Implicit Deep Adaptive Design: Policy-Based Experimental Design without Likelihoods

Anonymous Author(s)
Affiliation
Address
email

Abstract

We introduce implicit Deep Adaptive Design (iDAD), a new method for performing adaptive experiments for *implicit* models in *real-time*, based on the Bayesian optimal experimental design (BOED) framework. iDAD amortizes the cost of experimental design by learning a design *policy network* upfront, which can then be deployed quickly at the time of the experiment. The iDAD network can be trained on any model which simulates differentiable samples, unlike previous design policy work that requires a closed form likelihood and conditionally independent experiments. At deployment-time, iDAD allows design decisions to be made in milliseconds, in contrast to traditional approaches to adaptive BOED that require heavy computation during the experiment itself. We illustrate the applicability of iDAD on a number of experiments, and show that it provides a fast and effective mechanism for performing adaptive design with implicit models.

1 Introduction

Designing experiments to maximize the information gathered about an underlying process is a key challenge in science and engineering. Most such experiments are naturally *adaptive*—we can design later iterations on the basis of data already collected, refining our understanding of the process with each step [29, 38, 44]. For example, suppose that a chemical contaminant has accidentally been released and is rapidly spreading; we need to quickly discover its unknown source. To this end, we measure the contaminant concentration level at locations $\xi_1, \ldots, \xi_T$ (our experimental designs), obtaining observations $y_1, \ldots, y_T$. Provided we can perform the necessary computations sufficiently quickly, we can design each ξ_t using data from steps $1, \ldots, t-1$ to narrow in on the source.

Bayesian optimal experimental design (BOED) [5, 26] is a principled model-based framework for choosing designs optimally; it has been successfully adopted in a diverse range of scientific fields [45, 50, 52]. In BOED, the unknown quantity of interest (e.g. contaminant location) is encapsulated by a parameter θ, and our initial information about it by a prior $p(\theta)$. A simulator, or likelihood, model $y|\theta, \xi$ describes the relationship between θ, our controllable design ξ, and the experimental outcome y. To select designs *optimally*, the guiding principle is *information maximization*—we select the design ξ^* that maximizes the expected (Shannon) information gained about θ from the data y, or equivalently, that maximizes the mutual information between θ and y.

This naturally extends to adaptive settings by considering the *conditional* expected information gain given previously collected data. The traditional way to do this is to fit a posterior $p(\theta|\xi_{1:t-1}, y_{1:t-1})$ after each iteration, and then select ξ_t in a myopic fashion using the one-step mutual information [see 44, for a review]. Unfortunately, this approach necessitates significant computation at each t and does not lend itself to selecting optimal designs quickly and adaptively.

Recently, Foster et al. [13] proposed an exciting alternative approach, called Deep Adaptive Design (DAD), that is based on learning design *policies*. DAD provides a way to avoid significant computation at deployment-time by, prior to the experiment itself, learning a design policy network that takes

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

past design-outcome pairs and almost instantaneously returns the design for the next stage of the experiment. The required pre-training is done using simulated experimental histories, without the need to estimate any posterior or marginal distributions. DAD further only needs a single policy network to be trained for multiple experiments, further allowing for *amortization* of the adaptive design process. Unfortunately, DAD requires conditionally independent experiments and only works for the restricted class of models which have an explicit likelihood model that we can simulate from, evaluate the density of, and calculate derivatives for, which substantially reduces its applicability.

To address this shortfall, we instead consider a far more general class of models where we require only the ability to simulate $y|\theta, \xi$ and compute the derivative $\partial y/\partial \xi$, e.g. via automatic differentiation [3]. Such models are ubiquitous in scientific modelling and include differentiable *implicit models* [14], for which the likelihood density $p(y|\theta, \xi)$ is intractable. Examples include the Lotka Volterra model used in ecology [14], and models from chemistry and epidemiology [1].

Specifically, to perform rapid, adaptive experimentation with this large class of models, we introduce *implicit Deep Adaptive Design* (iDAD), a method for learning adaptive design policy networks without likelihoods. To achieve this, we introduce likelihood-free lower bounds on the total information gained from a sequence of experiments, which iDAD utilizes to learn a deep policy network that amortizes the cost of experimental design for implicit models and can be run in milliseconds at deployment-time. To this end, we show how the InfoNCE [49] and NWJ [30] bounds, popularized in representation learning, can be applied to the policy-based experimental design setting.

We also relax DAD's requirement for experiments to be conditionally independent, allowing its application in complex settings like time series data, and through innovative architecture adaptations also provide improvements in the conditionally independent setting as well. This further expands the model space for policy-based BOED, and leads to additional performance improvements.

Critically, iDAD forms the first method in the literature that can practically perform real-time adaptive BOED for implicit models: previous approaches are either not fast enough to run in real-time for non-trivial models, or require explicit likelihood models. We illustrate the applicability of iDAD on a range of experimental design problems, highlighting its benefits over existing baselines, even finding that it often outperforms costly non-amortized approaches.

2 Background

The BOED framework [26] begins by specifying a Bayesian model of the experimental process, consisting of a prior on the unknown parameters $p(\theta)$, a set of controllable designs ξ, and a data generating process that depends on them $y|\theta, \xi$; as usual in BOED, we assume that $p(\theta)$ does not depend on ξ. In this paper, we consider the situation where $y|\theta, \xi$ is specified *implicitly*. This means that it is defined by a deterministic transformation f of a base (or noise) random variable ε which is independent of the parameter θ and the design ξ; most commonly $\varepsilon \sim \mathcal{N}(\varepsilon; 0, I)$. The function $f(\varepsilon; \theta, \xi)$ is typically not known in closed form but implemented as a stochastic computer program (simulator) with input (θ, ξ) and random variables (random seed) ε. Even if this is not the case, the resulting induced likelihood density $p(y|\theta, \xi)$ is still generally intractable, but sampling is possible.

