Optimizing Sequential Experimental Design with Deep Reinforcement Learning

Tom Blau [1] **Edwin V. Bonilla** [1] **Iadine Chades** [2] **Amir Dezfouli** [1]

Abstract

Bayesian approaches developed to solve the optimal design of sequential experiments are mathematically elegant but computationally challenging. Recently, techniques using amortization have been proposed to make these Bayesian approaches practical, by training a parameterized policy that proposes designs efficiently at deployment time. However, these methods may not sufficiently explore the design space, require access to a differentiable probabilistic model and can only optimize over continuous design spaces. Here, we address these limitations by showing that the problem of optimizing policies can be reduced to solving a Markov decision process (MDP). We solve the equivalent MDP with modern deep reinforcement learning techniques. Our experiments show that our approach is also computationally efficient at deployment time and exhibits state-of-the-art performance on both continuous and discrete design spaces, even when the probabilistic model is a black box.

1. Introduction

One of the fundamental building blocks of scientific investigation is the development of predictive models that describe natural phenomena. Such models must be learned from data, which can only be acquired by conducting experiments. Since experiments are often costly and time-consuming, this naturally leads to the question of how to design experiments so that they are maximally informative. The formalism of Bayesian optimal experimental design (BOED, Lindley, 1956) casts this problem as optimization of expected information gain. BOED is a popular approach in many scientific fields (Chaloner & Verdinelli, 1995; Ryan et al., 2016).

[1]CSIRO's Data61, Australia [2]CSIRO's Land and Water, Australia. Correspondence to: Tom, Blau <tom.blau@data61.csiro.au>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

Following a Bayesian optimal experimental design (BOED) approach, let $p(y \mid \theta, d)$ be a probabilistic model describing some phenomenon. Here d represents our independent variables, or experimental *design*, y is the dependent variables, or experimental *outcome*, and θ the parameters of the model. We do not know the exact values of θ, but we have some prior belief about it represented as a distribution $p(\theta)$. The goal is then to find the design d^* that maximizes the expected information gain (EIG) from the experiment:

$$d^* = \arg\max_{d \in \mathcal{D}} \mathbb{E}_{p(y \mid d)} \left[H(p(\theta)) - H(p(\theta \mid y, d)) \right], \quad (1)$$

where $p(y \mid d)$ can be computed as an expectation over the original model, $H(\cdot)$ is the entropy of a distribution, and $\mathcal{D}$ is the space of possible designs. Simply put, this means maximizing the expected decrease in entropy from the prior to the posterior. In the sequential setting, after conducting an experiment and observing its outcome, the new posterior $p(\theta \mid y, d)$ is used instead of the prior and a new design is computed. This can be repeated for as long as resources allow, a process called *iterated experimental design*.

Evaluating the above expression requires computing an expectation w.r.t. the joint probability distribution $p(y, \theta \mid d) = p(y \mid d)p(\theta \mid y, d)$, which can be challenging by itself (Hong & Juneja, 2009). An even more difficult problem occurs when we have a budget of T experiments and wish to find the *non-myopically* optimal design that maximizes expected information gain w.r.t. all future experiments. This requires computing repeatedly nested expectations, with the depth of nesting growing at a rate of $O(T)$, and the convergence rate of evaluating this expectation being $O(n^{-\frac{1}{T+1}})$ (Rainforth et al., 2018), where n is the number of samples. Approaches to non-myopically optimal design that rely on estimating this expectation are therefore intractable.

Recently amortized methods have been introduced to solve the problem of non-myopic optimization (Foster et al., 2021; Ivanova et al., 2021). In this amortized approach a *design policy* is introduced that maps the history of designs and outcomes to the next design. A lower bound estimate of the non-myopic EIG is optimized w.r.t. the parameters of this policy, rather than optimizing the designs directly. After training, the policy achieves inference times orders of magnitude faster and EIG superior to (non-amortized) itera-

tive, myopically optimal methods. However, the approach involves backpropagation through a policy network, and hence requires that the probabilistic model be differentiable and the design space continuous. This makes these types of approaches inapplicable in problems where the design space is discrete, or where gradients of the probabilistic model are not available.

To address these shortcomings, we propose to formulate the sequential experimental design (SED) problem as a Markov decision process (MDP) (Bellman, 1957), and learn the design policy using reinforcement learning (RL). Since our RL agents exploit the Policy Gradient Theorem (Sutton et al., 1999), there is no need for the EIG objective to be differentiable. Furthermore, RL algorithms exist for optimizing agents in discrete decision spaces. Finally, modern RL techniques have many features such as stochastic policies and entropy-based regularization terms that help improve exploration of the solution space. We show that our approach, using deep RL methods, achieves state-of-the-art EIG in both continuous and discrete experimental design problems, even when the probabilistic model is a black-box, i.e. its gradients are not available. Further, we show that correct formulation of the MDP is critical to the success of our method, and naive conversions of SEDs to MDPs are significantly less performant.

2. Background

In the BOED framework, given a probabilistic model $p(y \mid \theta, d)$ and a prior distribution over the parameters $p(\theta)$ we aim to find a design d^* that maximizes an expected utility, where the expectation is taken over the joint distribution $p(y, \theta \mid d)$. It is a basic result in BOED that choosing the utility $\mathcal{U} = \log p(\theta \mid y, d) - \log p(\theta)$, leads to maximizing the EIG objective:

$$EIG(d) = \mathbb{E}_{p(y \mid d)} \left[H(p(\theta)) - H(p(\theta \mid y, d)) \right]. \tag{2}$$

Hence, solving a BOED problem maximizing utility $\mathcal{U}$ is equivalent to optimizing the EIG w.r.t. the design variables d. In general, the EIG cannot be evaluated in closed form, and must be estimated numerically. Foster et al. (2020) introduced the prior contrastive estimation (PCE) lower bound:

$$PCE(d, L) \equiv \mathbb{E}_{p(\theta_{0:L})p(y \mid \theta_0, d)} \left[\log \frac{p(y \mid \theta_0, d)}{\frac{1}{L+1} \sum_{l=0}^{L} p(y \mid \theta_l, d)} \right] \tag{3}$$

where $\theta_{0:L}$ are sampled i.i.d. from the prior $p(\theta)$ and $\theta_{1:L}$ are *contrastive samples* that re-weight the experiment outcome under alternative realizations of θ. With this estimator it is simple to optimize the design of a single experiment in isolation.

2.1. Sequential Experimental Design

The naive approach to SED is to iteratively optimize the PCE bound, conduct the experiment, update the posterior using Bayes' rule, and repeat. Let $h_t = (d_{1:t}, y_{1:t})$ be the history of experimental designs and outcomes at time t. At each iteration, the prior $p(\theta)$ is replaced by an intermediate posterior $p(\theta \mid h_t)$.

This approach has two significant drawbacks. The first is that computing posteriors is expensive in the general case, and here one must be computed at each time step. The second is that the approach is myopic: it only optimizes the immediate EIG from the next experiment, without taking into consideration how this would affect future experiments.

Suppose that rather than optimizing the designs directly, we instead optimize a policy $\pi : \mathcal{H} \to \mathcal{D}$ which maps the experimental history at time t to the next design, i.e., $d_t = \pi(h_{t-1})$. The expected total information gain of following this policy is (Foster et al., 2021):

$$EIG_T(\pi) \equiv \mathbb{E}_{p(\theta)p(h_T \mid \theta, \pi)} \left[\log \frac{p(h_T \mid \theta, \pi)}{p(h_T \mid \pi)} \right], \tag{4}$$

$$p(h_T \mid \theta, \pi) = \prod_{t=1}^{T} p(y_t \mid \theta, d_t), \tag{5}$$

where $p(h_T \mid \pi) = \mathbb{E}_{p(\theta)} p(h_T \mid \theta, \pi)$ is the marginal likelihood of the history. Note that the total EIG above does not require expectations w.r.t. the posterior over model parameters and, instead, the expectations are w.r.t. the prior $p(\theta)$. Equation (4) can be estimated using the sequential prior contrastive estimation (SPCE) lower bound:

$$\begin{aligned} sPCE(\pi, L, T) &\equiv \mathbb{E}_{p(\theta_{0:L})p(h_T \mid \theta_0, \pi)} \left[g(\theta, h_T) \right] \\ g(\theta, h_T) &= \log \frac{p(h_T \mid \theta_0, \pi)}{\frac{1}{L+1} \sum_{l=0}^{L} p(h_T \mid \theta_l, \pi)}, \end{aligned} \tag{6}$$

where the required expectations can be estimated numerically by sampling $\theta_{0:L}$ i.i.d. from $p(\theta)$ and h_T from Equation (5). A parameterized policy $\pi_{\boldsymbol{\phi}}(\cdot)$, given for example by a neural network, can then be found by maximizing Equation (6) using stochastic gradient ascent. This method of optimizing the policy w.r.t. the SPCE objective is known as deep adaptive design (DAD) (Foster et al., 2021).

DAD avoids the estimation of posteriors altogether and amortizes the cost at training time via the parameterized policy $\pi_{\boldsymbol{\phi}}$, thus achieving fast computation of designs at deployment time. However, as we shall see in the next section, we can leverage the SPCE bound above to propose a more general approach that can explore the design space more efficiently, and handle discrete designs and black-box probabilistic models. We will do this by reducing the policy optimization problem to that of solving a MDP.

2.2. Reinforcement Learning

Reinforcement learning is based on the Markov decision process (MDP) framework. An MDP is a tuple of the form $< \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \rho_0 >$ where $\mathcal{S}$ is the space of possible system states; $\mathcal{A}$ is the space of possible control actions, $\mathcal{T}$ is the transition dynamics $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{P}_s(\mathcal{S})$, giving for each state-action pair a distribution over states; $\mathcal{R}$ is a reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, giving the expected immediate reward gained for taking an action under a current state and ending up in another state; and ρ_0 is the distribution of initial states. In RL we are interested in policies $\pi : \mathcal{S} \to \mathcal{P}_a(\mathcal{A})$ that map the current state to a distribution over actions. Our goal is then to solve the MDP, i.e., to find an optimal policy π^* that maximizes the expected discounted return

$$J(\pi) \equiv \mathbb{E}_{\mathcal{T},\pi,\rho_0}\left[\sum_{t=1}^{} \gamma^{t-1}\mathcal{R}(s_{t-1}, a_{t-1}, s_t)\right], \quad (7)$$

with *discount factor* $\gamma \in [0, 1]$ and *time horizon* $T \in \mathbb{N}$.

