# Bayesian Optimal Experimental Design for Simulator Models of Cognition

**Simon Valentin***
University of Edinburgh

**Steven Kleinegesse***
University of Edinburgh

**Neil R. Bramley**
University of Edinburgh

**Michael U. Gutmann**
University of Edinburgh

**Christopher G. Lucas**
University of Edinburgh

## Abstract

Bayesian optimal experimental design (BOED) is a methodology to identify experiments that are expected to yield informative data. Recent work in cognitive science considered BOED for computational models of human behavior with tractable and known likelihood functions. However, tractability often comes at the cost of realism; simulator models that can capture the richness of human behavior are often intractable. In this work, we combine recent advances in BOED and approximate inference for intractable models, using machine-learning methods to find optimal experimental designs, approximate sufficient summary statistics and amortized posterior distributions. Our simulation experiments on multi-armed bandit tasks show that our method results in improved model discrimination and parameter estimation, as compared to experimental designs commonly used in the literature.

## 1 Introduction

Computational models provide a means to describe and study natural phenomena, with important applications ranging from understanding the spread of viruses [3] to discovering new molecules [5] and studying climate change [14]. A particular scientific domain where computational models are playing an increasingly prominent role is the study of human behavior. Here, computational models allow us to formalize theories about human cognition and thereby better understand and explain behavior. Often, models that are rich enough to capture realistic behavior have intractable likelihood functions, complicating common scientific tasks such as model comparison, parameter inference and designing informative experiments.

Gathering experimental data is generally costly and time-consuming. Meanwhile, behavioral experiments are typically designed based on prior work, intuitions and heuristics, which may yield data that are poor for resolving the researchers' theoretical questions. In particular, as psychological theories become more complex, designing informative experiments can be a difficult task.

As a potential solution to this problem, the field of Bayesian optimal experimental design (BOED) formalizes the design of experiments and treats it as an optimization problem. Concretely, the aim is to maximize a utility function that captures the worth of a particular experimental design. This utility function, however, usually depends on the posterior distribution and is therefore intractable for all but the simplest heuristic models of cognition.

Prior work in cognitive science has demonstrated the applicability and usefulness of BOED for parameter estimation and model comparison, albeit for simple models with known and tractable likelihood functions [e.g., 9, 21, 10]. There is, however, a lack of work that considers more realistic cognitive models in the context of BOED.

*Equal contribution. Correspondence to: `s.valentin@ed.ac.uk`.

NeurIPS 2021 AI for Science Workshop.

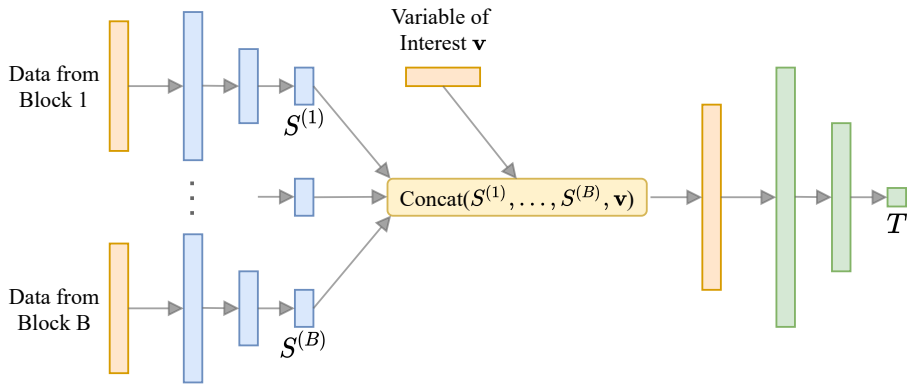


Figure 1: Neural network architecture for behavioral experiments. For each block of data we have a small sub-network (shown in blue) that outputs summary statistics $S$. These are concatenated with the variable of interest $\boldsymbol{v}$ and passed to a larger neural network (shown in green).

Nonetheless, in the recent years there have been significant methodological advances in BOED for simulator models [e.g. 4, 6, 11]. We here leverage these advances and utilize the MINEBED method of Kleinegesse and Gutmann [6]. This method performs BOED by maximizing a lower bound on the expected information gain at a particular experimental design. This lower bound is estimated by training a neural network on data generated by the computational models under consideration.