Having acquired a design-observation pair (ξ, y), we can quantify the amount of information we have gained about θ by calculating the reduction in entropy from the prior to the posterior. We can further assess the quality of a design ξ before acquiring y by computing the expected reduction in entropy with respect to the marginal distribution of the outcomes, $p(y|\xi) = \mathbb{E}_{p(\theta)}[p(y|\theta, \xi)]$. The resulting quantity $I(\xi)$ is of central interest in BOED and is called the *expected information gain* (EIG),

$$I(\xi) \coloneqq \mathbb{E}_{p(\theta)p(y|\theta,\xi)}\left[\log(p(\theta|\xi, y)/p(\theta))\right] = \mathbb{E}_{p(\theta)p(y|\theta,\xi)}\left[\log(p(y|\theta, \xi)/p(y|\xi))\right], \tag{1}$$

which is equivalent to the mutual information (MI) between the parameters θ and data y when performing experiment ξ. The optimal ξ^* is then the one that maximises the EIG, i.e. $\xi^* = \arg\max_\xi I(\xi)$. Performing this optimization is a major computational challenge, since the information objective is doubly intractable [39]. For implicit models, the cost becomes even greater as the likelihood itself is also not available in closed form, such that estimating it, along with the marginal likelihood, for any fixed value of ξ is already a major computational problem [9, 15, 27, 47].

Finding the ξ^* that maximises the mutual information in (1) is called *static* experimental design. In practice, however, we are often more interested in performing multiple experiments *adaptively* in a sequence $\xi_1, \dots, \xi_T$, so that the choice of each ξ_t can be guided by past experiments, namely the *histories* $h_{t-1} \coloneqq \{(\xi_i, y_i)\}_{i=1:t-1} \in \mathcal{H}^{t-1}$. The typical approach in such settings is to sequentially perform posterior inference followed by a one-step look ahead (myopic) BOED optimization. In other words, to determine the designs $\xi_1, \dots, \xi_T$, we sequentially optimize the objectives

$$I_{h_{t-1}}(\xi_t) := \mathbb{E}_{p(\theta|h_{t-1})p(y_t|\theta,\xi,h_{t-1})} \left[\log(p(y_t|\theta,\xi,h_{t-1})/p(y_t|\xi,h_{t-1}))\right], \quad t = 1, \dots, T. \quad (2)$$

However, such approaches incur significant computational cost during the experiment itself, particularly for implicit models [12, 16, 24]. This has critical consequences: in most cases they cannot be run in real-time, undermining one's ability to use them in practice.

2.1 Policy-based adaptive design with likelihoods

For tractable likelihood models, Foster et al. [13] proposed a new framework, called Deep Adaptive Design (DAD), for adaptive experimental design that avoids expensive computations during the experiment. To achieve this, they introduce a parameterized deterministic design function, or policy, π_ϕ that takes the history h_{t-1} as input and returns the design $\xi_t = \pi_\phi(h_{t-1})$ to be used for the next experiment as output. This set-up allows them to consider the objective

$$\mathcal{I}_T(\pi_\phi) = \mathbb{E}_{p(\theta)p(h_T|\theta,\pi_\phi)} \left[\sum\nolimits_{t=1}^{T} I_{h_{t-1}}(\xi_t)\right], \quad \xi_t = \pi_\phi(h_{t-1}), \quad (3)$$

which crucially depends on the policy π rather than the individual design ξ_t. Learning a policy up-front, rather than designs, is what allows adaptive experiments to be performed in real-time.

Under the assumption that y_t is independent of h_{t-1} conditional on θ and the design ξ_t so that $p(y_t|\theta,\xi,h_{t-1}) = p(y_t|\theta,\xi)$, Foster et al. [13] showed that the objective can be simplified to

$$\mathcal{I}_T(\pi_\phi) = \mathbb{E}_{p(\theta)p(h_T|\theta,\pi_\phi)} \left[\log \frac{p(h_T|\theta,\pi_\phi)}{p(h_T|\pi_\phi)}\right], \quad p(h_T|\theta,\pi_\phi) = \prod\nolimits_{t=1}^{T} p(y_t|\theta,\xi_t). \quad (4)$$

To deal with the marginal $p(h_T|\pi_\phi)$ in the denominator, the authors then derived several optimizable lower bounds on $\mathcal{I}_T(\pi_\phi)$, such as the sequential Prior Contrastive Estimation (sPCE) bound

$$\mathcal{L}_T^{sPCE}(\pi_\phi, L) = \mathbb{E}_{p(\theta_0)p(h_T|\theta,\pi_\phi)p(\theta_{1:L})} \left[\log \frac{p(h_T|\theta_0,\pi_\phi)}{\frac{1}{L+1}\sum_{\ell=0}^{L} p(h_T|\theta_\ell,\pi_\phi)}\right] \leq \mathcal{I}_T(\pi_\phi)\ \forall L \geq 1. \quad (5)$$

The parameters of the policy ϕ, which takes the form of a deep neural network, are then learned prior to the experiment(s) using stochastic gradient ascent on this bound with simulated experimental histories. Design decisions can then be made using a single forward pass of π_ϕ during deployment. Unfortunately, training the DAD network by optimizing (5) requires the likelihood density $p(h_T|\theta,\pi)$ to be analytically available—an assumption that is too restrictive in many practical situations. The architecture for DAD is also based on assuming conditionally independent designs, which is unsuitable in some settings like time-series data. Our method lifts both of these restrictions.

3 Implicit Deep Adaptive Design

We have seen that the traditional step-by-step approach to adaptive design for implicit models [12, 16, 24] is too costly for quick deployment, whilst the existing policy-based approach makes overly restrictive assumptions that prevent it being applied to implicit models. We aim to relax the restrictive assumptions on policy-based BOED, making it applicable to all differentiable implicit models. This requires a new training objective for the design policy network that is not based on conditionally independent experiments and does not involve an explicit likelihood, along with new neural architectures that work for non-exchangeable models like time series.

3.1 Information lower bounds for policy-based experimental design without likelihoods

To establish a suitable likelihood-free training objective for the implicit setting, our high-level idea is to leverage recent advances in variational mutual information [see 35, for an overview], which have shown promise for *static* BOED [12, 22, 23]. While using these bounds in the traditional sequential BOED framework of (2) would not permit real-time experiments, one could consider a naive application of them to the policy objective of (3) by replacing each $I_{h_{t-1}}$ with a suitable variational lower bound that uses a 'critic' $U_t : \mathcal{H}^{t-1} \times \Theta \to \mathbb{R}$ to avoid explicit likelihood evaluations. An effective critic successfully encapsulates the true likelihood, tightening the bound. Although its form depends on the choice of bound, critics are parametrized and trained in the same way, namely by a neural network U_{ϕ_t} which is optimized to tighten the bound. Unfortunately, replacing each $I_{h_{t-1}}$ involves learning T such critic networks and requires samples from all posteriors $p(\theta|h_{t-1})$, which will typically be impractically costly.