3. RL for Sequential Experiment Design

In order to optimize SED problems with RL algorithms, we must first formulate them as MDPs that satisfy the equality

$$J(\pi) = sPCE(\pi, L, T). \quad (8)$$

The naive approach is to assign the posterior as the state $s_t = p(\theta \,|\, h_t)$, the actions as designs, and the difference of entropies as the reward $r_t = H(s_{t-1}) - H(s_t)$. Unfortunately, this formulation would require the inference of a posterior in each timestep. We can avoid this expensive repeated inference by using $g(\theta, h_t)$ from Equation (6) as our reward instead. This implies that the state must include the true value of θ as it is an input to $g(\theta, h_t)$. This is troublesome in the MDP framework since θ is not observable at test time, and is not affected by actions. Thus, we resort to the Hidden Parameter MDP (HIP-MDP) framework for our formulation.

3.1. Hidden Parameter MDPs

A HIP-MDP (Doshi-Velez & Konidaris, 2016) is a tuple $< \mathcal{S}, \mathcal{A}, \Theta, \mathcal{T}, \mathcal{R}, \gamma, \rho_0, P_\Theta >$ where $\mathcal{S}, \mathcal{A}$ are the same as in a traditional MDP but we add a parameter space Θ that parameterizes the reward and transition functions, meaning $r_t = \mathcal{R}(s_{t-1}, a_{t-1}, s_t, \theta)$ and $s_t \sim \mathcal{T}(s_t \,|\, s_{t-1}, a_{t-1}, \theta)$. The realization θ is drawn once at the start of an episode from the prior P_Θ and is fixed until the end of the episode. Note the overlap in notation with θ from our probabilistic model is intentional, as both terms refer to the same variables. The expected return becomes:

$$J(\pi) \equiv \mathbb{E}_{\mathcal{T},\pi,\rho_0,P_\Theta}\left[\sum_{t=1}^{} \gamma^{t-1}\mathcal{R}(s_{t-1}, a_{t-1}, s_t, \theta)\right]. \quad (9)$$

This leads to a formulation of SED as a HIP-MDP where $P_\Theta = p(\theta)$, states are histories, actions are designs, and the reward and transition functions are $g(\cdot, \cdot)$ and the likelihood model, parameterised by θ. However, we cannot simply set $r_t = g(\theta, h_t)$ because this would count the contribution of the t^{th} experiment $T - t$ times. The naive solution is to set $r_t = 0 \; \forall t < T$ and compute $g(\cdot, \cdot)$ only for the terminal reward. Theorem 1 establishes that the solution to such a HIP-MDP also optimizes the corresponding SED problem.

Theorem 1. *Let $\mathcal{M}$ be a* HIP-MDP *where $s_t = h_t; a_{t-1} = d_t \; \forall t \in [1, T]$ and initial state distribution and transition dynamics are $\rho_0 = h_0 \sim \delta(\emptyset); P_\Theta = \theta_{0:L} \sim p(\theta)$ and $\mathcal{T} = p(y_t \,|\, h_{t-1}, d_t, \theta_0)$ respectively. If $\gamma = 1$ and the reward function is:*

$$\mathcal{R}(s_{t-1}, a_{t-1}, s_t, \theta) = \begin{cases} 0, & \text{if } t < T \\ g(\theta, h_t), & \text{if } t = T \end{cases} \quad (10)$$

then the expected return satisfies:

$$J(\pi) = sPCE(\pi, L, T). \quad (11)$$

For proof cf. Appendix A.1. We are therefore free to use any RL algorithm of our choosing to learn a policy that will optimize the return, and hence the SPCE.

3.2. The SED MDP

While the above treatment is sufficient for optimizing SED problems with RL algorithms, it still has a few shortcomings: the state is not truly Markovian as it contains the entire history, and the reward signal is sparse, which is generally known to hinder performance in RL settings. In this section we will take a closer look at the state and reward definition, in order to formulate a more precise HIP-MDP that addresses these problems. We turn first to the issue of the Markov property.

Compactly representing the history is a well-studied problem in partially observable MDPs. A common approach is to learn such representations from data (Gregor et al., 2019). Consider the permutation invariant representation proposed by Foster et al. (2021):

$$B_{\psi,t} \equiv \sum_{k=1}^{} ENC_\psi(d_k, y_k), \quad (12)$$

where ENC_ψ is an encoder network with parameters ψ. Although not originally intended to do so, this history summary induces a Markovian structure. Since it can be decomposed as $B_{\psi,t} = B_{\psi,t-1} + ENC_\psi(d_t, y_t)$, we do not need the whole history h_t to compute the next summary $B_{\psi,t+1}$. Rather, it is sufficient to store the most recent summary $B_{\psi,t}$. Note that at training time we will still need to keep entire histories in the replay buffer in order to learn ψ.

The history summary suf ces as an input for the policy, but rewards depend on $g(\cdot; h_t)$, which cannot be computed from $B_{;t}$. To address both the issues of the Markov property and of sparsity, we propose the following reward function, which estimates the marginal contribution of the t^{th} experiment to the cumulative EIG:

$$R(s_{t-1}, a_{t-1}, s_t; \cdot) = \log p(y_t \mid \theta; d_t) - \log(C_t \cdot 1) + \log(C_{t-1} \cdot 1), \tag{13}$$

$$\text{where } C_t = \left[\prod_{k=1}^{t} p(y_k \mid \theta; d_k) \right]_{l=0}^{L} \tag{14}$$

is a vector of history likelihoods, such that each element $c_{t;l}$ is the likelihood of observing the history h_t if the true model parameters were θ_l. We use $\cdot$ to denote a vector of ones of the appropriate length.

Our proposed reward has several elegant properties: first, it is in general nonzero for all timesteps, providing a dense learning signal for the RL agent. Second, the reward function assigns each experiment its immediate contribution to the final EIG. Therefore the sum of rewards over an entire trajectory (or any prefix of a trajectory) is exactly $g(\cdot; h_t)$ from Equation (6) for that trajectory (or for the prefix). In the special case of $t = 1$, the reward is exactly the non-sequential PCE. Third, we do not need to keep the entire history in memory in order to compute the reward—it is sufficient to store the most recent history likelihood C_t and experiment outcome y_t. This is because the history likelihood decomposes as $C_t = C_{t-1} \odot [p(y_t \mid \theta_l; d_t)]_{l=0}^{L}$, where $\odot$ denotes the Hadamard (or elementwise) product.

Note that the above formulation means we need to store different history representations for the policy and the reward function, and the state is thus a concatenation of both. The transition dynamics involve sampling the outcome of an experiment from the model, and updating the history summaries B and history likelihoods C accordingly:

$$\begin{aligned} y_t &\sim p(y_t \mid d_t; \theta), \\ B_{;t} &= B_{;t-1} + \text{ENC}(d_t, y_t) \text{ and} \\ C_t &= C_{t-1} \odot [p(y_t \mid \theta_l; d_t)]_{l=0}^{L}. \end{aligned} \tag{15}$$

Putting it all together, we have the following HIP-MDP:

- $\mathcal{S}$: the current experiment outcome y_t and the history summaries and likelihoods used by π and R, respectively. $s_t = (B_{;t}, C_t, y_t)\ \forall t \in [0, T]$.
- $\mathcal{A}$: is the design space with $d_{t-1} = d_t\ \forall t \in [1, T]$.
- R: the reward function of Equation (13).
- T: the transition dynamics of Equation (15).
- ρ_0: the initial history is always the empty set, thus $h_0 = (B_{;0}, C_0, y_0) = (\emptyset, 1, \cdot)$.
- Θ: is the space of model parameters and $\theta \sim p(\theta)$.
- γ: at time t, a policy $\pi(a_t \mid s_t)$ maps $B_{;t}$ to a distribution over designs d_{t+1}. Therefore, unlike Foster et al. (2021), our policy is stochastic.

Theorem 2 establishes that the solution to the above HIP-MDP also optimizes the corresponding SED problem:

Theorem 2. Let $\mathcal{M}$ be a HIP-MDP where $s_t = (B_{;t}, C_t, y_t)$; $a_{t-1} = d_t\ \forall t \in [1, T]$ and initial state distribution is $\rho_0 = (\emptyset, 1, \cdot)$; $P = \prod_{0:L} p(\theta)$ and reward and transition functions follow Equations (13) and (15), respectively. If $\gamma = 1$ then the expected return satisfies:

$$J(\pi) = \text{sPCE}(\pi; L, T). \tag{16}$$

Proof. Here we will provide a sketch of proof. For the full proof, see Appendix A.2. The proof relies on showing that the terms on both sides of Equation (16) are equal given the conditions of the theorem. As each side is an expectation, we show that the distributions on both sides are identical, and that the terms inside the expectation are equal for any realisation of the random variables. □

3.3. Advantages of the RL Formulation

Armed with this recipe for converting SED problems into MDPs, we can now deploy RL algorithms of our choosing to learn design policies off-line, and then use the policies for rapid on-line experiment design.

3.3.1. Discrete-Design Spaces and Black-Box Likelihoods

Learning design policies in this way has several advantages over the DAD approach. First, DAD is incapable of handling problems with discrete design spaces, a limitation explicitly acknowledged by Foster et al. (2021). In contrast, there are many RL algorithms that can efficiently solve MDPs with discrete action spaces. Another advantage of our method is that RL algorithms do not need the reward function to be differentiable, whereas DAD must have access to the gradients of the sPCE (w.r.t. policy parameters) in order to optimize the design policy. Hence, in cases where the likelihood is available but is not differentiable w.r.t. the design (for example, if the probabilistic model contains an argmax), RL can be used but DAD cannot.

3.3.2. Exploration vs Exploitation

Finally, the trade-off between exploration and exploitation is a significant and long-standing challenge in reinforcement learning (Sutton & Barto, 2018; Weng, 2020) as well as in

experiment design (Robbins, 1952). At any point in time, an agent must choose either to take an action that is optimal w.r.t. the expected return given current knowledge (exploitation), or an action that will lead to better knowledge and thus better returns in the future (exploration). Modern RL algorithms have many features that help improve exploration, such as random noise in the policy, regularization terms, and so forth (Ladosz et al., 2022). DAD, lacking these features, is a pure exploitation algorithm. Consequently, it may explore the design space insufficiently and thus get trapped in a poor local optimum. While some exploration features can perhaps be incorporated into the DAD framework, it would require significant engineering and scientific effort to do so. On the other hand, high-quality implementations of these features are already available in the RL framework, at no additional cost to practitioners.

4. Experimental Results

In this section we compare our approach to several baselines in order to empirically establish its theoretical benefits. We will examine experimental design problems with both continuous and discrete design spaces, as well as the case where the likelihood model is not differentiable. Our method[1], which we term the RL approach, is implemented in the Pyro (Bingham et al., 2018) and Garage (Garage Contributors, 2019) frameworks. We used the Randomized ensembled double q-learning (REDQ) (Chen et al., 2021) algorithm to train all of our policies. As an ablation, we include a version of our approach based on Theorem 1, denoted as naive reinforcement learning (NAIVE-RL). This ablation shares the permutation-invariant policy of the full RL approach, so that it evaluates only the effect of our original reward formulation and not the effect of the policy architecture we inherit from Foster et al. (2021).

