Interventions, Where and How? Bayesian Active Causal Discovery at Scale

Panagiotis Tigas **OATML, University of Oxford, UK*

Yashas Annadani **Max Planck Institute for Intelligent Systems and KTH Stockholm*

Andrew Jesson *OATML, University of Oxford, UK*

Bernhard Schölkopf *Max Planck Institute for Intelligent Systems*

Yarin Gal *OATML, University of Oxford, UK*

Stefan Bauer *KTH Stockholm and CIFAR Azrieli Global Scholar*

Abstract

Causal discovery from observational and interventional data is challenging due to limited data and non-identifiability: factors that introduce uncertainty in estimating the underlying structural causal model (SCM). Selecting experiments (interventions) based on the uncertainty arising from both the factors can expedite the identification of the SCM. Existing methods in experimental design for causal discovery from limited data either rely on linear assumptions for the SCM or select only the intervention target. This work incorporates recent advances in Bayesian causal discovery into the Bayesian optimal experimental design framework, allowing for active causal discovery of large, nonlinear SCMs while selecting both the interventional target and the value. We demonstrate the performance of the proposed method on synthetic graphs for nonlinear SCMs as well as on the *in-silico* single-cell gene regulatory network dataset, DREAM.

1. Introduction

What is the structure of the protein-signaling network derived from a single cell? How do different habits influence the presence of disease? Such questions refer to causal effects in complex systems governed by nonlinear, noisy processes. On most occasions, passive observation of such systems is insufficient to uncover the real cause-effect relationship and costly experimentation is required to disambiguate between competing hypotheses. As such, the design of experiments is of significant interest; an efficient experimentation protocol helps reduce the costs involved in experimentation while aiding the process of producing knowledge through the scientific method.

In the language of causality (Pearl, 2009), the causal relationships are represented qualitatively by a directed acyclic graph (DAG), where the nodes correspond to different variables of the system of study and the edges represent the flow of information between the variables. The abstraction of DAGs allows us to represent the space of possible explanations (hypotheses) for the observations at hand. Representing such hypotheses as Bayesian probabilities (beliefs) allows us to formalize the problem of the scientific method as one

∗. Equal contribution

of Bayesian inference, where the goal is to estimate the posterior distribution $p(\text{DAGs} \mid \text{Observations})$. A posterior distribution over the DAGs allows us to employ information-theoretic acquisition functions that guide experimentation towards the most informative variables for disambiguating between competing hypotheses. Such design procedures belong to the field of *Bayesian Optimal Experimental Design* (Lindley, 1956) for *Causal Discovery* (BOECD) (Tong and Koller, 2001; Murphy, 2001).

An intervention in a causal model refers to the variable (or target) we manipulate and the value (or strength) at which we set the variable. Hence, the design space in the case of learning causal models is the set of all subsets of the intervention targets and the possibly countably infinite set of intervention values of the chosen targets. The intervention value encapsulates important semantics in many causal inference applications. For instance, in medical applications, an intervention can correspond to the administration of different drugs and the intervention value takes the form of a dosage level for each drug. Even though the appropriate choice of this value is crucial for identifying the underlying causal model, existing work on active causal discovery focuses exclusively on selecting the intervention target (Agrawal et al., 2019; Cho et al., 2016). There, the intervention value is generally some arbitrary fixed value (like 0) which is suboptimal (see Fig. 1a). Hence, a holistic treatment of selecting the intervention value and the target in the general case of nonlinear causal models has been missing. We present a simple yet effective causal Bayesian experimental design method (`CBED` - pronounced "seabed") to acquire optimal intervention targets and values by performing Bayesian optimization. Additionally, we extend `CBED` to the batch setting and propose two different batching strategies for tractable, Bayes optimal acquisition of both intervention targets and values. The first strategy `Greedy-CBED` builds up the intervention set greedily. A greedy heuristic is still near-optimal due to submodularity properties of mutual information (Krause and Guestrin, 2012; Agrawal et al., 2019; Kirsch et al., 2019). The second strategy `Soft-CBED` constructs a set of interventions by stochastic sampling from a finite set of candidates, thereby significantly increasing computational efficiency while recovering the original SCM as fast as the greedy strategy. This strategy is well suited for resource-constrained settings.

Assumptions. Throughout this work, we make the following standard assumptions for causal discovery (Peters et al., 2017): **(Causal Sufficiency)** There are no hidden confounders, and all the random variables of interest are observable. **(Finite Samples)** There is a finite number of observational/ interventional samples available. **(Nonlinear SCM with Additive Noise)** The structural causal model has nonlinear conditional expectations with additive Gaussian noise. **(Single Target)** Each intervention is atomic and applied to a single target of the SCM. Additionally, we assume that interventions are planned and executed in batches of size $\mathcal{B}$, with a fixed budget of total interventions given by `Number of Batches` $\times \mathcal{B}$. We also assume that there is no model misspecification with respect to the class of conditional expectations and noise distributions. Finally, we are interested in recovering the full graph $\mathbf{G}$ with a small number of batches.

2. Method

Let $\mathbf{V} = \{1, \dots, d\}$ be the vertex set of any DAG $\mathbf{g} = (\mathbf{V}, E)$ and $\mathbf{X_V} = \{\mathrm{X}_1, \dots, \mathrm{X}_d\} \subseteq \mathcal{X}$ be the random variables of interest indexed by $\mathbf{V}$. Let $\boldsymbol{\theta}$ be the parameters of mechanisms

of SCM. We have an initial observational dataset $\mathcal{D} = \{\mathbf{x_V}^{(i)}\}_{i=1}^{n}$ comprised of instances $\mathbf{x_V} \sim P(\mathrm{X}_1 = \mathrm{x}_1, \ldots, \mathrm{X}_d = \mathrm{x}_d) = p(\mathrm{x}_1, \ldots, \mathrm{x}_d)$. The true SCM $\tilde{\phi} = (\tilde{\mathbf{g}}, \tilde{\boldsymbol{\theta}})$ over random variables $\mathbf{X_V}$ is a matter of fact, but our belief in $\tilde{\phi}$ is uncertain for many reasons. Primarily, it is only possible to learn the DAG $\tilde{\mathbf{g}}$ up to a Markov equivalence class (MEC) from observational data $\mathcal{D}$. Uncertainty also arises from $\mathcal{D}$ from being a finite sample, which we model by introducing the the random variable $\boldsymbol{\Phi}$. Let $\boldsymbol{\phi} \sim p(\boldsymbol{\phi}|\mathcal{D}) \propto p(\mathcal{D} \mid \boldsymbol{\phi})p(\boldsymbol{\phi})$ be an instance of the random variable $\boldsymbol{\Phi}$ that is sampled from our posterior over SCMs after observing the dataset $\mathcal{D}$.

We would like to design an experiment to identify an intervention $\xi \coloneqq \{(j, v)\} \coloneqq \mathrm{do}(\mathrm{X}_j = v)$ that maximizes the information gain about $\boldsymbol{\Phi}$ after observing the outcome of the intervention $\mathbf{y} \sim P(\mathrm{X}_1 = \mathrm{x}_1, \ldots, \mathrm{X}_d = \mathrm{x}_d \mid \mathrm{do}(\mathrm{X}_j = v) = p(\mathbf{y} \mid \xi)$. Here, $\mathbf{y}$ is an instance of the random variable $\mathbf{Y} \subseteq \mathcal{X}$ distributed according to the distribution specified by the mutilated true graph $\tilde{\mathbf{g}}'$ under intervention. Looking at one intervention at a time, one can formalize BOECD as gain in information about $\boldsymbol{\Phi}$ after observing the outcome of an experiment $\mathbf{y}$. The experiment $\xi \coloneqq \{(j, v)\}$ that maximizes the information gain is the experiment that maximizes the mutual information between $\boldsymbol{\Phi}$ and $\mathbf{Y}$: $\{(j^*, v^*)\} = \arg\max_{j,v} \{\mathrm{I}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j, v)\}, \mathcal{D})\}$. Note that the objective considers optimizing over not just the discrete set of intervention targets $j \in \mathbf{V}$, but also over the uncountable set of intervention values $v \subset \mathcal{X}_j$.

2.1 Single Design

To maximize the objective in Equation 2, we need to (1) estimate MI for candidate interventions and (2) maximize the estimated MI by optimizing over the domain of intervention value for every candidate interventional target.

Estimating the MI. As mutual information is intractable, there are various ways to estimate it depending on whether we can sample from the posterior and whether the likelihood can be evaluated (Foster et al., 2020; Poole et al., 2019; Houlsby et al., 2011). Since the models we consider allow both posterior sampling and likelihood evaluation, it suffices to obtain an estimator which requires only likelihood evaluation and Monte Carlo approximations of the expectations. To do so, we derive an estimator similar to Bayesian Active Learning by Disagreement (BALD) (Houlsby et al., 2011), which considers MI as a difference of conditional entropies over the outcomes $\mathbf{Y}$:

$$\begin{aligned}
&\mathrm{I}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j, v)\}, \mathcal{D}) = \mathrm{H}(\mathbf{Y} \mid \{(j, v)\}, \mathcal{D}) - \mathrm{H}(\mathbf{Y} \mid \boldsymbol{\Phi}, \{(j, v)\}, \mathcal{D}) \\
&= - \mathop{\mathbb{E}}_{p(\mathbf{y} \mid \{(j,v)\}, \mathcal{D})} \left[\log \left(\mathop{\mathbb{E}}_{p(\boldsymbol{\phi} \mid \mathcal{D})} \left[p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j, v)\}) \right] \right) \right] + \mathop{\mathbb{E}}_{p(\boldsymbol{\phi} \mid \mathcal{D})} \left[\mathop{\mathbb{E}}_{p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\})} \left[\log \left(p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j, v)\}) \right) \right] \right] \quad (1)
\end{aligned}$$

where $\mathrm{H}(\cdot)$ is the entropy. See Appendix B.1 for the derivation. A Monte Carlo estimator of the above equation can be used as an approximation (Appendix B.2). Equation (1) has an intuitive interpretation. It assigns high mutual information to interventions that the model disagrees the most regarding the outcome. We denote the MI for a single design as $\mathcal{I}(\{(j, v)\}) \coloneqq \mathrm{I}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j, v)\}, \mathcal{D})$.