To avoid this issue, we show that we can obtain a unified information objective similar to (4), *even without conditionally independent experiments*. The following proposition therefore marks the first key milestone in eliminating the restrictive assumptions of [13], by establishing a unified objective without intermediate posteriors that is valid even when the model itself changes between time steps.

Proposition 1 (Unified objective)**.** *Consider the data generating distribution* $p(h_T|\theta,\pi) = \prod_{t=1:T} p(y_t|\theta,\xi_t,h_{t-1})$, *where* $\xi_t = \pi(h_{t-1})$ *are the designs generated by the policy and, unlike in* (4), y_t *is allowed to depend on the complete history* h_{t-1}. *Then we can write* (3) *as*

$$\mathcal{I}_T(\pi) = \mathbb{E}_{p(\theta)p(h_T|\theta,\pi)}\left[\log p(h_T|\theta,\pi)\right] - \mathbb{E}_{p(h_T|\pi)}\left[\log p(h_T|\pi)\right]. \tag{6}$$

Proofs are presented in Appendix A. The advantage of (6) is that we can draw samples from $p(\theta)p(h_T|\theta,\pi)$ simply by sampling our model and taking forward passes through the design network. However, neither of the *densities* $p(h_T|\theta,\pi)$ nor $p(h_T|\pi)$ are tractable for implicit models.

To side-step this intractability, we observe that $\mathcal{I}_T(\pi)$ takes an analogous form to a mutual information between θ and h_T. For measure-theoretic reasons this is technically not correct because the $\xi_{1:T}$ are deterministic given $y_{1:T}$ (see Appendix A for a full discussion). However, the following two propositions show that we can treat $\mathcal{I}_T(\pi)$ *as if it were this mutual information*. Specifically, we show that the InfoNCE [49] and NWJ [30] bounds on the mutual information can be adapted to establish tractable lower bounds on our unified objective $\mathcal{I}_T(\pi)$. These two bounds both utilize a *single* auxiliary critic network U_ψ which is trained simultaneously with the design network.

Proposition 2 (NWJ bound for implicit policy-based BOED)**.** *For a design policy* π *and a critic function* $U : \mathcal{H}^T \times \Theta \to \mathbb{R}$, *let*

$$\mathcal{L}_T^{NWJ}(\pi, U) := \mathbb{E}_{p(\theta)p(h_T|\theta,\pi)}\left[U(h_T,\theta)\right] - e^{-1}\mathbb{E}_{p(\theta)p(h_T|\pi)}\left[\exp(U(h_T,\theta))\right]. \tag{7}$$

Then $\mathcal{I}_T(\pi) \geq \mathcal{L}_T^{NWJ}(\pi, U)$ *holds for any* U. *Further, the inequality is tight for the optimal critic* $U^*(h_T,\theta) = \log p(h_T|\theta,\pi) - \log p(h_T|\pi) + 1$.

Proposition 3 (InfoNCE bound for implicit policy-based BOED)**.** *Let* $\theta_{1:L} \sim p(\theta_{1:L}) = \prod_i p(\theta_i)$ *be a set of contrastive samples where* $L \geq 1$. *For design policy* π *and critic function* $U : \mathcal{H}^T \times \Theta \to \mathbb{R}$, *let*

$$\mathcal{L}_T^{NCE}(\pi, U; L) := \mathbb{E}_{p(\theta_0)p(h_T|\theta_0,\pi)}\mathbb{E}_{p(\theta_{1:L})}\left[\log \frac{\exp(U(h_T,\theta_0))}{\frac{1}{L+1}\sum_{i=0}^{L}\exp(U(h_T,\theta_i))}\right]. \tag{8}$$

Then $\mathcal{I}_T(\pi) \geq \mathcal{L}_T^{NCE}(\pi, U; L)$ *for any* U *and* $L \geq 1$. *Further, the inequality is tight in the limit as* $L \to \infty$ *for the optimal critic* $U^*(h_T,\theta) = \log p(h_T|\theta,\pi) + c(h_T)$, *where* $c(h_T)$ *is an arbitrary function depending only on the history. The optimal critic recovers the sPCE bound in* (5).

We propose these two alternative bounds due to their complementary properties: the NWJ bound can have large variance, but tends to be less biased (note $\mathcal{L}_T^{NCE} \leq \log(L+1)$ [35]). While the NWJ critic must learn to self-normalize, the InfoNCE bound avoids this issue but is still not tight even when the optimal critic is found (for finite L). Consequently, only the NWJ objective recovers the true optimal policy, i.e. $\pi^* = \arg\max_\pi \max_U \mathcal{L}_T^{NWJ}(\pi, U)$ which does not equal $\arg\max_\pi \max_U \mathcal{L}_T^{NCE}(\pi, U; L)$ in general. We provide further discussion in Appendix B.

In practice, we represent both the policy π and the critic U as neural networks, π_ϕ and U_ψ respectively, so that the lower bounds become a function $\mathcal{L}(\pi_\phi, U_\psi)$ of their parameters. By optimizing $\mathcal{L}(\pi_\phi, U_\psi)$ with respect to ϕ we improve the quality of the designs proposed by the design network, while optimizing ψ ensures that the bound becomes tighter, therefore resulting in more accurate estimates of the mutual information. Simultaneous optimization of ϕ and ψ, as discussed in the next section, therefore leads to a design network that can choose high-quality designs.

The propositions also show that an estimate of the likelihood function can be extracted from the final trained critic U_{ψ^*} for a fixed history h_T of real data. We can use this to compute an approximate posterior over θ given the collected real data in the designed experiment. This means that we can perform likelihood-free inference after training the critic, which importantly extends previous results [22, 23] from the static to the adaptive policy-based experimental design setting.