**Contributions** We demonstrate the applicability of modern machine learning and BOED methods to any computational models of cognition from which we can simulate data. To deal with data common in behavioral experiments, we construct a bespoke neural network architecture, visualized in Figure 1, that is trained on simulated data from our computational models. In addition to finding optimal designs and amortized posterior distributions, this allows us to extract approximate sufficient statistics at the same time. As a case study, we extend previously studied models of behavior in multi-armed bandit tasks to be more flexible and realistic, which results in intractable likelihood functions. In our simulation study, we demonstrate how to find optimal experimental designs for model discrimination and parameter estimation. We validate our results by comparing the performance of optimal and commonly-used experimental designs in statistical inference.

## 2 Models of Human Behavior in Bandit Tasks

We turn to models of human sequential decision making in multi-armed bandit tasks as an important area of research in cognitive science [e.g., 18]. Multi-armed bandits have a long history in statistics and machine learning [e.g., 13, 19, 18] and formalize a common class of decision problems — repeatedly choosing between a set of options under uncertainty. Here, we select three computational models that have previously been proposed as accounts of human choice behavior. We generalize them to accommodate richer and more realistic behavioral patterns, which results in intractable likelihoods.

Each of our computational models defines a generative model $\mathbf{y} \sim p(\mathbf{y}|\boldsymbol{\theta}, \mathbf{d})$, where the observed data $\mathbf{y}$ consist of the chosen bandit arms and received rewards in a particular bandit task. The model parameters $\boldsymbol{\theta}$ are treated as random variables with known prior distributions. The design vector $\mathbf{d}$ that we wish to optimize consists of the reward probabilities of the Bernoulli distributions associated with the arms in the bandit task. We provide high-level intuitions for our computational models below, and give detailed explanations in the Appendix.

**Win-Stay Lose-Thompson-Sample (WSLTS)** Here, we propose Win-Stay Lose-Thompson-Sample (WSLTS) as an amalgamation of Win-Stay Lose-Shift [WSLS; 13] and Thompson Sampling [20]. Upon observing a reward, the agent re-selects the previously chosen arm with a certain probability, and shifts to another arm if there was no reward (with a different probability). Here, the agent performs Thompson Sampling from a reshaped posterior instead of shifting to another arm uniformly at random, as would be the case in standard WSLS.

**Auto-regressive $\varepsilon$-Greedy (AEG)** $\varepsilon$-Greedy [e.g., 19] is a ubiquitous method in reinforcement learning in which the agent selects the arm with the highest expected reward with $p = 1 - \epsilon$ and a uniformly selected arm otherwise. Here, we propose Auto-regressive $\varepsilon$-Greedy (AEG) as a generalization of $\varepsilon$-Greedy, where the probability of selecting the previous arm is controlled by a separate parameter. This allows for modeling people's tendency towards auto-regressive behavior [16].

**Generalized Latent State (GLS)** Lee et al. [8] proposed a latent state model for bandit tasks whereby a learner can be in either an *explore* or an *exploit* state and switch between these as they go through the task. Here, we propose the Generalized Latent State (GLS) model, which unifies and extends latent-state and latent-switching models, previously studied in Lee et al. [8], allowing for more flexible and structured transitions (see the Appendix for more details).

## 3 Methods

**BOED Background** In BOED, we need to construct a utility function $U(\mathbf{d})$ that describes the worth of an experimental design $\mathbf{d}$. Finding an optimal design $\mathbf{d}^*$ then equates to maximizing this utility function, i.e. $\mathbf{d}^* = \arg\max_{\mathbf{d}} U(\mathbf{d})$. A prominent and principled choice of utility function is the *mutual information* (MI), which is equivalent to the expected information gain,

$$U(\mathbf{d}) = \text{MI}(\boldsymbol{v}; \mathbf{y}|\mathbf{d}) \coloneqq \mathbb{E}_{p(\mathbf{y}|\boldsymbol{v},\mathbf{d})p(\boldsymbol{v})} \left[ \log \frac{p(\boldsymbol{v}|\mathbf{y},\mathbf{d})}{p(\boldsymbol{v})} \right], \tag{1}$$

where $\boldsymbol{v}$ is a variable of interest that we wish to estimate.

Intuitively, mutual information quantifies the amount of information our experiment is expected to provide about the variables of interest. Unfortunately, computing the MI exactly is generally intractable, a difficulty that is exacerbated for intractable models. Below, we describe how the MI can be estimated and optimized effectively, by using the MINEBED methodology of Kleinegesse and Gutmann [6]. More details on the training procedure can be found in the Appendix.

**Variable of Interest** In this work we focus on two common scientific goals: (1) the task of model discrimination (MD), i.e. distinguishing between competing cognitive models, and (2) the task of parameter estimation (PE) of a given cognitive model. For MD, the variable of interest $\boldsymbol{v}$ in Equation 1 is a discrete model indicator $m$ that determines from which competing model the data originates. For PE, the variable of interest is the set of parameters $\boldsymbol{\theta}_m$ of a particular model $m$.

