explicitly using a neural network, which we now refer to as the *design network* π_ϕ, with trainable parameters ϕ. This policy-based approach marks a major break from existing methods, which do not represent design decisions explicitly as a function, but instead optimize designs on the fly during the experiment.

DAD amortizes the cost of experimental design—by training the network parameters ϕ, the design network is taught to make correct design decisions across a wide range of possible experimental outcomes. This removes the cost of adaptation for the live experiment itself: during deployment the design network will select the next design nearly instantaneously with a single forward pass of the network. Further, it offers a simplification and streamlining of the sequential BOED process: it only requires the upfront end-to-end training of a single neural network and thus negates the need to set up complex *automated* inference and optimization schemes that would otherwise have to run in the background during a live experiment. A high-level summary of the DAD approach is given in Algorithm 1.

Two key technical challenges still stand in the way of realizing the potential of adaptive BOED in real time. First, whilst the unified objective $\mathcal{I}_T(\pi)$ does not require the computation of intermediate posterior distributions, it remains an intractable objective due to the presence of $p(h_T|\pi)$. To deal with this, we derive a family of lower bounds that are appropriate for the policy-based setting and use them to construct stochastic gradient training schemes for ϕ. Second, to ensure that this network can efficiently learn a mapping from histories to designs, we require an effective architecture. As we show later, the optimal policy is invariant to the order of the history, and we use this key symmetry to architect an effective design network.

Algorithm 1 Deep Adaptive Design (DAD)

```
Input: Prior $p(\theta)$, likelihood $p(y|\theta,\xi)$, number of steps $T$
Output: Design network $\pi_\phi$
while training compute budget not exceeded do
    Sample $\theta_0 \sim p(\theta)$ and set $h_0 = \varnothing$
    for $t = 1, ..., T$ do
        Compute $\xi_t = \pi_\phi(h_{t-1})$
        Sample $y_t \sim p(y|\theta_0, \xi_t)$
        Set $h_t = \{(\xi_1, y_1), ..., (\xi_t, y_t)\}$
    end
    Compute estimate for $d\mathcal{L}_T/d\phi$ as per § 4.2
    Update $\phi$ using stochastic gradient ascent scheme
end
At deployment, $\pi_\phi$ is fixed, we take $\xi_t = \pi_\phi(h_{t-1})$, and
  each $y_t$ is obtained by running an experiment with $\xi_t$.
```

4.1. Contrastive bounds for sequential experiments

Our high-level aim is to train π_ϕ to maximize the mutual information $\mathcal{I}_T(\pi_\phi)$. In contrast to most machine learning tasks, this objective is *doubly* intractable and cannot be directly evaluated or even estimated with a conventional Monte Carlo estimator, except in very special cases (Rainforth et al., 2018). In fact, it is extremely challenging and costly to derive *any unbiased* estimate for it or its gradients. To train π_ϕ with stochastic gradient methods, we will therefore introduce and optimize *lower bounds* on $\mathcal{I}_T(\pi_\phi)$, building on the ideas of § 2.2.

Equation (8) shows that the objective function is the expected logarithm of a ratio of two terms. The first is the likelihood of the history, $p(h_T|\theta,\pi)$, and can be directly evaluated using (5). The second term is an intractable marginal $p(h_T|\pi)$ that is different for each sample of the outer expectation and must thus be estimated separately each time.

Given a sample $\theta_0, h_T \sim p(\theta, h_T|\pi)$, we can perform this estimation by introducing L independent *contrastive* samples $\theta_{1:L} \sim p(\theta)$. We can then approximate the log-ratio in two different ways, depending on whether or not we include θ_0 in our estimate for $p(h_T|\pi)$:

$$g_L(\theta_{0:L}, h_T) = \log \frac{p(h_T|\theta_0,\pi)}{\frac{1}{L+1}\sum_{\ell=0}^{L} p(h_T|\theta_\ell,\pi)} \tag{9}$$

$$f_L(\theta_{0:L}, h_T) = \log \frac{p(h_T|\theta_0,\pi)}{\frac{1}{L}\sum_{\ell=1}^{L} p(h_T|\theta_\ell,\pi)}. \tag{10}$$

These functions can both be evaluated by recomputing the likelihood of the history under each of the contrastive samples $\theta_{1:L}$. We note that g cannot exceed $\log(L+1)$, whereas f is potentially unbounded (see Appendix A for a proof).

We now show that using g to approximate the integrand leads to a *lower* bound on the overall objective $\mathcal{I}_T(\pi)$, whilst using f leads to an *upper* bound. During training, we focus on the lower bound, because it does not lead to unbounded ratio estimates and is therefore more numerically stable. We refer to this new lower bound as *sequential PCE* (sPCE).

Theorem 2 (Sequential PCE). *For a design function π and a number of contrastive samples $L \geq 0$, let*

$$\mathcal{L}_T(\pi, L) = \mathbb{E}_{p(\theta_0, h_T|\pi)p(\theta_{1:L})}\left[g_L(\theta_{0:L}, h_T)\right] \tag{11}$$

where $g_L(\theta_{0:L}, h_T)$ is as per (9), and $\theta_0, h_T \sim p(\theta, h_T|\pi)$, and $\theta_{1:L} \sim p(\theta)$ independently. Given minor technical assumptions discussed in the proof, we have[2]

$$\mathcal{L}_T(\pi, L) \uparrow \mathcal{I}_T(\pi) \text{ as } L \to \infty \tag{12}$$

at a rate $\mathcal{O}\left(L^{-1}\right)$.