Selecting the Intervention Value. Maximizing the objective is achieved not only by selecting the intervention target but also by setting the appropriate intervention value. Although optimizing the intervention target is tractable (discrete and finite number of nodes to select from),

selecting the value to intervene is usually intractable since they are continuous. For any given target node j, MI is a nonlinear function over $v \in \mathcal{X}_j$ and hence solving with gradient ascent techniques only yields a local maximum. Given that MI is expensive to evaluate, we treat MI for a given target node j as a black-box function and obtain its maximum using Bayesian Optimization (BO) (Kushner, 1964; Zhilinskas, 1975; Močkus, 1975). BO seeks to find the maximum of this function $max_{v \in \mathcal{X}_j} \mathcal{I}(\{(j, v)\})$ over the entire set $\mathcal{X}_j$ with as few evaluations as possible. See appendix E for details. In our work, we run GP-UCB (Srinivas et al., 2010) independently on every candidate intervention target $j = \{1, \ldots, d\}$ by querying points within a fixed domain $[-k, k] \subset \mathbb{R}$. Note that the domain can be chosen based on the application, for example, if we must constrain dosage levels within a fixed range. Each GP is one-dimensional in our setup; hence a few evaluations of UCB are sufficient to get a good value maxima candidate. Further, GP-UCB for each candidate target is parallelizable, making it efficient. We finally select the design with the highest MI across the candidate intervention targets.

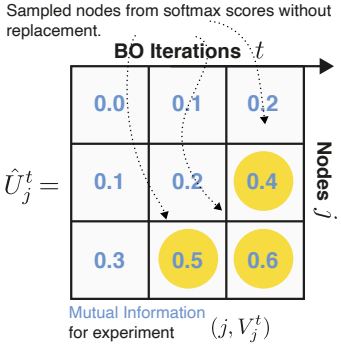


Figure 1: For each t iteration of the BO algorithm and each node j, we get a utility function evaluation $\hat{U}_j^t$ (the utility being the MI in our case). Then we sample without replacement proportionally to the scores to prepare a batch (2.2).

2.2 Batch Design

In many applications, it is desirable to select the most informative *set* of interventions instead of a single intervention at a time. Take, for example, a biologist entering a wet lab with a script of experiments to execute. Batching experiments removes the bottleneck of waiting for an experiment to finish and get analyzed until executing the next one. Given a budget per batch $\mathcal{B}$ which denotes the number of experiments in a batch, the problem of selecting the batch then becomes $\arg\max_{\boldsymbol{\Xi}} \mathrm{I}(\mathbf{Y}; \boldsymbol{\Phi} \mid \boldsymbol{\Xi}, \mathcal{D})$, such that $\texttt{cardinality}(\boldsymbol{\Xi}) = \mathcal{B}$, where $\boldsymbol{\Xi}$ is a set of interventions $\bigcup_{i=1}^{\mathcal{B}} (j_i, v_i)$ and $\mathbf{Y}$ denotes the random variable for the outcomes of the interventions of the batch. We denote the MI for a batch design as $\mathcal{I}(\boldsymbol{\Xi}) \coloneqq \mathrm{I}(\mathbf{Y}; \boldsymbol{\Phi} \mid \boldsymbol{\Xi}, \mathcal{D})$.

Greedy Algorithm. Computing the optimal solution $\mathcal{I}(\boldsymbol{\Xi}^*)$ is computationally infeasible. However, as the conditional mutual information is *submodular* and *non-decreasing* (see Appendix B.4 for proof), we can derive a simple greedy algorithm (Algorithm 1) that can achieve at least a $(1 - 1/e) \approx 0.64$ approximation of the optimal solution (Krause and Guestrin, 2012; Nemhauser et al., 1978). We denote this strategy as `Greedy-CBED`.

Soft Top-K. Although the greedy algorithm is tractable, it requires $O(\mathcal{B}d)$ instances of GP-UCB. Kirsch et al. (2021) show that a soft top-k selection strategy performs similarly to the greedy algorithm, reducing the computation requirements to $O(d)$ runs of GP-UCB. To achieve this, we construct a finite set of candidate intervention target-value pairs by keeping all the T evaluations of GP-UCB for each node $j = \{1, \ldots, d\}$. Therefore, for d nodes, we our candidate set is comprised of $d \times T$ experiments. We score each experiment in this candidate set using the MI estimate. We then sample *without replacement* $\mathcal{B}$ times proportionally to the *softmax* of the MI scores (Algorithm 2). We denote this strategy as `Soft-CBED`.

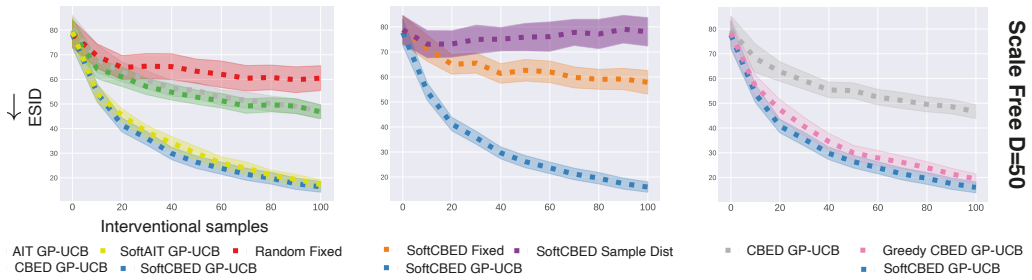


Figure 2: Results on the $\mathbb{E}$**-SID** $\downarrow$ metric (100 seeds, with standard error of the mean shaded) for 50 variables involving nonlinear functional relationships and additive Gaussian noise. (a) We show that `Soft-CBED` with GP-UCB value selection strategy significantly outperforms the baselines. (b) We isolate the effect of the value selection strategy. We show that intervening with a fixed value and sampling from the support of data both perform worse than having an optimizer like GP-UCB. (c) we compare non-batch (`CBED`) vs batch-based acquisition functions (`Greedy-CBED`, `Soft-CBED`). As we can see, `Soft-CBED` performs as well as `Greedy-CBED`. For all experiments, we use the DiBS (Lorch et al., 2021) posterior model.

3. Experiments and Results

Acquisition Functions. **(Random)** Random baseline acquires interventional targets at random. **(AIT / *soft*AIT)** active intervention targeting (AIT) (Scherrer et al., 2021) uses an f-score based acquisition strategy to select the intervention targets. See appendix **??** for more details. Since the original proposed approach does not consider a batch setting, we introduce a variant that augments AIT with the proposed soft batching, as described in section 2.2. **(CBED / GreedyCBED / SoftCBED)** These are the Monte Carlo estimates of MI, as described in section 2. `(CBED)` selects a single intervention (target and value) that maximizes the MI and this intervention is applied for the whole batch. In `Greedy-CBED` , the batch is built up in a greedy fashion selecting the target, value pairs one at a time (Algorithm 1). `Soft-CBED` is sampling (target, value) pairs proportionally to the MI scores to select a batch, as described in section 2.2 and Algorithm 2.

Value Selection Strategies. **(Fixed)** This value selection strategy assumes setting the value of the intervention to a fixed value. In the experiments, we fixed the value to 0. **(Sample-Dist)**This value selection strategy samples from the support of the observational data. **(GP-UCB)** This strategy uses the proposed GP-UCB Bayesian optimization strategy to select the value that maximizes MI.

Results. **(ER and SF task.)** We generate Erdős–Rényi Erdős and Rényi (1959) (ER) and Scale-Free (SF) graphs (Barabási and Albert, 1999) of size 20 and 50. We provide more details about the experiments in appendix D.1. **(Single-Cell Protein-Signalling Network.)** The DREAM family of benchmarks Greenfield et al. (2010) are designed to evaluate causal discovery algorithms of the regulatory networks of a single cell. Refer to appendix D.2 for details and exact settings. For each of the acquisition objectives and datasets, we present the mean and standard error of the expected structural hamming distance $\mathbb{E}$**-SHD**, expected structural interventional distance $\mathbb{E}$**-SID** (Peters and Bühlmann, 2015), area under the receiver operating characteristic curve **AUROC** and area under the precision-recall curve **AUPRC**. In Appendix H we describe the metrics in detail. We evaluate these metrics as

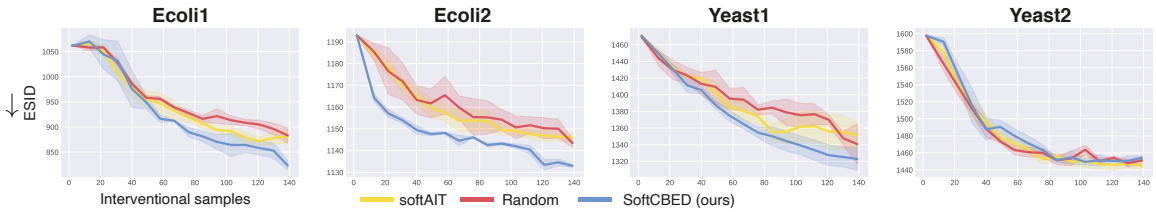


Figure 3: Comparison of acquisition functions on DREAM dataset, for 50 dimensions and batch size 10 on $\mathbb{E}$**-SID**$\downarrow$ metric (6 seeds, with standard error of the mean).

a function of the number of acquired interventional samples (or experiments), which helps quantitatively compare different acquisition strategies. Apart from $\mathbb{E}$**-SID**, we relegate results with other metrics to the appendix J.

