Cross-Entropy Estimators for Sequential Experiment Design with Reinforcement Learning

Tom Blau[*] TOMBLAU@GMAIL.COM
Nourish Ingredients
Sydney, Australia

Iadine Chades IADINE.CHADES@CSIRO.AU
CSIRO's Environment
Sydney, Australia

Amir Dezfouli[*] AKDEZFOULI@GMAIL.COM
BIMLOGIQ
Sydney, Australia

Daniel Steinberg DAN.STEINBERG@CSIRO.AU
CSIRO's Data61
Canberra, Australia

Edwin V. Bonilla EDWIN.BONILLA@CSIRO.AU
CSIRO's Data61
Sydney, Australia

Abstract

Reinforcement learning can learn amortised design policies for designing sequences of experiments. However, current methods rely on contrastive estimators of expected information gain, which require an exponential number of contrastive samples to achieve an unbiased estimation. We propose the use of an alternative lower bound estimator, based on the cross-entropy of the joint model distribution and a flexible proposal distribution. This proposal distribution approximates the true posterior of the model parameters given the experimental history and the design policy. Our method requires no contrastive samples, can achieve more accurate estimates of high information gains, allows learning of superior design policies, and is compatible with implicit probabilistic models. We assess our algorithm's performance in various tasks, including continuous and discrete designs and explicit and implicit likelihoods.

Keywords: Sequential design of experiments, reinforcement learning.

1. Introduction

A key challenge in science is to develop predictive models based on experimental observations. As far back as Lindley (1956) it has been recognised that experimental designs can be opti-

∗. Work done while at CSIRO's Data61.

mised to be maximally informative, under the assumptions of a Bayesian framework. Since then optimal experimental design has been applied to a wide variety of fields with different models and assumptions, including neuroscience (Shababo et al., 2013), biology (Treloar et al., 2022), ecology (Drovandi et al., 2014) and causal structure learning (Agrawal et al., 2019).

Under the framework of Bayesian optimal experimental design (BOED), we have a probabilistic model $p(y|\theta, d)$ where d is the design (e.g. where to measure), y is the outcome (e.g. the measurement value) and θ are parameters over which we have a prior belief $p(\theta)$. The objective is to find the optimal design d^* that maximises the expected information gain (EIG), formally:

$$EIG(d) \coloneqq \mathbb{E}_{p(y|d)} \left[H(p(\theta)) - H(p(\theta|y, d)) \right], \tag{1}$$

$$d^* = \arg\max_{d \in \mathcal{D}} EIG(d). \tag{2}$$

We see that naïve computation of the EIG requires an expectation over the marginal likelihood $p(y|d)$ and estimation of the posterior $p(\theta|y, d)$. Since sampling from $p(y|d)$ is typically intractable and there is usually no closed-form solution for the posterior, minimising this expression involves estimating a nested expectation numerically, which is challenging (Rainforth et al., 2018). Furthermore, we are often interested in conducting more than one experiment, in which case optimal designs must incorporate the outcomes of previous experiments sequentially (Krause and Guestrin, 2007).

In settings where computational or application-specific constraints demand fast deployment times, *amortised* methods have proved successful, as they learn an optimal design policy as a function of the experimental history instead of optimising each design in turn (Blau et al., 2022; Foster et al., 2021; Ivanova et al., 2021). Once trained, a policy can be reused to design experiments as many times as desired, thus amortising the cost of training. However, all amortised methods introduced thus far have the drawback that they rely on maximising contrastive lower bounds of the objective. To achieve an unbiased estimate of the EIG, these contrastive bounds require a number of samples that is exponential in its magnitude (McAllester and Stratos, 2020; Poole et al., 2019). Thus their performance degrades in cases where the EIG is large.

To address this limitation, we propose a new amortised method, using reinforcement learning (RL) and a non-contrastive bound based on the cross-entropy of the joint model distribution and a flexible proposal distribution. This proposal approximates the true posterior of the model parameters given the experimental history and the design policy. Our method does not suffer from exponential sample complexity and is thus able to achieve higher EIG than prior art, especially in settings where the information gain of the optimal policy is large. Furthermore, unlike previous amortised methods, our method is generally applicable to continuous and discrete design spaces, non-differentiable likelihoods, and even implicit likelihoods. Our experiments show the benefits of our approach when compared to previous methods in these settings.

2. Amortised design of experiments

In Bayesian optimal experimental design (BOED) the goal is to identify the parameters of a probabilistic model by sending queries to that model. Let $p(y|\theta, d)$ be the model of concern,

with some prior belief $p(\theta)$ regarding the value of parameters θ. As described above, an optimal design d^* is one that maximises the EIG as given by Equations (1) and (2), where computational intractabilities readily appear in the estimation of the marginal likelihood and the posterior distribution.

Furthermore, more challenging than optimising a single experiment is the problem of optimising an entire sequence of experiments $d_{1:T}$ where $T \in \mathbb{N}$ is some fixed budget. One promising approach for settings under strong computational constraints at deployment time is to optimise a design policy $\pi : \mathcal{H} \to \mathcal{D}$ that designs experiments conditioned on a history $h_t = (d_i, y_i)_{i=1}^t$. The computational cost of learning such a policy is high, but designing experiments with a trained policy is computationally efficient, requiring only a single forward pass of a neural network. Therefore the training cost is amortised over the lifetime of the policy, and this class of algorithms is known as *amortised* design of experiments. Current amortised methods all use contrastive bounds to optimise the policy. However, such bounds require a number of contrastive samples L that is exponential in the magnitude of the quantity being estimated (McAllester and Stratos, 2020). In other words, if the EIG of a policy is large, then computing an accurate contrastive bound is intractable.

3. The sequential cross-entropy estimator

To resolve this limitation of requiring an exponentially large number of samples, we introduce a proposal distribution $q(\theta|h_T, \pi)$ that approximates the true posterior $p(\theta|h_T, \pi)$. Using the cross-entropy of the two, we use the following estimator, which we refer to as the sequential cross-entropy estimator (sCEE).

$$sCEE(\pi, T) \coloneqq \mathbb{E}_{p(\theta, h_T|\pi)} \left[\log q(\theta|h_T, \pi)\right] + H\left[p(\theta)\right]. \tag{3}$$

From Jensen's inequality it follows that the cross-entropy of two random variables is a lower bound for the self-entropy of either variable. By extending this to the sequential case, the following theorem shows that the sCEE is a lower bound of the true EIG:

Theorem 1 *Let $p(y|\theta, d)$ be a probabilistic model with prior $p(\theta)$. For an arbitrary fixed design policy π and sequence length T, the EIG of using π to design T experiments is denoted $EIG(\pi, T)$. Let $q(\theta|h_T, \pi)$ be a proposal distribution over parameters θ conditioned on experimental history h_T, and the sCEE bound is*

$$sCEE(\pi, T) \coloneqq \mathbb{E}_{p(\theta, h_T|\pi)} \left[\log q(\theta|h_T, \pi)\right] + H\left[p(\theta)\right], \tag{4}$$

we have that

$$sCEE(\pi, T) \leq EIG(\pi, T). \tag{5}$$

Proof A sketch of proof follows here, with the full proof in Appendix A.1. The main idea is to rewrite the EIG as an expectation w.r.t. distribution $p(h_T, \theta|\pi)$, and then show that the difference between EIG and sCEE is an expectation over KL divergences. ■

It is easy to show that the above bound is tight if and only if $q(\theta|h_T, \pi) = p(\theta|h_T, \pi)$ and that the bias of the sCEE estimator is $-\mathbb{E}_{h_T}\left[\mathrm{KL}\left[p(\theta|h_T, \pi) \parallel q(\theta|h_T, \pi)\right]\right]$. In other words, the quality of the estimation rests on how well the proposal distribution can match the true posterior, in terms of KL divergence.