4.1. Continuous Design Space

We begin with an examination of continuous design problems. This is the only scenario in which the DAD baseline is applicable. Additionally, we have the variational prior contrastive estimation (VPCE) baseline, which involves iteratively optimizing the myopic PCE bound by gradient descent w.r.t. the next design, then estimating a posterior with variational inference (Foster et al., 2020). Finally, a random policy is included to provide a naive baseline.

4.1.1. SOURCE LOCATION FINDING

The first problem we consider is source location, previously studied by Foster et al. (2021). In this problem there are 2 signal sources positioned in 2-dimensional space. The strength of the signal decays with distance from its respective sources, and the signals of the different sources combine additively. The signal can be measured with some observation noise at arbitrary points in the plane, and the design is the 2-d coordinates at which to take this sample. For full details see Appendix B.1.

Table 1. Lower and upper bounds on EIG at $t = 30$ for the source location problem, computed using SPCE and SNMC with $L = 1e6$ respectively. Means and standard errors from 2000 (RL and DAD) or 1000 (VPCE and random) rollouts.

METHOD	LOWER BOUND		UPPER BOUND	
RANDOM	1.624	0.053	1.639	0.057
VPCE	7.766	0.069	7.802	0.072
DAD	10.965	0.041	12.380	0.086
RL	11.73	0.040	12.362	0.062
NAIVE RL	9.789	0.045	9.898	0.049

Figure 1 shows the expected information gain of the various methods for an experimental budget of 30 experiments, where EIG was estimated using the SPCE lower bound with $1e6$ contrastive samples. Trendlines are the means and shaded regions are the standard errors aggregated from 2000 rollouts of RL and DAD or 1000 rollouts for VPCE and random. The standard errors are very small due to the large number of rollouts. Since RL is notorious for being highly sensitive to the random seed, we split the 2000 rollouts of RL among 10 agents each trained with its own random seed.

The RL method outperforms all baselines by a considerable margin, and the effect size is several times the magnitude of the standard error. Interestingly, DAD underperforms the myopic methods in the first few experiments before overtaking them. This is unsurprising, as DAD prioritizes high EIG at $t = 30$ at the expense of EIG at any timestep $t < 30$. What is notable is that the RL method, which similarly prioritizes long-term over short-term gain, is nonetheless able to achieve better EIG than the other baselines at every timestep. This may be due to the superior exploration capabilities that RL provides.

Table 1 focuses on the EIG at $t = 30$, including both lower and upper bound estimates. The upper bound is based on the sequential nested monte carlo (SNMC) estimator (Foster et al., 2021). These results emphasize the significance of the gap between RL and the baselines. The improvement in RL's lower bound over DAD's is an order of magnitude bigger than the standard error, and the improvement over the non-amortized methods is even bigger. In terms of the upper bound, RL performs comparably to DAD and significantly better than all other baselines. It is important to note, however, that both amortized methods are trained to maximize the lower bound, not the upper bound.

[1] See https://github.com/csiro-mlai/RL-BOED for source code.

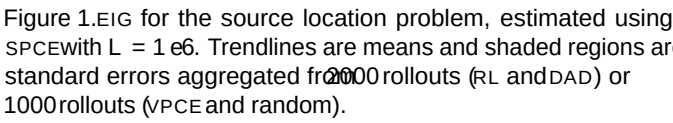

Figure 1. EIG for the source location problem, estimated using SPCE with $L = 1e6$. Trendlines are means and shaded regions are standard errors aggregated from 2000 rollouts (RL and DAD) or 1000 rollouts (VPCE and random).

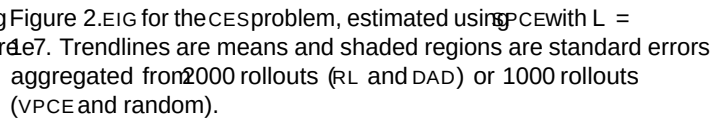

Figure 2. EIG for the CES problem, estimated using SPCE with $L = 1e7$. Trendlines are means and shaded regions are standard errors aggregated from 2000 rollouts (RL and DAD) or 1000 rollouts (VPCE and random).

4.1.2. Constant Elasticity of Substitution

The next problem we examine is the constant elasticity of substitution (CES), previously studied by Foster et al. (2020). This is a behavioral economics problem in which a participant compares 2 baskets of goods and rates the subjective difference in utility between the baskets on a sliding scale from 0 to 1. A probabilistic model with latent variables (α, β, u) represents the response function, and the goal is to design pairs of baskets in order to infer the value of the latent variables. Each basket consists of 3 constrained values, so that the design space is 6-dimensional. For full details see Appendix B.2.

In Figure 2 we see the EIG for our proposed RL method and the baselines over the course of 10 experiments. EIG was estimated using SPCE with $1e7$ contrastive samples, and again we show means and standard errors. As in the source location case, the RL approach outperforms all baselines, and both amortized methods outperform the non-amortized myopic methods. However, a notable difference is that in the CES problem the performance of DAD begins to converge much earlier in the sequence of experiments than it does. This is likely to be an exploration issue, which arises from the nature of the CES problem. The experiment outcome y is the result of a censored sigmoid function, and as a consequence of this, much of the design space maps to an outcome on the boundary of the observation space. This leads to a challenging exploration problem, as the surface of the objective function has many suboptimal local maxima in which to be trapped. The RL algorithm has a stochastic policy with Gaussian noise that diminishes as performance improves, and this allows it to explore the design space more fully and find better maxima. The DAD algorithm, on the other hand, uses a deterministic policy and struggles to explore regions of design space far from where that policy was initialized.

To illustrate the exploratory behavior of both algorithms, we plot the distributions over designs proposed by the trained policies of both methods. As can be seen in Figure 3, the DAD policies only propose designs in a narrow range of the design space, relatively near to its center. Consequently, for some samples of $p(\theta)$ the DAD policies will not be able to propose effective designs, which limits their EIG. The RL policies, on the other hand, are capable of proposing designs from all over the space. Because the RL algorithm has good exploration, it encountered situations in training where diverse designs have lead to improved outcomes. Consequently, the learned policies propose such designs when needed.

Table 2 shows the performance advantage of the RL approach even more clearly than Figure 2. At $t = 10$, the lower bound of EIG for RL is greater than the upper bound for DAD, and this difference is several times greater than the standard errors. The gap between RL and the myopic baselines is even bigger.

4.2. Discrete Design Space

We now turn to an examination of experimental design problems where the design space is discrete. As mentioned previously, DAD is inapplicable in this situation. The VPCE baseline also needs to be modified: instead of using gradient descent to optimize the (non-sequential) PCE estimator, we compute it for every possible design and take the argmax.

is further emphasized by Table 3, which lists the lower and upper bound estimates for each method at $t = 10$. The gap in the lower bound between SMC and RL is smaller than the standard error, in spite of the large number of replicates. It is possible that with additional replication a statistically significant difference would emerge, but the relative effect size would still be on the order of 1%.

Figure 3. Distributions of designs proposed by trained RL (green boxes) and DAD (blue boxes) policies for the CES problem. One box plot for each of the 6 elements of the design vector. Data collected from 2000 rollouts. Boxes indicate the interquartile range, with a notch at the median. Whiskers extend up to 1.5 times the interquartile range.

Table 2. Lower and upper bounds on EIG at $t = 10$ for the CES problem, computed using SPCE and SNMC with $L = 1e7$ respectively. Means and standard errors from 2000 (RL and DAD) or 1000 (VPCE and random) rollouts.

METHOD	LOWER BOUND	UPPER BOUND
RANDOM	8.099 0.153	16.451 0.685
VPCE	9.547 0.137	24.396 2.024
DAD	10.774 0.077	13.374 0.150
RL	13.965 0.064	17.794 0.226
NAIVE RL	12.131 0.058	15.641 0.166

Note that this modification is tractable because the VPCE baseline is myopic, and we only need to compute $|A|$ designs. A similar adjustment to DAD is not tractable, as we would need to compute the SPCE for $|A|^T$ designs.

We consider a prey population problem, where we control the initial population of a prey species and measure the number of individuals that were consumed by predators after a period of 24 hours. The goal is to estimate the attack rate and handling time of the predator species. For full details see Appendix B.3. This setting was previously studied by Moffat et al. (2020), and we include their sequential Monte Carlo (SMC) method as a baseline.

EIG for this problem, given a budget of 10 experiments, is shown in Figure 4. All 3 methods outperform the naive random baseline. The RL method achieves similar results to the SMC and VPCE baselines. SMC has a small performance advantage, but the difference is within the error bars. This

Figure 4. EIG for the prey population problem, estimated using SPCE with $L = 1e6$. Trendlines are means and shaded regions are standard errors aggregated from 2000 rollouts (RL), 1000 rollouts (VPCE and random) or 500 rollouts (SMC).

It is noteworthy that the RL method does not outperform the myopic baselines, since at least in theory it has the capability to find non-myopic solutions. There are a number of possible explanations for this result. The first is that the myopic baselines compute an explicit posterior after each experiment, and as such have a more accurate basis to perform each optimization than the RL method, which must implicitly encapsulate this information in its policy and critic networks. It is possible that the state representation being learned by the RL agent is degrading its overall performance, and that better representation learning would lead to improvement over the myopic baselines. Another possibility is that the optimality gap between the myopically optimal solution and the non-myopically optimal solution is small for the prey population problem. It is impossible to know for sure if this is the case as no theoretical bounds are known for either type of optimal solution.

4.3. Computational Benefits of the RL Appproach at Deployment

Ultimately, RL can achieve comparable EIG to state-of-the-art methods which compute explicit posteriors, and require considerably more computation time at deployment. Table 4 shows the time in seconds that each method needs to compute a single proposed design. For the prey population

Table 3. Lower and upper bounds on EIG at $t = 10$ for the prey population problem, computed using SPCE and SNMC with $L = 1e6$ respectively. Means and standard errors from 2000 rollouts (RL), 1000 rollouts (VPCE and random) or 500 rollouts (SMC).

METHOD	LOWER BOUND	UPPER BOUND
RANDOM	3.923 0.042	3.925 0.043
VPCE	4.396 0.046	4.42 0.050
SMC	4.521 0.065	4.523 0.063
RL	4.456 0.032	4.459 0.033
NAIVE RL	4.375 0.032	4.376 0.032

Table 4. Deployment time in seconds by method and problem. Means and standard errors computed from 100 replications.

