Interventions, Where and How?
Experimental Design for Causal Models at Scale

Panagiotis Tigas[*1]   **Yashas Annadani**[*2,3]   **Andrew Jesson**[1]   **Bernhard Schölkopf**[3]
Yarin Gal[1]   **Stefan Bauer**[2,4]

[1]OATML, University of Oxford   [2]KTH Royal Institute of Technology, Stockholm
[3]Max Planck Institute for Intelligent Systems   [4]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 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 (Erdos-Rènyi, Scale Free) for both linear and 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 (closed-loop / policy-driven) scientific method (Fig. 1).

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 of
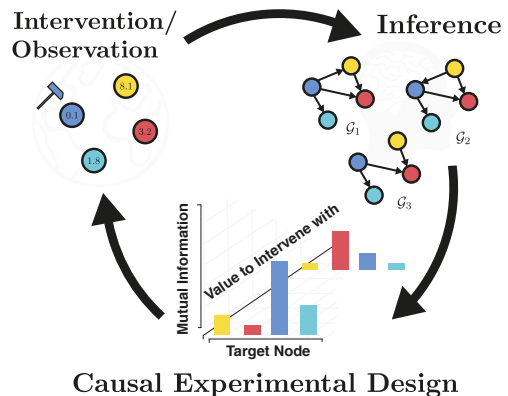
Figure 1: Intervention-Inference-Design loop of Bayesian Optimal Experimental Design for Causal Discovery framework.

---

*Equal contribution. Correspondence to `ptigas@robots.ox.ac.uk`, `yashas.annadani@gmail.com`.
Implementation available at: https://github.com/yannadani/cbed

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

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).

In the *Bayesian Optimal Experimental Design* (BOED) (Lindley, 1956) framework, one seeks the experiment that maximizes the *expected information gain* about some parameter(s) of interest. In *causal discovery*, an experiment takes the form of a causal intervention, and the parameters of interest are the Structural Causal Model (SCM) and its associated DAG.

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. 2a). 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 Bayesian experimental design method (CBED - pronounced "seabed") to acquire optimal intervention targets and values by performing Bayesian optimization.

Additionally, some settings call for the selection of a batch of interventions. The problem of batched interventions is computationally expensive as it requires evaluating all possible combinations of interventions. 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 DAG structure and the parameters of the SCM as fast as the greedy strategy. This strategy is well suited for resource-constrained settings.

Throughout this work, we make the following standard assumptions for causal discovery (Peters et al., 2017):

Assumption 1 (Causal Sufficiency). *There are no hidden confounders, and all the random variables of interest are observable.*

Assumption 2 (Finite Samples). *There is a finite number of observational/ interventional samples available.*

Assumption 3 (Nonlinear SCM with Additive Noise). *The structural causal model has nonlinear conditional expectations with additive Gaussian noise.*

Assumption 4 (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 the underlying graph is sparse, as is the case in all the real-world settings (Bengio et al., 2019, Schmidt et al., 2007). Experimental design is preferable in sparse graph settings as the number of informative intervention targets and values would be significantly less compared to dense graphs. Many nodes corresponding to a sparse graph would have very less probability of having parent sets, and hence preforming experiments with a random policy is not maximally informative. Finally, we are interested in recovering the full graph $\mathbf{G}$ with a small number of batches. As with all causal inference tasks, the assumptions that we make above have to be carefully verified for the application of interest.

We show that our methods, Greedy-CBED and Soft-CBED, perform better than the state-of-the-art active causal discovery baselines in linear and nonlinear SCM settings. In addition, our approach achieves superior results in the real-world inspired nonlinear dataset, DREAM (Greenfield et al., 2010).