We note here that our sCEE bound is the sequential version of the bound proposed by Barber and Agakov (2004), who used it for estimating mutual information in the context of information transmission over noisy channels. This bound is also referred in Foster et al. (2019) as the variational posterior estimator, who used it for gradient-based experimental design in a non-sequential setting.

3.1 Proposal parameterization

To evaluate the sCEE, we sample from the joint $p(\theta, h_T|\pi)$ simply by rolling out the policy. Under mild assumptions, it can be shown that this Monte Carlo estimation approaches the true value of the sCEE at a rate of $O(\frac{1}{\sqrt{n}})$, where n is the number of samples (cf. Appendix A.3). We will parameterise the proposal distribution by a conditional normalising flow (Winkler et al., 2019) with parameters κ and, therefore, refer to it using $q_\kappa(\cdot)$. Thus, we can maximise the sCEE w.r.t. κ using stochastic gradient descent. Note that we only need to optimise the negative cross-entropy term $\mathbb{E}_{p(\theta,h_T|\pi)}\left[\log q(\theta|h_T,\pi)\right]$, since the prior entropy is constant. Details of the normalising flows used in our experiments are in Appendix F.

4. Experiment design with sCEE and reinforcement learning

We implement the sCEE bound in the context of an RL algorithm by using it in the formulation of the reward function within the RL framework defined by Blau et al. (2022),

$$\mathcal{R}(s_{t-1}, a_{t-1}, s_t, \theta) = \log q(\theta|B_{\psi,t}) - \log q(\theta|B_{\psi,t-1}), \tag{6}$$

where the key idea is to map experimental designs to policy actions $a_{t-1} = d_t$, and the history information to the system states s_t with a parameterized summary from an encoder network given by $B_{\psi,t}$. Details of the RL formulation are given in Appendix C.

4.1 Advantages of sCEE-RL

Our method based on the sCEE lower bound and RL delivers a number of advantages. **(i) Better sample complexity**: it does not require the use of contrastive samples, and hence does not suffer from the exponential sample complexity issue of the sPCE bound. Thus, sCEE can more closely estimate EIG when the true quantity is large, although estimation accuracy depends on learning a good posterior network $q_\kappa(\cdot)$. **(ii) Applicable to implicit models**: Furthermore, we see that the sCEE estimator, as defined in Equation (3), only requires sampling of the model distribution and avoids explicit log-likelihood computations $\log p(h_T|\theta,\pi)$. This means that our method is compatible with implicit likelihood models where the likelihood is a black-box or intractable and, therefore, can only be sampled but not evaluated explicitly. Interestingly, the sCEE bound is closely related to the sACE bound introduced in the appendices of Foster et al. (2021). We discuss this relationship in Appendix B. **(iii) Suitable for continuous and discrete design spaces**: Finally, similar to the method proposed in Blau et al. (2022), our approach using the sCEE estimator along with reinforcement learning, as described in Algorithm 1 (in Appendix C), can handle both continuous and discrete design spaces.

Table 1: Different estimators for EIG of increasing magnitudes in **synthetic data** problems with conjugate priors. Averages computed over 1000 samples. k is the number of random variable dimensions, σ_0 is prior variance, and σ is likelihood variance.

Method	$k = 10$ $\sigma_0 = 0.5$ $\sigma = 5$	$k = 10$ $\sigma_0 = 0.5$ $\sigma = 1$	$k = 10$ $\sigma_0 = 1$ $\sigma = 1$	$k = 10$ $\sigma_0 = 2$ $\sigma = 1$	$k = 10$ $\sigma_0 = 2$ $\sigma = 0.5$	$k = 10$ $\sigma_0 = 4$ $\sigma = 0.5$	$k = 20$ $\sigma_0 = 4$ $\sigma = 0.5$
True EIG	3.47	8.96	11.99	15.22	18.57	21.97	43.94
sCEE	3.40	8.90	**11.92**	**15.07**	**18.41**	**20.47**	**43.89**
sPCE(L = 1E4)	3.45	7.92	8.95	9.18	9.21	9.21	9.21
sPCE(L = 1E6)	**3.48**	8.89	11.45	13.18	13.75	13.81	13.81
sPCE(L = 1E8)	**3.48**	**8.97**	11.85	14.35	16.71	18.08	18.42

5. Experimental results

We evaluate our proposed method on (i) synthetic data; (ii) continuous designs and implicit likelihoods[1] in behavioural economics under a constant elasticity of substitution (CES) problem and a (iii) source location problem; and (iv) discrete designs in a prey population problem. Description and details of these problems and their mathematical models are given in Appendix E. We compare our method (RL-sCEE) with a number of baselines, including RL with the sPCE bound (RL-sPCE; Blau et al., 2022), Deep Adaptive Design (DAD; Foster et al., 2021), implicit Deep Adaptive Design (iDAD; Ivanova et al., 2021), and a non-amortised sequential Monte Carlo experiment design approach (SMC-ED; Moffat et al., 2020).

Results are shown on Tables 1 and 2. We see on Table 1 that, on the synthetic data, when the EIG is small enough, sPCE can provide a better estimate than sCEE (note that the sPCE at times slightly overestimates the EIG due to variance in estimating the expectation with Monte Carlo samples). However, as the EIG becomes large relative to $\log(L)$, the underestimation of sPCE becomes more severe, and for the right-most columns all sPCE variants have reached their upper limit. Meanwhile, sCEE consistently provides good estimates regardless of the magnitude of the EIG.

The results on the real datasets on Table 2 show that for both the CES and source location problem, our proposed RL-sCEE method outperforms all baselines, in spite of not having access to explicit likelihoods. Furthermore, on the prey population problem (rightmost column in Table 2), we note that DAD and iDAD cannot optimise over discrete design spaces and, therefore, we added the sequential Monte Carlo design algorithm proposed by Moffat et al. (2020) as a baseline. Note that this method is not amortised, and requires considerable computation time to design each experiment ($>$ 1 minute per design, whereas amortised policies take milliseconds). We see that RL-sCEE performs similarly to the baselines while using orders of magnitude less time to compute designs than the SMC-ED baseline.

1. We simulate an implicit likelihood by withholding the explicit likelihood values $p(y|\theta, d)$ from the RL-sCEE and iDAD agents.

Table 2: Lower and upper bounds for the EIG computed using the sPCE and sNMC estimators, respectively. $L = 1E8$ contrastive samples were used for the **CES and Source Location problems**, and $L = 1E6$ for the **Prey Population problem**. Means and standard errors aggregated from 1000 rollouts.