PROBLEM	METHOD	DEPLOYMENT TIME (S)
CES	RANDOM	2.37E-5 1.51E-7
	VPCE	146.944 1.397
	PCE-BO	23.830 0.771
	DAD	1.25E-4 3.37E-7
	RL	1.35E-3 4.18E-6
PREY POPULATION	RANDOM	1.29E-4 4.94E-7
	VPCE	20.550 1.893
	SMC	81.252 4.310
	RL	1.50E-3 3.23E-6

problem, the trained RL agents propose designs orders of magnitude faster than the SMC and VPCE baselines. Indeed, the RL agents are close to the deployment time of the random baseline, which is so fast and trivial that no algorithm can be reasonably expected to beat it. In the context of continuous design spaces, RL achieves comparable deployment time to DAD, in the sense that the deployment time of both is imperceptible to humans. This is unsurprising, as both approaches compute the design by means of a forward pass through a neural network. RL is slightly slower because it must also sample from a probability distribution.

4.4. Non-differentiable Likelihood

One of the claimed benefits of the RL approach is that it can learn design policies even without access to gradients of the likelihood. To demonstrate this, we re-run the CES problem with a pytorch `no_grad` context around the likelihood function. This means that gradients were not computed for the likelihood model.

The results of this experiment are shown in Figure 5. Note that the DAD and VPCE baselines are absent from the results, as both of these methods do not function in the absence of likelihood model gradients. Indeed, trying to run them with the `no_grad` statement results in an error as pytorch tries to perform automatic differentiation without a gradient. VPCE has been replaced with a baseline that uses Bayesian optimization to maximize the myopic PCE bound (PCE-BO). No such replacement has been made for DAD, as this would involve high-dimensional Bayesian optimization (BO) of the policy parameters, which is a challenging problem (Greenhill et al., 2020).

The performance of the RL agents is nearly identical to what is seen in Figure 2. This is because we used the same random seeds, and because RL does not rely on gradients from the likelihood model. RL outperforms all of the gradient-free baselines by a considerable margin, starting from the very first experiment, and is orders of magnitude faster at deployment time as per Table 4.

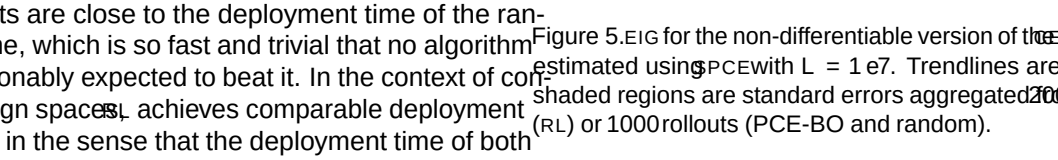

Figure 5. EIG for the non-differentiable version of the CES problem, estimated using SPCE with $L = 1e7$. Trendlines are means and shaded regions are standard errors aggregated from 2000 rollouts (RL) or 1000 rollouts (PCE-BO and random).

5. Related Work

The classic approach to Bayesian optimal experimental design is to iteratively optimize the immediate EIG, run the experiment, and infer a posterior. The EIG has been estimated by particle filters (Cavagnaro et al., 2010), sequential Monte Carlo (Drovandi et al., 2014; Moffat et al., 2020), nested Monte Carlo (Myung et al., 2013), multilevel Monte Carlo (Goda et al., 2020), Markov Chain Monte Carlo (Müller et al., 2004), ratio estimation (Kleinegesse & Gutmann, 2019), variational bounds (Foster et al., 2019; 2020), neural estimation (Kleinegesse & Gutmann, 2020), Laplace importance sampling (Beck et al., 2018) and more. Methods for specific models have been developed that exploit unique properties, such as in the case of linear models (Verdinelli, 1996), Gaussian Process models (Pandita et al., 2021), and polynomial models (Rainforth et al., 2018).

Posterior inference has been done by sequential Monte

Carlo (Moffat et al., 2020; Kleinegesse et al., 2021), Markov Chain Monte Carlo (Myung & Pitt, 2009; Huan & Marzouk, 2013), and variational inference (Foster et al., 2020). More recently, alternatives have been proposed that avoid explicit computation of a posterior. These include adversarial methods (Prangle et al., 2019; Overstall, 2022), and amortized methods (Foster et al., 2021; Ivanova et al., 2021).

Deep connections exist between the fields of experimental design and active learning. Gal et al. (2017) introduced the DBALD estimator for information gain, which is in fact a special case of PCE (Foster, 2022). Non-myopic methods have also been developed in the classification context, using n-step look-ahead (Zhao et al., 2021b) or compressing the joint of multiple experiments into a one-dimensional integral (Zhao et al., 2021a).

The principles of optimal experimental design have also been applied in the context of reinforcement learning. Sun et al. (2011) derived "curiosity Q-learning" based on the information gain quantity, but the resulting algorithm is only tractable in the case of tabular RL, where the spaces of observations and actions are finite. Houthooft et al. (2016) proposed to maximize the variational information of a predictive model as an auxiliary objective to the typical reward function. Shyam et al. (2019) explicitly use the information gain of an ensemble of world models to optimize an exploration policy in each iteration, which is then used to guide the next action in the real world. Finally, the field of active inference (Friston et al., 2009) seeks to solve RL problems by minimizing the variational free energy, which is equivalent to maximizing the expected information gain of the posterior.

In contrast, this work seeks to go in the reverse direction and apply RL to solve experimental design problems. To the best of our knowledge, this is the first work that makes generic RL algorithms applicable for optimal design of experiments. The work that comes nearest is Huan & Marzouk (2016); Shen & Huan (2021), which proposes a specific Dynamic Programming algorithm and requires the inference of explicit posteriors, approximated on a discretized grid.

6. Discussion

In this work we proposed a method for transforming sequential experimental design (SED) problems into Markov decision processes (MDPs) such that each pair of problems are solved by the same optimal policy. This allowed us to learn design policies using off-the-shelf reinforcement learning (RL) algorithms, resulting in an amortized algorithm in the spirit of deep adaptive design (DAD). Our approach improves on the state-of-the-art in several important ways.

First, in the case where design spaces are continuous, our approach significantly outperformed all baselines, including both conventional methods that do not use amortization as well as the amortized DAD. Our analyses indicate that this is due to the superior ability of RL to explore the space of possible designs and thus discover better solutions. Indeed, investigation of the designs proposed by trained policies confirms that RL policies have learned to propose much more diverse designs than the DAD policies. However, the importance of exploration will vary between different SED problems. Second, in the case where design spaces are discrete, our method achieves comparable performance to the best non-amortized baselines. The amortized baselines simply do not work in this setting, because they depend on backpropagating gradients through the design. Finally, in the case where the probabilistic model is not differentiable, the RL approach achieves the best performance by far, whereas many of the baselines are inapplicable as they rely on gradient information which is not available. Thus our proposed method is applicable to a broad spectrum of design problems, is able to match or outperform the state-of-the-art in all of them, and still enjoys the fast deployment times of amortized methods that are restricted to continuous design spaces with differentiable probabilistic models.

Of note, our results cannot be achieved by naively using the SPCE objective as a reward function. Ablations show that correct factorization of the rewards is critical to the performance of the resulting RL policies.

Beyond the promising experimental results, it is important to note that our method opens the door to deploying existing algorithms on sequential experimental design problems. The RL literature is rich with techniques for handling problems with unique characteristics that make policy learning difficult, e.g. sparse rewards (Zhang et al., 2021) or high dimensionality (Laskin et al., 2020). Many of these techniques may prove to be useful for specific experimental design problems. With our approach, they can be applied in a straightforward manner by smartly transforming the design problem to an MDP.

References

Beck, J., Dia, B. M., Espath, L. F., Long, Q., and Tempone, R. Fast bayesian experimental design: Laplace-based importance sampling for the expected information gain. Computer Methods in Applied Mechanics and Engineering, 334:523–553, 2018.

Bellman, R. A markovian decision process. Journal of mathematics and mechanics, pp. 679–684, 1957.

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.

Cavagnaro, D. R., Myung, J. I., Pitt, M. A., and Kujala, J. V. Adaptive design optimization: A mutual information-based approach to model discrimination in cognitive science. Neural computation, 22(4):887–905, 2010.

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

Chen, X., Wang, C., Zhou, Z., and Ross, K. W. Randomized ensembled double q-learning: Learning fast without a model. In International Conference on Learning Representations, 2021.

Doshi-Velez, F. and Konidaris, G. Hidden parameter markov decision processes: A semiparametric regression approach for discovering latent task parametrizations. In International Joint Conference on Arti cial Intelligence, 2016.

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.

Foster, A. Variational, Monte Carlo and Policy-Based Approaches to Bayesian Experimental Design. PhD thesis, University of Oxford, 2022.

Foster, A., Jankowiak, M., Bingham, E., Horsfall, P., Teh, Y. W., Rainforth, T., and Goodman, N. Variational bayesian optimal experimental design. Advances in Neural Information Processing Systems, pp. 14036–14047, 2019.

Foster, A., Jankowiak, M., O'Meara, M., Teh, Y. W., and Rainforth, T. A uni ed stochastic gradient approach to designing bayesian-optimal experiments. In International Conference on Arti cial Intelligence and Statistics, pp. 2959–2969. PMLR, 2020.

Foster, A., Ivanova, D. R., Malik, I., and Rainforth, T. Deep adaptive design: Amortizing sequential bayesian experimental design. International Conference on Machine Learning, 2021.

Friston, K. J., Daunizeau, J., and Kiebel, S. J. Reinforcement learning or active inference? PloS one, 4(7):e6421, 2009.

Gal, Y., Islam, R., and Ghahramani, Z. Deep bayesian active learning with image data. In International Conference on Machine Learning, 2017.

Garage Contributors. Garage: A toolkit for reproducible reinforcement learning research. https://github.com/rlworkgroup/garage, 2019.

Goda, T., Hironaka, T., and Iwamoto, T. Multilevel monte carlo estimation of expected information gains. Stochastic Analysis and Applications, 38(4):581–600, 2020.

Greenhill, S., Rana, S., Gupta, S., Vellanki, P., and Venkatesh, S. Bayesian optimization for adaptive experimental design: A review. IEEE access, 8:13937–13948, 2020.

Gregor, K., Jimenez Rezende, D., Besse, F., Wu, Y., Merzic, H., and van den Oord, A. Shaping belief states with generative environment models for rl. Advances in Neural Information Processing Systems, 2019.

Guez, A. Sample-based search methods for Bayes-adaptive planning. PhD thesis, UCL (University College London), 2015.

Hong, L. J. and Juneja, S. Estimating the mean of a non-linear function of conditional expectation. In Winter Simulation Conference. IEEE, 2009.

Houthooft, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., and Abbeel, P. Vime: Variational information maximizing exploration. In Advances in Neural Information Processing Systems, 2016.

Huan, X. and Marzouk, Y. M. Simulation-based optimal bayesian experimental design for nonlinear systems. Journal of Computational Physics, 232(1):288–317, 2013.

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

Ivanova, D., Foster, A., Kleinegesse, S., Gutmann, M. U., and Rainforth, T. Implicit deep adaptive design: Policy-based experimental design without likelihoods. Advances in Neural Information Processing Systems, 2021.