On the synthetic graphs (Figure 2), we can see that for SF graphs with $50D$ variables and nonlinear functional relationships, the proposed approach based on soft top-k to select a batch with GP-UCB outperforms all the baselines in terms of the $\mathbb{E}$**-SID** metric. On the other hand, AIT alone does not converge to the ground truth graph fast even after combining with the proposed value acquisition, but when further augmented with the proposed soft strategy, the *soft*AIT recovers the ground truth causal graph upto 4 times faster and performs competitively to `Soft-CBED`. We observe similar performance across other metrics as well, including for ER graphs. Full results are presented in the appendix J.

Next, we examine the importance of having a value selection strategy for active causal discovery. We use the MI estimator in Equation 1; moreover, we test the proposed GP-UCB with two heuristics - the fixed value strategy and sampling values from the support. As we can see in Figure 2(b), selecting the value using GP-UCB clearly benefits the causal discovery process. We expect this finding as the mutual information is not constant with respect to the intervened value. To make this point clear, we demonstrate in the appendix G the influence of the value in a simple two variables graph. In addition, we note that naively sampling from the support of the observed dataset performs worse than fixing the value to 0. We hypothesize that this is due to lower epistemic uncertainty in the high density regions of the support, hinting that these regions might be less informative.

In order to further understand how the soft batch strategy compares with other batch selection strategies, we compare the results of `Soft-CBED` with `Greedy-CBED` and `CBED`. We observe (Figure 2(c)) that `Greedy-CBED` and `Soft-CBED` give very similar results overall. While `Greedy-CBED` is optimal under certain conditions (Kirsch et al., 2019), `Soft-CBED` remains competitive and has the advantage that the batch can be selected in a one-shot manner. This is also evident from the runtime performance of both these batching strategies in Table 5 (appendix I). Both these batch selection strategies perform significantly better than selecting one intervention target/value pair, and executing them $\mathcal{B}$ times (`CBED`).

Finally, on the DREAM task, we see that our method outperforms *soft*AIT and random baselines on the $\mathbb{E}$**-SID** metric (see Figure 3). In these experiments, since the intervention is emulating the gene knockout setting, we only use the fixed value strategy, with a value of 0.0. Although random baseline still remains a competitive choice, in certain settings, `Soft-CBED` objective is significantly better (`Ecoli1`, `Ecoli2` datasets).

References

Raj Agrawal, Chandler Squires, Karren Yang, Karthikeyan Shanmugam, and Caroline Uhler. Abcd-strategy: Budgeted experimental design for targeted causal structure discovery. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3400–3409. PMLR, 2019.

Yashas Annadani, Jonas Rothfuss, Alexandre Lacoste, Nino Scherrer, Anirudh Goyal, Yoshua Bengio, and Stefan Bauer. Variational causal networks: Approximate bayesian inference over causal structures. *arXiv preprint arXiv:2106.07635*, 2021.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

Vladimir Batagelj and Ulrik Brandes. Efficient generation of large random networks. *Physical Review E*, 71(3):036113, 2005.

Philippe Brouillard, Sébastien Lachapelle, Alexandre Lacoste, Simon Lacoste-Julien, and Alexandre Drouin. Differentiable causal discovery from interventional data. *arXiv preprint arXiv:2007.01754*, 2020.

Hyunghoon Cho, Bonnie Berger, and Jian Peng. Reconstructing causal biological networks through active learning. *PloS one*, 11(3):e0150611, 2016.

Chris Cundy, Aditya Grover, and Stefano Ermon. Bcd nets: Scalable variational approaches for bayesian causal discovery. *Advances in Neural Information Processing Systems*, 34, 2021.

P Erdős and A Rényi. On random graphs i. publicationes mathematicae (debrecen). 1959.

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

Nir Friedman, Moises Goldszmidt, and Abraham Wyner. Data analysis with bayesian networks: A bootstrap approach. *arXiv preprint arXiv:1301.6695*, 2013.

Juan L Gamella and Christina Heinze-Deml. Active invariant causal prediction: Experiment selection through stability. *arXiv preprint arXiv:2006.05690*, 2020.

Alex Greenfield, Aviv Madar, Harry Ostrer, and Richard Bonneau. Dream4: Combining genetic and dynamic information to identify biological networks and dynamical models. *PloS one*, 5(10):e13397, 2010.

Yang-Bo He and Zhi Geng. Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research*, 9(Nov):2523–2547, 2008.

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

Nan Rosemary Ke, Olexa Bilaniuk, Anirudh Goyal, Stefan Bauer, Hugo Larochelle, Bernhard Schölkopf, Michael C Mozer, Chris Pal, and Yoshua Bengio. Learning neural causal models from unknown interventions. *arXiv preprint arXiv:1910.01075*, 2019.

Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32:7026–7037, 2019.

Andreas Kirsch, Sebastian Farquhar, and Yarin Gal. A simple baseline for batch active learning with stochastic acquisition functions. *arXiv preprint arXiv:2106.12059*, 2021.

Andreas Krause and Carlos E Guestrin. Near-optimal nonmyopic value of information in graphical models. *arXiv preprint arXiv:1207.1394*, 2012.

Harold J Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. 1964.

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

Lars Lorch, Jonas Rothfuss, Bernhard Schölkopf, and Andreas Krause. Dibs: Differentiable bayesian structure learning. *arXiv preprint arXiv:2105.11839*, 2021.

Jonas Močkus. On bayesian methods for seeking the extremum. In *Optimization techniques IFIP technical conference*, pages 400–404. Springer, 1975.

Kevin P Murphy. Active learning of causal bayes net structure. 2001.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1): 265–294, 1978.

Robert Osazuwa Ness, Karen Sachs, Parag Mallick, and Olga Vitek. A bayesian active learning experimental design for inferring signaling networks. In *International Conference on Research in Computational Molecular Biology*, pages 134–156. Springer, 2017.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Jonas Peters and Peter Bühlmann. Structural intervention distance for evaluating causal graphs. *Neural computation*, 27(3):771–799, 2015.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pages 947–1012, 2016.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

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*, pages 5171–5180. PMLR, 2019.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308 (5721):523–529, 2005.

Thomas Schaffter, Daniel Marbach, and Dario Floreano. Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics*, 27 (16):2263–2270, 2011.

Nino Scherrer, Olexa Bilaniuk, Yashas Annadani, Anirudh Goyal, Patrick Schwab, Bernhard Schölkopf, Michael C Mozer, Yoshua Bengio, Stefan Bauer, and Nan Rosemary Ke. Learning neural causal models with active interventions. *arXiv preprint arXiv:2109.02429*, 2021.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias W Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *ICML*, 2010.

Scott Sussex, Andreas Krause, and Caroline Uhler. Near-optimal multi-perturbation experimental design for causal structure learning. *arXiv preprint arXiv:2105.14024*, 2021.

MTCAJ Thomas and A Thomas Joy. *Elements of information theory*. Wiley-Interscience, 2006.

Simon Tong and Daphne Koller. Active learning for structure in bayesian networks. In *International joint conference on artificial intelligence*, volume 17, pages 863–869. Citeseer, 2001.

Julius von Kügelgen, Paul K Rubenstein, Bernhard Schölkopf, and Adrian Weller. Optimal experimental design via bayesian optimization: active causal structure learning for gaussian process networks. *arXiv preprint arXiv:1910.03962*, 2019.

Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *arXiv preprint arXiv:1803.01422*, 2018.

AG Zhilinskas. Single-step bayesian search method for an extremum of functions of a single variable. *Cybernetics*, 11(1):160–166, 1975.