Method	CES ($T = 10$)		Source Location ($T = 30$)		Prey Population ($T = 10$)	
	Lower bound	Upper bound	Lower bound	Upper bound	Lower bound	Upper bound
RL-sCEE	$\mathbf{15.91} \pm 0.10$	20.78 ± 0.43	$\mathbf{13.37} \pm 0.07$	13.42 ± 0.08	4.41 ± 0.05	4.41 ± 0.05
RL-sPCE	14.81 ± 0.12	15.56 ± 0.17	11.65 ± 0.06	12.01 ± 0.07	4.38 ± 0.05	4.41 ± 0.04
DAD	10.77 ± 0.08	13.20 ± 0.68	11.22 ± 0.07	11.29 ± 0.07	N/A	N/A
iDAD	9.67 ± 0.08	10.63 ± 0.52	10.37 ± 0.07	10.41 ± 0.08	N/A	N/A
SMC-ED	N/A	N/A	N/A	N/A	$\mathbf{4.52} \pm 0.07$	4.52 ± 0.06

6. Related work

Considerable work has been done on BOED (Chaloner and Verdinelli, 1995; Ryan et al., 2016), and particularly on using machine learning to optimise experimental designs (Rainforth et al., 2023). Greedy algorithms have been developed based on variational bounds (Foster et al., 2019, 2020) or neural network estimates (Kleinegesse and Gutmann, 2020) of the EIG. In the active learning literature, the BALD (Houlsby et al., 2011) score is equivalent to EIG, and can be estimated using Monte Carlo dropout neural networks (Gal et al., 2017). Other works attempt a non-greedy approach, i.e. they can sacrifice information gain in the current experiment in exchange for higher information gain in future experiments. Such approaches include n-step look-ahead (Zhao et al., 2021; Yue and Kontar, 2020) or using batch designs as a lower bound for the utility of sequential designs (Jiang et al., 2020). Foster et al. (2021) were the first to propose an amortised method for sequential experiment design, and showed empirically that the learned policies can exhibit non-myopic behaviour. This was extended to the case of implicit likelihood models by Ivanova et al. (2021). Blau et al. (2022) formulated the sequential experimental design (SED) problem as a special Markov decision process (MDP), and showed that design policies can be learned with RL algorithms.

7. Conclusion

We have introduced the sequential Cross-Entropy Estimator (sCEE), a lower bound estimate for the EIG of an experiment design policy, as well as a reinforcement learning algorithm (RL-sCEE) that uses it to optimise policies. Experiments show the sCEE is capable of estimating large EIGs that are intractable to estimate with contrastive estimators, which are the state of the art. In tasks where EIG is large, RL-sCEE significantly outperforms all baselines and learns policies whose lower bound EIG estimates exceed the upper bound estimate of the strongest alternative. In tasks where EIG is small, RL-sCEE matches the performance of state-of-the-art baselines.

References

Raj Agrawal, Chandler Squires, Karren Yang, Karthikeyan Shanmugam, and Caroline Uhler. Abcd-strategy: Budgeted experimental design for targeted causal structure discovery. In *International Conference on Artificial Intelligence and Statistics*, 2019.

George Baltas. Utility-consistent brand demand systems with endogenous category consumption: principles and marketing applications. *Decision Sciences*, 32(3):399–422, 2001.

David Barber and Felix Agakov. The IM algorithm: a variational approach to information maximization. In *Advances in neural information processing systems*, 2004.

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.

Tom Blau, Edwin V Bonilla, Iadine Chades, and Amir Dezfouli. Optimizing sequential experimental design with deep reinforcement learning. In *International Conference on Machine Learning*, 2022.

Adrienne Bloss, Paul Hudak, and Jonathan Young. Code optimizations for lazy evaluation. *Lisp and Symbolic Computation*, 1(2):147–164, 1988.

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

Xinyue Chen, Che Wang, Zijian Zhou, and Keith W Ross. Randomized ensembled double q-learning: Learning fast without a model. In *International Conference on Learning Representations*, 2021.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *International Conference on Learning Representations*, 2017.

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

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*, 2019.

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*, 2020.

Adam Foster, Desi R Ivanova, Ilyas Malik, and Tom Rainforth. Deep adaptive design: Amortizing sequential Bayesian experimental design. *International Conference on Machine Learning*, 2021.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 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.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

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

Shali Jiang, Henry Chai, Javier Gonzalez, and Roman Garnett. Binoculars for efficient, nonmyopic sequential experimental design. In *International Conference on Machine Learning*, 2020.

Steven Kleinegesse and Michael U Gutmann. Bayesian experimental design for implicit models by mutual information neural estimation. In *International Conference on Machine Learning*, 2020.

Andreas Krause and Carlos Guestrin. Nonmyopic active learning of gaussian processes: an exploration-exploitation approach. In *International Conference on Machine learning*, 2007.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *International Conference on Learning Representations*, 2016.

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

David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Hayden Moffat, Markus Hainy, Nikos E Papanikolaou, and Christopher Drovandi. Sequential experimental design for predator–prey functional response experiments. *Journal of the Royal Society Interface*, 17(166), 2020.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, 2019.

Tom Rainforth, Rob Cornish, Hongseok Yang, Andrew Warrington, and Frank Wood. On nesting monte carlo estimators. In *International Conference on Machine Learning*, 2018.

Tom Rainforth, Adam Foster, Desi R Ivanova, and Freddie Bickford Smith. Modern Bayesian experimental design. *arXiv preprint arXiv:2302.14545*, 2023.

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

Ben Shababo, Brooks Paige, Ari Pakman, and Liam Paninski. Bayesian inference and online experimental design for mapping neural microcircuits. *Advances in Neural Information Processing Systems*, 2013.

Vincent Stimper, David Liu, Andrew Campbell, Vincent Berenz, Lukas Ryll, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Normflows: A PyTorch Package for Normalizing Flows. *arXiv preprint arXiv:2302.12014*, 2023.

Neythen J Treloar, Nathan Braniff, Brian Ingalls, and Chris P Barnes. Deep reinforcement learning for optimal experimental design in biology. *PLOS Computational Biology*, 18(11), 2022.

Hado Van Hasselt, Yotam Doron, Florian Strub, Matteo Hessel, Nicolas Sonnerat, and Joseph Modayil. Deep reinforcement learning and the deadly triad. In *NeurIPS Deep Reinforcement Learning Workshop*, 2018.

Christina Winkler, Daniel Worrall, Emiel Hoogeboom, and Max Welling. Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*, 2019.

Xubo Yue and Raed AL Kontar. Why non-myopic Bayesian optimization is promising and how far should we look-ahead? a study via rollout. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Guang Zhao, Edward Dougherty, Byung-Jun Yoon, Francis Alexander, and Xiaoning Qian. Uncertainty-aware active learning for optimal Bayesian classifier. In *International Conference on Learning Representations*, 2021.

Appendix A. Proofs

This appendix enumerates the proofs for the theorems, corollaries and other claims made in the main paper.