Algorithm 1: Implicit Deep Adaptive Design with (iDAD)

Input: Differentiable simulator f, prior $p(\theta)$, number of experimental steps T
Output: Design network π_ϕ, critic network U_ψ
while Computational training budget not exceeded **do**
  Sample $\theta \sim p(\theta)$ and set $h_0 = \varnothing$
  for $t = 1, ..., T$ **do**
    Compute $\xi_t = \pi_\phi(h_{t-1})$
    Sample $\varepsilon_t \sim p(\varepsilon)$ and compute $y_t = f(\varepsilon_t; \xi_t, \theta, h_{t-1})$
    Set $h_t = \{(\xi_1, y_1), ..., (\xi_t, y_t)\}$
  end
  Estimate $\nabla_{\phi,\psi}\mathcal{L}_T(\pi_\phi, U_\psi)$ as per (10) where $\mathcal{L}_T(\pi_\phi, U_\psi)$ is either NWJ (7), or InfoNCE (8)
  Update the parameters (ϕ, ψ) using stochastic gradient ascent scheme
end
For deployment, use the deterministic trained design network π_ϕ to obtain a designs ξ_t directly.

To use these bounds in practice, two key challenges remain. First, we must set up a scheme to optimize our chosen lower bound with respect to ϕ, ψ using stochastic optimization methods [20, 42]. Second, we must choose a suitable neural architecture for the critic U_ψ and the design network π_ϕ.

3.2 Gradient estimation

To optimize these bounds using stochastic gradients, we must account for the fact that the parameter ϕ affects the probability distributions with respect to which expectations are taken. We deal with this problem by utilizing the reparametrization trick [28, 41], for which we assume that design space Ξ and observation space $\mathcal{Y}$ are continuous. To this end, we first formalize the notion of a differentiable implicit model in the adaptive design setting as

$$y_t = f(\varepsilon_t; \xi_t(h_{t-1}), \theta, h_{t-1}), \quad \text{where} \quad \theta \sim p(\theta), \quad \varepsilon_t \sim p(\varepsilon) \;\; \forall t \in \{1, \dots, T\} \tag{9}$$

and we assume that we can compute the derivatives $\partial f/\partial \xi$ and $\partial f/\partial h$. Interestingly, it is possible to use an implicit prior without access to the density $p(\theta)$, and we do not need access to $\partial f/\partial \theta$.

Under these conditions, we can express the bounds in terms of expectations that do not depend on ϕ or ψ, and hence move the gradient operator inside. For $\mathcal{L}_T^{NCE}(\pi_\phi, U_\psi; L)$, for example, we have

$$\nabla_{\phi,\psi}\mathcal{L}_T^{NCE} = \mathbb{E}_{p(\theta_{0:L})p(\varepsilon_{1:T})}\left[\nabla_{\phi,\psi} \log \frac{\exp(U_\psi(h_T(\varepsilon_{1:T}, \pi_\phi), \theta_0))}{\frac{1}{L+1}\sum_{i=0}^{L}\exp(U_\psi(h_T(\varepsilon_{1:T}, \pi_\phi), \theta_i))}\right]. \tag{10}$$

While each element of the history h_T depends on ϕ in a possibly nested manner, we do not need to explicitly keep track of these dependencies thanks to automatic differentiation [3, 34].

Like DAD, our new method—which we call *implicit Deep Adaptive Design* (iDAD)—is trained with simulated histories $h_T = \{(\xi_i, y_i)\}_{i=1:T}$ prior to the actual experiment, allowing design decision to be made using a single forward pass during deployment. Unlike DAD, however, it does not require knowledge of the likelihood function, nor the assumption of conditionally independent designs, which significantly broadens its applicability. A summary of the iDAD approach is given in Algorithm 1.

3.3 Network architectures

Having established that policy-based BOED with implicit models is possible to do with a single critic network U_ψ, it is essential to choose its architecture carefully as that will not only facilitate training, but also help achieve tighter bounds which will in turn lead to better designs. The critic network, unlike the design policy, takes two inputs—a *complete* history h_T and a parameter θ, which belong to different spaces and typically have very different dimensions. To deal with this, we first represent the inputs h_T and θ as vectors in $\mathbb{R}^d$, using two *encoder networks* E_{ψ_h} and E_{ψ_θ}, which, once trained, correspond to approximate sufficient statistics [7]. We then define our critic to be the dot product between the two vector representations, $U_\psi(h_T, \theta) = E_{\psi_h}(h_T)^\top E_{\psi_\theta}(\theta)$, corresponding to a *separable* critic architecture typically used in the representation learning literature [2, 6, 49].

Rather than encoding the entire history h_T immediately, we first encode individual design–outcome pairs with a network E_{ψ_0} and then concatenate the encodings into a vector $R_{cat}(h_T) =$

$(E_{\psi_0}(\xi_1, y_1), \ldots, E_{\psi_0}(\xi_T, y_T))$. This is then passed through final fully connected *head* layers, H_{ψ_1}. Our history encoder is therefore $E_{\psi_h} = H_{\psi_1}(R_{cat}(h_T))$. This generic architecture does not assume conditional independence of the data and is applicable to all models we consider. However, as the following proposition shows, when conditional independence does hold, the critic is still invariant to the order of the history—something which can be utilized to construct more efficient architectures.

Proposition 4 (Permutation invariance)**.** *Let $U : \mathcal{H}^T \times \Theta \to \mathbb{R}$ be a critic and let σ be a permutation acting on a history h^1_T yielding $h^2_T = \{(\xi_{\sigma(i)}, y_{\sigma(i)})\}_{i=1}^T$. If the data generating process is conditionally independent of its past, then the critic is invariant under permutations of the history, i.e.*

$$p(\theta) \prod_{t=1}^{T} p(y_t|\theta, \xi_t(h_{t-1}), h_{t-1}) = p(\theta) \prod_{t=1}^{T} p(y_t|\theta, \xi_t) \implies U(h^1_T, \theta) = U(h^2_T, \theta). \quad (11)$$

Attention to history. We propose utilizing a more advanced permutation invariant lower level architecture based on self-attention [51]—a popular deep learning module [10, 19, 32, 40]. Namely, we incorporate self-attention mechanisms, inspired by the Image Transformer of [33] in *both* the design and critic networks. As we show later, this provides notable further empirical gains.

Design network for non-exchangeable data. The proposed design network architecture in [13] is based on pooling [53], and requires conditional independence. The concatenation approach described above is also not appropriate for the *design* network, since the π_ϕ takes intermediate histories h_t as input. To ensure the inputs are of equal length, we use zero-padding so that $R_{cat}(h_t) = (E_{\phi_0}(\xi_1, y_1), \ldots, E_{\phi_0}(\xi_t, y_t), 0 \ldots, 0)$. We note that the design–outcome encoder can be shared between the critic and the design network, i.e. setting $E_{\psi_0} = E_{\phi_0}$, which can bring further efficiency.