The proof is presented in Appendix A. For evaluation purposes, it is helpful to pair sPCE with an upper bound, which we obtain by using f as our estimate of the integrand

$$\mathcal{U}_T(\pi, L) = \mathbb{E}_{p(\theta_0, h_T|\pi)p(\theta_{1:L})}\left[f_L(\theta_{0:L}, h_T)\right]. \tag{13}$$

We refer to this bound as sequential Nested Monte Carlo (sNMC). Theorem 4 in Appendix A shows that $\mathcal{U}_T(\pi, L)$ satisfies complementary properties to $\mathcal{L}_T(\pi, L)$. In particular, $\mathcal{L}_T(\pi, L) \leq \mathcal{I}_T(\pi) \leq \mathcal{U}_T(\pi, L)$ and both bounds become monotonically tighter as L increases, becoming exact as $L \to \infty$ at a rate $\mathcal{O}(1/L)$. We can thus directly control the trade-off between bias in our objective and the computational cost of training. Note that increasing L has no impact on the cost at deployment time. Critically, as we will see in our experiments, we tend to only need relatively modest values of L for $\mathcal{L}_T(\pi, L)$ to be an effective objective.

If using a sufficiently large L proves problematic (e.g. our available training time is strictly limited), one can further tighten these bounds for a fixed L by introducing an amortized proposal, $q(\theta; h_T)$, for the contrastive samples $\theta_{1:L}$, rather than drawing them from the prior, as in Foster et al. (2020). By appropriately adapting $\mathcal{L}_T(\pi, L)$, the proposal and the design network can then be trained simultaneously with a single unified objective, in a manner similar to a variational autoencoder (Kingma & Welling, 2014), allowing the bound itself to get tighter during training. The resulting more general class of bounds are described in detail in Appendix B and may offer further improvements for the DAD approach. We focus on training with sPCE here in the interest of simplicity of both exposition and implementation.

4.2. Gradient estimation

The design network parameters ϕ can be optimized using a stochastic optimization scheme such as Adam (Kingma & Ba, 2014). Such methods require us to compute unbiased gradient estimates of the sPCE objective (11). Throughout, we assume that the design space Ξ is continuous.

We first consider the case when the observation space $\mathcal{Y}$ is also continuous and the likelihood $p(y|\theta, \xi)$ is reparametrizable. This means that we can introduce random variables $\epsilon_{1:T} \sim p(\epsilon)$, which are independent of $\xi_{1:T}$ and $\theta_{0:L}$, such that $y_t = y(\theta_0, \xi_t, \epsilon_t)$. As we already have that $\xi_t = \pi_\phi(h_{t-1})$, we see that h_t becomes a deterministic function of h_{t-1} given ϵ_t and θ_0. Under these assumptions we can take the gradient operator inside the expectation and apply the law of the unconscious statistician to write[3]

$$\frac{d\mathcal{L}_T}{d\phi} = \mathbb{E}_{p(\theta_{0:L})p(\epsilon_{1:T})}\left[\frac{d}{d\phi} g_L(\theta_{0:L}, h_T)\right]. \tag{14}$$

We can now construct unbiased gradient estimates by sampling from $p(\theta_{0:L})p(\epsilon_{1:T})$ and evaluating, $dg_L(\theta_{0:L}, h_T)/d\phi$. This gradient can be easily computed via an automatic differentiation framework (Baydin et al., 2018; Paszke et al., 2019).

For the case of discrete observations $y \in \mathcal{Y}$, first note that given a policy π_ϕ, the only randomness in the history h_T comes from the observations $y_1, \ldots, y_T$, since the designs are computed deterministically from past histories. One approach to computing the gradient of (11) in this case is to sum over all possible histories h_T, integrating out the variables $y_{1:T}$, and take gradients with respect to ϕ to give

$$\frac{d\mathcal{L}_T}{d\phi} = \mathbb{E}\left[\sum_{h_T} \frac{d}{d\phi}\Big(p(h_T|\theta_0) g_L(\theta_{0:L}, h_T)\Big)\right], \tag{15}$$

where the expectation is over $\theta_{0:L} \sim p(\theta)$. Unbiased gradient estimates can be computed using samples from the prior. Unfortunately, this gradient estimator has a computational cost $\mathcal{O}(|\mathcal{Y}|^T)$ and is therefore only applicable when both the number of experiments T and the number of possible outcomes $|\mathcal{Y}|$ are relatively small.

To deal with the cases when it is either impractical to enumerate all possible histories, or $\mathcal{Y}$ is continuous but the likelihood $p(h_T|\theta, \pi_\phi)$ is non-reparametrizable, we propose using the score function gradient estimator, which is also known as the REINFORCE estimator (Williams, 1992). The score function gradient, is given by

$$\frac{d\mathcal{L}_T}{d\phi} = \mathbb{E}\left[\left(\log \frac{p(h_T|\theta_0, \pi_\phi)}{\sum_{\ell=0}^{L} p(h_T|\theta_\ell, \pi_\phi)}\right) \frac{d}{d\phi} \log p(h_T|\theta_0, \pi_\phi) - \frac{d}{d\phi} \log \sum_{\ell=0}^{L} p(h_T|\theta_\ell, \pi_\phi)\right] \tag{16}$$

[2] $x_L \uparrow x$ means that x_L is a monotonically increasing sequence in L with limit x.