A.1 Proof of Theorem 1

Here we prove the main theorem of the paper, which is restated for convenience

Theorem 1 *Let $p(y|\theta, d)$ be a probabilistic model with prior $p(\theta)$. For an arbitrary fixed design policy π and sequence length T, the EIG of using π to design T experiments is denoted $EIG(\pi, T)$. Let $q(\theta|h_T, \pi)$ be a proposal distribution over parameters θ conditioned on experimental history h_T, and the sCEE bound is*

$$sCEE(\pi, T) \coloneqq \mathbb{E}_{p(\theta, h_T|\pi)} \left[\log q(\theta|h_T, \pi)\right] + H\left[p(\theta)\right] \tag{7}$$

we have that

$$sCEE(\pi, T) \leq EIG(\pi, T) \tag{8}$$

Proof From Theorem 1 of Foster et al. (2021) we have that the EIG is:

$$EIG(\pi, T) = \mathbb{E}_{p(h_T, \theta|\pi)} \left[\log p(h_T|\theta, \pi) - \log p(h_T|\pi)\right] \tag{9}$$

This can be rewritten into a more convenient form:

$$EIG(\pi, T) = \mathbb{E}_{p(h_T, \theta|\pi)} \left[\log \frac{p(h_T|\theta, \pi)}{p(h_T|\pi)}\right] = \mathbb{E}_{p(h_T, \theta|\pi)} \left[\log \frac{p(h_T, \theta|\pi)}{p(h_T|\pi)p(\theta)}\right] \tag{10}$$

$$= \mathbb{E}_{p(h_T, \theta|\pi)} \left[\log \frac{p(\theta|h_T, \pi)\cancel{p(h_T|\pi)}}{p(\theta)\cancel{p(h_T|\pi)}}\right] = \mathbb{E}_{p(h_T, \theta|\pi)} \left[\log \frac{p(\theta|h_T, \pi)}{p(\theta)}\right] \tag{11}$$

$$= \mathbb{E}_{p(h_T, \theta|\pi)} \left[\log p(\theta|h_T, \pi) - \log p(\theta)\right] \tag{12}$$

$$= \mathbb{E}_{p(h_T, \theta|\pi)} \left[\log p(\theta|h_T, \pi)\right] + H\left[p(\theta)\right]. \tag{13}$$

We proceed to show that sCEE lower bounds this form. Consider the KL divergence between 2 conditional distributions given a fixed value y:

$$\mathrm{KL}\left[p(x|y) \parallel q(x|y)\right] = \mathbb{E}_{p(x|y)} \left[\log \frac{p(x|y)}{q(x|y)}\right] \tag{14}$$

If y is not fixed but random we then take an expectation:

$$\mathbb{E}_{p(y)} \left[\mathrm{KL}\left[p(x|y) \parallel q(x|y)\right]\right] = \mathbb{E}_{p(x|y)p(y)} \left[\log \frac{p(x|y)}{q(x|y)}\right] \tag{15}$$

$$= \mathbb{E}_{p(x,y)} \left[\log \frac{p(x|y)}{q(x|y)}\right] \tag{16}$$

$$= \mathbb{E}_{p(x,y)} \left[\log p(x|y) - \log q(x|y)\right] \tag{17}$$

rearranging the sides gives

$$\mathbb{E}_{p(x,y)} \left[\log q(x|y)\right] = \mathbb{E}_{p(x,y)} \left[\log p(x|y)\right] - \mathbb{E}_{p(y)} \left[\mathrm{KL}\left[p(x|y) \parallel q(x|y)\right]\right] \tag{18}$$

$$\leq \mathbb{E}_{p(x,y)} \left[\log p(x|y)\right] \tag{19}$$

where the last line exploits the fact that the KL divergence is always non-negative. Plugging in $x = \theta; y = (h_T; \pi)$ yields the lower bound:

$$\mathbb{E}_{p(\theta,h_T,\pi)}\left[\log q(\theta|h_T,\pi)\right] \leq \mathbb{E}_{p(\theta,h_T,\pi)}\left[\log p(\theta|h_T,\pi)\right] \tag{20}$$

For a known policy π this becomes:

$$\mathbb{E}_{p(\theta,h_T|\pi)}\left[\log q(\theta|h_T,\pi)\right] \leq \mathbb{E}_{p(\theta,h_T|\pi)}\left[\log p(\theta|h_T,\pi)\right] \tag{21}$$

Adding the prior entropy to both sides yields:

$$\mathbb{E}_{p(\theta,h_T|\pi)}\left[\log q(\theta|h_T,\pi)\right] + H\left[p(\theta)\right] \leq \mathbb{E}_{p(\theta,h_T|\pi)}\left[\log p(\theta|h_T,\pi)\right] + H\left[p(\theta)\right]. \tag{22}$$

Finally, plugging in Equations (7) and (13) completes the proof:

$$sCEE(\pi,T) \leq EIG(\pi,T) \tag{23}$$

■

A.2 Proof of corollaries

In the main paper we state 2 corollaries of the above theorem:

Corollary 2 *The bound is tight if and only if* $p(\theta|h_T,\pi) = q(\theta|h_T,\pi)$

Corollary 3 *The bias of the sCEE estimator is* $-\mathbb{E}_{h_T}\left[\mathrm{KL}\left[p(\theta|h_T,\pi) \parallel q(\theta|h_T,\pi)\right]\right]$

If we subtract the lower bound from the EIG we get the difference:

$$\mathbb{E}_{p(\theta,h_T|\pi)}\left[\log p(\theta|h_T,\pi)\right] - \mathbb{E}_{p(\theta,h_T|\pi)}\left[\log q(\theta|h_T,\pi)\right]. \tag{24}$$

From Equation (18) it follows that this difference is

$$-\mathbb{E}_{h_T}\left[\mathrm{KL}\left[p(\theta|h_T,\pi) \parallel q(\theta|h_T,\pi)\right]\right] \tag{25}$$

Since the KL divergence is always non-negative, this difference is 0 and the bound is tight if and only if $\mathrm{KL}\left[p(\theta|h_T,\pi) \parallel q(\theta|h_T,\pi)\right] = 0$ for all realisations of h_T. This establishes both corollaries.

A.3 Proof of convergence

In the main paper we make the claim that a Monte Carlo estimator of the sCEE converges at a rate of $O(\frac{1}{\sqrt{n}})$, where n is the number of MC samples. Since the prior is known, we can rely on standard MC convergence proofs for the prior entropy component. Thus we need only worry about a convergence proof for estimating the cross-entropy component

$\mathbb{E}_{p(\theta,h_T|\pi)}[\log q(\theta|h_T,\pi)]$. We denote the cross-entropy as $H[p(\theta|h_T,\pi), q(\theta|h_T,\pi)]$ and the MC estimator as

$$\hat{H}[p(\theta|h_T,\pi), q(\theta|h_T,\pi)] = \frac{1}{n}\sum_{i=1}^{n} -\log q(\theta^i|h_T^i,\pi) \tag{26}$$

According to Theorem 5.1 of McAllester and Stratos (2020), if there is a minimum log-likelihood F_{max} such that $\log q(\theta|h_T,\pi) \geq F_{max}$, then with probability at least $1-\delta$ we have that

$$|H[p(\theta|h_T,\pi), q(\theta|h_T,\pi)] - \hat{H}[p(\theta|h_T,\pi), q(\theta|h_T,\pi)]| \leq F_{max}\sqrt{\frac{\log(\frac{2}{\delta})}{2n}} \tag{27}$$

Thus the MC estimator converges to the true sCEE with high probability at the desired rate of $O(\frac{1}{\sqrt{n}})$.