4 Related work

Adaptive policy-based BOED has only recently been introduced [13] and has not yet been extended to implicit models—the gap that this work addresses. Previous approaches to adaptive experiments usually follow the two-step greedy procedure described in Section 2. Methods for MI/EIG estimation without likelihoods include the use of variational bounds [11, 12, 22] and ratio estimation [21, 24]; approximate Bayesian computation together with kernel density estimation [36]; and approximating the intractable likelihood first, for example via polynomial chaos expansion [18], followed by applying likelihood-based estimators, such as nested Monte Carlo [39]. The maximization step in more traditional methods tends to rely on gradient-free optimization, including grid-search, evolutionary algorithms [37], Bayesian optimization [11, 24], or Gaussian process surrogates [31]. More recently, gradient-based approaches have been introduced [11, 22], some of which allow the estimation and optimization simultaneously in a single stochastic-gradient scheme [12, 17, 23]. From a posterior estimation perspective, likelihood-free inference can be performed via approximate Bayesian computation [27, 47], ratio estimation [48], conventional MCMC for methods that make tractable approximation to the likelihood [17, 18], or as a byproduct of MI estimation [12, 21, 23, 24].

5 Experiments

Table 1: Key properties of considered methods

	Adaptive	Real-time	Implicit
Random	✗	N/A	✓
Equal interval	✗	N/A	✓
MINEBED	✗	N/A	✓
SG-BOED	✗	N/A	✓
Variational	✓	✗	✓
DAD	✓	✓	✗
iDAD	✓	✓	✓

We evaluate the performance of **iDAD** on a number of real-world experimental design problems and a range of baselines (summarized in Table 1). Since we aim to perform adaptive experiments in *real-time*, we focus mostly on baselines that do not require significant computational time during the experiment. These include heuristic approaches that require no training, namely **equal** interval designs (when possible) and **random** designs, as well as static BOED approaches. The latter are also non-adaptive strategies, which learn a set of designs $\xi_1, ..., \xi_T$ prior to the experiment by optimising the mutual information objective of Equation (1). The static BOED approaches we consider are the **MINEBED** method of [22] and the likelihood-free ACE approach of [12], where we use the prior as a proposal distribution, referring to this baseline as **SG-BOED**. We also implement the expensive traditional non-amortized myopic strategy described in Section 2, for which we use the mean-field **variational** posterior estimator of [11] at each experiment step. Finally, where possible, we compare our method with DAD [13], in order to assess the performance gap that would arise if we had an analytic likelihood. This comparison is done primarily for evaluation

Table 2: Upper and lower bounds on the total information, $\mathcal{I}_{10}(\pi)$, for the location finding experiment in Section 5.1. The bounds were estimated using $L = 5 \times 10^5$ contrastive samples. Errors indicate ± 1 s.e. estimated over 4096 histories (128 for variational). Deployment time was measured on a CPU (GPU for variational) and errors were calculated on the basis of 10 runs.

Method	Lower bound	Upper bound	Deployment time (sec.)
Random	4.7914 ± 0.0403	4.7941 ± 0.0405	N/A
MINEBED	5.5183 ± 0.0283	5.5217 ± 0.0284	N/A
SG-BOED	5.5466 ± 0.0280	5.5490 ± 0.0281	N/A
Variational	4.6385 ± 0.1440	4.6438 ± 0.1456	$758.4 \pm 1\%$
iDAD (NWJ)	$\mathbf{7.6942 \pm 0.0448}$	$\mathbf{7.8061 \pm 0.0495}$	$0.0167 \pm 2\%$
iDAD (InfoNCE)	$\mathbf{7.7500 \pm 0.0386}$	$\mathbf{7.8631 \pm 0.0425}$	$0.0168 \pm 2\%$
DAD	7.9669 ± 0.0342	8.0335 ± 0.0375	$0.0070 \pm 6\%$

purposes—as it has access to the likelihood density DAD serves as an upper bound on the performance iDAD can achieve; one should use explicit likelihood methods whenever possible.

The main performance metric that we focus on is the total EIG, $\mathcal{I}_T(\pi)$, as given in (4). In cases where the likelihood is available we estimate the total EIG using the sPCE lower bound in (5) and its corresponding upper bound, the sequential Nested Monte Carlo bound [sNMC; 13]. To ensure that the bounds are tight, we evaluate them with a large number of contrastive samples, i.e. $L \geq 10^5$. Where the likelihood is truly intractable, we assess the iDAD strategy in a more qualitative manner, by looking at the optimal designs and approximate posteriors.

For the adaptive experiments, we further consider the deployment time (i.e. the time required to propose a design), which is a critical metric for our aims. All deployment times exclude the time needed to determine the first experiment as that can be computed up-front, during the training phase.

We implement iDAD by extending PyTorch [34] and Pyro [4] to provide an implementation that is abstracted from the specific probabilistic model. Code is provided in the Supplement and full experiment details are given in Appendix C.

5.1 Location Finding in 2D

We first demonstrate our approach on the location finding experiment from [13]. Inspired by the acoustic energy attenuation model of Sheng and Hu [46], this experiment involves finding the locations of multiple hidden sources, each emitting a signal with intensity that decreases according to the inverse-square law. The *total intensity*—a superposition of these signals—is measured with noise. The design problem is choosing where to measure the total signal in order to uncover the sources.

We train iDAD networks to perform $T = 10$ experiments to locate 2 sources (see Appendix C.3 for additional results). We incorporate attention mechanisms in both the design and the critic networks. Table 2 shows the performance of each method. We can see that iDAD has a very small performance gap to DAD and substantially outperforms all baselines, including, perhaps surprisingly, the traditional (non-amortized) adaptive variational approach, despite its large computational budget. The particularly poor performance of the variational approach is likely driven by the inability of the mean-field variational family to capture the highly non-Gaussian true posterior, highlighting the detrimental effect wrong posteriors can have on determining optimal designs in the traditional sequential BOED setting. Overall, this experiment demonstrates that iDAD is able to learn near-optimal amortized design policies without likelihoods and can be run in milliseconds at deployment.