[3] We use $\partial a/\partial b$ and da/db to represent the Jacobian matrices of partial and total derivatives respectively for vectors a and b.

where the expectation is over $\theta_0, h_T \sim p(\theta, h_T|\pi)$ and $\theta_{1:L} \sim p(\theta)$, and unbiased estimates may again be obtained using samples. This gradient is amenable to the wide range of existing variance reduction methods such as control variates (Tucker et al., 2017; Mohamed et al., 2020). In our experiments, however, we found the standard score function gradient to be sufficiently low variance. For complete derivations of the gradients estimators we use, see Appendix C.

4.3. Architecture

Finally, we discuss the deep learning architecture used for π_ϕ. To allow efficient and effective training, we take into account a key permutation invariance of the BOED problem as highlighted by the following result (proved in Appendix A).

Theorem 3 (Permutation invariance)**.** *Consider a permutation $\sigma \in S_k$ acting on a history h_k^1, yielding $h_k^2 = (\xi_{\sigma(1)}, y_{\sigma(1)}), ..., (\xi_{\sigma(k)}, y_{\sigma(k)})$. For all such σ, we have*

$$\mathbb{E}\left[\sum_{t=1}^{T} I_{h_{t-1}}(\xi_t) \middle| h_k = h_k^1\right] = \mathbb{E}\left[\sum_{t=1}^{T} I_{h_{t-1}}(\xi_t) \middle| h_k = h_k^2\right]$$

such that the EIG is unchanged under permutation. Further, the optimal policies starting in h_k^1 and h_k^2 are the same.

This permutation invariance is an important and well-studied property of many machine learning problems (Bloem-Reddy & Teh, 2019). The knowledge that a system exhibits permutation invariance can be exploited in neural architecture design to enable significant *weight sharing*. One common approach is pooling (Edwards & Storkey, 2016; Zaheer et al., 2017; Garnelo et al., 2018a;b). This involves summing or otherwise combining representations of multiple inputs into a single representation that is invariant to their order.

Using this idea, we represent the history h_t with a fixed dimensional representation that is formed by pooling representations of the distinct design-outcome pairs of the history

$$R(h_t) := \sum\nolimits_{k=1}^{t} E_{\phi_1}(\xi_k, y_k), \tag{17}$$

where E_{ϕ_1} is a neural network *encoder* with parameters ϕ_1 to be learned. Note that this pooled representation is the same if we reorder the labels $1, ..., t$. By convention, the sum of an empty sequence is 0.

We then construct our design network to make decisions based on the pooled representation $R(h_t)$ by setting $\pi_\phi(h_t) = F_{\phi_2}(R(h_t))$, where F_{ϕ_2} is a learned *emitter* network. The trainable parameters are $\phi = \{\phi_1, \phi_2\}$. By combining simple networks in a way that is sensitive to the permutation invariance of the problem, we facilitate parameter sharing in which the network E_{ϕ_1} is re-used for each input pair and for each time step t. This results in significantly improved performance compared to networks that are forced to *learn* the relevant symmetries of the problem.

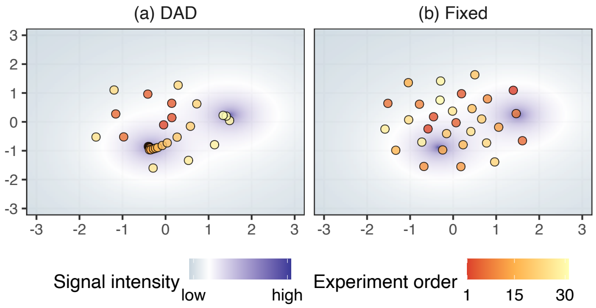

Figure 1. An example of the designs learnt by (a) the DAD network and (b) the fixed baseline for a given θ sampled from the prior.

5. Related Work

Existing approaches to sequential BOED typically follow the path outlined in § 2.1. The posterior inference performed at each stage of the conventional approach has been done using sequential Monte Carlo (Del Moral et al., 2006; Drovandi et al., 2014), population Monte Carlo (Rainforth, 2017), variational inference (Foster et al., 2019; 2020), and Laplace approximation (Lewi et al., 2009; Long et al., 2013).

The estimation of the mutual information objective at each step has been performed by nested Monte Carlo (Myung et al., 2013; Vincent & Rainforth, 2017), variational bounds (Foster et al., 2019; 2020), Laplace approximation (Lewi et al., 2009), ratio estimation (Kleinegesse et al., 2020), and hybrid methods (Senarathne et al., 2020). The optimization over designs has been performed by Bayesian optimization (Foster et al., 2019; Kleinegesse et al., 2020), interacting particle systems (Amzal et al., 2006), simulated annealing (Müller, 2005), utilizing regret bounds (Zheng et al., 2020), or bandit methods (Rainforth, 2017).

There are approaches that simultaneously estimate the mutual information and optimize it, using a single stochastic gradient procedure. Examples include perturbation analysis (Huan & Marzouk, 2014), variational lower bounds (Foster et al., 2020), or multi-level Monte Carlo (Goda et al., 2020).