Appendix B. Relationship between sCEE and sACE

Foster et al. (2021) propose in the appendices a lower bound EIG estimator that relies on a parameterised proposal distribution that approximates the posterior $p(\theta|h_T)$. They called this the *sequential Adaptive Constrastive Estimation* (sACE):

$$\mathbb{E}_{p(\theta_0,h_T|\pi)q(\theta_{1:L};h_T)}\left[\log \frac{p(h_T|\theta_0,\pi)}{\frac{1}{L+1}\sum_{l=0}^{L}\frac{p(h_T|\theta_l,\pi)p(\theta_l)}{q(\theta_l;h_T)}}\right] \tag{28}$$

This is a contrastive bound where the contrastive samples are distributed according to the proposal distribution $\theta_{1:l} \sim q(\theta|h_T)$. The construction and proof assume a minimum of 1 contrastive sample. However, if we set $L=0$ in this expression, the sampling of contrastive samples from $q(\theta|h_T)$ disappears and we get:

$$\mathbb{E}_{p(\theta_0,h_T|\pi)}\left[\log \frac{p(h_T|\theta_0,\pi)}{\frac{1}{0+1}\sum_{l=0}^{0}\frac{p(h_T|\theta_l,\pi)p(\theta_l)}{q(\theta_l;h_T)}}\right] = \mathbb{E}_{p(\theta_0,h_T|\pi)}\left[\log \frac{p(h_T|\theta_0,\pi)}{\frac{1}{0+1}\frac{p(h_T|\theta_0,\pi)p(\theta_0)}{q(\theta_0;h_T)}}\right] \tag{29}$$

$$= \mathbb{E}_{p(\theta_0,h_T|\pi)}\left[\log \frac{\cancel{p(h_T|\theta_0,\pi)}}{\cancel{p(h_T|\theta_0,\pi)}}\frac{q(\theta_0;h_T)}{p(\theta_0)}\right] \tag{30}$$

$$= \mathbb{E}_{p(\theta_0,h_T|\pi)}\left[\log \frac{q(\theta_0;h_T)}{p(\theta_0)}\right], \tag{31}$$

which is equivalent to the sCEE. Note that by avoiding the need for contrastive samples, the sCEE gains a considerable computational advantage. In the RL setting, the rewards depend on $q(\theta|h_T)$ and hence need to be recomputed every time q is updated. With the sACE estimator, this recomputation requires resampling the contrastive samples, increasing the computational effort by a factor of $O(L)$. Indeed, depending on memory constraints, it may not be possible to recompute an entire batch of rewards in a single vectorised operation. With the sCEE, however, reward recomputation requires only a single neural network pass.

In addition to the computational benefits, sCEE has the further advantage that it is compatible with implicit likelihood models, wherease sACE requires explicit models, since it includes the term $p(h_T|\theta_0,\pi)$ in the numerator.

Algorithm 1: sCEE-RL

Input: $\mathcal{M}$: SED-MDP, L_π: policy loss function, $L_\mathcal{C}$: critic loss function
Initialise replay buffer $\mathcal{B}$
while convergence criterion not reached **do**
  Generate rollouts $(s_{0:T}, a_{0:T}, \theta)^{1:N}$ using $\mathcal{M}$ and π and push to $\mathcal{B}$.
  Sample mini-batch of size mb from $\mathcal{B}$
  Compute posterior loss $L_q = -\frac{1}{mb}\sum_{i=1}^{mb} \log q_\kappa(\theta^i | B^i_{\psi,t})$
  Take gradient step to minimise $\nabla_\kappa L_q$
  Compute rewards for mini-batch using Equation (6)
  Use mini-batch and rewards to compute L_π and $L_\mathcal{C}$
  Take gradient step to minimise $\nabla_\phi L_\pi$ and $\nabla_\chi L_\mathcal{C}$
end while

Appendix C. Reinforcement learning algorithm

Blau et al. (2022) have shown that the problem of learning an experiment design policy can be formulated as a special case of a MDP called the SED-MDP. We follow their formulation for the reinforcement learning algorithm in this paper, with the main difference being the use of the sCEE reward and consequently the use of an approximate proposal $q_\kappa(\theta|h_T)$ parameterised as a conditional normalising flows neural network (Winkler et al., 2019) with parameters κ. This posterior network $q_\kappa(\cdot)$ can be updated by using the same mini-batches to maximise the log-likelihood of the observations under our posterior model. Note that this means rewards are now no longer fixed but depend on $q_\kappa(\cdot)$, and change with every update of κ. The computational cost thus incurred can be minimised by lazy evaluation (Bloss et al., 1988): we only update each reward when we are about to use it to update the policy and critic networks of the RL agent. The procedure is summarised in Algorithm 1, and we give more details about this procedure in the following sections.

C.1 Simultaneous policy and reward learning

We propose to learn the design policy network π_ϕ and the proposal distribution q_κ from data simultaneously. Since the reward function depends on q_κ, and the objective function of q_κ in turn depends on π_ϕ, this leads to inherent instability, similar to the "deadly triad" that is often observed in value-based reinforcement learning (Van Hasselt et al., 2018). We therefore apply several stabilisation mechanisms to prevent the neural network estimators from diverging.

Target posterior network: similar to the use of target Q-networks as introduced by Lillicrap et al. (2016), we maintain a primary posterior network q_κ and a target network q'_κ. The primary network q_κ is updated using gradient descent in every iteration of the algorithm, but is not used directly to compute rewards. Instead, the target network q'_κ is used to compute Equation (6), and its weights are periodically updated to maintain a moving average:

$$\kappa' \leftarrow \kappa' \cdot (1-\tau) + \kappa \cdot \tau \tag{32}$$

where $\tau \in (0, 1)$ is a constant controlling the rate of change.

Fixed initial posterior: the reward definition of Equation (6) assigns each experiment its own (estimated) information gain. The return of an entire trajectory is a telescoping sum that reduces to $\log q(\theta|B_{\psi,T}) - \log q(\theta|B_{\psi,0})$, and the expected return over infinitely many trajectories recovers the sCEE. Therefore, the component $q(\theta|B_{\psi,0})$ of the first reward r_0 is the only contributor to the prior entropy term $H[p(\theta)]$ of the sCEE. Since this term is constant w.r.t. all networks, we can simply ignore it when training as it does not change the optimal policy. Furthermore, learning the correct estimator for $q(\theta|B_{\psi,0})$ that maps the null inputs to the prior $p(\theta)$ can be challenging. Therefore instead of learning this mapping for the empty first state, we assigned it a fixed value of $\log q(\theta|B_{\psi,0}) \coloneqq 0$.

Appendix D. Normalising flows on the probability simplex

If we have a random variable with support on the canonical (open) simplex Δ_{k-1} rather than in $\mathcal{R}^k$, additional caution is required for fitting a normalising flow to this RV. Since the k^{th} dimension of the RV is fully determined by the first $k-1$ dimensions, the NF is free to fit this dimension with extremely high confidence, leading to an overestimation of log-likelihood of the entire RV.