**MI Lower Bound** We are ultimately interested in maximizing the MI with respect to the designs $\mathbf{d}$, not estimating it accurately everywhere in the design domain. Recent advances in BOED for simulator models thus advocate the use of cheaper MI lower bounds instead [e.g. 4, 6]. We shall here use the MINEBED method of Kleinegesse and Gutmann [6], which works by training a neural network $T_{\boldsymbol{\psi}}(\boldsymbol{v}, \mathbf{y})$ using stochastic gradient-ascent, where $\boldsymbol{\psi}$ are the neural network parameters and the data $\mathbf{y}$ is simulated at design $\mathbf{d}$ with samples from the prior $p(\boldsymbol{v})$.

In particular, the neural network is trained by using the MI lower bound as an objective function (see the Appendix for the exact form of the lower bound).

The resulting trained neural network and the final lower bound estimate can thus be used to compute an estimate of the MI at design $\mathbf{d}$.

**Network Architecture** We propose an effective architecture choice for the neural network $T_{\boldsymbol{\psi}}(\boldsymbol{v}, \mathbf{y})$, devised specifically with applications to behavioral experiments in mind (but may also be effective in other applications). Our architecture, summarized in Figure 1, incorporates sub-networks $\mathbf{S}_{\boldsymbol{\phi}}(\boldsymbol{v}, \mathbf{y})$ for each block of behavioral data in a bandit task. The outputs of these sub-networks are then concatenated and passed as input to the network $T_{\boldsymbol{\psi}}(\boldsymbol{v}, \mathbf{y})$. Following Chen et al. [2], each sub-network is learning approximate sufficient statistics of the data from a particular block in the bandit task.

**Gradient-Free Optimization** The MINEBED method originally optimizes the utility function $U(\mathbf{d})$ by means of gradient-ascent. This is however not possible when the simulated data $\mathbf{y}$ is discrete, as is commonly the case with behavioral data. We thus follow the third experiment in Kleinegesse

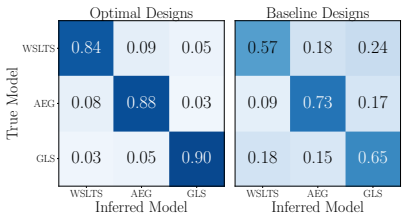


Figure 2: Confusion matrices of the inferred behavioral models, for optimal (left) and baseline (right) designs.

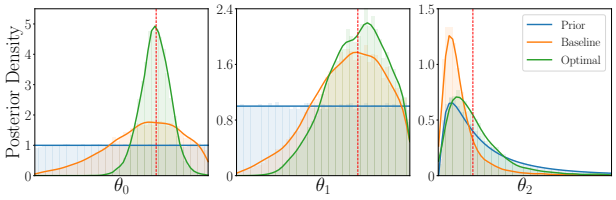


Figure 3: Marginal posterior distributions of the three WSLTS model parameters for optimal (green) and baseline (orange) designs, averaged over 1,000 observations.

and Gutmann [6] and optimize $U(\mathbf{d})$ with respect to $\mathbf{d}$ using Bayesian Optimization (BO). We use a Gaussian Process (GP) as our probabilistic surrogate model with a Matérn-5/2 kernel and Expected Improvement as the acquisition function (these are standard choices, see Shahriari et al. [17] for a review on BO).

## 4 Experiments

In this section we demonstrate the optimization of reward probabilities for multi-armed bandit tasks, with the scientific goals of (1) model discrimination (MD) and (2) parameter estimation (PE). We consider multi-armed bandits with three arms and 30 trials per block, where each block may have different reward probabilities. For the MD task we use two blocks of behavioral data, whereas we use three blocks for the PE task. As part of our methodology, we use relatively small architectures with 2 hidden layers for all neural networks. More information about the experimental settings, descriptions of the algorithms and additional discussions can be found in the Appendix.

**Baseline design** As our baseline designs, we sample all reward probabilities from a Beta$(2, 2)$ distribution, following the, to the best of our knowledge, largest behavioral experiments on bandit problems in the literature with 451 participants [18].