Some recent work has focused specifically on models with intractable likelihoods (Hainy et al., 2016; Kleinegesse & Gutmann, 2020; Kleinegesse et al., 2020). Other work has sought to learn a non-myopic strategy focusing on specific tractable cases (Huan & Marzouk, 2016; Jiang et al., 2020).

6. Experiments

We now compare **DAD** to a number of baselines across a range of experimental design problems. We implement DAD by extending PyTorch (Paszke et al., 2019) and Pyro (Bingham et al., 2018) to provide an implementation that is abstracted from the specific problem. Code is publicly available at `https://github.com/ae-foster/dad`.

Method	Lower bound, $\mathcal{L}_{30}$	Upper bound, $\mathcal{U}_{30}$
Random	8.303 ± 0.043	8.322 ± 0.045
Fixed	8.838 ± 0.039	8.914 ± 0.038
DAD	$\mathbf{10.926 \pm 0.036}$	$\mathbf{12.382 \pm 0.095}$
Variational	8.776 ± 0.143	9.064 ± 0.187

Table 1. Upper and lower bounds on the total EIG, $\mathcal{I}_{30}(\pi)$, for the location finding experiment. Errors indicate ±1 s.e. estimated over 256 (variational) or 2048 (others) rollouts.

As our aim is to adapt designs in *real-time*, we primarily compare to strategies that are fast at deployment time. The simplest baseline is **random** design, which selects designs uniformly at random. The **fixed** baseline completely ignores the opportunity for adaptation and uses static design to learn a fixed $\xi_1, ..., \xi_T$ before the experiment. We use the SG-BOED approach of Foster et al. (2020) with the PCE bound to optimize the fixed design $\xi_{1:T}$. We also compare to tailor-made heuristics for particular models as appropriate.

Similarly to the notion of the amortization gap in amortized inference (Cremer et al., 2018), one might initially expect to a drop in performance of DAD compared to conventional (non-amortized) BOED methods that use the traditional iterative approach of § 2.1. To assess this we also consider using the SG-BOED approach of Foster et al. (2020) in a traditional iterative manner to approximate π_s, referring to this as the **variational** baseline, noting this requires significant runtime computation. We also look at several iterative BOED baselines that are specifically tailored to the examples that we choose (Vincent & Rainforth, 2017; Kleinegesse et al., 2020). Perhaps surprisingly, we find that DAD is not only competitive compared to these non-amortized methods, but often outperforms them. This is discussed in § 7.

The first performance metric that we focus on is total EIG, $\mathcal{I}_T(\pi)$. When no direct estimate of $\mathcal{I}_T(\pi)$ is available, we estimate both the sPCE lower bound and sNMC upper bound. We also present the standard error to indicate how the performance varies between different experiment realizations (rollouts). We further consider the deployment time (i.e. the time to run the experiment itself, after pre-training); a critical metric for our aims. Full experiment details are given in Appendix D.

6.1. Location finding in 2D

Inspired by the acoustic energy attenuation model of Sheng & Hu (2005), we consider the problem of finding the locations of multiple hidden sources which each emits a signal whose intensity attenuates according to the inverse-square law. The *total intensity* is a superposition of these signals. The design problem is to choose where to make observations of the total signal to learn the locations of the sources.

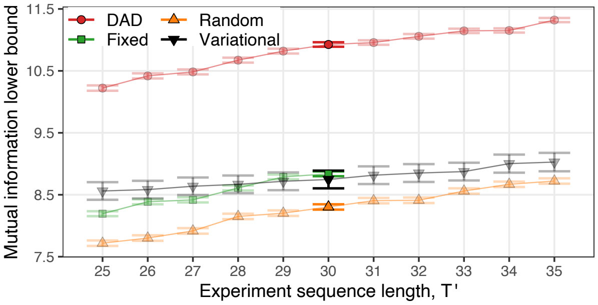

Figure 2. Generalizing sequence length for the location finding experiment. The DAD network and the fixed strategy were trained to perform $T = 30$ experiments, whilst other strategies do not require pre-training. The fixed strategy cannot be generalized to sequences longer than its training regime. We present sPCE estimates with error bars computed as in Table 1.

We train a DAD network to perform $T = 30$ experiments with $K = 2$ sources. The designs learned by DAD are visualized in Figure 1(a). Here our network learns a complex strategy that initially explores in a spiral pattern. Once it detects a strong signal, multiple experiments are performed close together to refine knowledge of that location (note the high density of evaluations near the sources). The fixed design strategy, displayed in Figure 1(b) must choose all design locations up front, leading to an evenly dispersed strategy that cannot "hone in" on the critical areas, thus gathering less information.

Table 1 reports upper and lower bounds on $\mathcal{I}_T(\pi)$ for each strategy and confirms that DAD significantly outperforms all the considered baselines. DAD is also orders of magnitude faster to deploy than the variational baseline, the other adaptive method, with DAD taking 0.0474 ± 0.0003 secs to make all 30 design decisions on a lightweight CPU, compared to 8963 secs for the variational method.

Varying the Design Horizon In practical situations the exact number of experiments to perform may be unknown. Figure 2 indicates that our DAD network that is pretrained to perform $T = 30$ experiments can generalize well to perform $T' \neq 30$ experiments at deployment time, still outperforming the baselines, indicating that DAD is robust to the length of training sequences.