Appendix A. Algorithms

Algorithm 1: `Greedy-CBED`

Input : $\mathcal{E}$ environment, N initial observational samples, $\mathcal{B}$ batch Size, d number of nodes

▷ Initialize set of experiments $\boldsymbol{\Xi}$ to empty

$\boldsymbol{\Xi} \leftarrow \varnothing$
for $n = 1 \dots \mathcal{B}$ **do**
  for $j = 1 \dots d$ **do**
    ▷ Select optimal intervention value per node j using GP-UCB
    $V_j \leftarrow \arg\max_v \mathcal{I}(\boldsymbol{\Xi} \cup \{(j, v)\})$
    $U_j \leftarrow \mathcal{I}(\boldsymbol{\Xi} \cup \{(j, V_j)\})$
  $j^* \leftarrow \arg\max_j U_j$
  $v^* \leftarrow V_{j^*}$
  $\boldsymbol{\Xi} \leftarrow \boldsymbol{\Xi} \cup \{(j^*, v^*)\}$
return $\boldsymbol{\Xi}$

Algorithm 2: `Soft-CBED`

Input : $\mathcal{E}$ environment, N initial observational samples, $\mathcal{B}$ batch Size, d number of nodes, ζ softmax temperature

for $j = 1 \dots d$ **do**
  ▷ Select candidate intervention values per node j using GP-UCB
  Initialize μ_j^0 and σ_j^0
  for $t = 1 \dots T$ **do**
    $V_j^t \leftarrow \arg\max_v \mu_j^{t-1}(v) + \sqrt{\beta^t}\sigma_j^{t-1}(v)$
    $\hat{U}_j^t \leftarrow \mathcal{I}(\{(j, V_j^t)\})$
    Update the GP to obtain μ_j^t and σ_j^t
$\{(t_i, j_i)\}_{i \in \{1,\dots,\mathcal{B}\}} \leftarrow \mathcal{B}$ samples *without* replacement $\propto \exp(\hat{U}_j^t / \zeta)$
$\boldsymbol{\Xi} \leftarrow \{(j_i, V_{j_i}^{t_i})\}_{i \in \{1,\dots,\mathcal{B}\}}$
return $\boldsymbol{\Xi}$

Appendix B. Theoretical Results

B.1 Deriving the Mutual Information over Outcomes

In the following lemma, we derive the mutual information over outcomes given in (1).

Lemma 1

$$\begin{aligned}
&\mathrm{I}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j, v)\}, \mathcal{D}) \\
&= - \mathop{\mathbb{E}}_{p(\mathbf{y} \mid \{(j,v)\}, \mathcal{D})} \left[\log \left(\mathop{\mathbb{E}}_{p(\boldsymbol{\phi} \mid \{(j,v)\}, \mathcal{D})} \left[p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j, v)\}) \right] \right) \right] + \mathop{\mathbb{E}}_{p(\boldsymbol{\phi} \mid \mathcal{D})} \left[\mathop{\mathbb{E}}_{p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\})} \left[\log \left(p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j, v)\}) \right) \right] \right]
\end{aligned} \tag{2}$$

Proof

$$\mathrm{I}(\mathbf{Y};\mathbf{\Phi} \mid \{(j,v)\},\mathcal{D}) = \mathrm{H}(\mathbf{Y} \mid \{(j,v)\},\mathcal{D}) - \mathrm{H}(\mathbf{Y} \mid \mathbf{\Phi},\{(j,v)\},\mathcal{D}) \tag{3a}$$

$$= \mathrm{H}(\mathbf{Y} \mid \{(j,v)\},\mathcal{D}) - \mathop{\mathbb{E}}_{p(\boldsymbol{\phi}|\mathcal{D})}\left[\mathrm{H}(\mathbf{Y} \mid \boldsymbol{\phi},\{(j,v)\})\right] \tag{3b}$$

$$= -\mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\},\mathcal{D})}\left[\log\left(p(\mathbf{y} \mid \{(j,v)\},\mathcal{D})\right)\right] + \mathop{\mathbb{E}}_{p(\boldsymbol{\phi}|\mathcal{D})}\left[\mathop{\mathbb{E}}_{p(\mathbf{y}|\boldsymbol{\phi},\{(j,v)\})}\left[\log\left(p(\mathbf{y} \mid \boldsymbol{\phi},\{(j,v)\})\right)\right]\right] \tag{3c}$$

$$= -\mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\},\mathcal{D})}\left[\log\left(\int_{\boldsymbol{\phi}} p(\mathbf{y},\boldsymbol{\phi} \mid \{(j,v)\},\mathcal{D})d\boldsymbol{\phi}\right)\right] + \mathop{\mathbb{E}}_{p(\boldsymbol{\phi}|\mathcal{D})}\left[\mathop{\mathbb{E}}_{p(\mathbf{y}|\boldsymbol{\phi},\{(j,v)\})}\left[\log\left(p(\mathbf{y} \mid \boldsymbol{\phi},\{(j,v)\})\right)\right]\right] \tag{3d}$$

$$= -\mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\},\mathcal{D})}\left[\log\left(\int_{\boldsymbol{\phi}} p(\boldsymbol{\phi} \mid \{(j,v)\},\mathcal{D})p(\mathbf{y} \mid \boldsymbol{\phi},\{(j,v)\},\mathcal{D})d\boldsymbol{\phi}\right)\right] + \mathop{\mathbb{E}}_{p(\boldsymbol{\phi},|\mathcal{D})}\left[\mathop{\mathbb{E}}_{p(\mathbf{y}|\boldsymbol{\phi},\{(j,v)\})}\left[\log\left(p(\mathbf{y} \mid \boldsymbol{\phi},\{(j,v)\})\right)\right]\right] \tag{3e}$$

$$= -\mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\},\mathcal{D})}\left[\log\left(\mathop{\mathbb{E}}_{p(\boldsymbol{\phi}|\{(j,v)\},\mathcal{D})}\left[p(\mathbf{y} \mid \boldsymbol{\phi},\{(j,v)\})\right]\right)\right] + \mathop{\mathbb{E}}_{p(\boldsymbol{\phi}|\mathcal{D})}\left[\mathop{\mathbb{E}}_{p(\mathbf{y}|\boldsymbol{\phi},\{(j,v)\})}\left[\log\left(p(\mathbf{y} \mid \boldsymbol{\phi},\{(j,v)\})\right)\right]\right]. \tag{3f}$$

■

B.2 Estimating the Mutual Information over Outcomes

For models that allow for evaluation of the experimental outcome density (likelihood), $p(\mathbf{y} \mid \boldsymbol{\phi},\{(j,v)\})$, we can use the following estimator for $\mathrm{I}(\mathbf{Y};\mathbf{\Phi} \mid \{(j,v)\},\mathcal{D})$:

$$\widehat{\mathrm{I}}(\mathbf{Y};\mathbf{\Phi} \mid \{(j,v)\},\mathcal{D}) = \widehat{\mathrm{H}}(\mathbf{Y} \mid \{(j,v)\},\mathcal{D}) - \widehat{\mathrm{H}}(\mathbf{Y} \mid \mathbf{\Phi},\{(j,v)\},\mathcal{D}) \tag{4}$$

Algorithm 3: Mutual Information Computation

Input : Posterior $q(\boldsymbol{\phi} \mid \mathcal{D}_{\text{👁}} \cup \mathcal{D}_{\text{🔨}})$, Number of posterior samples c, Number of interventional samples m, Intervention $\{(j,v)\}$. We notate as 🔨 the interventional and Y the observational data.

▷Sample from the posterior

$\{\widehat{\boldsymbol{\phi}}_i \sim q(\boldsymbol{\phi} \mid \mathcal{D}_{\text{👁}} \cup \mathcal{D}_{\text{🔨}})\}_{i=1}^{c}$

▷Sample from mutilated SCMs

$\{\widehat{\mathbf{y}}_{i,j,k} \sim p(\mathbf{y} \mid \widehat{\boldsymbol{\phi}}_i,\{(j,v)\})\}_{k=1}^{m}$

return $-\frac{1}{c\times m}\sum_{i=1}^{c}\sum_{k=1}^{m}\log\left(\frac{1}{c}\sum_{l=1}^{c}p(\widehat{\mathbf{y}}_{i,k} \mid \widehat{\boldsymbol{\phi}}_l,\{(j,v)\})\right)$
$+\frac{1}{c\times m}\sum_{i=1}^{c}\sum_{k=1}^{m}\log\left(p(\widehat{\mathbf{y}}_{i,k} \mid \widehat{\boldsymbol{\phi}}_i,\{(j,v)\})\right)$

Definition 2 *The Monte Carlo estimator,* $\widehat{\mathrm{H}}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D})$*, of the marginal entropy of the experimental outcomes,* $\mathrm{H}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D})$*, is given by:*

$$-\frac{1}{c_o \times m} \sum_{i=1}^{c_o} \sum_{k=1}^{m} \log \left(\frac{1}{c_{in}} \sum_{l=1}^{c_{in}} p(\widehat{\mathbf{y}}_{i,k} \mid \widehat{\boldsymbol{\phi}}_l, \{(j,v)\}) \right), \tag{5}$$

where $\widehat{\mathbf{y}}_{i,k} \sim p(\mathbf{y} \mid \widehat{\boldsymbol{\phi}}_l, \{(j,v)\})$ *is one of* m *samples from the density parameterised by the* i*th of* c_o *SCMs* $\widehat{\boldsymbol{\phi}}_i \sim p(\boldsymbol{\phi} \mid \mathcal{D})$ *augmented by intervention* $\{(j,v)\}$*. The likelihood of the sample* $\widehat{\mathbf{y}}_{i,k}$ *is then evaluated under the parameterisation of the* l*th of* c_{in} *additional SCMs* $\widehat{\boldsymbol{\phi}}_l \sim p(\boldsymbol{\phi} \mid \mathcal{D})$ *augmented by intervention* $\{(j,v)\}$*.*

$\widehat{\mathrm{H}}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D})$ is a consistent but biased estimator of $\mathrm{H}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D})$ due to the expectation inside of the nonlinear log function. Alternatively, we can look at the following lower bound on $\mathrm{H}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D})$:

$$\begin{aligned}
\mathrm{H}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D}) &= - \mathop{\mathbb{E}}_{p(\mathbf{y} \mid \{(j,v)\}, \mathcal{D})} \left[\log \left(\mathop{\mathbb{E}}_{p(\boldsymbol{\phi} \mid \mathcal{D})} \left[p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}) \right] \right) \right], \\
&\leq - \mathop{\mathbb{E}}_{p(\mathbf{y} \mid \{(j,v)\}, \mathcal{D})} \left[\mathop{\mathbb{E}}_{p(\boldsymbol{\phi} \mid \mathcal{D})} \left[\log \left(p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}) \right) \right] \right],
\end{aligned}$$

by Jensen's inequality. We can then define an unbiased estimator of this lower bound.