Kleinegesse, S. and Gutmann, M. U. Ef cient bayesian experimental design for implicit models. In The 22nd International Conference on Arti cial Intelligence and Statistics, pp. 476–485, 2019.

Kleinegesse, S. and Gutmann, M. U. Bayesian experimental design for implicit models by mutual information neural estimation. In International Conference on Machine Learning, pp. 5316–5326. PMLR, 2020.

Kleinegesse, S., Drovandi, C., and Gutmann, M. U. Sequential bayesian experimental design for implicit models via mutual information. Bayesian Analysis, 1(1):1–30, 2021.

Ladosz, P., Weng, L., Kim, M., and Oh, H. Exploration in deep reinforcement learning: A survey. Information Fusion, 2022.

Laskin, M., Srinivas, A., and Abbeel, P. Curl: Contrastive unsupervised representations for reinforcement learning. In International Conference on Machine Learning, pp. 5639–5650. PMLR, 2020.

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

Moffat, H., Hainy, M., Papanikolaou, N. E., and Drovandi, C. Sequential experimental design for predator–prey functional response experiments. Journal of the Royal Society Interface, 17(166), 2020.

Müller, P., Sansó, B., and De Iorio, M. Optimal bayesian design by inhomogeneous markov chain simulation. Journal of the American Statistical Association, 99(467):788–798, 2004.

Myung, J. I. and Pitt, M. A. Optimal experimental design for model discrimination. Psychological review, 116(3): 499, 2009.

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.

Overstall, A. M. Properties of fisher information gain for bayesian design of experiments. Journal of Statistical Planning and Inference, 218:138–146, 2022.

Pandita, P., Tsilifis, P., Awalgaonkar, N. M., Bilionis, I., and Panchal, J. Surrogate-based sequential bayesian experimental design using non-stationary gaussian processes. Computer Methods in Applied Mechanics and Engineering, 385:114007, 2021.

Prangle, D., Harbisher, S., and Gillespie, C. S. Bayesian experimental design without posterior calculations: an adversarial approach. arXiv preprint arXiv:1904.05703, 2019.

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

Robbins, H. Some aspects of the sequential design of experiments. Bulletin of the American Mathematical Society, 58(5):527–535, 1952.

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. doi: 10.1111/insr.12107. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/insr.12107.

Shen, W. and Huan, X. Bayesian sequential optimal experimental design for nonlinear models using policy gradient reinforcement learning. arXiv preprint arXiv:2110.15335, 2021.

Shyam, P., Jaśkowski, W., and Gomez, F. Model-based active exploration. In International Conference on Machine Learning, 2019.

Sun, Y., Gomez, F., and Schmidhuber, J. Planning to be surprised: Optimal bayesian exploration in dynamic environments. In International Conference on Artificial General Intelligence, 2011.

Sutton, R. S. and Barto, A. G. Reinforcement learning: An introduction. MIT press, 2018.

Sutton, R. S., McAllester, D., Singh, S., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. Advances in neural information processing systems, 1999.

Verdinelli, I. Bayesian design of experiments for the linear model. PhD thesis, Carnegie Mellon University, 1996.

Weng, L. Exploration strategies in deep reinforcement learning. lilianweng.github.io/lil-log, 2020. URL https://lilianweng.github.io/lil-log/2020/06/07/exploration-strategies-in-deep-reinforcement-learning.html.

Zhang, T., Rashidinejad, P., Jiao, J., Tian, Y., Gonzalez, J. E., and Russell, S. Made: Exploration via maximizing deviation from explored regions. Advances in Neural Information Processing Systems, 34, 2021.

Zhao, G., Dougherty, E., Yoon, B.-J., Alexander, F., and Qian, X. Efficient active learning for gaussian process classification by error reduction. Advances in Neural Information Processing Systems, 34, 2021a.

Zhao, G., Dougherty, E., Yoon, B.-J., Alexander, F., and Qian, X. Uncertainty-aware active learning for optimal bayesian classifier. In International Conference on Learning Representations, 2021b.

A. Theorem Proofs

In this appendix we provide proofs for the theorems in our paper.

A.1. Proof of Theorem 1

Here we prove our first theorem, showing that even a naive formulation allows us to construct an equivalent HIP-MDP to any sequential experiment optimization problem.

Theorem 1. Let M be a HIP-MDP where $s_t = h_t; a_{t-1} = d_t \; \forall t \in [1; T]$ and initial state distribution and transition dynamics are $_0 = h_0 \quad (;)$; $P = \prod_{0:L} p(\,)$ and $T = p(y_t \mid h_{t-1}; d_t; \,_0)$ respectively. If $= 1$ and the reward function is:

$$R(s_{t-1}; a_{t-1}; s_t; \,) = \begin{cases} 0; & \text{if } t < T \\ g(\,; h_t); & \text{if } t = T \end{cases} \tag{17}$$

then the expected return satisfies:

$$J(\,) = \text{sPCE}(\,; L; T): \tag{18}$$

Proof. We need to show that $J(\,) = \text{sPCE}(\,; L; T)$ under the conditions of the theorem. Plugging in the definitions of return and sPCE, this means showing that

$$E_{T;\,;\,_0;P}\left[\sum_{t=1}^{T} \,^{t-1} R(s_{t-1}; a_{t-1}; s_t; \,)\right] = E_{p(\,_0;h_T \mid \,)p(\,_{1:L})}\left[\log \frac{p(h_T \mid \,_0; \,)}{\frac{1}{L+1}\sum_{l=0}^{L} p(h_T \mid \,_l; \,)}\right]: \tag{19}$$

Plugging in the definitions of T; ; $_0$; P, this is equivalent to

$$E_{\prod_{t=1}^{T}[p(y_t \mid h_{t-1}; d_t; \,_0)\;(d_t \mid h_{t-1})]\,p(\,_{0:L})}\left[\sum_{t=1}^{T} \,^{t-1} R(s_{t-1}; a_{t-1}; s_t; \,)\right] = E_{p(\,_0;h_T \mid \,)p(\,_{1:L})}\left[\log \frac{p(h_T \mid \,_0; \,)}{\frac{1}{L+1}\sum_{l=0}^{L} p(h_T \mid \,_l; \,)}\right] \tag{20}$$

$$E_{p(h_T \mid \,_0; \,)p(\,_{0:L})}\left[\sum_{t=1}^{T} \,^{t-1} R(s_{t-1}; a_{t-1}; s_t; \,)\right] = E_{p(\,_0;h_T \mid \,)p(\,_{1:L})}[g(\,; h_T)] \tag{21}$$

$$E_{p(\,_0;h_T \mid \,)p(\,_{1:L})}\left[\sum_{t=1}^{T} \,^{t-1} R(s_{t-1}; a_{t-1}; s_t; \,)\right] = E_{p(\,_0;h_T \mid \,)p(\,_{1:L})}[g(\,; h_T)]: \tag{22}$$

Note that the distributions on both sides are identical. Therefore, if the term inside the expectation on the LHS is equal to the term inside the expectation on the RHS for all possible realizations of h_T; and , then the expectations must be equal. Thus we seek to show that:

$$\sum_{t=1}^{T} \,^{t-1} R(s_{t-1}; a_{t-1}; s_t; \,) = g(\,; h_T): \tag{23}$$

By the conditions of the theorem $= 1$ so this simplifies to:

$$\sum_{t=1}^{T} R(s_{t-1}; a_{t-1}; s_t; \,) = g(\,; h_T): \tag{24}$$

Plugging in the definition of the reward function:

$$0 + 0 + \quad + g(\,; h_T) = g(\,; h_T); \tag{25}$$

which is trivially true. Therefore $J(\,) = \text{sPCE}(\,; L; T)$. □

A.2. Proof of Theorem 2

In this appendix we prove the main theorem of the paper, which shows that we can construct an equivalent HIP-MDP with dense rewards to any sequential experiment optimization problem. For convenience, we restate the theorem here along with some relevant de nitions:

$$B_{;t} \quad \sum_{k=1}^{t} ENC\ (d_k; y_k) \tag{26}$$

$$C_t \quad \left[\prod_{k=1}^{t} p(y_k \mid {}_l; d_k) \right]_{l=0}^{L} \tag{27}$$

Theorem 2. Let M be a HIP-MDP where $s_t = (B_{;t}; C_t; y_t)$; $a_{t\ 1} = d_t\ 8t\ 2\ [1; T]$ and initial state distribution is ${}_0 = (0; 1; ;)$; $P = {}_{0:L}\ p(\ )$, the reward function is

$$R(s_{t\ 1}; a_{t\ 1}; s_t; \ ) = \log p(y_t \mid {}_0; d_t) \quad \log(C_t \quad 1) + \log(C_{t\ 1} \quad 1); \tag{28}$$

and transition function is

$$\begin{aligned} y_t &\quad p(y_t \mid d_t; {}_0) \\ B_{;t} &= B_{;t\ 1} + ENC\ (d_t; y_t) \\ C_t &= C_{t\ 1} \quad [p(y_t \mid {}_l; d_t)]_{l=0}^{L} : \end{aligned} \tag{29}$$

If $= 1$ then the expected return satis es:

$$J(\ ) = sPCE(\ ; L; T\ ): \tag{30}$$

Proof. We need to show that $J(\ ) = sPCE(\ ; L; T\ )$ under the conditions of the theorem. Plugging in the de nitions of return and SPCE, this means showing that

$$E_{T; ; {}_0; P}\left[\sum_{t=1}^{T} {}^{t\ 1} R(s_{t\ 1}; a_{t\ 1}; s_t; \ ) \right] = E_{p({}_0; h_T \mid)p({}_{1:L})}\left[\log \frac{p(h_T \mid {}_0; \ )}{\frac{1}{L+1} \sum_{l=0}^{L} p(h_T \mid {}_l; \ )} \right] : \tag{31}$$

Plugging in the de nitions of $T; ; {}_0; P$ we get:

$$\begin{aligned} &E_{\prod_{t=1}^{T} [p(B_{;t}; C_t; y_t \mid B_{;t\ 1}; C_{t\ 1}; y_{t\ 1}; d_t; {}_0)\ (d_t \mid B_{;t\ 1})]\, p({}_{0:L})}\left[\sum_{t=1}^{T} {}^{t\ 1} R(s_{t\ 1}; a_{t\ 1}; s_t; \ ) \right] \\ &= E_{p({}_0; h_T \mid)p({}_{1:L})}\left[\log \frac{p(h_T \mid {}_0; \ )}{\frac{1}{L+1} \sum_{l=0}^{L} p(h_T \mid {}_l; \ )} \right] \end{aligned} \tag{32}$$

$$\begin{aligned} &E_{\prod_{t=1}^{T} [p(B_{;t}; C_t \mid B_{;t\ 1}; C_{t\ 1}; d_t; y_t) p(y_t \mid d_t; {}_0)\ (d_t \mid B_{;t\ 1})]\, p({}_{0:L})}\left[\sum_{t=1}^{T} {}^{t\ 1} R(s_{t\ 1}; a_{t\ 1}; s_t; \ ) \right] \\ &= E_{p({}_0; h_T \mid)p({}_{1:L})}\left[\log \frac{p(h_T \mid {}_0; \ )}{\frac{1}{L+1} \sum_{l=0}^{L} p(h_T \mid {}_l; \ )} \right] \end{aligned} : \tag{33}$$