Ablation: attention to history. We next assess the benefit of utilizing our more sophisticated permutation invariant architectures, compared to the simple pooling of [53] used in [13]. Our approach incorporates attention layers into both networks that we train. This leads us to four possible combinations of network architectures. Table 3 compares the efficacy of the resulting design policies and strongly suggests that incorporating attention mechanisms in either and/or both networks improves performance, with inclusion in the design network particularly important.

5.2 Pharmacokinetic model

Our next experiment is taken from the pharmacokinetics literature and has been studied in other recent works on BOED for implicit models [22, 54]. Specifically, we consider the compartmental

Table 3: Lower and upper bounds on mutual information $\mathcal{I}_{10}(\pi)$ for different network architectures on location finding experiment using the InfoNCE bound. All estimates obtained as in Table 2.

Design	Critic	Lower bound	Upper bound
Attention	**Attention**	$\mathbf{7.7500 \pm 0.0386}$	$\mathbf{7.8631 \pm 0.0425}$
Attention	Pooling	7.5670 ± 0.0366	7.6317 ± 0.0386
Pooling	Attention	7.3981 ± 0.0398	7.4701 ± 0.0424
Pooling	Pooling	7.1346 ± 0.0374	7.1921 ± 0.0405

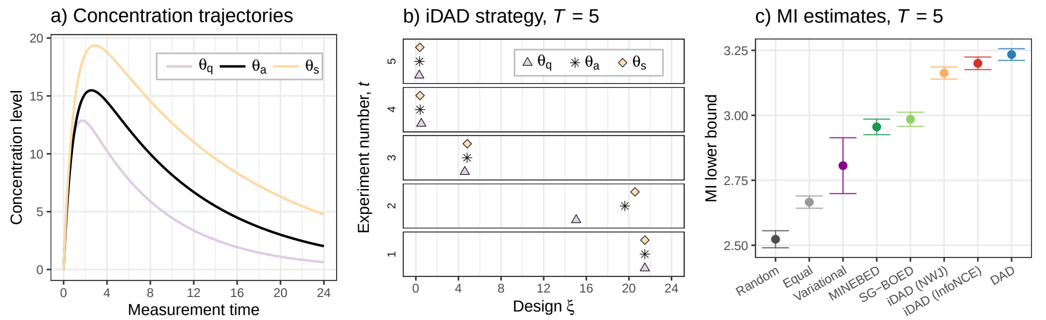

Figure 1: Plots for pharmacokinetics experiment. a) Visualisation of model showing concentration level as a function of measurement time for 3 values of θ, resulting in a quick (θ_q), average (θ_a), or slow (θ_s) trajectory. b) Designs selected by an iDAD policy trained with InfoNCE. c) Mutual information lower bounds achieved by iDAD and baselines. All estimates obtained as in Table 2.

model of [43], for which the distribution of an administered drug through the body is governed by three parameters: absorption rate k_α, elimination rate k_e and volume V, which form the parameters of interest, i.e. $\theta = (k_\alpha, k_e, V)$. Given $T = 5$ patients, the design problem is to adaptively choose blood sampling times, $0 \leq \xi_t \leq 24$ hours, for each, measured from the the point the drug was administered (with patient 2 not being administered until after sampling patient 1 etc). Plausible concentration trajectories are shown in Figure 1a). Full details and further results are given in Appendix C.4.

We first qualitatively consider the design policy of iDAD (trained with the InfoNCE objective) in Figure 1b). As we have not yet observed any data, the optimal design for the first patient (bottom row) is the same for all θ. For the second patient, only guided by ξ_1 and the outcome y_1, iDAD is already able to distinguish between quickly and slowly decaying concentration trajectories: it proposes a significantly earlier measurement time for the quickly decaying trajectory (purple triangle, θ_q) and later time for the slowly decaying one (yellow diamond, θ_s). For the third patient, iDAD always targets the peak of the drug concentration distribution which is quite similar for all θ. Measurements for the last two patients are made soon after the drug has been administered ($\sim 15 - 30$ min), when concentration levels increase rapidly, to capture information about how quickly the drug is absorbed.

To provide more quantitative assessment and compare to our baselines, we again consider the final EIG values as shown in Figure 1c). This reveals that the iDAD strategies perform best among the methods that are applicable to implicit models, confirming that the learnt policies propose superior designs. The performance gap to DAD, which relies on explicit likelihoods, is not statistically significant (at the 5% level) for iDAD trained with InfoNCE, while significant, but still small, for NWJ.

Finally, we consider the convergence of the iDAD networks under different training objective and compare to DAD for reference. As shown in Figure 2, although all three converge to approximately the same value, they do so at rather different speeds: while DAD requires about 5000 gradient updates, implicit methods need longer training and tend to exhibit higher variance, particularly NWJ.

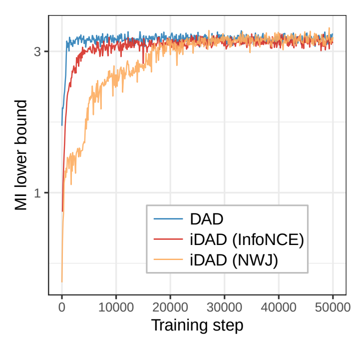

Figure 2: Convergence of mutual information lower bounds.

5.3 SIR Model

In this experiment we demonstrate our approach on an implicit model from epidemiology. Namely, we consider a formulation of the stochastic SIR model [8] that is based on stochastic differential equations

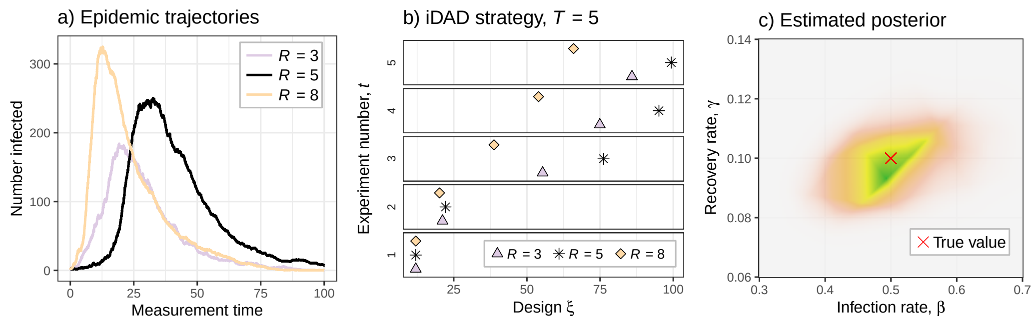

Figure 3: a) Epidemic trajectories for 3 realization of (β, γ) with different reproduction numbers $R = \beta/\gamma$. b) Designs selected by an iDAD policy trained with NWJ. c) Example posterior estimates from the critic network given data generated with the ground-truth parameters shown by the red cross.