**Model discrimination** The optimal reward probabilities for the MD task found using our methodology are $[0, 0, 0.6]$ for the first block of trials and $[1, 1, 0]$ for the other second block. These optimal designs stand in stark contrast to the usual reward probabilities used in such behavioral experiments, which almost always take non-extreme values [18]. Using the neural network that was trained at that optimal design, we can compute posterior distributions of the model indicator (as described in the appendix). This yields the confusion matrices in Figure 2, which show that our optimal designs result in considerably better model recovery than the baseline designs.

**Parameter estimation** We here discuss the PE results for the WSLTS model; the results for the AEG and GLS model can be found in the appendix. We find that the optimal reward probabilities for the WSLTS model are $[0, 1, 0]$, $[0, 1, 1]$ and $[1, 0, 1]$ for the first, second and third block, respectively. Similarly to the MD task, these optimal designs take extreme values, unlike commonly-used reward probabilities in the literature. In Figure 3 we show posterior distributions of the WSLTS model parameters for optimal and baseline designs. We find that optimal designs yield data that result in considerably improved parameter recovery.

## 5 Conclusions

Our experiments demonstrate that our methodology allows us to effectively design optimal experiments in cognitive science, resulting in considerably better model and parameter recovery than for designs commonly-used in the literature. In particular, the combination of a lower bound on the MI, parameterized by a bespoke neural network, allows us to scale to realistic behavioral experiments. It would be interesting to see how our approach can be adapted to the sequential BOED setting with intractable models [e.g., 7]. Additionally, our experiments only included synthetic data and no real-world data, and therefore it would be useful to apply our approach to real participants as well.

## Acknowledgments

SV was supported by a Principal's Career Development Scholarship, awarded by the University of Edinburgh. SK was supported in part by the EPSRC Centre for Doctoral Training in Data Science, funded by the UK Engineering and Physical Sciences Research Council (grant EP/L016427/1) and the University of Edinburgh.

## References

[1] O. Chapelle and L. Li. An empirical evaluation of thompson sampling. *Advances in neural information processing systems*, 24:2249–2257, 2011.

[2] Y. Chen, D. Zhang, M. U. Gutmann, A. Courville, and Z. Zhu. Neural approximate sufficient statistics for implicit models. In *International Conference on Learning Representations (ICLR)*, 2021.

[3] C. S. Currie, J. W. Fowler, K. Kotiadis, T. Monks, B. S. Onggo, D. A. Robertson, and A. A. Tako. How simulation modelling can help reduce the impact of covid-19. *Journal of Simulation*, 14(2):83–97, 2020.

[4] A. Foster, M. Jankowiak, E. Bingham, P. Horsfall, Y. W. Teh, T. Rainforth, and N. Goodman. Variational Bayesian Optimal Experimental Design. In *Advances in Neural Information Processing Systems*, 2019.

[5] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, pages 1–11, 2021.

[6] S. Kleinegesse and M. U. Gutmann. Bayesian Experimental Design for Implicit Models by Mutual Information Neural Estimation. In *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2020.

[7] S. Kleinegesse, C. Drovandi, and M. U. Gutmann. Sequential Bayesian experimental design for implicit models via mutual information. *Bayesian Analysis*, 2020.

[8] M. D. Lee, S. Zhang, M. Munro, and M. Steyvers. Psychological models of human and optimal performance in bandit problems. *Cog. Sys. Research*, 12(2), 2011.

[9] J. I. Myung and M. A. Pitt. Optimal experimental design for model discrimination. *Psych. Review*, 116(3), 2009.

[10] L. Ouyang, M. H. Tessler, D. Ly, and N. D. Goodman. webppl-oed: A practical optimal experiment design system. In *Proceedings of the annual meeting of the cognitive science society*, 2018.

[11] A. Overstall and J. McGree. Bayesian design of experiments for intractable likelihood models using coupled auxiliary models and multivariate emulation. *Bayesian Anal.*, 15(1), 2020.

[12] B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker. On variational bounds of mutual information. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5171–5180. PMLR, 09–15 Jun 2019.

[13] H. Robbins. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.*, 58(5), 09 1952.

[14] J. Runge, S. Bathiany, E. Bollt, G. Camps-Valls, D. Coumou, E. Deyle, C. Glymour, M. Kretschmer, M. D. Mahecha, J. Muñoz-Marí, et al. Inferring causation from time series in earth system sciences. *Nature communications*, 10(1):1–13, 2019.

[15] D. Russo, B. Van Roy, A. Kazerouni, I. Osband, and Z. Wen. A tutorial on thompson sampling. *arXiv preprint arXiv:1707.02038*, 2017.

[16] E. Schulz, N. T. Franklin, and S. J. Gershman. Finding structure in multi-armed bandits. *Cognitive psychology*, 119:101261, 2020.