Training Stability To assess the stability between different training runs, we trained 16 different DAD networks. Computing the mean and standard error of the lower bound on $\mathcal{I}_T(\pi)$ over these 16 runs gave 10.91 ± 0.014, and the matching upper bounds were 12.47 ± 0.046. We see that the variance across different training seeds is modest, indicating that DAD reaches designs of a similar quality each time. Comparing with Table 1, we see that the natural variability across rollouts (i.e. different θ) with a single DAD network tends to be larger than the variance between the average performance of different DAD networks.

Method	Deployment time (s)
Frye et al. (2016)	0.0902 ± 0.0003
Kirby (2009)	N/A
Fixed	N/A
DAD	0.0901 ± 0.0007
Badapted	25.2679 ± 0.1854

Table 2. Deployment times for Hyperbolic Temporal Discounting methods. We present the total design time for $T = 20$ questions, taking the mean and ±1 s.e. over 10 realizations. Tests were conducted on a lightweight CPU (see Appendix D).

Method	Lower bound	Upper bound
Frye et al. (2016)	3.500 ± 0.029	3.513 ± 0.029
Kirby (2009)	1.861 ± 0.008	1.864 ± 0.009
Fixed	2.518 ± 0.007	2.524 ± 0.007
DAD	**5.021 ± 0.013**	**5.123 ± 0.015**
Badapted	4.454 ± 0.016	4.536 ± 0.018

Table 3. Final lower and upper bounds on the total information $\mathcal{I}_T(\pi)$ for the Hyperbolic Temporal Discounting experiment. The bounds are finite sample estimates of $\mathcal{L}_T(\pi, L)$ and $\mathcal{U}_T(\pi, L)$ with $L = 5000$. The errors indicate ±1 s.e. over the sampled histories.

6.2. Hyperbolic temporal discounting

In psychology, temporal discounting is the phenomenon that the utility people attribute to a reward typically decreases the longer they have to wait to receive it (Critchfield & Kollins, 2001; Green & Myerson, 2004). For example, a participant might be willing to trade £90 today for £100 in a month's time, but not for £100 in a year. A common parametric model for temporal discounting in humans is the hyperbolic model (Mazur, 1987); we study a specific form of this model proposed by Vincent (2016).

We design a sequence of $T = 20$ experiments, each taking the form of a binary question "Would you prefer £R today, or £100 in D days?" with design $\xi = (R, D)$ that must be chosen at each stage. As real applications of this model would involve human participants, the available time to choose designs is strictly limited. We consider DAD, the aforementioned fixed design policy, and strategies that have been used specifically for experiments of this kind: Kirby (2009), a human constructed fixed set of designs; Frye et al. (2016), a problem-specific adaptive strategy; and Vincent & Rainforth (2017), a partially customized sequential BOED method, called Badapted, that uses population Monte Carlo (Cappé et al., 2004) to approximate the posterior distribution at each step and a bandit approach to optimize the EIG over possible designs.

We begin by investigating the time required to deploy each of these methods. As shown in Table 2, the non-amortized Badapted method takes the longest time, while for DAD, the total deployment time is less than 0.1 seconds—totally imperceptible to a participant.

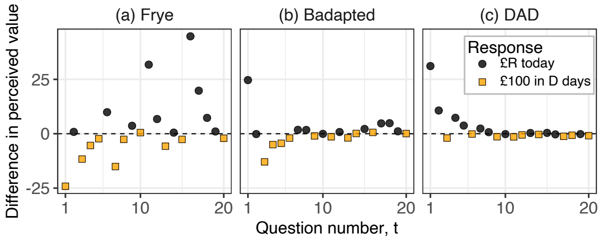

Figure 3. An example of the designs learnt by two of the problem-specific baselines and DAD. We plot the difference in perceived value of the two propositions "£R today" and "£100 in D days" for a certain participant, represented by a specific value of the latent variable θ. A difference of 0 indicates that the participant is indifferent between the two offers.

Table 3 shows the performance of each method. We see that DAD performs best, surpassing bespoke design methods that have been proposed for this problem, including Badapted which has a considerably larger computation budget. Figure 3 demonstrates how the designs learnt by DAD compare qualitatively with the two most competitive problem-specific baselines. As with Badapted, DAD designs rapidly cluster near the indifference point.

This experiment demonstrates that DAD can successfully amortize the process of experimental design in a real application setting. It outperforms some of the most successful non-amortized and highly problem-specific approaches with a fraction of the cost during the real experiment.

6.3. Death process

We conclude with an example from epidemiology (Cook et al., 2008) in which healthy individuals become infected at rate θ. The design problem is to choose observations times $\xi > 0$ at which to observe the number of infected individuals: we select $T = 4$ designs sequentially with an independent stochastic process observed at each iteration. We compare to our fixed and variational baselines, along with the adaptive SeqBED approach of Kleinegesse et al. (2020).

First, we examine the compute time required to deploy each method for a single run of the sequential experiment. The times illustrated in Table 4 show that the adaptive strategy learned by DAD can be deployed in under 0.01 seconds, many orders of magnitude faster than the non-amortized methods, with SeqBED taking hours for *one* rollout.