(SDEs), as done by [23]. Here, individuals in a fixed population can move from a susceptible state $S(\tau)$ to an infected state $I(\tau)$, after which they can move to a recovered state $R(\tau)$. The dynamics of these two events are governed by two model parameters, the infection rate β and the recovery rate γ, which we wish to estimate. Our aim is to determine the optimal measurement times at which to measure the state populations, in particular the number of infected $I(\tau)$. This implicit model is challenging because data simulation is expensive, since we need to solve many SDEs, and experimental designs have a time-dependency. See Appendix C.5 for full details.

Table 4: MI lower bounds (± 1 s.e.).

Method	Lower bound
Random	1.9049 ± 0.0317
Equal interval	2.8670 ± 0.0031
MINEBED	3.0058 ± 0.0030
iDAD (NWJ)	$\mathbf{3.0429 \pm 0.0024}$

We here perform $T = 5$ experiments and train iDAD with the NWJ bound as done previously. Results for InfoNCE are discussed in the Appendix. We compare our iDAD approach to random designs, equidistant designs, and the static MINEBED approach (DAD cannot be run because the problem corresponds to a true implicit model). This yields the lower bound estimates presented in Table 4, which show that iDAD outperforms all compared methods, though it should be noted that these results are also influenced by the biases in the estimation process that are difficult to avoid because the model is implicit. In general though, iDAD should be more adversely effected by this bias than the baselines, see the Appendix C.5 for discussion.

Figure 3 further summarizes important qualitative results for this model. Figure 3a) shows different epidemic trajectories, i.e. number of infected $I(\tau)$, as a function of measurement time τ. Figure 3b) shows the learned iDAD policy for the same three underlying true parameters considered in Figure 3a). Importantly, diseases with a significantly different profile, e.g. a slow or a fast spread, result in different sets of optimal designs, highlighting the adaptivity of iDAD. Finally, Figure 3c) shows an example posterior distribution estimate from the learnt iDAD critic network, which we see is consistent with the ground truth parameters.

6 Discussion

Limitations. The benefit that iDAD can be used in live experiments comes at the cost of substantial pre-training which can be computationally expensive. This though is mitigated by its amortization of the adaptive design process, such that only one network needs training even if we have multiple experiment instances. The cost–performance trade–off can also be directly controlled by judicious choices of architecture. Another natural limitation is that the use of gradients naturally restricts the approach to continuous design settings, something which future work might look to address.

Conclusions. In this paper we introduced iDAD—the first policy-based adaptive BOED method that can be applied to implicit models. By training a design network without likelihoods upfront, iDAD is also the first method that allows real-time adaptive experiments for simulator-based models. In our experiments iDAD performed significantly better than all likelihood-free baselines. Further, in models where the likelihood is available, it was able to almost match likelihood-based adaptive approaches, which act as an upper bound on what can be achieved by an implicit method. In conclusion, we believe iDAD marks a step change in Bayesian experimental design for *implicit* models, allowing designs to be proposed quickly, adaptively, and non-myopically during the live experiment.

References

[1] Edward J. Allen, Linda J. S. Allen, Armando Arciniega, and Priscilla E. Greenwood. Construction of equivalent stochastic differential equation models. *Stochastic Analysis and Applications*, 26(2):274–297, 2008.

[2] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019.

[3] Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *Journal of machine learning research*, 18, 2018.

[4] Eli Bingham, Jonathan P Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D Goodman. Pyro: Deep universal probabilistic programming. *Journal of Machine Learning Research*, 2018.

[5] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, pages 273–304, 1995.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020.

[7] Yanzhi Chen, Dinghuai Zhang, Michael U. Gutmann, Aaron Courville, and Zhanxing Zhu. Neural approximate sufficient statistics for implicit models. In *International Conference on Learning Representations*, 2021.

[8] Alex R Cook, Gavin J Gibson, and Christopher A Gilligan. Optimal observation times in experimental epidemic processes. *Biometrics*, 64(3):860–868, 2008.

[9] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 2020.

[10] Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 4171–4186, 2019.

[11] Adam Foster, Martin Jankowiak, Elias Bingham, Paul Horsfall, Yee Whye Teh, Thomas Rainforth, and Noah Goodman. Variational Bayesian Optimal Experimental Design. In *Advances in Neural Information Processing Systems 32*, pages 14036–14047. Curran Associates, Inc., 2019.

[12] Adam Foster, Martin Jankowiak, Matthew O'Meara, Yee Whye Teh, and Tom Rainforth. A unified stochastic gradient approach to designing bayesian-optimal experiments. In *International Conference on Artificial Intelligence and Statistics*, pages 2959–2969. PMLR, 2020.

[13] Adam Foster, Desi R Ivanova, Ilyas Malik, and Tom Rainforth. Deep adaptive design: Amortizing sequential bayesian experimental design. *arXiv preprint arXiv:2103.02438*, 2021.

[14] Matthew Graham and Amos Storkey. Asymptotically exact inference in differentiable generative models. In *Artificial Intelligence and Statistics*, pages 499–508. PMLR, 2017.

[15] PeterJ. Green, Krzysztof Latuszynski, Marcelo Pereyra, and Christian P. Robert. Bayesian computation: a summary of the current state, and samples backwards and forwards. *Statistics and Computing*, 25(4):835–862, 2015. doi: 10.1007/s11222-015-9574-5.

[16] Markus Hainy, Christopher C. Drovandi, and James M. McGree. Likelihood-free extensions for bayesian sequentially designed experiments. In Joachim Kunert, Christine H. Müller, and Anthony C. Atkinson, editors, *mODa 11 - Advances in Model-Oriented Design and Analysis*, pages 153–161. Springer International Publishing, 2016.

[17] Xun Huan and Youssef Marzouk. Gradient-based stochastic optimization methods in bayesian experimental design. *International Journal for Uncertainty Quantification*, 4(6), 2014.

[18] Xun Huan and Youssef M Marzouk. Simulation-based optimal bayesian experimental design for nonlinear systems. *Journal of Computational Physics*, 232(1):288–317, 2013.