[17] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas. Taking the Human Out of the Loop: A Review of Bayesian Optimization. In *Proceedings of the IEEE*, 2015.

[18] M. Steyvers, M. D. Lee, and E.-J. Wagenmakers. A Bayesian analysis of human decision-making on bandit problems. *Journal of Mathematical Psychology*, 53(3), 2009.

[19] R. S. Sutton and A. G. Barto. *Reinforcement learning: an introduction*. Adaptive computation and machine learning series. The MIT Press, 2nd edition, 2018.

[20] W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[21] S. Zhang and M. D. Lee. Optimal experimental design for a class of bandit problems. *Journal of Mathematical Psychology*, 54(6), 2010.

# 6 Appendix

## 6.1 Computational models

### 6.1.1 Win-Stay Lose-Thompson-Sample (WSLTS)

WSLTS performs Thompson Sampling using the reshaped posterior, excluding the previously selected arm. That is, WSLTS follows standard Thompson Sampling for Bernoulli bandits, as described in [15], with two important differences: First, as opposed to sampling from the posterior over all arms, the sampled reward probability of the previously selected arm is set to zero. This is to ensure that WSLTS follows the core semantics of WSLS as a strict generalization that allows for more sophisticated exploration/exploitation mechanisms. Second, we include a temperature parameter $\lambda$ that is used to reshape the posterior over reward probabilities, to widen or sharpen the posterior, conceptually following Chapelle and Li [1]. That is, we draw rewards for the $k$-th arm from Beta$(\alpha_k^\lambda, \beta_k^\lambda)$. We treat the model parameter $\lambda$ as a random variable with a log-normal$(0, 1)$ prior distribution. The WSLTS model has three parameters: $\gamma_w$, which controls the probability of staying after winning, $\gamma_l$, which controls the probability of shifting after losing, and $\lambda$, for posterior reshaping.

### 6.1.2 Autoregressive $\varepsilon$-Greedy (AEG)

The chance of selecting the previously selected arm is controlled by the $\varphi$ parameter. That is, the probability of selecting the previous arm is given by $\varphi + \frac{1-\varphi}{m}$, where on exploration choices, $m$ is the number of bandit arms, and on greedy choices, $m$ is the number of arms that have the maximal expected reward probability, thereby breaking possible ties. $\varphi$ can thus be thought of as a "stickiness" parameter [16] that favors the previous arm on both greedy as well as exploration choices. The AEG model thus has two parameters, $\varepsilon$ and $\varphi$.

### 6.1.3 Generalized Latent State (GLS)

Lee et al. [8] proposed a latent state model for bandit tasks, which includes a latent *explore/exploit* state $z_i$ for each trial $i$. This latent state determines which arm is chosen to resolve the explore-exploit dilemma whenever one arm has more observed wins but also more losses than the other arm(s). Their model proposes that the latent state changes stochastically on a trial-by-trial basis, where transitions between latent states are modeled using a Bernoulli$(0.5)$ distribution.

The authors demonstrate that this model captures people's behavior in an empirical evaluation, but they also show that most participants could be modeled by a simplified account, in which there is only one switch point from exploration to exploitation, without the possibility of transitioning back to exploration [for details, see 8].

The originally proposed latent state model by [8] only dealt with 2-armed bandits. Here we describe the generalization used for k-armed bandits. [2] We distinguish between the following situations the agent can be in [8].

**Same** If two or more arms have the maximum number of wins and the minimum number of losses, choose one of these at random.

**Better-worse** If only one arm has the maximum number of wins and the minimum number of losses, choose this arm.

**Explore-exploit** If neither *same* nor *better-worse* applies, the agent faces an exploration-exploitation dilemma, which is resolved based on the latent state of the agent as follows:

- **Exploit state** If the latent state is *exploit*, then, if there is at least one arm with more wins than all other arms, choose the arm with the maximum number of wins that has the minimum number of losses out the set of arms that have the maximum number of wins (or choose uniformly at random from the set of equivalent options).
- **Explore state** If the latent state is *explore*, then, if there is at least one arm with fewer losses than all other arms, choose the arm with the maximum number of wins out of the set of arms with the minimum number of failures (or choose uniformly at random from the set of equivalent options).

[2] We thank Patrick Laverty for contributing towards this.

We propose a simple extension of this model, which we call the generalized latent state (GLS) model and that includes the original latent state model and the kind of switch-point model as special cases, but also accommodates more flexible latent state transitions. The GLS models the probability of being in a latent exploit (as opposed to explore) state as being dependent on whether the last latent state was explore or exploit, and on whether a reward was observed in the previous trial or not.