Definition 3 *The unbiased Monte Carlo estimator,* $\widehat{\mathrm{H}}^*(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D})$*, of the lower bound on the marginal entropy of the experimental outcomes,* $\mathbb{E}_{p(\mathbf{y} \mid \{(j,v)\}, \mathcal{D})} \left[\mathbb{E}_{p(\boldsymbol{\phi}, \mid \mathcal{D})} \left[\log \left(p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}) \right) \right] \right]$*, is given by:*

$$-\frac{1}{c_o \times c_{in} \times m} \sum_{i=1}^{c_o} \sum_{k=1}^{m} \sum_{l=1}^{c_{in}} \log \Big(p(\widehat{\mathbf{y}}_{i,k} \mid \widehat{\boldsymbol{\phi}}_l, \{(j,v)\}) \Big), \tag{6}$$

Finally, we define our estimator for $\mathrm{H}(\mathbf{Y} \mid \boldsymbol{\Phi}, \{(j,v)\}, \mathcal{D})$.

Definition 4 *The Monte Carlo estimator,* $\widehat{\mathrm{H}}(\mathbf{Y} \mid \boldsymbol{\Phi}, \{(j,v)\}, \mathcal{D})$*, of the entropy of the experimental outcomes conditioned on* $\boldsymbol{\Phi}$*,* $\mathrm{H}(\mathbf{Y} \mid \boldsymbol{\Phi}, \{(j,v)\}, \mathcal{D})$*, is given by:*

$$-\frac{1}{c_o \times m} \sum_{i=1}^{c_o} \sum_{k=1}^{m} \log \Big(p(\widehat{\mathbf{y}}_{i,k} \mid \widehat{\boldsymbol{\phi}}_i, \{(j,v)\}) \Big), \tag{7}$$

where $\widehat{\mathbf{y}}_{i,k} \sim p(\mathbf{y} \mid \widehat{\boldsymbol{\phi}}_i, \{(j,v)\})$ *is one of* m *samples from the density parameterised by the* i*th of* c_o *graphs* $\widehat{\boldsymbol{\phi}}_i \sim p(\boldsymbol{\phi} \mid \mathcal{D})$ *augmented by intervention* $\{(j,v)\}$*.*

B.3 Monte Carlo Estimator of the Batch Mutual Information

While Equation 1 pertains to MI for a single design, we present here the MI estimator for the batch design.

$$
\begin{aligned}
\mathrm{I}(\mathbf{Y}; \boldsymbol{\Phi} \mid \boldsymbol{\Xi}, \mathcal{D}) &= \sum_{\{(j,v)\} \in \boldsymbol{\Xi}} \mathrm{I}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) \\
&= \sum_{\{(j,v)\} \in \boldsymbol{\Xi}} \mathrm{H}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D}) - \mathrm{H}(\mathbf{Y} \mid \boldsymbol{\Phi}, \{(j,v)\}, \mathcal{D}) \\
&= - \sum_{\{(j,v)\} \in \boldsymbol{\Xi}} \mathop{\mathbb{E}}_{p(\mathbf{y} \mid \{(j,v)\}, \mathcal{D})} \left[\log \left(\mathop{\mathbb{E}}_{p(\boldsymbol{\phi} \mid \mathcal{D})} \left[p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}) \right] \right) \right] + \mathop{\mathbb{E}}_{p(\boldsymbol{\phi} \mid \mathcal{D})} \left[\mathop{\mathbb{E}}_{p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\})} \left[\log \left(p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}) \right) \right] \right]
\end{aligned}
\tag{8}
$$

B.4 Mutual Information Submodularity and Monotonicity Proofs

Theorem 5 $\mathbf{I}(Y; \omega \mid X)$ *is submodular.*

Proof The proof follows the structure of (Kirsch et al., 2019, Appendix A).

$$
\begin{aligned}
&\mathbf{I}(Y \cup \{y_1\}; \omega \mid X \cup \{x_1\}) + \mathbf{I}(Y \cup \{y_2\}; \omega \mid X \cup \{x_2\}) \geq \\
&\mathbf{I}(Y \cup \{y_1, y_2\}; \omega \mid X \cup \{x_1, x_2\}) + \mathbf{I}(Y; \omega \mid X) \\
&\text{(conditioning on RVs that are independent of the non-conditioning RVs)} \Leftrightarrow \\
&\mathbf{I}(Y \cup \{y_1\}; \omega \mid X \cup \{x_1, x_2\}) + \mathbf{I}(Y \cup \{y_2\}; \omega \mid X \cup \{x_1, x_2\}) \geq \\
&\mathbf{I}(Y \cup \{y_1, y_2\}; \omega \mid X \cup \{x_1, x_2\}) + \mathbf{I}(Y; \omega \mid X \cup \{x_1, x_2\}) \\
&\text{(substituting X } \cup \{x_1, x_2\} \text{ with } X^+) \Leftrightarrow \\
&\mathbf{I}(Y \cup \{y_1\}; \omega \mid X^+) + \mathbf{I}(Y \cup \{y_2\}; \omega \mid X^+) \geq \\
&\mathbf{I}(Y \cup \{y_1, y_2\}; \omega \mid X^+) + \mathbf{I}(Y; \omega \mid X^+) \\
&\text{(subtract } 2 * \mathbf{I}(Y; \omega \mid X^+) \text{ from both sides} \\
&\text{and use the identity } \mathbf{I}(A, B; C) - \mathbf{I}(B; C) = \mathbf{I}(A; C \mid B) \text{)} \Leftrightarrow \\
&\mathbf{I}(y_1; \omega \mid Y, X^+) + \mathbf{I}(y_2; \omega \mid Y, X^+) \geq \mathbf{I}(y_1, y_2; \omega \mid Y, X^+) \\
&\Leftrightarrow \\
&\mathbf{I}(y_1; \omega \mid Y, X^+) + \mathbf{I}(y_2; \omega \mid Y, X^+) = \\
&\underbrace{(h(y_1 \mid Y, X^+) + h(y_2 \mid Y, X^+))}_{\geq h(y_1, y_2 \mid Y, X^+) \text{ (Thomas and Joy, 2006, p.253)}} \quad \underbrace{- (h(y_1 \mid Y, X^+, \omega) + h(y_2 \mid Y, X^+, \omega))}_{= h(y_1, y_2 \mid \omega, Y, X^+) \text{ (because } y_1 \perp\!\!\!\perp y_2 \mid \omega)} \geq \\
&h(y_1, y_2 \mid Y, X^+) - h(y_1, y_2 \mid \omega, Y, X^+) = \mathbf{I}(y_1, y_2; \omega \mid Y, X^+)
\end{aligned}
$$

■

Theorem 6 $\mathbf{I}(Y; \omega \mid X)$ *is non-decreasing.*

Proof

$$\mathbf{I}(Y \cup \{y\}; \omega \mid X \cup \{x\}) - \mathbf{I}(Y; \omega \mid X) =$$
(conditioning on RVs that are independent of the non-conditioning RVs)
$$\mathbf{I}(Y \cup \{y\}; \omega \mid X \cup \{x\}) - \mathbf{I}(Y; \omega \mid X \cup \{x\}) =$$
(use the identity $\mathbf{I}(A, B; C) - \mathbf{I}(B; C) = \mathbf{I}(A; C \mid B)$)
$$\mathbf{I}(\{y\}; \omega \mid Y, X \cup \{x\}) \geq 0$$

■

B.5 Relation to MI Approximation in ABCD

Here we demonstrate that though ABCD (Agrawal et al., 2019) uses an importance weighted estimate of mutual information, for the specific choice of importance weights used in ABCD, the MI estimate turns out to be the same as the one used in this work.

We note that ABCD decomposes the MI as *entropy over the SCM* as opposed to the *entropy over outcomes* used in this work.