Next, we estimate the objective $\mathcal{I}_T(\pi)$ by averaging the information gain over simulated rollouts. The results in Table 4 reveal that DAD designs are superior to both fixed design and variational adaptive design, tending to uncover more information about the latent θ across many possible experimental trajectories. For comparison with SeqBED, we were unable to perform sufficient rollouts to obtain a high

Method	Deployment time (s)	$\mathcal{I}_T(\pi)$
Fixed	N/A	2.023 ± 0.007
DAD	$0.0051 \pm 12\%$	$\mathbf{2.113 \pm 0.008}$
Variational	$1935.0 \quad \pm \quad 2\%$	2.076 ± 0.034
SeqBED*	25911.0	1.590

Table 4. Total EIG $\mathcal{I}_T(\pi)$ and deployment times for the Death Process. We present the EIG ±1 s.e. over 10,000 rollouts (fixed and DAD), 500 rollouts (variational) or *1 rollout (SeqBED). The IG can be efficiently evaluated in this case (see Appendix D). Runtimes computed as per Table 2.

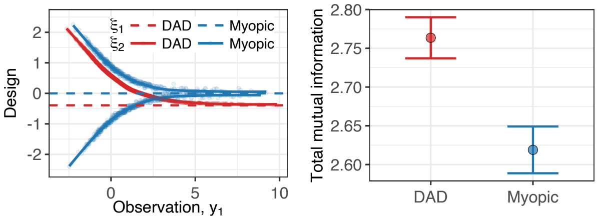

Figure 4. 1D location finding with 1 source, $T = 2$. [Left] the design function, dashed lines correspond to the first design ξ_1, which is independent of y_1. [Right] $\mathcal{I}_2(\pi)$, the total EIG ±1 s.e.

quality estimate of $\mathcal{I}_T(\pi)$. Instead, we conducted a single rollout of each method with $\theta = 1.5$ fixed. The resulting information gains for this one rollout were: 1.590 (SeqBED), 1.719 (Variational), 1.678 (Fixed), **1.779 (DAD)**.

7. Discussion

In this paper we introduced DAD—a new method utilizing the power of deep learning to amortize the cost of sequential BOED and allow adaptive experiments to be run in real time. In all experiments DAD performed significantly better than baselines with a comparable deployment time. Further, DAD showed competitive performance against conventional BOED approaches that do not use amortization, but make costly computations at each stage of the experiment.

Surprisingly, we found DAD was often able to outperform these non-amortized approaches despite using a tiny fraction of the resources at deployment time. We suggest two reasons for this. Firstly, conventional methods must approximate the posterior $p(\theta|h_t)$ at each stage. If this approximation is poor, the resulting design optimization will yield poor results regardless of the EIG optimization approach chosen. Careful tuning of the posterior approximation could alleviate this, but would increase computational time further and it is difficult to do this in the required automated manner. DAD sidesteps this problem altogether by eliminating the need for directly approximating a posterior distribution.

Secondly, the policy learnt by DAD has the potential to be *non-myopic*: it does not choose a design that is optimal for the current experiment in isolation, but takes into account the fact that there are more experiments to be performed in the future. We can see this in practice in a simple experiment using the location finding example with one source in 1D with prior $\theta \sim N(0, 1)$ and with $T = 2$ steps. This setting is simple enough to compute the *exact* one-step optimal design via numerical integration. Figure 4 [Left] shows the design function learnt by DAD alongside the exact optimal myopic design. The optimal myopic strategy for $t = 1$ is to sample at the prior mean $\xi_1 = 0$. At time $t = 2$ the myopic strategy selects a positive or negative design with equal probability. In contrast, the policy learnt by DAD is to sample at $\xi_1 \approx -0.4$, which does not optimize the EIG for $T = 1$ in isolation, but leads to a better *overall* design strategy that focuses on searching the positive regime $\xi_2 > \xi_1$ in the second experiment. Figure 4 [Right] confirms that the policy learned by DAD achieves higher total EIG from the two step experiment than the *exact* myopic approach.

Limitations and Future Work The present form of DAD still possesses some restrictions that future work might look to address. Firstly, it requires the likelihood model to be *explicit*, i.e. that we can evaluate the density $p(y_t|\theta, \xi_t)$. Secondly, it requires the experiments to be conditionally independent given θ, i.e. $p(y_{1:T}|\theta, \xi_{1:T}) = \prod_{t=1}^{T} p(y_t|\theta, \xi_t)$, which may not be the case for, e.g. time series models. Thirdly, it requires the designs themselves, ξ_t, to be continuous to allow for gradient-based optimization. On another note, DAD's use of a policy to make design decisions establishes a critical link between experimental design and model-based reinforcement learning (Sekar et al., 2020). Though DAD is distinct in several important ways (e.g. the lack of observed rewards), investigating these links further might provide an interesting avenue for future work.

Conclusions DAD represents a new conception of adaptive experimentation that focuses on learning a design *policy* network offline, then deploying it during the live experiment to quickly make adaptive design decisions. This marks a departure from the well-worn path of myopic adaptive BOED (Sec. 2), eliminating the need to estimate intermediate posterior distributions or optimize over designs during the live experiment itself; it represents the first approach to allow adaptive BOED to be run in real-time for general problems. As such, we believe it may be beneficial to practitioners in a number of fields, from online surveys to clinical trials.