The generalization to dependencies in the latent transitions allows the model to account for several further psychologically plausible mechanisms that may be at play. First, as suggested by the switch-point model, people may have a "stickiness" (or "anti-stickiness") in their latent state, such that the probability of being in an exploit-state at trial $t$ depends on the previous latent state. Note that the stickiness in the latent state may be different for explore as opposed to exploit states, and we therefore include these two transition probabilities as two separate model parameters. Second, switches between the latent explore/exploit state may not be symmetric in regard to the previously observed reward. That is, people may, e.g., be more willing to switch from exploration to exploitation when observing a win than when observing a loss, as would be suggested by the WSLS heuristic. The transition probabilities when observing a win and when observing a loss are thus also treated as two separate model parameters.

In total, the GLS has an accuracy of execution parameter $\gamma_{\text{GLS}}$ (conceptually identical to the one in the latent state model of Lee et al. [8]) which accounts for random behavior, as well as four parameters controlling the latent state transitions described above. The initial latent state is sampled from a Bernoulli$(0.5)$ distribution for simplicity. Note that our proposed GLS model can recover the latent state model by [8] by setting all latent transition probabilities to $0.5$, rendering the latent state independent of the preceding state and reward. Similarly, we could treat the latent exploitation state as absorbing, such that once a transition to the exploit state has happened, the agent never goes back to exploring.

### 6.2 MINEBED method

In their MINEBED method, [6] proposed to maximize a cheaper, tractable lower bound on the MI, as opposed to spending resources on estimating the MI to a high accuracy. This lower bound is parametrized by a neural network $T_{\boldsymbol{\psi}}(\boldsymbol{v}, \mathbf{y})$, where $\boldsymbol{\psi}$ are the neural network parameters.[3] The MINEBED method uses the NWJ lower bound [see e.g. 12], which is now a function of $\mathbf{d}$ and $\boldsymbol{\psi}$, given by

$$U(\mathbf{d}; \boldsymbol{\psi}) = \mathbb{E}_{p(\mathbf{y}|\boldsymbol{v},\mathbf{d})p(\boldsymbol{v})}\left[T_{\boldsymbol{\psi}}(\boldsymbol{v}, \mathbf{y})\right] - e^{-1}\mathbb{E}_{p(\mathbf{y}|\mathbf{d})p(\boldsymbol{v})}\left[e^{T_{\boldsymbol{\psi}}(\boldsymbol{v},\mathbf{y})}\right] \tag{2}$$

The expectations in Equation 2 are approximated via sample averages.

The lower bound shown in Equation 2 is a function of the experimental designs $\mathbf{d}$ and the neural network parameters $\boldsymbol{\psi}$. For a fixed design, we can tighten the lower bound, i.e. let it approach the true MI value, by optimizing the neural network with respect to $\boldsymbol{\psi}$ by means of stochastic gradient-ascent, using Equation 2 as an objective function. The gradients of $U(\mathbf{d}; \boldsymbol{\psi})$ with respect to $\boldsymbol{\psi}$ can be easily obtained via automatic differentiation available in common machine-learning libraries. This allows us to obtain an estimate of the mutual information at a fixed design, i.e.

$$U(\mathbf{d}) = \max_{\boldsymbol{\psi}} U(\mathbf{d}; \boldsymbol{\psi}) \tag{3}$$

In order to then optimize the MI with respect to the designs $\mathbf{d}$ we can use any gradient-free optimization technique. In our case, we use Bayesian optimization (BO) [17]. As a probabilistic surrogate model we use Gaussian Processes (GPs) and as an acquisition function we use Expected Improvement (EI). We present a short summary of the gradient-free MINEBED method in Algorithm 1. Note that a variant of the MINEBED method deals with gradient-based design optimization [see 6]. However, this is not applicable in our experiments with bandit tasks, where gradients with respect to the discrete choices and rewards $\mathbf{y}$ are undefined.

**Posterior Estimation** By looking at Equation 2, we can see that the NWJ lower bound is tight when $T_{\boldsymbol{\psi}^*}(\boldsymbol{v}, \mathbf{y}) = 1 + \log p(\boldsymbol{v}|\mathbf{y}, \mathbf{d}^*)/p(\boldsymbol{v})$. By rearranging this, we can thus use our trained neural network $T_{\boldsymbol{\psi}^*}(\boldsymbol{v}, \mathbf{y})$ to compute a (normalized) estimate of the posterior distribution,

$$p(\boldsymbol{v}|\mathbf{y}, \mathbf{d}^*) = p(\boldsymbol{v})e^{T_{\boldsymbol{\psi}^*}(\boldsymbol{v},\mathbf{y})-1}. \tag{4}$$

[3] This neural network takes as input $\boldsymbol{v}$ and $\mathbf{y}$, which are concatenated, and returns a scalar value.