[19] Cheng Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer: Generating music with long-term structure, 2019. ISSN 23318422.

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[21] S. Kleinegesse and M.U. Gutmann. Efficient Bayesian experimental design for implicit models. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 89 of *Proceedings of Machine Learning Research*, pages 1584–1592. PMLR, 2019.

[22] Steven Kleinegesse and Michael Gutmann. 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, pages 5316–5326. PMLR, 2020.

[23] Steven Kleinegesse and Michael U. Gutmann. Gradient-based bayesian experimental design for implicit models using mutual information lower bounds. *arXiv preprint arXiv:2105.04379*, 2021.

[24] Steven Kleinegesse, Christopher Drovandi, and Michael U. Gutmann. Sequential Bayesian Experimental Design for Implicit Models via Mutual Information. *Bayesian Analysis*, pages 1 – 30, 2021. doi: 10.1214/20-BA1225.

[25] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019.

[26] Dennis V Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, pages 986–1005, 1956.

[27] J. Lintusaari, M.U. Gutmann, R. Dutta, S. Kaski, and J. Corander. Fundamentals and recent developments in approximate Bayesian computation. *Systematic Biology*, 66(1):e66–e82, January 2017.

[28] Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 21(132):1–62, 2020.

[29] Jay I Myung, Daniel R Cavagnaro, and Mark A Pitt. A tutorial on adaptive design optimization. *Journal of mathematical psychology*, 57(3-4):53–67, 2013.

[30] Xuanlong Nguyen, Martin J. Wainwright, and Michael I. Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11), 2010. ISSN 00189448. doi: 10.1109/TIT.2010.2068870.

[31] Antony Overstall and James McGree. Bayesian Design of Experiments for Intractable Likelihood Models Using Coupled Auxiliary Models and Multivariate Emulation. *Bayesian Analysis*, 15(1):103 – 131, 2020. doi: 10.1214/19-BA1144.

[32] Emilio Parisotto, Francis Song, Jack Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant Jayakumar, Max Jaderberg, Raphaël Lopez Kaufman, Aidan Clark, Seb Noury, Matthew Botvinick, Nicolas Heess, and Raia Hadsell. Stabilizing transformers for reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7487–7498. PMLR, 2020.

[33] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018.

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[35] Ben Poole, Sherjil Ozair, Aäron van den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180, 2019.

[36] David J. Price, Nigel G. Bean, Joshua V. Ross, and Jonathan Tuke. On the efficient determination of optimal bayesian experimental designs using abc: A case study in optimal observation of epidemics. *Journal of Statistical Planning and Inference*, 172:1–15, May 2016.

[37] David J Price, Nigel G Bean, Joshua V Ross, and Jonathan Tuke. An induced natural selection heuristic for finding optimal bayesian experimental designs. *Computational Statistics & Data Analysis*, 126:112–124, 2018.

[38] Tom Rainforth. *Automating Inference, Learning, and Design using Probabilistic Programming*. PhD thesis, University of Oxford, 2017.

[39] Tom Rainforth, Rob Cornish, Hongseok Yang, Andrew Warrington, and Frank Wood. On nesting monte carlo estimators. In *International Conference on Machine Learning*, pages 4267–4276. PMLR, 2018.

[40] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[41] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pages 1278–1286, 2014.

[42] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[43] Elizabeth G. Ryan, Christopher C. Drovandi, M. Helen Thompson, and Anthony N. Pettitt. Towards bayesian experimental design for nonlinear models that require a large number of sampling times. *Computational Statistics & Data Analysis*, 70:45–60, 2014. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2013.08.017.

[44] Elizabeth G Ryan, Christopher C Drovandi, James M McGree, and Anthony N Pettitt. A review of modern computational algorithms for bayesian optimal design. *International Statistical Review*, 84(1):128–154, 2016.

[45] Ben Shababo, Brooks Paige, Ari Pakman, and Liam Paninski. Bayesian inference and online experimental design for mapping neural microcircuits. In *Advances in Neural Information Processing Systems*, pages 1304–1312, 2013.

[46] Xiaohong Sheng and Yu Hen Hu. 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.

[47] S.A. Sisson, Y. Fan, and M. Beaumont. *Handbook of Approximate Bayesian Computation*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. CRC Press, 2018. ISBN 9781351643467.

[48] Owen Thomas, Ritabrata Dutta, Jukka Corander, Samuel Kaski, and Michael U Gutmann. Likelihood-free inference by ratio estimation. *arXiv preprint arXiv:1611.10242*, 2016.

[49] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[50] Joep Vanlier, Christian A Tiemann, Peter AJ Hilbers, and Natal AW van Riel. A Bayesian approach to targeted experiment design. *Bioinformatics*, 28(8):1136–1142, 2012.

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[52] Benjamin T Vincent and Tom Rainforth. The DARC toolbox: automated, flexible, and efficient delayed and risky choice experiments using bayesian adaptive design. 2017.

[53] Manzil Zaheer, Satwik Kottur, Siamak Ravanbhakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J Smola. Deep sets. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, 2017.

[54] Jiaxin Zhang, Sirui Bi, and Guannan Zhang. A stochastic approximate gradient ascent method for bayesian experimental design with implicit models. In *The 24nd International Conference on Artificial Intelligence and Statistics*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] Claims we make match theoretical and experimental results. Contributions and the overarching assumptions (e.g. types of models we consider) are clearly stated in the introduction.
   (b) Did you describe the limitations of your work? [Yes] Limitations are discussed throughout the paper (e.g. Section 3 discusses limitations of the bounds used) and in the conclusion.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] We thought about this issue and did not establish potential negative societal impact, ethical or environmental harm that our work could be a cause of.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] Detailed proofs are provided in the Appendix which also includes details on how our results relate to previous results in the field.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] In the supplemental material and includes instructions how to set-up an environment to reproduce the results.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In the appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] In the appendix, for one of the models as training is computationally intensive.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Resources used are described in the appendix, together with approximate time required to train a model.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Deep learning and probabilistic programming frameworks that were used for the experimental part of this work were cited. Details on versions used are available in the appendix.
   (b) Did you mention the license of the assets? [Yes] In appendix along with the details on computational resources.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Code is provided in the supplement
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] No actual data was used. All experiments were done using simulators.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]