The fix to this issue is rather involved. First, we exclude the k^{th} dimension as input to the NF. Then, at the penultimate layer of a normalising flow, it implements the diffeomorphism $\mathcal{F} : \mathcal{R}^{k-1} \to \mathcal{R}^{k-1}$ i.e. the base distribution is a standard Gaussian and the resulting distribution can have support in the entire real space. Now we add a series of bijections that will produce a map $\mathcal{G} : \mathcal{R}^{k-1} \to \Delta_{k-1}$. Note that it is not enough simply to concatenate $1 - \sum_{i=1}^{k-1} \mathcal{F}(x)_i$ with the intermediate vector $\mathcal{F}(x)$ because we are not guaranteed that $0 \leq \mathcal{F}(x)_k \leq 1 \; \forall k$ and that $\sum_{i=1}^{k-1} \mathcal{F}(x)_i \leq 1$. First we must transform the output to ensure these properties:

$$u = \mathcal{F}(x) \tag{33}$$

$$v_i = \sigma(u_i) \tag{34}$$

$$w_i = v_i \left(1 - \sum_{j=1}^{i-1} w_j\right) \quad \forall i \in [1, k-1] \tag{35}$$

$$\theta_i = \frac{w_i}{1-\epsilon}. \tag{36}$$

Equation (34) projects $\mathcal{R}^{k-1}$ to the semi-open box $[0,1)^{k-1}$. Equation (35) projects this box to the $k-1$ dimensional simplex $\mathbf{s} = \{x : \sum_{i=1}^{k-1} x_i < 1 \text{ and } 0 \leq x_k < 1 \; \forall i \in [1, k-1]\}$. This non-canonical simplex is in fact the equivalent of projecting the k-dimensional canonical simplex Δ_{k-1} down to $k-1$ dimensions. The simplex $\mathbf{s}$ can be lifted to Δ_{k-1} by assigning $w_k = 1 - \sum_{j=1}^{k-1} w_j$. However, we won't include this in the mapping $\mathcal{G}$ because it makes the Jacobian low-rank and hence the inverse ill-defined. To avoid floating-point errors, each element of the RV actually has to be in the range $[\epsilon, 1-\epsilon]$ where ϵ is the machine epsilon. Equation (36) maps between this space and the actual canonical simplex.

The corresponding log-det-Jacobians are:

$$\sum_{i=1}^{k-1} \log(0.99 \cdot v_i(1-v_i)), \tag{37}$$

$$\sum_{i=1}^{k-1} \log(1 - \sum_{j=1}^{i-1} w_j), \tag{38}$$

$$(1-k) \cdot \log(1-\epsilon). \tag{39}$$

The inverse $\mathcal{G}^{-1}$ can be written compactly as:

$$u_i = \sigma^{-1}\left(\frac{(1-\epsilon)\cdot\theta_i}{1-\sum_{j=1}^{i-1}(1-\epsilon)\cdot\theta_j}\right) \quad \forall i \in [1, k-1]. \tag{40}$$

Appendix E. Experiment details

This appendix describes the probabilistic models, hyperparameters, and all other details relating to the experiment design problems appearing in the paper.

We implemented our algorithm using Pyro (Bingham et al., 2018) and normflows Stimper et al. (2023) along with the Garage framework (Garage Contributors, 2019) and the REDQ algorithm (Chen et al., 2021) for reinforcement learning. For complete details about algorithms and hyperparameters, see Appendix F. To evaluate the EIG we used contrastive estimators with $L = 1\text{E}8$, a number of contrastive samples that is impractical for learning, but achieves better estimation than sCEE in most problems we investigated.

E.1 Synthetic data – EIG for conjugate priors

Given the theoretical results about sCEE and contrastive bounds, we expect that sCEE should perform well in situations where the EIG is large and $q_\kappa(\cdot)$ is easy to learn. To assess this, we evaluate the estimator on 7 experimental tasks which allow us to know the true EIG in closed form. The priors are isotropic Gaussians of the form $\mathcal{N}(\mu_0, \sigma_0\mathbf{I}_k)$ and the likelihoods are similarly Gaussian with known isotropic covariance $\sigma\mathbf{I}_k$, where k is the number of dimensions. Each task has an experimental budget of $T = 10$ experiments. Thus we can manipulate k, σ_0 and σ to create tasks where the EIG of the optimal design is known exactly.

Table 1 enumerates these tasks, alongside the optimal EIG and the estimates of sCEE and sPCE with different numbers of contrastive samples. As can be seen from the left-most columns of the table, when the EIG is small enough, sPCE can provide a better estimate than sCEE (note that the sPCE at times slightly overestimates the EIG due to variance in estimating the expectation with Monte Carlo samples). However, as the EIG becomes large relative to $\log(L)$, the underestimation of sPCE becomes more severe, and for the right-most columns all sPCE variants have reached their upper limit. Meanwhile, sCEE consistently provides good estimates regardless of the magnitude of the EIG. It should be noted, however, that this is in part because the posterior is easy to learn from data. A more complex posterior, or less training, would worsen the underestimation.

For an isotropic Gaussian prior $\mathcal{N}(\mu_0, \sigma_0\mathbf{I_k})$ and Gaussian likelihood with known isotropic covariance $\sigma\mathbf{I_k}$, the posterior after n observations is an isotropic Gaussian with covariance:

$$\Sigma_{post} = (\sigma_0^{-1}\mathbf{I}_k + n\sigma^{-1}\mathbf{I}_k)^{-1} \tag{41}$$

$$= (\sigma_0^{-1} + n\sigma^{-1})^{-1}\mathbf{I}_k \tag{42}$$

The mean of the posterior is unimportant to us as it does not affect the entropy:

$$H_{post} = \frac{k}{2} + \frac{k}{2}\log(2\pi) + \frac{1}{2}\log(|\Sigma_{post}|) \tag{43}$$

$$= \frac{k}{2} + \frac{k}{2}\log(2\pi) - \frac{k}{2}\log(\sigma_0^{-1} + n\sigma^{-1}). \tag{44}$$

Therefore the entropy is independent of the designs and we can compute the entropy of the "optimal" policy by subtracting the posterior entropy from the prior entropy:

$$I_n(\pi) = H[\mathcal{N}(\mu_0, \sigma_0\mathbf{I_k})] - H_{post} \tag{45}$$

$$= \cancel{\frac{k}{2}} + \cancel{\frac{k}{2}\log(2\pi)} + \frac{k}{2}\log(\sigma_0) - \cancel{\frac{k}{2}} - \cancel{\frac{k}{2}\log(2\pi)} + \frac{k}{2}\log(\sigma_0^{-1} + n\sigma^{-1}) \tag{46}$$

$$= \frac{k}{2}(\log(\sigma_0) + \log(\sigma_0^{-1} + n\sigma^{-1})) \tag{47}$$

$$= \frac{k}{2}\log(1 + n\frac{\sigma_0}{\sigma}) \tag{48}$$

Thus we can create an EIG estimation problem with an EIG of our choice by setting k, n, σ_0 and σ appropriately. In our experiments sCEE was trained for $10,000$ epochs, and was exposed to $10,000$ data points in each epoch. Each estimator was evaluated using $1,000$ Monte Carlo samples.

E.2 Constant elasticity of substitution

We evaluate a design problem in behavioural economics where we must estimate the parameters of a Constant Elasticity of Substitution (CES) utility function Baltas (2001). In this experiment economic agents compare 2 baskets of goods and give a rating on a sliding scale from 0 to 1. Each basket consists of k different goods with different value. We set $k = 3$.

The outcome is the relative preference of a test subject in the range $[0, 1]$, as determined by the agent's CES utility function, and the specific values of its parameters $\theta = \{\rho, \alpha, u\}$, with $\rho \in [0, 1]$, $\alpha \in \Delta_3$ and $u > 0$.