**Algorithm 1** Gradient-Free MINEBED

**Input:** Implicit simulator model $\mathbf{y} \sim p(\mathbf{y}|\boldsymbol{v}, \mathbf{d})$, prior distribution $p(\boldsymbol{v})$, neural network architecture for $T_{\boldsymbol{\psi}}(\boldsymbol{v}, \mathbf{y})$

**Output:** Optimal design $\mathbf{d}^*$, trained neural network $T_{\boldsymbol{\psi}^*}(\boldsymbol{v}, \mathbf{y})$ at $\mathbf{d}^*$

Randomly initialize the experimental designs $\mathbf{d}_n \leftarrow \mathbf{d}_0$
Initialize the Gaussian Process for BO
**while** $U(\mathbf{d}_n)$ not converged **do**
    Sample from the prior: $\boldsymbol{v}^{(i)} \sim p(\boldsymbol{v})$ for $i = 1, \ldots, N$
    Sample from the simulator: $\mathbf{y}^{(i)} \sim p(\mathbf{y}|\boldsymbol{v}^{(i)}, \mathbf{d})$ for $i = 1, \ldots, N$
    Randomly initialize the neural network parameters $\boldsymbol{\psi}_n \leftarrow \boldsymbol{\psi}_0$
    **while** $U(\mathbf{d}_n; \boldsymbol{\psi}_n)$ with fixed $\mathbf{d}_n$ not converged **do**
        Compute a sample average of the lower bound in Equation 2
        Estimate gradients of sample average with respect to $\boldsymbol{\psi}_n$
        Update $\boldsymbol{\psi}_n$ using any gradient-based optimizer
    Use $\mathbf{d}_n$ and $U(\mathbf{d}_n)$ to update the Gaussian Process
    Use BO to find out at which $\mathbf{d}_{n+1}$ to evaluate next

### 6.3 Experimental Details

We here provide details about our experiments.

**Priors** We use a uniform categorical prior over the model indicator $m$, i.e. $p(m) = \mathcal{U}(\{1, 2, 3\})$. We generally use uninformative priors $\mathcal{U}(0, 1)$ for all model parameters, except for the temperature parameter of the WSLTS model that has a LogNorm$(0, 1)$ prior, as it acts as an exponent in reshaping the posterior. We generate $50{,}000$ samples from the prior and then simulate corresponding synthetic data $\mathbf{y}|\boldsymbol{\theta}, \mathbf{d}$ at every design $\mathbf{d}$.

**Sub-networks** For all of our experiments, we use sub-networks $\mathbf{S}_{\boldsymbol{\psi}}(\boldsymbol{v}, \mathbf{y})$ that consist of two hidden layers with 64 and 32 hidden units, respectively, and ReLU activation functions. The number of sufficient statistics we wish to learn for each block of behavioral data is given by the number of dimensions in the output layer of the sub-networks. These are $6$, $8$, $6$ and $8$ units for the MD, PE WSLTS, PE AEG and PE GLS experiments, respectively. The flexibility of a sub-network is naturally increased when increasing the number of desired summary statistics, but the cost increases accordingly. When the number of summary statistics is too low, the summary statistics we learn may not be sufficient. We have found the above number of summary statistics to be effective middle-grounds.

**Main Network** The main network $T_{\boldsymbol{\psi}}(\boldsymbol{v}, \mathbf{y})$ consists of the concatenated outputs of the sub-networks for each block of behavioral data and the variable of interest. This is then followed by two fully-connected layers with ReLU activation functions. For the MD experiment we use $32$ hidden units for the two hidden layers, while we use $64$ and $32$ hidden units for the PE experiments. See Figure 1 for a visualization of this bespoke neural network architecture.

**Training** We use the Adam optimizer to maximize the lower bound shown in Equation 2, with a learning rate of $10^{-3}$ and a weight decay of $10^{-3}$ (except for the PE WSLTS experiments where we use a weight decay of $10^{-4}$). We additionally use a plateau learning rate scheduler with a decay factor of $0.5$ and a patience of $25$ epochs. We train the neural network for $200$, $400$, $300$ and $300$ epochs for the MD, PE WSLTS, PE AEG and PE GLS experiments, respectively. At every design we simulate $50{,}000$ samples from the data-generating distribution (one for every prior sample) and randomly hold out $10{,}000$ of those as a validation set, which are then used to compute an estimate of the mutual information via Equation 2. During the BO procedure we select an experimental budget of $400$ $U(\mathbf{d})$ evaluations ($80$ of which were initial evaluations), which is more than double needed to converge.