2

2 Background

Notation. Let $\mathbf{V} = \{1, \ldots, d\}$ be the vertex set of any DAG $\mathbf{g} = (\mathbf{V}, E)$ and $\mathbf{X_V} = \{X_1, \ldots, X_d\} \subseteq \mathcal{X}$ be the random variables of interest indexed by $\mathbf{V}$. We have an initial observational dataset $\mathcal{D} = \{\mathbf{x_V}^{(i)}\}_{i=1}^{n}$ comprised of instances $\mathbf{x_V} \sim P(X_1 = x_1, \ldots, X_d = x_d) = p(x_1, \ldots, x_d)$.

Causal Bayesian Network. A causal Bayesian network (CBN) is the pair $(\mathbf{g}, P)$ such that for any $\mathbf{W} \subset \mathbf{V}$,

$$P(\mathbf{X_V}|\mathrm{do}(\mathbf{X_W} = \mathbf{x_W'})) = \prod_{i \in \mathbf{V} \backslash \mathbf{W}} P(X_i \mid X_{\mathrm{pa}_{\mathbf{g}}(i)}) \mathbb{1}(\mathbf{X_W} = \mathbf{x_W'})$$

where $\mathrm{do}(X_{\mathbf{W}})$ represents a hypothetical intervention on the variables $X_{\mathbf{W}}$, $\mathbb{1}(\cdot)$ is an indicator function and $\mathrm{pa}_{\mathbf{g}}(i)$ denotes parents of variable X_i in DAG $\mathbf{g}$. A perfect intervention on any variable X_j completely removes all dependencies with its parents, i.e. $P(X_j \mid X_{\mathrm{pa}_{\mathbf{g}}(j)}) = P(X_j)$ thereby resulting in a mutilated DAG $\mathbf{g}' = (\mathbf{V}, E \backslash (pa_{\mathbf{g}}(j), j))$.

Structural Causal Model. From the data generative mechanism point of view, the DAG $\mathbf{g}$ on $\mathbf{X_V}$ matches a set of *structural equations*:

$$X_i := f_i(X_{\mathrm{pa}_{\mathbf{g}}(i)}, \epsilon_i) \quad \forall i \in \mathbf{V} \tag{1}$$

where f_i's are (potentially nonlinear) causal mechanisms that remain invariant when intervening on any variable $X_j \neq X_i$. ϵ_i's are exogenous noise variables with arbitrary distribution that are mutually independent, i.e $\epsilon_i \perp\!\!\!\perp \epsilon_j \forall i \neq j$. (1) represents the conditional distributions in a Causal Bayesian Network and can additionally reveal the effect of interventions if the mechanisms are known (Peters et al., 2017, Pearl, 2009). These equations together form the structural causal model (SCM), with an associated DAG $\mathbf{g}$. Though the mechanisms f can be nonparametric in the general case, we assume that there exists a parametric approximation to these mechanisms with parameters $\boldsymbol{\gamma} \in \Gamma$. In the case of linear SCMs, $\boldsymbol{\gamma}$ corresponds to the weights of the edges in E. In the nonlinear case, they could represent the parameters of a nonlinear function that parameterizes the mean of a Gaussian distribution.

A common form of (1) corresponds to Gaussian additive noise models (ANM)[2]:

$$X_i := f_i(X_{\mathrm{pa}_{\mathbf{g}}(i)}; \boldsymbol{\gamma}_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_i^2) \tag{2}$$

An ANM is fully specified by a a DAG $\mathbf{g}$, mechanisms, $f(\cdot; \boldsymbol{\gamma}) = [f_1(\cdot; \gamma_1), \ldots, f_d(\cdot; \gamma_d)]$, parameterized by $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_d]$, and variances, $\sigma^2 = [\sigma_1^2, \ldots, \sigma_d^2]$. For notational brevity, henceforth we denote $\boldsymbol{\theta} = (\boldsymbol{\gamma}, \sigma^2)$ and all the parameters of interest with $\boldsymbol{\phi} = (\mathbf{g}, \boldsymbol{\theta})$.