The designs are vectors $d = (x, x')$ where $x, x' \in [0, 100]^k$ are the baskets of goods. The latent parameters of the likelihood and their priors are:

$$\rho \sim Beta(1, 1) \tag{49}$$

$$\alpha \sim Dirichlet(\mathbf{1}_k) \tag{50}$$

$$\log u \sim \mathcal{N}(1, 3). \tag{51}$$

The probabilistic model is:

$$U(x) = \left(\sum_i x_i^\rho \alpha_i \right)^{1/\rho} \tag{52}$$

$$\mu_\eta = (U(x) - U(x'))u \tag{53}$$

$$\sigma_\eta = (1 + ||x - x'||)\tau \cdot u \tag{54}$$

$$\eta \sim \mathcal{N}(\mu_\eta, \sigma_\eta^2) \tag{55}$$

$$y = clip(sigmoid(\eta), \epsilon, 1 - \epsilon), \tag{56}$$

In our experiments we used the following hyperparameters:

Parameter	Value
k	3
τ	0.005
ϵ	2^{-22}

E.3 Prey population

To evaluate our method in tasks with discrete design spaces, we consider the prey population problem from Moffat et al. (2020). Designs are the initial population of a prey species, limited to the discrete interval $\mathcal{D} = 1, 2, \ldots, 300$. The outcome is the number of individual who were consumed by predators at the end of a 24 hour period, based on the attack rate and handling time of the predators.

In this experiment an initial population of prey animals is left to survive for $\mathcal{T}$ hours, and we measure the number of individuals consumed by predators at the end of the experiment. The designs are the initial populations $d = N_0 \in 1, 2, \ldots, 300$. The latent parameters and priors are:

$$\log a \sim \mathcal{N}(-1.4, 1.35) \tag{57}$$

$$\log T_h \sim \mathcal{N}(-1.4, 1.35), \tag{58}$$

where a represents the attack rate and T_h is the handling time.

The population changes over time according to a Holling's Type III model, which is a differential equation:

$$\frac{dN}{d\tau} = -\frac{aN^2}{1 + aT_h N^2}. \tag{59}$$

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

$$p_\mathcal{T} = \frac{d - N_\mathcal{T}}{d} \tag{60}$$

$$y \sim Binom(d, p_\mathcal{T}). \tag{61}$$

We used a simulation time of $\mathcal{T} = 24$ hours.

E.4 Source location

In this experiment there are n sources embedded in k-dimensional space that emit independent signals. the designs are the co-ordinates at which to measure signal intensity, and we restrict the space to $d \in [-4, 4]^k$. The total intensity at any given co-ordinate d in the plane is given the sum of individual signals:

$$\mu(\theta, d) = b + \sum_i \frac{1}{m + ||\theta_i - d||^2}, \tag{62}$$

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

$$\theta_i \sim \mathcal{N}(0, I); \quad \log y|\theta, d \sim \mathcal{N}(\log(\mu(\theta, d), \sigma), \tag{63}$$

i.e. the prior is unit Gaussian and we observe the log of the total signal intensity with some Gaussian observation noise σ. The hyperparameters we used are

Parameter	Value
n	2
k	2
b	$1\text{E}-1$
m	$1\text{E}-4$
σ	0.5

We trained policies to design sequences of $T = 30$ experiments. Table 2 shows that the lower bound estimate for our proposed method exceeds the upper bound estimate for all other methods. As with the CES problem, we can examine the posteriors obtained from $q_\kappa(\theta|h_T)$. This time, however, we plotted the marginals $q(x_1, x_2)$ and $q(y_1, y_2)$, i.e. the x and y co-ordinates of the 2 signal sources, using x_1 (respectively y_1) as the X-axis and x_2 (respectively y_2) as the Y-axis. The sources are exchangeable, i.e. $p(y|d, s_1 = (x_1, y_1), s_2 = (x_2, y_2)) = p(y|d, s_1 = (x_2, y_2), s_2 = (x_1, y_1))$. Therefore the marginals of the true Bayesian posterior, $p(x_1, x_2)$, should be symmetric w.r.t. the line $x_2 = x_1$ in the x_1x_2-plane. Indeed, the plots in Appendix I exhibit this symmetry, which has been learned entirely from data, without providing any inductive bias.

Appendix F. Algorithm experimental details

This appendix provides the implementation details for all design of experiment algorithms used in the paper.

F.1 RL-sCEE

We used the implementation of REDQ from Blau et al. (2022) as the basis of our algorithm, although we limited the ensemble size to $N = 2$. Normalising flows were implemented using

the normflows Stimper et al. (2023) package, which we extended to create a conditioned version of realNVP Dinh et al. (2017). In every experiment we used a normalising flow with 6 layers, and the parameter map is a 2-layer neural network with sizes $(128, 128)$. Both normalising flows and policies use a permutation invariant representation similar to Ivanova et al. (2021), including a single self-attention layer with 8 attention heads.

Additional hyperparameters are listed in the table below, and are largerly derived from Blau et al. (2022):

Parameter	Source Location	CES	Prey Population
training iterations	1E5	1E5	2E4
T	30	10	10
γ	0.9	0.9	0.95
τ	$1\text{E}-3$	$5\text{E}-3$	$1\text{E}-2$
policy learning rate	$1\text{E}-4$	$3\text{E}-4$	$1\text{E}-4$
critic learning rate	$3\text{E}-4$	$3\text{E}-4$	$1\text{E}-3$
buffer size	1E7	1E7	1E6

F.2 RL-sPCE

We used the implementation of Blau et al. (2022), which is available at `https://github.com/csiro-mlai/rl-boed`. We kept all hyperparameters and network architectures the same, with the exception of adding a self-attention layer to the policy network. This layer is identical to the one described in the previous section. We did not find that adding attention lead to significant change in performance, but included it in order to maintain a fair comparison with the RL-sCEE implementation.

In particular, we used $L = 1\text{E}5$ contrastive samples for training. Not only is it the value used by Blau et al. (2022), but is also pushing the limits of the number of samples that can be used in a reasonable amount of time. Since tens of millions of simulated experiments have to be run to train a single agent, we must leverage vectorisation over multiple sequences of experiments in parallel. Although in the evaluation we used $L = 1\text{E}8$ samples, this only allows a single experiment at a time to fit in a GPU, and requires multiple seconds per experiment. It would require several years to train a single agent in this manner.

F.3 DAD and iDAD

For these baselines we used the implementations of the original papers, which are available at `https://github.com/ae-foster/dad` and `https://github.com/desi-ivanova/idad`, respectively. We kept the default hyperparameters of those implementations. The only exception is for iDAD on the source location problem, which we found unstable for a sequence of $T = 30$ experiments. We therefore used early stopping, and stopped learning at $40k$ iterations instead of the original $100k$.

F.4 SMC-ED

We used the implementation made available in `https://github.com/csiro-mlai/rl-boed`, which in turn uses the R language implementation of Moffat et al. (2020) and executes it from within a Python script by using the *rpy*2 bindings. The original R code is available at `https://github.com/haydenmoffat/sequential_design_for_predator_prey_experiments`.

Appendix G. Hardware details

SMC-ED experiments were run on a desktop machine with an Intel i7-10610U CPU and no GPU. All other experiments were run in a high-performance computing cluster, using a single node each with 4 cores of an Intel Xeon E5-2690 CPU and an Nvidia Tesla P100 GPU.

Appendix H. Ablation Study

To evaluated the stabilisation mechanisms incorporated in the implementation of RL-sCEE, we conduct an ablation study where we remove either the target posterior network, the fixed initial posterior, or both. The results can be seen in Figure 1, with each variant replicated 10 times, using common random seeds between different variants (e.g. the blue trendline labeled "0" represents the same random seed in all 4 plots).