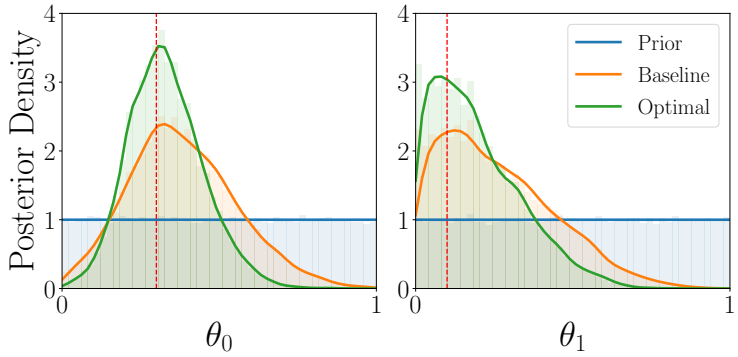


Figure 4: Marginal posterior distributions of the two AEG model parameters for optimal (green) and baseline (orange) designs, averaged over 1,000 observations.

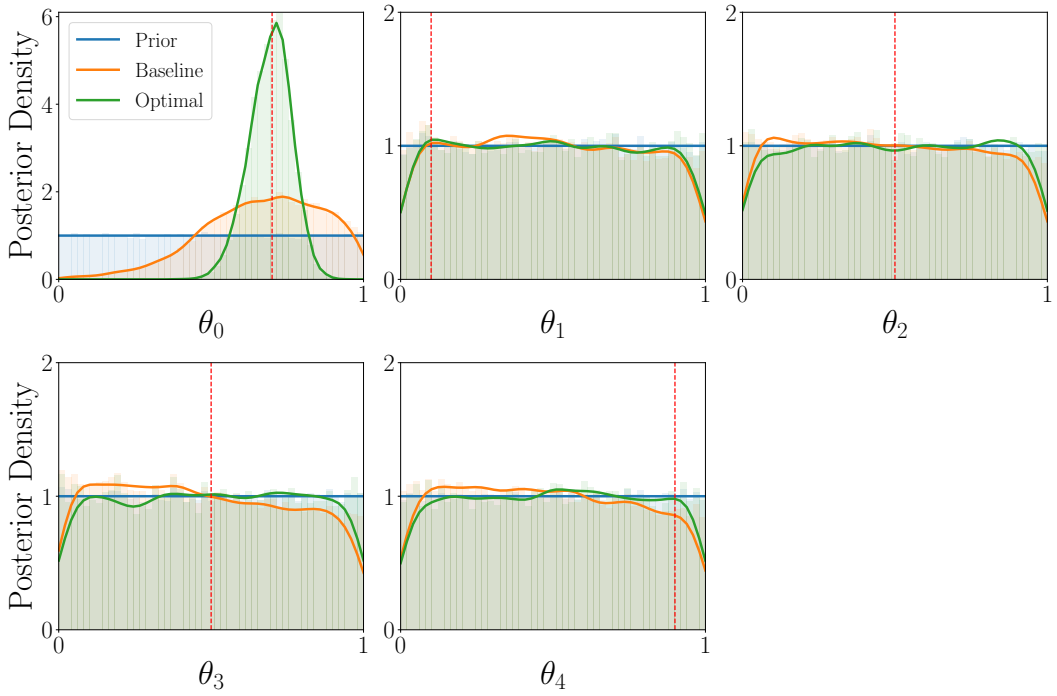


Figure 5: Marginal posterior distributions of the five GLS model parameters for optimal (green) and baseline (orange) designs, averaged over 1,000 observations. Note that the downwards behavior at the boundaries is due to taking a Gaussian kernel density estimation of a nearly-uniform distribution.

### 6.4 Additional Results

We here present additional parameter estimation results for the AEG in Figure 4 and for the GLS model in Figure 5.

Similarly to the parameter estimation task of the WSLTS model (see Figure 3), our optimal designs yield data that are more informative than baseline designs from literature, which can be seen from sharper posterior distributions that are centered around the ground-truths (shown in red-dotted lines). We note that the transition parameters of the GLS model are not well-estimated for any of the designs due to the uninformative priors and small number of trials in the behavioral blocks.