Acknowledgements

AF gratefully acknowledges funding from EPSRC grant no. EP/N509711/1. DRI is supported by EPSRC through the Modern Statistics and Statistical Machine Learning (StatML) CDT programme, grant no. EP/S023151/1. AF would like to thank Martin Jankowiak and Adam Golinski for helpful discussions about amortizing BOED.

References

Amzal, B., Bois, F. Y., Parent, E., and Robert, C. P. Bayesian-optimal design via interacting particle systems. *Journal of the American Statistical association*, 101(474):773–785, 2006.

Angelova, J. A. On moments of sample mean and variance. *Int. J. Pure Appl. Math*, 79(1):67–85, 2012.

Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M. Automatic differentiation in machine learning: a survey. *Journal of machine learning research*, 18, 2018.

Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P., Horsfall, P., and Goodman, N. D. Pyro: Deep universal probabilistic programming. *Journal of Machine Learning Research*, 2018.

Bloem-Reddy, B. and Teh, Y. W. Probabilistic symmetry and invariant neural networks. *arXiv preprint arXiv:1901.06082*, 2019.

Cappé, O., Guillin, A., Marin, J.-M., and Robert, C. P. Population monte carlo. *Journal of Computational and Graphical Statistics*, 13(4):907–929, 2004.

Chaloner, K. and Verdinelli, I. Bayesian experimental design: A review. *Statistical Science*, pp. 273–304, 1995.

Cook, A. R., Gibson, G. J., and Gilligan, C. A. Optimal observation times in experimental epidemic processes. *Biometrics*, 64(3):860–868, 2008.

Cremer, C., Li, X., and Duvenaud, D. Inference suboptimality in variational autoencoders. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1078–1086. PMLR, 2018.

Critchfield, T. S. and Kollins, S. H. Temporal discounting: Basic research and the analysis of socially important behavior. *Journal of applied behavior analysis*, 34(1): 101–122, 2001.

Del Moral, P., Doucet, A., and Jasra, A. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

Drovandi, C. C., McGree, J. M., and Pettitt, A. N. A sequential monte carlo algorithm to incorporate model uncertainty in bayesian sequential design. *Journal of Computational and Graphical Statistics*, 23(1):3–24, 2014.

Dushenko, S., Ambal, K., and McMichael, R. D. Sequential bayesian experiment design for optically detected magnetic resonance of nitrogen-vacancy centers. *Physical Review Applied*, 14(5):054036, 2020.

Edwards, H. and Storkey, A. Towards a neural statistician. *arXiv preprint arXiv:1606.02185*, 2016.

Evans, J. R. and Mathur, A. The value of online surveys. *Internet research*, 2005.

Foster, A., Jankowiak, M., Bingham, E., Horsfall, P., Teh, Y. W., Rainforth, T., and Goodman, N. Variational Bayesian Optimal Experimental Design. In *Advances in Neural Information Processing Systems 32*, pp. 14036–14047. Curran Associates, Inc., 2019.

Foster, A., Jankowiak, M., O'Meara, M., Teh, Y. W., and Rainforth, T. A unified stochastic gradient approach to designing bayesian-optimal experiments. volume 108 of *Proceedings of Machine Learning Research*, pp. 2959–2969, Online, 26–28 Aug 2020. PMLR.

Frye, C. C., Galizio, A., Friedel, J. E., DeHart, W. B., and Odum, A. L. Measuring delay discounting in humans using an adjusting amount task. *JoVE (Journal of Visualized Experiments)*, (107):e53584, 2016.

Garnelo, M., Rosenbaum, D., Maddison, C. J., Ramalho, T., Saxton, D., Shanahan, M., Teh, Y. W., Rezende, D. J., and Eslami, S. Conditional neural processes. *arXiv preprint arXiv:1807.01613*, 2018a.

Garnelo, M., Schwarz, J., Rosenbaum, D., Viola, F., Rezende, D. J., Eslami, S., and Teh, Y. W. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018b.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. *Bayesian data analysis*. Chapman and Hall/CRC, 2013.

Goda, T., Hironaka, T., and Kitade, W. Unbiased mlmc stochastic gradient-based optimization of bayesian experimental designs. *arXiv preprint arXiv:2005.08414*, 2020.

González, J., Osborne, M., and Lawrence, N. Glasses: Relieving the myopia of bayesian optimisation. In *Artificial Intelligence and Statistics*, pp. 790–799. PMLR, 2016.

Green, L. and Myerson, J. A discounting framework for choice with delayed and probabilistic rewards. *Psychological bulletin*, 130(5):769, 2004.

Hainy, M., Drovandi, C. C., and McGree, J. M. Likelihood-free extensions for bayesian sequentially designed experiments. In Kunert, J., Müller, C. H., and Atkinson, A. C. (eds.), *mODa 11 - Advances in Model-Oriented Design and Analysis*, pp. 153–161. Springer International Publishing, 2016.

Huan, X. and Marzouk, Y. Gradient-based stochastic optimization methods in bayesian experimental design. *International Journal for Uncertainty Quantification*, 4(6), 2014.

Huan, X. and Marzouk, Y. M. Sequential bayesian optimal experimental design via approximate dynamic programming. *arXiv preprint arXiv:1604.08320*, 2016.