Noting that $B_{\cdot;t}$ and C_t can be computed exactly from h_t, we can simplify the LHS further:

$$E_{\prod_{t=1}^{T}[p(h_t \mid h_{t-1}, d_t, y_t)p(y_t \mid d_t; \theta)\,\pi(d_t \mid h_{t-1})]\,p(\theta_{0:L})}\left[\sum_{t=1}^{T} \gamma^{t-1} R(s_{t-1}, a_{t-1}, s_t; \theta)\right] \tag{34}$$

$$= E_{p(\theta_0, h_T \mid \theta)p(\theta_{1:L})}\left[\log \frac{p(h_T \mid \theta_0; \theta)}{\frac{1}{L+1}\sum_{l=0}^{L} p(h_T \mid \theta_l; \theta)}\right]$$

$$E_{p(h_T \mid \theta_0; \theta)p(\theta_{0:L})}\left[\sum_{t=1}^{T} \gamma^{t-1} R(s_{t-1}, a_{t-1}, s_t; \theta)\right] = E_{p(\theta_0, h_T \mid \theta)p(\theta_{1:L})}\left[\log \frac{p(h_T \mid \theta_0; \theta)}{\frac{1}{L+1}\sum_{l=0}^{L} p(h_T \mid \theta_l; \theta)}\right] \tag{35}$$

$$E_{p(\theta_0, h_T \mid \theta)p(\theta_{1:L})}\left[\sum_{t=1}^{T} \gamma^{t-1} R(s_{t-1}, a_{t-1}, s_t; \theta)\right] = E_{p(\theta_0, h_T \mid \theta)p(\theta_{1:L})}\left[\log \frac{p(h_T \mid \theta_0; \theta)}{\frac{1}{L+1}\sum_{l=0}^{L} p(h_T \mid \theta_l; \theta)}\right]: \tag{36}$$

Note that the distributions on both sides are identical. Therefore, if the term inside the expectation on the LHS is equal to the term inside the expectation on the RHS for all possible realizations of h_T and θ, then the expectations must be equal. Thus we seek to show that:

$$\sum_{t=1}^{T} \gamma^{t-1} R(s_{t-1}, a_{t-1}, s_t; \theta) = \log \frac{p(h_T \mid \theta_0; \theta)}{\frac{1}{L+1}\sum_{l=0}^{L} p(h_T \mid \theta_l; \theta)}: \tag{37}$$

Recall that we have made the same assumption as Foster et al. (2021) that $p(h_t \mid \theta; \pi) = \prod_{t=1}^{T} p(h_t \mid \theta_0; \pi; h_{t-1})$. We can therefore rewrite the RHS:

$$\text{RHS} = \log \frac{p(h_T \mid \theta_0; \theta)}{\frac{1}{L+1}\sum_{l=0}^{L} p(h_T \mid \theta_l; \theta)} \tag{38}$$

$$= \log \frac{\prod_{t=1}^{T} p(h_t \mid \theta_0; \pi; h_{t-1})}{\frac{1}{L+1}\sum_{l=0}^{L} \prod_{t=1}^{T} p(h_t \mid \theta_l; \pi; h_{t-1})} \tag{39}$$

$$= \log \frac{\prod_{t=1}^{T} p(y_t \mid \theta_0; d_t)\,\pi(d_t \mid h_{t-1})}{\frac{1}{L+1}\sum_{l=0}^{L} \prod_{t=1}^{T} p(y_t \mid \theta_l; d_t)\,\pi(d_t \mid h_{t-1})} \tag{40}$$

$$= \log \frac{\prod_{t=1}^{T} p(y_t \mid \theta_0; d_t)}{\frac{1}{L+1}\sum_{l=0}^{L} \prod_{t=1}^{T} p(y_t \mid \theta_l; d_t)}; \tag{41}$$

where the last line uses the fact that $\pi(d_t \mid h_{t-1})$ is independent of θ. Applying log rules, we get:

$$= \sum_{t=1}^{T} \log p(y_t \mid \theta_0; d_t) - \log \frac{1}{L+1} \sum_{l=0}^{L} \prod_{t=1}^{T} p(y_t \mid \theta_l; d_t) \tag{42}$$

$$= \sum_{t=1}^{T} \log p(y_t \mid \theta_0; d_t) - \log \frac{1}{L+1} \sum_{l=0}^{L} \exp \sum_{t=1}^{T} \log p(y_t \mid \theta_l; d_t) \tag{43}$$

$$= \sum_{t=1}^{T} \log p(y_t \mid \theta_0; d_t) + \log(L+1) \sum_{t=1}^{T} \log \sum_{l=0}^{L} \prod_{k=1}^{t} p(y_k \mid \theta_l; d_k) - \log \sum_{l=0}^{L} \prod_{k=1}^{t-1} p(y_k \mid \theta_l; d_k) - \log \sum_{l=0}^{L} \prod_{k=1}^{0} p(y_k \mid \theta_l; d_k); \tag{44}$$

where the last step breaks $\log \sum_{l=0}^{L} \prod_{t=1}^{T} p(y_t \mid \theta_l; d_t)$ into a telescoping sum where the terms cancel out. Now, noting that

the product of an empty sequence is by de nition 1. We get:

$$= \sum_{t=1}^{T} \log p(y_t \mid \theta_0; d_t) + \log(L+1) \left[\sum_{t=1}^{T} \log \sum_{l=0}^{t-1} \sum_{k=1}^{Y} p(y_k \mid \theta_l; d_k) - \log \sum_{l=0}^{t-1} \sum_{k=1}^{Y} p(y_k \mid \theta_l; d_k) - \log(L+1) \right] \tag{45}$$

$$= \sum_{t=1}^{T} \log p(y_t \mid \theta_0; d_t) + \left[\sum_{t=1}^{T} \log \sum_{l=0}^{t-1} \sum_{k=1}^{Y} p(y_k \mid \theta_l; d_k) - \log \sum_{l=0}^{t-1} \sum_{k=1}^{Y} p(y_k \mid \theta_l; d_k) \right] \tag{46}$$

$$= \sum_{t=1}^{T} \log p(y_t \mid \theta_0; d_t) - \log \sum_{l=0}^{t-1} \sum_{k=1}^{Y} p(y_k \mid \theta_l; d_k) + \log \sum_{l=0}^{t-1} \sum_{k=1}^{Y} p(y_k \mid \theta_l; d_k) : \tag{47}$$

By the conditions of the theorem, we have that

$$= 1 \tag{48}$$

$$R(s_{t-1}; a_{t-1}; s_t; \theta) = \log p(y_t \mid \theta_0; d_t) - \log \sum_{l=0}^{t-1} \sum_{k=1}^{Y} p(y_k \mid \theta_l; d_k) + \log \sum_{l=0}^{t-1} \sum_{k=1}^{Y} p(y_k \mid \theta_l; d_k): \tag{49}$$

Plugging this into Equation (47), we get:

$$RHS = \sum_{t=1}^{T} R(s_{t-1}; a_{t-1}; s_t; \theta) \tag{50}$$

$$RHS = \sum_{t=1}^{T} \gamma^{t-1} R(s_{t-1}; a_{t-1}; s_t; \theta) \tag{51}$$

$$\log \frac{p(h_T \mid \theta_0; \theta)}{\frac{1}{L+1} \sum_{l=0}^{L} p(h_T \mid \theta_l; \theta)} = \sum_{t=1}^{T} \gamma^{t-1} R(s_{t-1}; a_{t-1}; s_t; \theta): \tag{52}$$

Which is the equivalence we sought to show in Equation (37). Therefore $\mathrm{IG}(\theta) = \mathrm{sPCE}(\theta; L; T)$. □

A.3. Q-Learning from Priors

In this appendix we prove an additional theorem that shows we can learn Q-function approximators without needing to infer posteriors. It is based in part on a related proof from Section 2.1.3 of Guez (2015), which establishes a recursion for the state value function in Bayes-Adaptive MDPs. However, Guez (2015) does not show that the value function can be computed using only samples from the prior.

Theorem 3. Let M be a HIP-MDP where $s_t = f_1(h_t)$; $a_t = f_2(d_{t+1})$ for some deterministic functions $f_1; f_2$ and the transition dynamics are some deterministic function γ_t of $p(y_t \mid h_{t-1}; d_t; \theta)$. For an arbitrary policy π, the Q-function estimator $\hat{Q}$ can be learned using trajectories sampled from the joint $p(h_T \mid \theta)\pi(h_T)$.

Proof. To begin the proof we must first derive a Bellman equation for the Q-function in the HIP-MDP M. In other words, we need to show that

$$Q^{\pi}(s_i; a_i) = E[R(s_i; a_i; s_{i+1}; \theta) + Q^{\pi}(s_{i+1}; a_{i+1})] \tag{53}$$

for some expectation. Since the state and action are deterministic functions of the history and design, respectively, we can without loss of generality replace s_t with h_t and a_t with d_{t+1} in the following derivation, with the functions f_1 and f_2 being implicitly subsumed into the reward and value functions. Thus we seek to show that

$$Q^{\pi}(h_i; d_{i+1}) = E[R(h_{i+1}; \theta) + Q^{\pi}(h_{i+1}; d_{i+2})]: \tag{54}$$

Recall that a HIP-MDP is a distribution over MDPs, where a particular realization of θ de nes an MDP sampled from that distribution. We therefore de ne the state-action-parameter value function

$$U_\pi(h_i, d_{i+1}, \theta) = \mathbb{E}_{\prod_{t=i+1}^{T}[y_t \sim p(y_t \mid d_t, \theta); d_{t+1} \sim \pi(d_{t+1} \mid h_t)]} \left[\sum_{t=i}^{T} \gamma^{t-i} \mathcal{R}(d_{1:t+1}, y_{1:t+1}, \theta) \right] \tag{55}$$

$$= \mathbb{E}_{y_{i+1} \sim p(y_{i+1} \mid d_{i+1}, \theta); d_{i+2} \sim \pi(d_{i+2} \mid h_{i+1})} \left[\mathcal{R}(d_{1:i+1}, y_{1:i+1}, \theta) \right. \\ \left. + \gamma \cdot \mathbb{E}_{\prod_{t=i+2}^{T}[y_t \sim p(y_t \mid d_t, \theta); d_{t+1} \sim \pi(d_{t+1} \mid h_t)]} \left[\sum_{t=i+1}^{T} \gamma^{t-i} \mathcal{R}(d_{1:t+1}, y_{1:t+1}, \theta) \right] \right] \tag{56}$$

$$= \mathbb{E}_{y_{i+1} \sim p(y_{i+1} \mid d_{i+1}, \theta); d_{i+2} \sim \pi(d_{i+2} \mid h_{i+1})} \left[\mathcal{R}(d_{1:i+1}, y_{1:i+1}, \theta) + \gamma \cdot U_\pi(h_{i+1}, d_{i+2}, \theta) \right], \tag{57}$$

and we note that this is equivalent to the state-action value function in the MDP defined by θ. By construction, the Q-function is the expectation of the U-function:

$$Q_\pi(h_i, d_{i+1}) = \mathbb{E}_{\theta \sim p(\theta \mid h_i)} U_\pi(h_i, d_{i+1}, \theta). \tag{58}$$