Bayesian Causal Discovery. A common assumption in causal inference is that causal relations are known qualitatively and can be represented by a DAG. While this qualitative information can be obtained from domain knowledge in some scenarios, it's infeasible in most applications. The goal of causal discovery is to recover the SCM and the associated DAG, given a dataset $\mathcal{D}$. In general, without further assumptions about the nature of mechanisms f (e.g., linear vs. nonlinear), the true SCM may not be *identifiable* (Peters et al., 2012) from observational data alone. This non-identifiability is because there could be multiple DAGs (and hence multiple factorizations of $P(\mathbf{X_V})$) which explain the data equally well. Such DAGs are said to be *Markov Equivalent*. Interventions can improve identifiability. In addition to identifiability issues, estimating the functional relationships between nodes using finite data is another source of uncertainty. Bayesian parameter estimation over the unknown SCM provides a principled way to quantify these uncertainties and obtain a posterior distribution over the SCM given observational data. An experimenter can then use the knowledge encoded by the posterior to design informative experiments that efficiently *acquire interventional data to resolve unknown edge orientations and functional uncertainty.*

---

[2]ANM's can have noise variables that are non-Gaussian as well, but we restrict our exposition to the Gaussian case.

3

Bayesian Inference of SCMs and DAGs. The key challenge in performing Bayesian inference jointly over SCMs and DAGs is that the space of DAGs is discrete and superexponential in the number of variables (Peters et al., 2017). However, recent techniques based on variational inference (Annadani et al., 2021, Lorch et al., 2021, Cundy et al., 2021) provide a tractable and scalable way of performing posterior inference of these parameters. Given a tractable distribution $q_\psi(\phi)$ which approximates the posterior $p(\phi \mid \mathcal{D})$, variational inference maximizes a lower bound on the (log-) evidence:

$$\log p(\mathcal{D}) \geq \mathcal{L}(\psi \in \Psi) = \underset{q_\psi(\phi)}{\mathbb{E}} \left[\log p(\mathcal{D} \mid \phi)\right] - D_{\mathrm{KL}}(q_\psi(\phi)||p(\phi))$$

The key idea in these techniques is the way the variational family Ψ for DAGs is parameterized. The variational family for the Variational Causal Network (VCN) method (Annadani et al., 2021) is an autoregressive Bernoulli distribution over the adjacency matrix. They further enforce the acyclicity constraint (Zheng et al., 2018) through the prior. BCD-Nets (Cundy et al., 2021) consider a distribution over node orderings through a Boltzmann distribution and perform inference with Gumbel-Sinkhorn (Mena et al., 2018) operator. DiBS (Lorch et al., 2021) consider latent variables over entries of adjacency matrix and perform inference over these latent variables using SVGD (Liu and Wang, 2016). We demonstrate empirical results in BOECD using the DiBS model in this work because it is easily extendable to nonlinear SCMs.

Bayesian Optimal Experimental Design. *Bayesian Optimal Experimental Design* (BOED) (Lindley, 1956, Chaloner and Verdinelli, 1995) is an information theoretic approach to the problem of selecting the optimal experiment to estimate any parameter θ. For BOED, the *utility* of the experiment ξ is the mutual information (MI) between the observation $\mathbf{y}$ and θ:

$$U_{\mathrm{BOED}}(\xi) \triangleq I(\mathbf{Y}; \theta \mid \xi, \mathcal{D}) = \underset{p(\mathbf{y}|\theta,\xi)p(\theta|\mathcal{D})}{\mathbb{E}} \left[\log p(\mathbf{y} \mid \xi, \mathcal{D}) - \log p(\mathbf{y} \mid \theta, \xi, \mathcal{D})\right]$$

The goal of BOED is to select the experiment that maximizes this objective $\xi^* = \arg\max_\xi U_{\mathrm{BOED}}(\xi)$. Unfortunately, evaluating and optimizing this objective is challenging because of the nested expectations (Rainforth et al., 2018) and several estimators have been introduced (Foster et al., 2019, Kleinegesse and Gutmann, 2019) that lower bound the BOED objective which then can be combined with various optimization methods to select the designs (Foster et al., 2020, Ivanova et al., 2021, Foster et al., 2021, Lim et al., 2022).