B.5.1 Entropy Over SCM

The mutual information in (2) can be written as:

$$\mathrm{I}(\mathbf{Y}; \mathbf{\Phi} \mid \{(j, v)\}, \mathcal{D}) = \mathrm{H}(\mathbf{\Phi} \mid \{(j, v)\}, \mathcal{D}) - \mathrm{H}(\mathbf{\Phi} \mid \mathbf{Y}, \{(j, v)\}, \mathcal{D}) \tag{9}$$

where $\mathrm{H}(\cdot)$ is the *expected entropy*. As the posterior $p(\mathbf{g}, \boldsymbol{\theta}|\mathcal{D})$ does not change as a result of conditioning on the design choice $\{(j, v)\}$, the first entropy term is constant wrt $\{(j, v)\}$. Hence, selecting the most informative target corresponds to minimising the conditional entropy of the parameters $\mathbf{\Phi}$.

$$\begin{aligned} &\mathrm{H}(\mathbf{\Phi} \mid \mathbf{Y}, \{(j, v)\}, \mathcal{D}) \\ &= - \mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\},\mathcal{D})} \left[\mathop{\mathbb{E}}_{p(\boldsymbol{\phi}|\mathbf{y},\{(j,v)\},\mathcal{D})} \left[\log p(\boldsymbol{\phi} \mid \mathbf{y}, \{(j, v)\}, \mathcal{D}) \right] \right] \end{aligned} \tag{10}$$

The above equation cannot be estimated from samples of $q(\boldsymbol{\phi} \mid \mathcal{D}) \approx p(\boldsymbol{\phi} \mid \mathcal{D})$ since the posterior of the SCM would change when the interventional outcome $\mathbf{y}$ is conditioned on. To address this problem, ABCD Agrawal et al. (2019) proposes to use weighted importance sampling with weights $w = p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j, v)\}, \mathcal{D})$ and use samples from $q(\boldsymbol{\phi} \mid \mathcal{D})$.

Definition 7 *The weighted importance sampling estimate of entropy over SCM* (9) *with weights* $w(\boldsymbol{\phi})$ *is given by*

$$\widehat{\mathrm{I}}_{WIS} = \frac{1}{c_o} \sum_{i=1}^{c_o} \mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\},\mathcal{D})} \left[\log w(\widehat{\boldsymbol{\phi}}_i) \right] - \mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\},\mathcal{D})} \left[\log \left[\mathop{\mathbb{E}}_{p(\boldsymbol{\phi}|\mathcal{D},\{(j,v)\})} w(\boldsymbol{\phi}) \right] \right] \tag{11}$$

B.5.2 Entropy Over Outcomes.

We can instead consider an alternative factorisation of (2) which would not require importance sampling and also compute entropies in the lower dimensional space of experimental outcomes, as given in Equation 1.

Definition 8 *The Monte Carlo estimate of entropy over outcomes* (1) *is given by*

$$\widehat{\mathrm{I}}_{MC} = \frac{1}{c_o \times m}\sum_{i=1}^{c_o}\sum_{k=1}^{m}\log\Big(p(\widehat{\mathbf{y}}_{i,k} \mid \widehat{\boldsymbol{\phi}}_i, \{(j,v)\})\Big) - \frac{1}{c_o \times m}\sum_{i=1}^{c_o}\sum_{k=1}^{m}\log\left(\frac{1}{c_{in}}\sum_{l=1}^{c_{in}} p(\widehat{\mathbf{y}}_{i,k} \mid \widehat{\boldsymbol{\phi}}_l, \{(j,v)\})\right) \tag{12}$$

B.5.3 Relation between Approximations with Entropy over SCM and Entropy over Outcomes

We prove below that for specific choice of importance weights $w(\boldsymbol{\phi}) :== p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}, \mathcal{D})$ used in ABCD, the MI approximations due to the above two factorizations are the same..

Theorem B.1 *Let* $\widehat{\mathrm{I}}_{WIS}$ (11) *be the weighted importance sampling estimate of entropy over SCM* (9) *with weights* $w(\boldsymbol{\phi})$ *and* $\widehat{\mathrm{I}}_{MC}$ (12) *be the Monte Carlo estimate of entropy over outcomes* (1)*. Then,* $\widehat{\mathrm{I}}_{WIS} = \widehat{\mathrm{I}}_{MC}$ *if* $w(\boldsymbol{\phi}) = p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}, \mathcal{D})$*.*

Proof Consider the entropy over SCM:

$$\mathrm{I}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) = \mathrm{H}(\boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) - \mathrm{H}(\boldsymbol{\Phi} \mid \mathbf{Y}, \{(j,v)\}, \mathcal{D})$$

$$\mathrm{I}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) = \mathrm{H}(\boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) + \mathop{\mathbb{E}}_{p(\mathbf{y}\mid\{(j,v)\},\mathcal{D})}\left[\mathop{\mathbb{E}}_{p(\boldsymbol{\phi}\mid\mathbf{y},\{(j,v)\},\mathcal{D})}\left[\log p(\boldsymbol{\phi} \mid \mathbf{y}, \{(j,v)\}, \mathcal{D})\right]\right] \tag{13}$$

Consider the importance weighted estimate of the above equation with weights $w(\boldsymbol{\phi})$. We can rewrite $p(\boldsymbol{\phi} \mid \mathbf{y}, \{(j,v)\}, \mathcal{D})$ as:

$$p(\boldsymbol{\phi} \mid \mathbf{y}, \{(j,v)\}, \mathcal{D}) = \frac{w(\boldsymbol{\phi})p(\boldsymbol{\phi} \mid \mathcal{D}, \{(j,v)\})}{\mathbb{E}_{p(\boldsymbol{\phi}\mid\mathcal{D},\{(j,v)\})}\left[w(\boldsymbol{\phi})\right]} \tag{14}$$

Let $\{\widehat{\boldsymbol{\phi}}_i \sim p(\boldsymbol{\phi} \mid \mathcal{D})\}_{i=1}^{c_o}$, using (14) in (13),

$$\widehat{\mathrm{I}}_{\mathrm{WIS}}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) = \mathrm{H}(\boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) + \frac{1}{c_o}\sum_{i=1}^{c_o}\mathop{\mathbb{E}}_{p(\mathbf{y}\mid\{(j,v)\},\mathcal{D})}\log\left[\frac{w(\widehat{\boldsymbol{\phi}}_i)p(\widehat{\boldsymbol{\phi}}_i \mid \mathcal{D}, \{(j,v)\})}{\mathbb{E}_{p(\boldsymbol{\phi}\mid\mathcal{D},\{(j,v)\})}\left[w(\boldsymbol{\phi})\right]}\right] \tag{15a}$$

Furthermore, using a Monte-Carlo estimate on first term with $\widehat{\boldsymbol{\phi}}_i$, we get

$$\widehat{\mathrm{I}}_{\mathrm{WIS}}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) = \frac{1}{c_o}\sum_{i=1}^{c_o}\left[-\log p(\widehat{\boldsymbol{\phi}}_i \mid \mathcal{D}, \{(j,v)\}) + \mathop{\mathbb{E}}_{p(\mathbf{y}\mid\{(j,v)\},\mathcal{D})}\log\left(\frac{w(\widehat{\boldsymbol{\phi}}_i)p(\widehat{\boldsymbol{\phi}}_i \mid \mathcal{D}, \{(j,v)\})}{\mathbb{E}_{p(\boldsymbol{\phi}\mid\mathcal{D},\{(j,v)\})}\left[w(\boldsymbol{\phi})\right]}\right)\right] \tag{15b}$$

Focusing on the second term,

$$\mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\},\mathcal{D})} \log \left[\frac{w(\widehat{\boldsymbol{\phi}}_i) p(\widehat{\boldsymbol{\phi}}_i \mid \mathcal{D}, \{(j,v)\})}{\mathbb{E}_{p(\boldsymbol{\phi}|\mathcal{D},\{(j,v)\})}\left[w(\boldsymbol{\phi})\right]} \right]$$
$$= \mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\},\mathcal{D})} \left[\log w(\widehat{\boldsymbol{\phi}}_i)\right] + \log p(\widehat{\boldsymbol{\phi}}_i \mid \mathcal{D}, \{(j,v)\}) - \mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\},\mathcal{D})} \left[\log \left[\mathop{\mathbb{E}}_{p(\boldsymbol{\phi}|\mathcal{D},\{(j,v)\})} w(\boldsymbol{\phi}) \right]\right] \tag{16}$$

Plugging the above result back in (15b) and noticing that second term in the above equation cancels with first term in (15b), we get:

$$\widehat{\mathrm{I}}_{\mathrm{WIS}} = \frac{1}{c_o} \sum_{i=1}^{c_o} \mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\},\mathcal{D})} \left[\log w(\widehat{\boldsymbol{\phi}}_i)\right] - \mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\},\mathcal{D})} \left[\log \left[\mathop{\mathbb{E}}_{p(\boldsymbol{\phi}|\mathcal{D},\{(j,v)\})} w(\boldsymbol{\phi}) \right]\right] \tag{17}$$

$\widehat{\mathrm{I}}_{\mathrm{MC}}$ is given by (5)+(7). We can notice that $\widehat{\mathrm{I}}_{\mathrm{WIS}} = \widehat{\mathrm{I}}_{\mathrm{MC}}$ if $w(\boldsymbol{\phi}) = p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}, \mathcal{D})$ and approximating remaining expectations in the above equation with Monte Carlo samples. ■

Appendix C. Models

C.1 DiBS Hypermarameters

For optimizing DiBS Lorch et al. (2021) we used RMSProp with learning rate 0.005. Additionally, per dataset we set the following hyperparameters:

Nodes	Dataset	Graph Prior	Particles Transportation Steps	Number of Particles	Kernel
20	Scale Free	Scale Free	20000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
	Erdős-Rényi	Erdős-Rényi	20000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
50	Scale Free	Scale Free	20000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
	Erdős-Rényi	Erdős-Rényi	20000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
10	Ecoli1	Erdős-Rényi	10000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
	Ecoli2	Erdős-Rényi	10000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
	Yeast1	Erdős-Rényi	10000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
	Yeast2	Erdős-Rényi	10000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
50	Ecoli1	Erdős-Rényi	10000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
	Ecoli2	Erdős-Rényi	10000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
	Yeast1	Erdős-Rényi	10000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)
	Yeast2	Erdős-Rényi	10000	20	Frobenius Squared Exponential ($h_{\text{latent}} = 5.0, h_{\text{theta}} = 500$)

Table 1: Settings of DREAM experiments for nodes 10 and 50.