Plugging in the recursive relationship of U_π into this expression, we get

$$Q_\pi(h_i, d_{i+1}) = \mathbb{E}_{\theta \sim p(\theta \mid h_i); y_{i+1} \sim p(y_{i+1} \mid d_{i+1}, \theta); d_{i+2} \sim \pi(d_{i+2} \mid h_{i+1})} \left[\mathcal{R}(d_{1:i+1}, y_{1:i+1}, \theta) + \gamma \cdot U_\pi(h_{i+1}, d_{i+2}, \theta) \right] \tag{59}$$

$$= \mathbb{E}_{\theta \sim p(\theta \mid h_i); y_{i+1} \sim p(y_{i+1} \mid d_{i+1}, \theta); d_{i+2} \sim \pi(d_{i+2} \mid h_{i+1})} \left[\mathcal{R}(d_{1:i+1}, y_{1:i+1}, \theta) \right] \\ + \gamma \cdot \mathbb{E}_{\theta \sim p(\theta \mid h_i); y_{i+1} \sim p(y_{i+1} \mid d_{i+1}, \theta); d_{i+2} \sim \pi(d_{i+2} \mid h_{i+1})} [U_\pi(h_{i+1}, d_{i+2}, \theta)]. \tag{60}$$

We want a recursive expression for the Q-function, so we need to show that $U_\pi(h_{i+1}, d_{i+2}, \theta)$ is equal in expectation to $Q_\pi(h_{i+1}, d_{i+2})$. Using the identity $p(\theta \mid h_i) p(y_{i+1} \mid d_{i+1}, \theta) = p(\theta \mid h_{i+1}) p(y_{i+1} \mid h_i, d_{i+1})$ we get:

$$\mathbb{E}_{p(\theta \mid h_i) p(y_{i+1} \mid d_{i+1}, \theta) \pi(d_{i+2} \mid h_{i+1})} [U_\pi(h_{i+1}, d_{i+2}, \theta)] = \mathbb{E}_{p(\theta \mid h_{i+1}) p(y_{i+1} \mid h_i, d_{i+1}) \pi(d_{i+2} \mid h_{i+1})} [U_\pi(h_{i+1}, d_{i+2}, \theta)] \tag{61}$$

$$= \mathbb{E}_{p(y_{i+1} \mid h_i, d_{i+1}) \pi(d_{i+2} \mid h_{i+1})} \mathbb{E}_{p(\theta \mid h_{i+1})} [U_\pi(h_{i+1}, d_{i+2}, \theta)] \tag{62}$$

$$= \mathbb{E}_{p(y_{i+1} \mid h_i, d_{i+1}) \pi(d_{i+2} \mid h_{i+1})} [Q_\pi(h_{i+1}, d_{i+2})] \tag{63}$$

$$= \mathbb{E}_{p(y_{i+1} \mid h_i, d_{i+1}) \pi(d_{i+2} \mid h_{i+1})} \mathbb{E}_{p(\theta \mid h_{i+1})} [Q_\pi(h_{i+1}, d_{i+2})] \tag{64}$$

$$= \mathbb{E}_{p(\theta \mid h_i) p(y_{i+1} \mid d_{i+1}, \theta) \pi(d_{i+2} \mid h_{i+1})} \left[Q_\pi(h_{i+1}, d_{i+2}) \right]. \tag{65}$$

Now we plug this back into Equation (60):

$$Q_\pi(h_i, d_{i+1}) = \mathbb{E}_{p(\theta \mid h_i) p(y_{i+1} \mid d_{i+1}, \theta) \pi(d_{i+2} \mid h_{i+1})} \left[\mathcal{R}(d_{1:i+1}, y_{1:i+1}, \theta) \right] \\ + \gamma \cdot \mathbb{E}_{p(\theta \mid h_i) p(y_{i+1} \mid d_{i+1}, \theta) \pi(d_{i+2} \mid h_{i+1})} \left[U_\pi(h_{i+1}, d_{i+2}, \theta) \right] \tag{66}$$

$$= \mathbb{E}_{p(\theta \mid h_i) p(y_{i+1} \mid d_{i+1}, \theta) \pi(d_{i+2} \mid h_{i+1})} \left[\mathcal{R}(d_{1:i+1}, y_{1:i+1}, \theta) \right] \\ + \gamma \cdot \mathbb{E}_{p(\theta \mid h_i) p(y_{i+1} \mid d_{i+1}, \theta) \pi(d_{i+2} \mid h_{i+1})} \left[Q_\pi(h_{i+1}, d_{i+2}) \right] \tag{67}$$

$$= \mathbb{E}_{p(\theta \mid h_i) p(y_{i+1} \mid d_{i+1}, \theta) \pi(d_{i+2} \mid h_{i+1})} \left[\mathcal{R}(d_{1:i+1}, y_{1:i+1}, \theta) + Q_\pi(h_{i+1}, d_{i+1}) \right], \tag{68}$$

which gives us a recursive expression of the form in Equation (54). To train the estimator $\hat{Q}_\pi$ we need to minimize some expected loss w.r.t. the distribution of trajectories generated by the policy, i.e.

$$\mathbb{E}_{p(h_i, d_{i+1} \mid \pi)} f_{loss}(\hat{Q}_\pi(h_i, d_{i+1}), Q_\pi(h_i, d_{i+1})), \tag{69}$$

where the loss is a deterministic function, for example the mean square error. We can use Monte Carlo sampling to estimate this expectation. $\hat{Q}_\pi$ is available in functional form (e.g. as a neural net) and can be easily evaluated for any input. To evaluate

Q_π, however, we would have to sample from the distribution $p(h_i, d_{i+1} \mid \pi)p(\theta \mid h_i)p(y_{i+1} \mid d_{i+1}, \theta)\pi(d_{i+2} \mid h_{i+1})$. This requires sampling from several difficult distributions, such as the marginal $p(h_i, d_i \mid \pi)$ and the posterior $p(\theta \mid h_i)$. However, recalling the identity $p(\theta \mid h_i)p(y_{i+1} \mid d_{i+1}, \theta) = p(\theta \mid h_{i+1})p(y_{i+1} \mid h_i, d_{i+1})$, we can simplify the above distribution:

$$
\begin{aligned}
p(h_i, d_{i+1} \mid \pi)p(\theta \mid h_i)p(y_{i+1} \mid d_{i+1}, \theta)\pi(d_{i+2} \mid h_{i+1}) = & && (70)\\
= p(\theta \mid h_i)p(y_{i+1} \mid d_{i+1}, \theta)p(h_i, d_{i+1} \mid \pi)\pi(d_{i+2} \mid h_{i+1}) & && (71)\\
= p(\theta \mid h_i)p(y_{i+1} \mid d_{i+1}, \theta)p(d_{i+1} \mid h_i, \pi)p(h_i \mid \pi)\pi(d_{i+2} \mid h_{i+1}) & && (72)\\
= p(\theta \mid h_{i+1})p(y_{i+1} \mid h_i, d_{i+1})p(d_{i+1} \mid h_i, \pi)p(h_i \mid \pi)\pi(d_{i+2} \mid h_{i+1}) & && (73)\\
= p(\theta \mid h_{i+1})p(y_{i+1}, d_{i+1} \mid h_i, \pi)p(h_i \mid \pi)\pi(d_{i+2} \mid h_{i+1}) & && (74)\\
= p(\theta \mid h_{i+1})p(h_{i+1} \mid h_i, \pi)p(h_i \mid \pi)\pi(d_{i+2} \mid h_{i+1}) & && (75)\\
= p(\theta \mid h_{i+1})p(h_{i+1}, h_i \mid \pi)\pi(d_{i+2} \mid h_{i+1}) & && (76)\\
= p(\theta \mid h_{i+1})p(h_{i+1} \mid \pi)\pi(d_{i+2} \mid h_{i+1}) & && (77)\\
= p(\theta, h_{i+1} \mid \pi)\pi(d_{i+2} \mid h_{i+1}). & && (78)
\end{aligned}
$$

Hence we can generate the necessary samples by drawing θ from the prior and rolling out the policy, i.e. sampling from the joint $p(\theta, h_T \mid \pi)$.

□

B. Experiment Details

In this section we include the full description of all the experimental design problems considered in the paper.

B.1. Source Location

In this experiment there are 2 sources embedded in the 2-dimensional plane that emit a signal. The total intensity at any given coordinate d in the plane is:

$$\mu(\theta, d) = b + \frac{1}{m + ||\theta_1 - d||^2} + \frac{1}{m + ||\theta_2 - d||^2}, \tag{79}$$

where $b, m > 0$ are the background and maximum signals, respectively, $||\cdot||^2$ is the squared Euclidean norm, and θ_i are the coordinates of the i^{th} signal source. The probabilistic model is:

$$\theta_i \sim \mathcal{N}(0, I); \quad \log y \mid \theta, d \sim \mathcal{N}(log(\mu(\theta, d), \sigma), \tag{80}$$

i.e. the prior is unit Gaussian and we observe the log of the total signal intensity with some Gaussian observation noise σ. The design space is restricted to $\mathcal{A} = [-4, 4]^2$. The hyperparameters we used are

PARAMETER	VALUE
b	$1e$ 1
m	$1e$ 4
σ	0.5

B.2. Constant Elasticity of Substitution

In this experiment subjects compare 2 baskets of goods and give a rating on a sliding scale from 0 to 1. The designs are vectors $d = (x, x')$ where $x, x' \in [0, 100]^3$ are the baskets of goods. The latent parameters and priors are:

$$\rho \sim Beta(1,1) \tag{81}$$
$$\alpha \sim Dirichlet([1,1,1]) \tag{82}$$
$$\log u \sim \mathcal{N}(1,3). \tag{83}$$

The probabilistic model is:

$$U(x) = \left(\sum_i x_i^\rho \alpha_i \right)^{1/\rho} \tag{84}$$
$$\mu_\eta = (U(x) - U(x'))u \tag{85}$$
$$\sigma_\eta = (1 + ||x - x'||)\tau \cdot u \tag{86}$$
$$\eta \sim \mathcal{N}(\mu_\eta, \sigma_\eta^2) \tag{87}$$
$$y = clip(sigmoid(\eta), \epsilon, 1-\epsilon), \tag{88}$$

where $U(\cdot)$ is a subjective utility function. The hyperparameter values are:

PARAMETER	VALUE
τ	0.005
ϵ	2^{-22}

B.3. Prey Population

In this experiment an initial population of prey animals is left to survive for $\mathcal{T} = 24$ hours, and we measure the number of individuals consumed by predators at the end of the experiment. The designs are the initial populations $d = N_0 \in [1, 300]$. The latent parameters and priors are:

$$\log a \sim \mathcal{N}(-1.4, 1.35) \tag{89}$$
$$\log T_h \sim \mathcal{N}(-1.4, 1.35). \tag{90}$$

The population changes over time according to the differential equation:

$$\frac{dN}{d\tau} = -\frac{aN^2}{1 + aT_h N^2}. \tag{91}$$

And the population $N_\mathcal{T}$ is thus the solution of an initial value problem. The probabilistic model is:

$$p_\mathcal{T} = \frac{d - N_\mathcal{T}}{d} \tag{92}$$
$$y \sim Binom(d, p_\mathcal{T}). \tag{93}$$

C. Algorithm Details

In this section we provide implementation details and hyperparameter settings for the various algorithms used in the paper. All of our code was written in Python using the Pytorch framework for automatic differentiation.