A common setting, called *static*, *fixed* or *batch* design, is to optimize $\mathcal{B}$ designs $\{\xi_1, \ldots, \xi_\mathcal{B}\}$ at the same time. The designs are then executed and the experimental outcomes are collected to update the model parameters in a Bayesian fashion.

3 Method

The true SCM and the associated DAG $\tilde{\phi} = (\tilde{\mathbf{g}}, \tilde{\boldsymbol{\theta}})$ 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 $\boldsymbol{\Phi}$, of which ϕ is an outcome. Let $\phi \sim p(\phi|\mathcal{D}) \propto p(\mathcal{D} \mid \phi)p(\phi)$ 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 $\xi := \{(j, v)\} := \mathrm{do}(X_j = v)$ that maximizes the information gain about $\boldsymbol{\Phi}$ after observing the outcome of the intervention $\mathbf{y} \sim P(X_1 = x_1, \ldots, X_d = x_d \mid \mathrm{do}(X_j = v) = p(\mathbf{y} \mid \xi)$. 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 $\xi := \{(j, v)\}$ that maximizes the information gain is the experiment that maximizes the mutual information between $\boldsymbol{\Phi}$ and $\mathbf{Y}$:

$$\{(j^*, v^*)\} = \underset{j,v}{\arg\max} \left\{I(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j, v)\}, \mathcal{D})\right\} \tag{3}$$

The above objective considers taking `arg max` over not just the discrete set of intervention targets $j \in \mathbf{V}$, but also over the uncountable set of intervention values $v \subset \mathcal{X}_j$. While the existing works in BOECD consider only the design of intervention targets to limit the complexity (Tong and Koller, 2001, Murphy, 2001, Agrawal et al., 2019), our approach tackles both the problems. We first outline the methodology for a single design and in Section 3.2 demonstrate how to extend this single design to a batch setting.

4
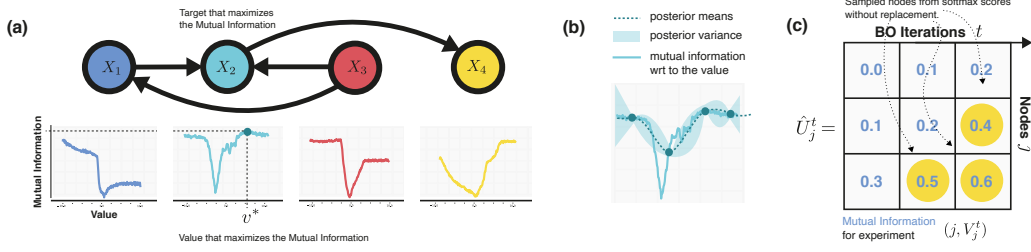
Figure 2: (a) Each graph shows how the **mutual information** (MI) (y-axis) changes for intervening on that node (plot color matching the node color) with different **values** (x-axis). The SCM in this example is a nonlinear SCM with Additive Gaussian noise. We can see that by intervening on node X_2 with the value v^*, the mutual information gets maximized. **(b)** The posterior distribution of a GP on the Mutual Information function as a response to different intervention values after four Bayesian Optimization (BO) steps. **(c)** 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 (3.2).