It is clear that the removal of the target network causes significant degradation in performance, with many replications converging to a lower final performance or even peaking early and then decreasing in EIG. On the other hand, the use of a fixed initial posterior doesn't seem to offer a clear advantage over a learned one.

Appendix I. Additional results

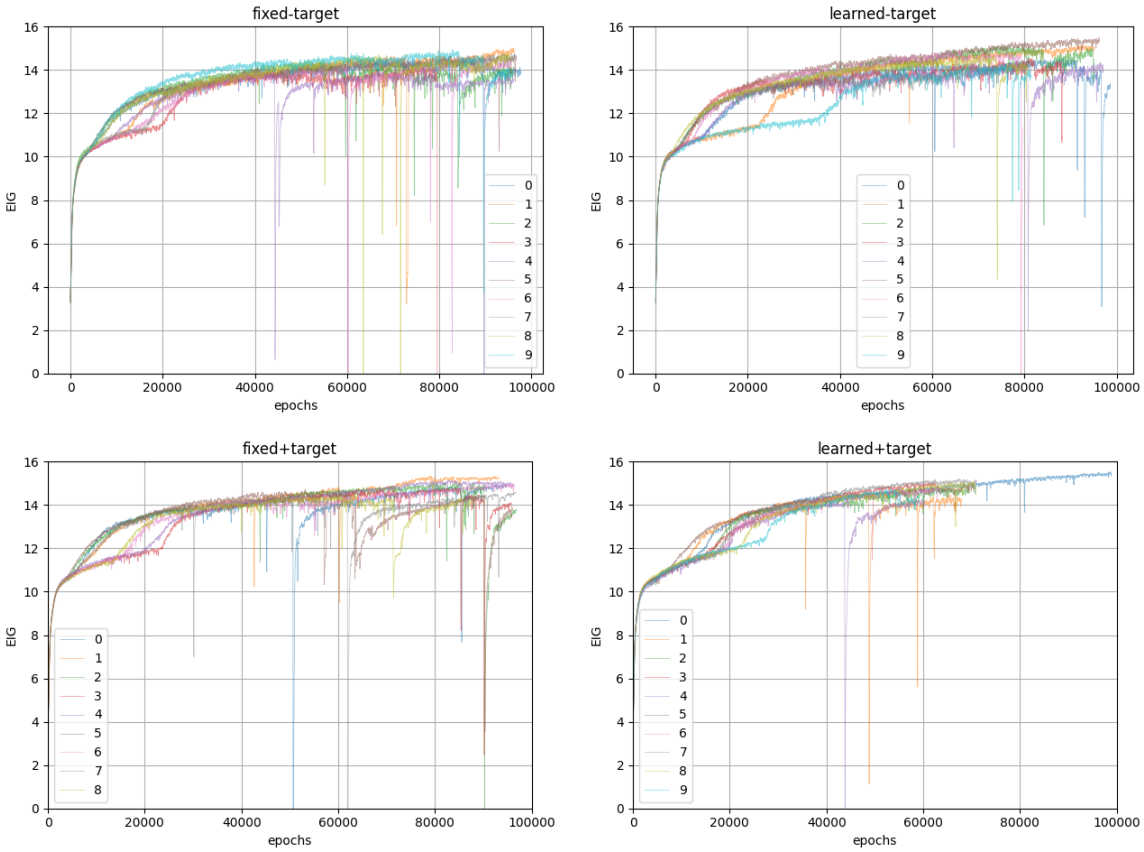

Figure 1: Ablation studies for the stabilisation mechanisms.

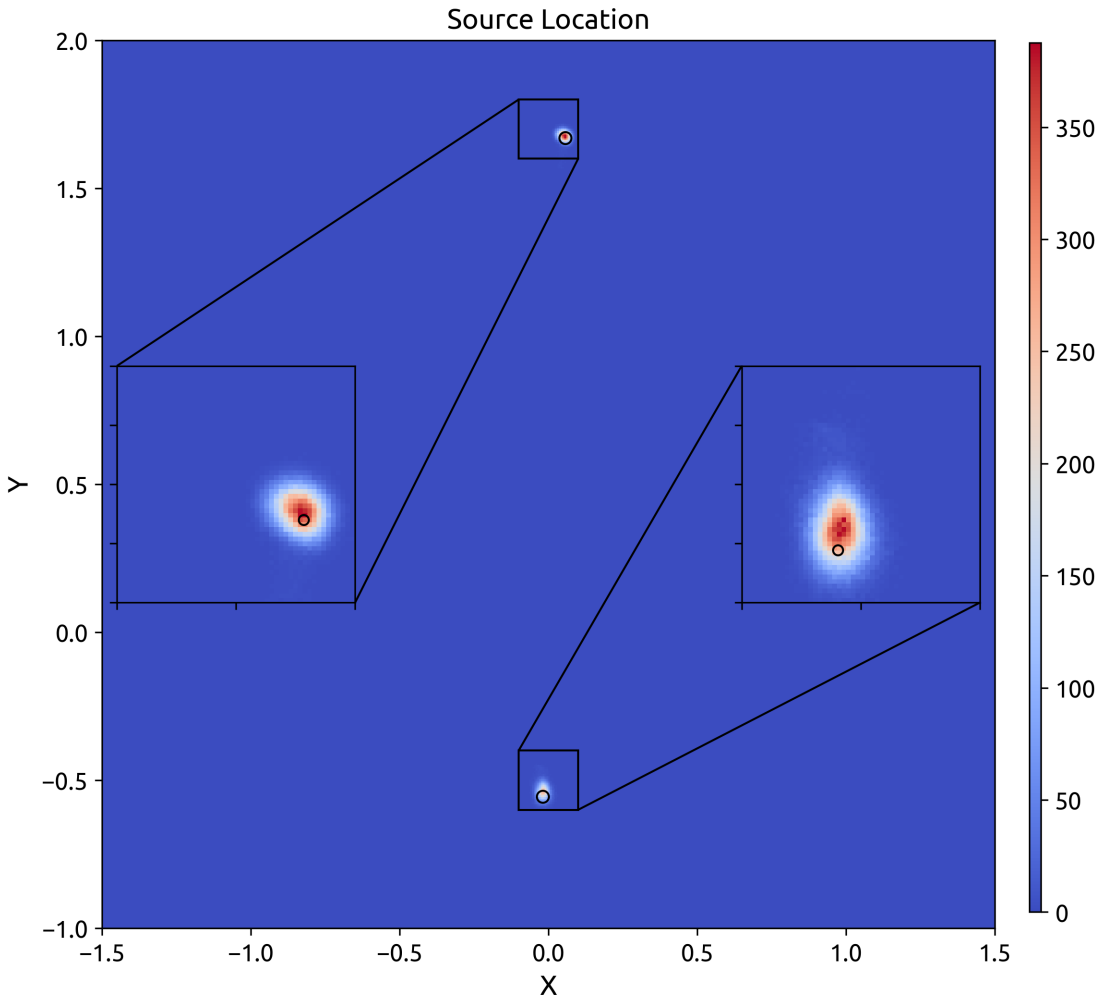

Figure 2: Posterior for the source location problem. Computed from 1E5 samples. Black rings denote the true co-ordinates of signal sources.