C.1. RL

The REDQ (Chen et al., 2021) algorithm is a generalisation of the earlier Soft Actor-Critic, where instead of 2 critic networks we train an ensemble of N critics and estimate the Q value by a random sample of $M < N$ members from this

ensemble. We therefore modified the implementation of Soft Actor-Critic available in Garage (https://github.com/rlworkgroup/garage). Hyparparameter choices for the different SED problems are enumerated in the table below. Values were chosen by linear search over each hyperparameter independently, with the following search spaces: target update rate $\tau \in \{1e-3, 5e-3, 1e-2\}$; policy learning rate $LR_\pi \in \{1e-5, 1e-4, 3e-4, 1e-3\}$; q-function learning rate $LR_{qf} \in \{1e-5, 1e-4, 3e-4, 1e-3\}$; replay buffer size $|buffer| \in \{1e5, 1e6, 1e7\}$; discount factor $\gamma \in \{0.95, 0.99, 1\}$. Note that we chose $\gamma \neq 1$ as we empirically observed this leads to improved performance.

PARAMETER	SOURCE LOCATION	CES	PREY POPULATION
N	2	2	10
M	2	2	2
TRAINING ITERATIONS	$2e4$	$2e4$	$4e4$
CONTRASTIVE SAMPLES	$1e5$	$1e5$	$1e4$
T	30	10	10
γ	0.9	0.9	0.95
τ	$1e-3$	$5e-3$	$1e-2$
POLICY LEARNING RATE	$1e-4$	$3e-4$	$1e-4$
CRITIC LEARNING RATE	$3e-4$	$3e-4$	$1e-3$
BUFFER SIZE	$1e7$	$1e6$	$1e6$

The summary network B_ψ consisted of 2 fully connected layers with 128 units and ReLU activation, followed by an output layer with 64 units and no activation function. The policy network π_ϕ has a similar architecture, but its output layer depends on the experiment. For the source location and CES experiments, the output layer has $2|\mathcal{A}|$ dimensions, representing the mean and log-variance of $|\mathcal{A}|$ independent Tanh-Gaussian distributions, one for each dimension of the design space. For the prey population experiment, the output layer has $|\mathcal{A}|$ dimensions, representing the logits of a Gumbel-Softmax distribution with $|\mathcal{A}|$ classes.

C.2. DAD

We used the official DAD implementation published by Foster et al. (2021) and available at https://github.com/ae-foster/dad. We verified that the authors' evaluation code produces the same results as ours for the same trained DAD policy, and subsequently used their code to evaluate all DAD policies.

C.3. SMC

For SMC we used the implementation of Moffat et al. (2020) (https://github.com/haydenmoffat/sequential_design_for_predator_prey_experiments). We connected to the R code by means of the $rpy2$ Python package. All hyperparameters are identical to the original paper. We used the parameter estimation utility and the model was Holling's type-II Binomial.

C.4. VPCE

VPCE consists of maximizing the PCE lower bound by gradient descent w.r.t. the experimental design and inferring the posterior through variational inference, then repeating the process. We ran this algorithm with the hyperparameters in the table below. Hyperparameters were chosen based on the recommendations of Foster et al. (2020) where relevant.

PARAMETER	SOURCE LOCATION	CES	PREY POPULATION
T	30	10	10
DESIGN GRADIENT STEPS	2500	2500	2500
CONTRASTIVE SAMPLES	500	10	100
EXPECTATION SAMPLES	500	10	100
VI SAMPLES	10	10	10
VI GRADIENT STEPS	1000	1000	1000
VI LEARNING RATE	$4e-2$	$4e-2$	$4e-2$

For PCE-BO, we used a Matern $\frac{5}{2}$ kernel with a lengthscale of 10 and 4 steps of BO.

D. Hardware details

Except for SMC, all experiments were run on a Slurm cluster node with a single Nvidia Tesla P100 GPU and 4 cores of an Intel Xeon E5-2690 CPU. The SMC experiments did not run on a GPU as the original authors' R code is not written to utilize GPU acceleration.

E. Non-Myopic Solutions

In general it is not feasible to determine the myopically optimal solution to a SED problem, and therefore it is unclear whether an agent is actually finding a non-myopic solution or just a better myopic solution than the baselines. Foster et al. (2021) proposed to use a simplified variant of the source location problem, i.e. setting $d = 1; k = 1; T = 2$. In this variant the EIG of a design can be computed through numerical integration, and an optimal design can be approximated with high accuracy via grid search. We executed an experiment in this setting with 2 RL agents: a myopic agent (derived by setting $\gamma = 0$) and a non-myopic agent ($\gamma = 1$).

The results are shown in Figure 6. Note that after the second experiment, our myopic agent achieves a very similar result to the optimal myopic strategy of Foster et al. (2021), and the non-myopic agent outperforms both by a significant margin. However, at $t = 1$ the non-myopic agent does worse than the myopic agent. This shows that the agent has indeed learned to sacrifice EIG in earlier experiments in order to set itself up for superior EIG at $t = T$.

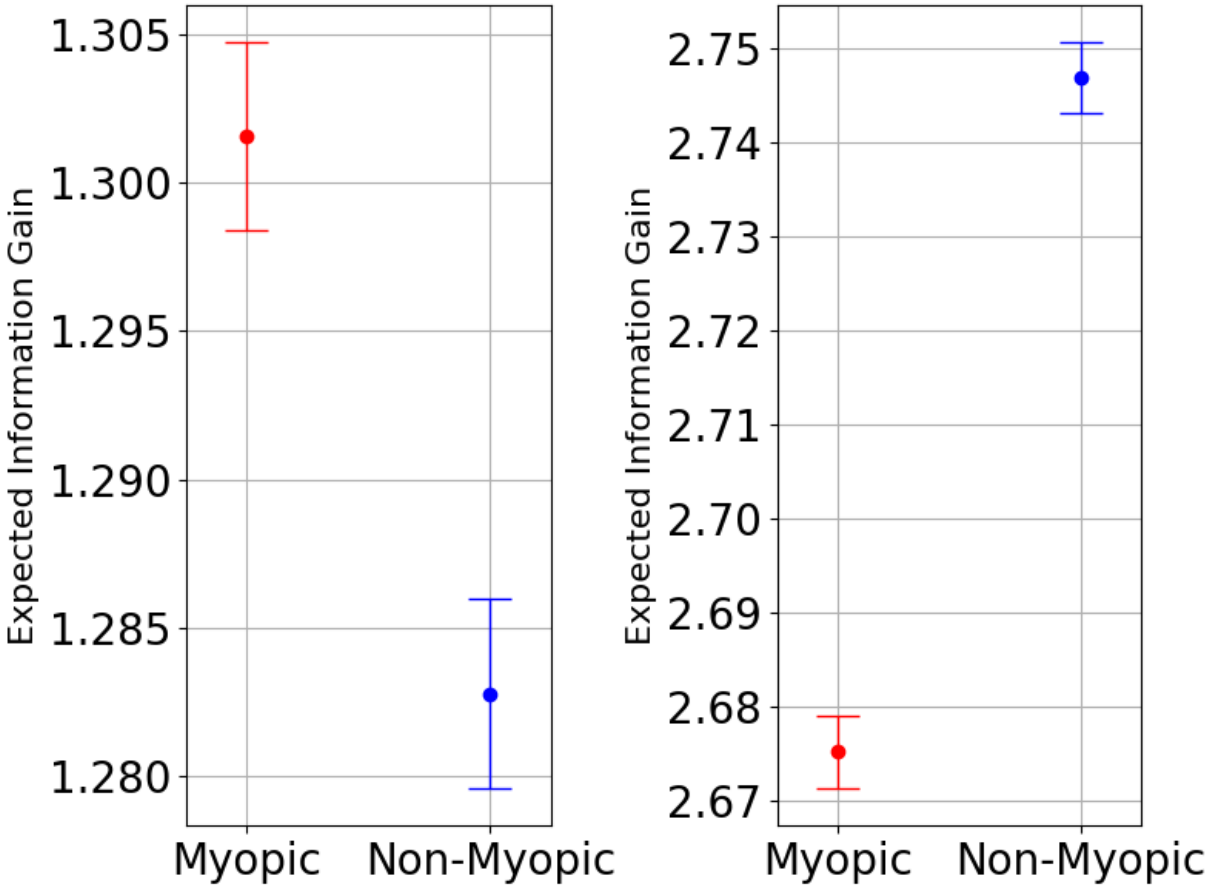

Figure 6. Mean and standard error of expected information gain for the 1-dimensional source location problem, aggregated from $1e5$ rollouts with $L = 1e4$ contrastive samples. (Left) information gain after the first experiment. (Right) information gain after the second experiment.

F. Learning Curves

Plotted below are the learning curves for the reinforcement learning agents in the 3 SED problems investigated in this paper. The plots include mean and standard errors of the rewards for both the full method and the naive baseline, aggregated over 10 replications. Note that the reward is computed with a smaller number of contrastive samples than was used in the final evaluation. For exact values cf. Appendix C.

These learning curves illustrate how our factorized reward formulation provides a more informative learning signal to the agent, allowing it to improve its performance faster (in a smaller number of iterations) and converge to higher rewards. This is particularly notable in the source location problem, where the high experimental budget (30 experiments as opposed to only 10 in the other tasks) makes the problem of credit assignment exceptionally challenging.