Jiang, S., Chai, H., Gonzalez, J., and Garnett, R. Binoculars for efficient, nonmyopic sequential experimental design. In *International Conference on Machine Learning*, pp. 4794–4803. PMLR, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *ICLR*, 2014.

Kirby, K. N. One-year temporal stability of delay-discount rates. *Psychonomic bulletin & review*, 16(3):457–462, 2009.

Kleinegesse, S. and Gutmann, M. Bayesian experimental design for implicit models by mutual information neural estimation. In *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research, pp. 5316–5326. PMLR, 2020. URL http://proceedings.mlr.press/v119/kleinegesse20a.html.

Kleinegesse, S., Drovandi, C., and Gutmann, M. U. Sequential bayesian experimental design for implicit models via mutual information. *arXiv preprint arXiv:2003.09379*, 2020.

Kruschke, J. Doing bayesian data analysis: A tutorial with r, jags, and stan. 2014.

Lewi, J., Butera, R., and Paninski, L. Sequential optimal design of neurophysiology experiments. *Neural Computation*, 21(3):619–687, 2009.

Lindley, D. V. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, pp. 986–1005, 1956.

Long, Q., Scavino, M., Tempone, R., and Wang, S. Fast estimation of expected information gains for Bayesian experimental designs based on Laplace approximations. *Computer Methods in Applied Mechanics and Engineering*, 259:24–39, 2013.

Lyu, J., Wang, S., Balius, T. E., Singh, I., Levit, A., Moroz, Y. S., OMeara, M. J., Che, T., Algaa, E., Tolmachova, K., et al. Ultra-large library docking for discovering new chemotypes. *Nature*, 566(7743):224, 2019.

Mazur, J. E. An adjusting procedure for studying delayed reinforcement. *Commons, ML.; Mazur, JE.; Nevin, JA*, pp. 55–73, 1987.

Mohamed, S., Rosca, M., Figurnov, M., and Mnih, A. Monte carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 21(132):1–62, 2020.

Müller, P. Simulation based optimal design. *Handbook of Statistics*, 25:509–518, 2005.

Myung, J. I., Cavagnaro, D. R., and Pitt, M. A. A tutorial on adaptive design optimization. *Journal of mathematical psychology*, 57(3-4):53–67, 2013.

Nowozin, S. Debiasing evidence approximations: On importance-weighted autoencoders and jackknife variational inference. In *International Conference on Learning Representations*, 2018.

Pasek, J. and Krosnick, J. A. Optimizing survey questionnaire design in political science. In *The Oxford handbook of American elections and political behavior*. 2010.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Poole, B., Ozair, S., van den Oord, A., Alemi, A., and Tucker, G. On variational bounds of mutual information. In *International Conference on Machine Learning*, pp. 5171–5180, 2019.

Rainforth, T. *Automating Inference, Learning, and Design using Probabilistic Programming*. PhD thesis, University of Oxford, 2017.

Rainforth, T., Cornish, R., Yang, H., Warrington, A., and Wood, F. On nesting monte carlo estimators. In *International Conference on Machine Learning*, pp. 4267–4276. PMLR, 2018.

Robbins, H. and Monro, S. A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407, 1951.

Ryan, E. G., Drovandi, C. C., McGree, J. M., and Pettitt, A. N. A review of modern computational algorithms for bayesian optimal design. *International Statistical Review*, 84(1):128–154, 2016.

Sekar, R., Rybkin, O., Daniilidis, K., Abbeel, P., Hafner, D., and Pathak, D. Planning to explore via self-supervised world models. In *International Conference on Machine Learning*, pp. 8583–8592. PMLR, 2020.

Senarathne, S., Drovandi, C., and McGree, J. A laplace-based algorithm for bayesian adaptive design. *Statistics and Computing*, 30(5):1183–1208, 2020.

Sheng, X. and Hu, Y. H. Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks. *IEEE Transactions on Signal Processing*, 2005. ISSN 1053587X. doi: 10.1109/TSP.2004.838930.

Stuhlmüller, A., Taylor, J., and Goodman, N. Learning stochastic inverses. In *Advances in neural information processing systems*, pp. 3048–3056, 2013.

Tucker, G., Mnih, A., Maddison, C. J., Lawson, D., and Sohl-Dickstein, J. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. *arXiv preprint arXiv:1703.07370*, 2017.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Vanlier, J., Tiemann, C. A., Hilbers, P. A., and van Riel, N. A. A Bayesian approach to targeted experiment design. *Bioinformatics*, 28(8):1136–1142, 2012.

Vincent, B. T. Hierarchical bayesian estimation and hypothesis testing for delay discounting tasks. *Behavior research methods*, 48(4):1608–1620, 2016.

Vincent, B. T. and Rainforth, T. The DARC toolbox: automated, flexible, and efficient delayed and risky choice experiments using bayesian adaptive design. 2017.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 1992. ISSN 0885-6125. doi: 10.1007/bf00992696.

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R., and Smola, A. Deep sets. *arXiv preprint arXiv:1703.06114*, 2017.

Zheng, S., Pacheco, J., and Fisher, J. A robust approach to sequential information theoretic planning. In *International Conference on Machine Learning*, pp. 5941–5949, 2018.

Zheng, S., Hayden, D., Pacheco, J., and Fisher III, J. W. Sequential bayesian experimental design with variable cost structure. *Advances in Neural Information Processing Systems*, 33, 2020.