Appendix D. Datasets and Experiment details

D.1 Synthetic Graphs Experiments

In the synthetic data experiments, we focus on two types of graphs. The Erdős-Rényi and Scale Free. For linear SCMs, we sample the edge weights γ uniformly at random. For the nonlinear SCM, we parameterize each variable to be a Gaussian whose mean is a nonlinear

function of its parents. We model the nonlinear function with a neural network. In all settings, we set noise variance $\sigma^2 = 0.1$. For both types of graphs, we set the expected number of edges per vertex to 1.

Erdos-Renyi model:

We used `networkx`[1] and method `fast_gnp_random_graph` Batagelj and Brandes (2005) to generate graphs based on the Erdős-Rényi model. We set expected number of edges per vertex to 1.

Scale Free (Barabasi-Albert) graphs:

We used igraph[2] package to generate the graphs. We set the expected number of edges per vertex to 1.

D.2 DREAM Experiments

For the DREAM experiments, we used GeneNetWeaver Schaffter et al. (2011), a simulator of gene regulatory networks, based on stochastic differential equations. This simulator was used to generate data for *Dialogue for Reverse Engineering Assessments and Methods* (DREAM) Sachs et al. (2005) competition with three network inference challenges (`DREAM3`, `DREAM4` and `DREAM5`). We used the `GeneNetWeaver v3.1`[3].

Each experiment is parametrized as an xml file describing the network topology but also the crucial parameters of the stochastic differential equation that `GeneNetWeaver` simulates. In our experiments, we used `Ecoli1`, `Ecoli2`, `Yeast1` and `Yeast2` networks for 10 and 50 nodes.

Each experiment was initialized with 100 observational data. For the observational data, we used the steady state [4] of wild-type experiments. For the interventional data, we used the steady-state of knock-out experiments. Each observational or interventional sample was conducted by running the simulator with a different seed per draw.

	Dataset	Model	Starting Observational Samples	Batch Size	Number of Batches
10 nodes	`Ecoli1`	DiBS non linear	100	5	20
	`Ecoli2`	DiBS non linear	100	5	20
	`Yeast1`	DiBS non linear	100	5	20
	`Yeast2`	DiBS non linear	100	5	20
50 nodes	`Ecoli1`	DiBS non linear	100	20	20
	`Ecoli2`	DiBS non linear	100	20	20
	`Yeast1`	DiBS non linear	100	20	20
	`Yeast2`	DiBS non linear	100	20	20

Table 2: Settings of DREAM experiments for nodes 10 and 50.

1. https://networkx.org/documentation/networkx-1.10/reference/generated/networkx.generators.random_graphs.fast_gnp_random_graph.html
2. https://igraph.org/python/api/latest/igraph._igraph.GraphBase.html#Barabasi
3. https://github.com/tschaffter/genenetweaver
4. Steady state is considered the result of the simulation of the SDE for maximum 2000 steps.

Appendix E. Bayesian Optimisation

Bayesian Optimisation (BO) Kushner (1964); Zhilinskas (1975); Močkus (1975) is a global optimisation technique for optimising black-box functions. More formally, for any function U defined on a set $\mathcal{X}$ which is expensive to evaluate, BO seeks to find the maximum of the function over the entire set $\mathcal{X}$ with as few evaluations as possible.

$$\max_{x \in \mathcal{X}} \mathrm{U}(x)$$

BO typically proceeds by placing a prior on the unknown function and obtaining the posterior over this function with the queried points $\mathbf{x}^* = \{x_1^*, \dots, x_t^*\}$. A common prior is a Gaussian Process (GP) Rasmussen (2003) with mean 0 and covariance function defined by a kernel $k(x, x')$. Let $\mathrm{U}_{\mathbf{x}^*} = [\mathrm{U}(x_1^*), \dots, \mathrm{U}(x_t^*)]$ denote the vector of function evaluations, $\mathbf{K}$ the kernel matrix with $\mathbf{K}_{i,j} = k(x_i^*, x_j^*)$ and $\mathbf{k}_{t+1} = [k(x_1^*, x_{t+1}), \dots, k(x_t^*, x_{t+1})]$. The posterior predictive of point x_{t+1} can be obtained in closed form:

$$\begin{aligned}
p(\mathrm{U}) &\sim \mathcal{GP}(0, k) \\
p(\mathrm{U} \mid \mathbf{x}^*, \mathrm{U}_{\mathbf{x}^*}, x_{t+1}) &= \mathcal{N}(\boldsymbol{\mu}(x_{t+1}), \boldsymbol{\sigma}^2(x_{t+1})) \\
\boldsymbol{\mu}(x_{t+1}) &= \mathbf{k}_{t+1}^T (\mathbf{K} + \mathrm{I})^{-1} \mathrm{U}_{\mathbf{x}^*} \\
\boldsymbol{\sigma}^2(x_{t+1}) &= k(x_{t+1}, x_{t+1}) - \mathbf{k}_{t+1}^T (\mathbf{K} + \mathrm{I})^{-1} \mathbf{k}_{t+1}
\end{aligned}$$

Appendix F. Related Work

Early efforts of using *Bayesian Optimal Experimental Design for Causal Discovery* (BOECD) can be found in the works of Murphy (2001) and Tong and Koller (2001). However, these approaches deal with simple settings like limiting the graphs to topologically ordered structures, intervening sequentially, linear models, and discrete variables.

In Cho et al. (2016) and Ness et al. (2017), BOECD was applied for learning biological networks structure. More recently, ABCD framework Agrawal et al. (2019) extended the work of Murphy (2001) and Tong and Koller (2001) in the setting where interventions can be applied in batches with continuous variables. To achieve this, they (approximately) solve the submodular problem of maximizing the batched mutual information between interventions (experiments), outcomes, and observational data, given a DAG. DAG hypotheses are sampled using *DAG-bootstrap* (Friedman et al., 2013). Our work differs from ABCD in a few ways: we work with both linear and nonlinear SCMs by using state-of-the-art posterior models over DAGs Lorch et al. (2021), we apply BO to select the value to intervene with, but we also prepare the batch using *soft*BALD Kirsch et al. (2021) which is significantly faster than the greedy approximation of ABCD method.

In von Kügelgen et al. (2019) the authors proposed the use of *Gaussian Processes* to model the posterior over DAGs and then use BO to identify the value to intervene with, however, this method was not shown to be scalable for larger than bivariate graphs since they rely on multi-dimensional Gaussian Processes for modeling the conditional distributions.

A new body of work has emerged in the field of differentiable causal discovery, where the problem of finding the structure, usually from observational data, is solved with gradient ascent and functional approximators, like neural networks (Zheng et al., 2018; Ke et al., 2019;

Brouillard et al., 2020). In recent works (Cundy et al., 2021; Lorch et al., 2021; Annadani et al., 2021), the authors proposed a variational approximation of the posterior over the DAGs which allowed for modeling a distribution rather than a point estimate of the DAG that best explains the observational data $\mathcal{D}$. Such work can be used to replace *DAG-bootstrap* (Friedman et al., 2013), allowing for the modeling of posterior distributions with greater support.

Besides the BOECD-based approaches, a few active causal learning works have been proposed He and Geng (2008); Gamella and Heinze-Deml (2020); Scherrer et al. (2021). Active ICP Gamella and Heinze-Deml (2020) uses ICP Peters et al. (2016) for causal learning while using an active policy to select the target, however, this work is not applicable in the setting where the full graph needs to be recovered. Closer to our proposal belongs AIT Scherrer et al. (2021), which uses a neural network-based posterior model over the graphs but evaluates the F-score to select the interventions.

Table 3: Comparison of the proposed experimental design for causal discovery with existing experimental design for causal discovery techniques.

Method	Nonlinear	BOED	Scalable	Continuous	Finite Data	Setting the value
Murphy, Tong and Koller Murphy (2001); Tong and Koller (2001)		✓			✓	
ABCD Agrawal et al. (2019)		✓	✓	✓	✓	
Active NCM Scherrer et al. (2021)	✓			✓	✓	
Active ICP Gamella and Heinze-Deml (2020)	✓			✓	✓	
GP-UCB von Kügelgen et al. (2019)	✓	✓		✓	✓	✓
Sussex Sussex et al. (2021)		✓		✓		
Ours	✓	✓	✓	✓	✓	✓

Appendix G. Mutual Information per value for two Variables graph

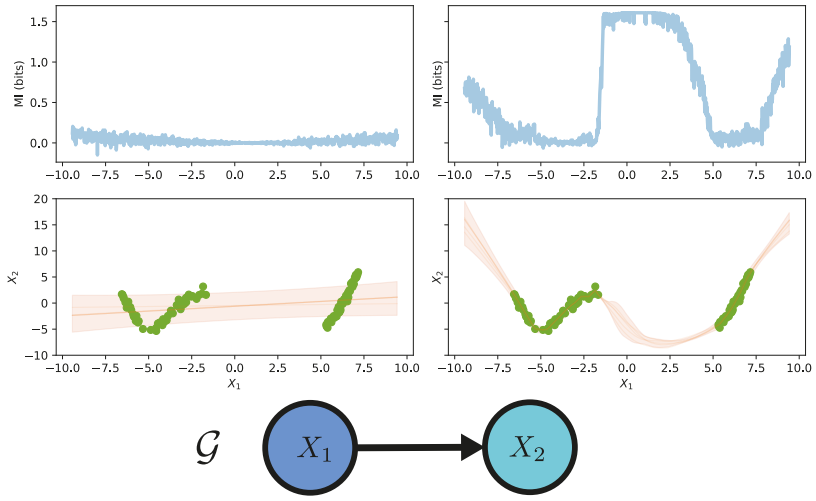


Figure 4: Estimation of the Mutual Information using two variables model $\mathcal{G}$. In green we represent the interventional data. We train an ensemble of a linear (left plot) and a non-linear (right plot) function approximator (NN) parametrizing a Gaussian Distribution. We can see that in both cases, MI is influenced by the value of intervention $\text{do}(X_1 = x_1)$. In this experiment we used the BALD estimator of the MI.

Appendix H. metrics

$\mathbb{E}$-SHD: Defined as the *expected structural hamming distance* between samples from the posterior model over graphs and the true graph $\mathbb{E}\text{-SHD} \coloneqq \mathbb{E}_{\mathbf{g}\sim p(\mathcal{G}|\mathcal{D})}\big[\text{SHD}(\mathbf{g}, \tilde{\mathbf{g}})\big]$
$\mathbb{E}$-SID: As the SHD is agnostic to the notion of intervention, Peters and Bühlmann (2015) proposed the *expected structural interventional distance* (**$\mathbb{E}$-SID**) which quantifies the differences between graphs with respect to the causal inference statements and interventional distributions.
AUROC: The *area under the receiver operating characteristic curve* of the binary classification task of predicting the presence/ absence of all edges.
AUPRC: The *area under the precision-recall curve* of the binary classification task of predicting the presence/ absence of all edges.

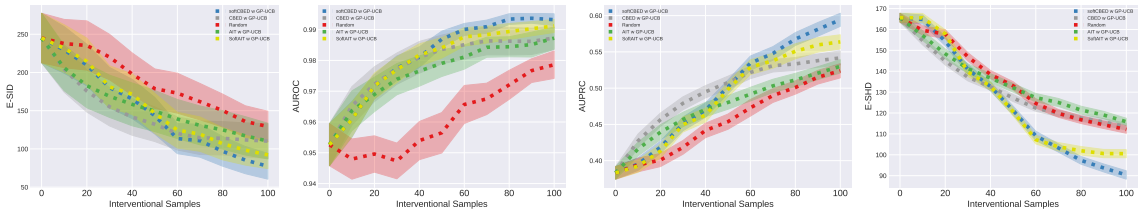
Appendix I. Walltime Performance of CBED vs Soft-CBED

Figure 5: Performance comparison between different value selection and batch strategies for `CBED`. Experiments are performed using an AMD EPYC 7662 64-Core CPU and Tesla V100 GPU.

Strategy		
Value	**Batch**	**Runtime(s)**
Fixed	Greedy	32.56
	Soft	6.42
GP-UCB	Greedy	284.98
	Soft	24.17

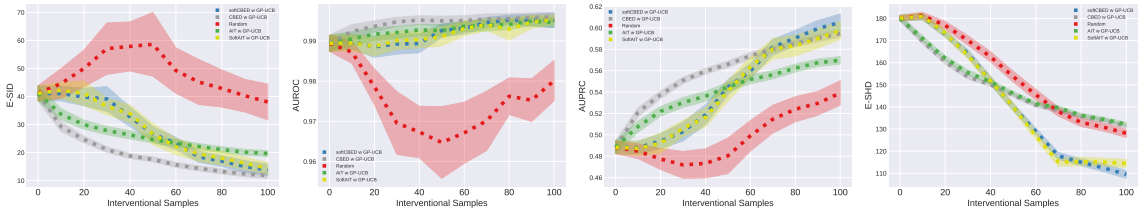
Appendix J. Complete list of Synthetic task results

Unless stated otherwise, for all the synthetic experiments we run 100 seeds, with standard error of the mean shaded.



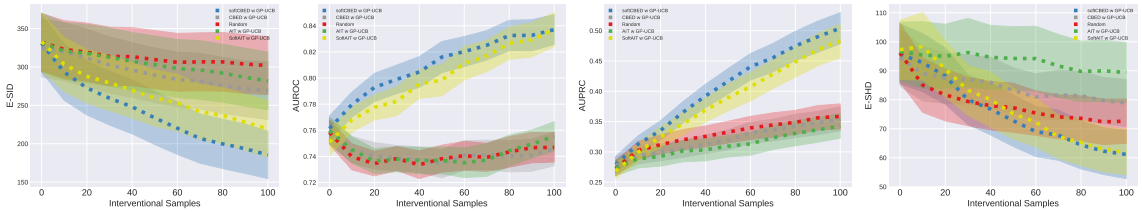
a ESID results (↓) b AUROC results (↑) c AUPRC results (↑) d ESHD results (↓)

Figure 6: Results of Erdős–Rényi Erdős and Rényi (1959) linear SCMs with 50 variables. Experiments were performed with DAG Bootstrap as the underlying posterior model.



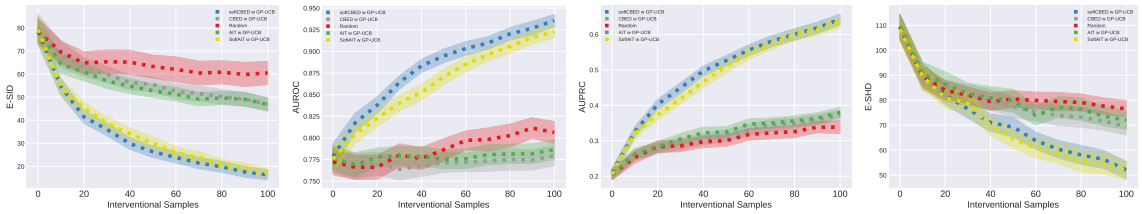
a ESID results (↓) b AUROC results (↑) c AUPRC results (↑) d ESHD results (↓)

Figure 7: Results of scale-free linear SCMs with 50 variables. Experiments were performed with DAG Bootstrap as the underlying posterior model.



a ESID results (↓) b AUROC results (↑) c AUPRC results (↑) d ESHD results (↓)

Figure 8: Results of Erdős–Rényi Erdős and Rényi (1959) nonlinear SCMs with 50 variables. Experiments were performed with DiBS as the underlying posterior model.



a ESID results (↓) b AUROC results (↑) c AUPRC results (↑) d ESHD results (↓)

Figure 9: Results of scale-free nonlinear SCMs with 50 variables. Experiments were performed with DiBS as the underlying posterior model.

Appendix K. Code Dependencies

We are using the following dependencies.

Table 4: Set-up and dataset details for non-convex, non-linear regression prblem.

Name	URL	License
jaxlib/jax	https://jax.readthedocs.io/en/latest/	Apache
causaldag	https://github.com/FenTechSolutions/CausalDiscoveryToolbox	MIT
pytorch	https://github.com/pytorch/pytorch	BSD
xarray	https://github.com/pydata/xarray	Apache
cdt	https://github.com/FenTechSolutions/CausalDiscoveryToolbox	MIT
bayesian-optimization	https://github.com/fmfn/BayesianOptimization	MIT
pgmpy	https://github.com/pgmpy/pgmpy	MIT
igraph	https://github.com/igraph/igraph	GPL-2.0
numpy	https://github.com/numpy/numpy	BSD
SciPy	https://github.com/scipy/scipy	BSD
scikit-learn	https://github.com/scikit-learn/scikit-learn	BSD
networkx	https://github.com/networkx/networkx	BSD

Appendix L. Computation requirements

Table 5: Total number of GPU hours (back-of-the-envelope estimation). Experiments are performed on an AMD EPYC 7662 64-core CPU and Tesla V100 GPU.

		Runtime per acq.	**Iterations**	**Seeds**	**Experiments**	**total (hou**
D=50	greedy ucb (CBED)	284.98	20	100	2	316.64
	greedy fixed (CBED)	32.56	20	100	2	36.17
	soft ucb (CBED)	24.17	20	100	2	26.85
	soft fixed (CBED)	6.42	20	100	2	7.13
		6.42	20	100	2	7.13
	greedy ucb (AIT)	284.98	20	100	2	316.64
	soft ucb (AIT)	24.17	20	100	2	26.85
D=20	greedy ucb (CBED)	113.992	20	100	2	126.65
	greedy fixed (CBED)	13.024	20	100	2	14.47
	soft ucb (CBED)	9.668	20	100	2	10.74
	soft fixed (CBED)	2.568	20	100	2	2.85
	soft sampled (CBED)	2.568	20	100	2	2.85
	greedy ucb (AIT)	113.992	20	100	2	126.65
	soft ucb (AIT)	9.668	20	100	2	10.74
DREAM	soft fixed (CBED)	6.42	20	6	4	0.856
	soft fixed (AIT)	6.42	20	6	4	0.856
					sum	889.31

Appendix M. License

We summarize the licenses on table 4.