3.1 Single Design

To maximize the objective in Equation 3, 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 $\mathbf{Y}$:

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

$$= - \underset{p(\mathbf{y}\mid\{(j,v)\},\mathcal{D})}{\mathbb{E}} \left[\log \left(\underset{p(\boldsymbol{\phi}\mid\mathcal{D})}{\mathbb{E}} [p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\})] \right) \right] + \underset{p(\boldsymbol{\phi}\mid\mathcal{D})}{\mathbb{E}} \left[\underset{p(\mathbf{y}\mid\boldsymbol{\phi},\{(j,v)\})}{\mathbb{E}} [\log \left(p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\})\right)] \right] \tag{4}$$

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 (4) 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. As shown in (3), 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 $v \in \mathcal{X}_j$ (See Fig 2) 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.

BO typically proceeds by placing a Gaussian Process (GP) (Rasmussen, 2003) prior on the function $\mathcal{I}(\{j, \cdot\})$ and obtain the posterior of this function with the queried points $\mathbf{v}^* = \{v^{(1)*}, \ldots, v^{(T)*}\}$. Let the value of the mutual information queried at each optimization step t be $\hat{\mathrm{U}}_j^t = \mathcal{I}(\{(j, v^{(t)*}\})$. The posterior *predictive* of a point $v^{(t+1)}$ can be obtained in closed form as a Gaussian with mean $\boldsymbol{\mu}_j^{(t+1)}(v)$ and variance $\boldsymbol{\sigma}_j^{(t+1)}(v)$. Subscript j signifies the fact that we maintain different μ and σ per intervention target and superscript t represents the BO step. Querying proceeds by having an acquisition function defined on this posterior, which suggests the next point to query. For BO, we use

5

an acquisition function called as the Upper Confidence Bound (UCB) (Srinivas et al., 2010) which suggests the next point to query by trading-off *exploration* and *exploitation* with a hyperparameter β_j^t: $v_j^{(t+1)*} = \arg\max_v \boldsymbol{\mu}_j^t(v) + \sqrt{\beta_j^{t+1}} \boldsymbol{\sigma}_j^t(v)$.

We run GP-UCB 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.

3.2 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, 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`.

3.3 Comparison with existing active causal discovery methods

We outline how our approach compares with two main existing active causal discovery methods.

ABCD (Agrawal et al., 2019). The estimator of MI used in ABCD is based on weighted importance sampling. However, for the specific choice of the importance sampling weights used in ABCD, their MI estimator ends up with the same approximation as in our method (see Appendix B.5). Nevertheless, ABCD does not select intervention values but suboptimally sets them to a fixed value. In addition, our proposed `Soft-CBED` is a faster and more efficient batch strategy, especially when values also have to be acquired. From this perspective, our approach is an extension of ABCD with nonlinear assumptions, value acquisition, and a soft top-k batching strategy.

AIT (Scherrer et al., 2021). AIT is an F-score-based intervention target acquisition strategy. Although it is not a BOECD method, we prove here that it can be viewed as a Monte Carlo estimate of the approximation to MI when the outcomes $\mathbf{Y}$ are Gaussian. Nevertheless, AIT does not select intervention values like ABCD and does not have a batch strategy.

Theorem 3.1. *Let $\mathbf{Y}$ be a Gaussian random variable. Then the discrepancy score of Scherrer et al. (2021) is a Monte Carlo estimate of an approximation to mutual information (Eq. (4)). See Appendix B.6 for proof.*

6

Algorithm 1: `Greedy-CBED`	**Algorithm 2:** `Soft-CBED`
Input : $\mathcal{E}$ environment, N initial observational samples, $\mathcal{B}$ batch Size, d number of nodes	**Input** : $\mathcal{E}$ environment, N initial observational samples, $\mathcal{B}$ batch Size, d number of nodes, ζ softmax temperature
▷ Initialize set of experiments Ξ to empty	**1** **for** $j = 1 \ldots d$ **do**
1 $\Xi \leftarrow \varnothing$	 ▷ Select candidate intervention values per node j using GP-UCB
2 **for** $n = 1 \ldots \mathcal{B}$ **do**	**2** Initialize μ_j^0 and σ_j^0
3 **for** $j = 1 \ldots d$ **do**	**3** **for** $t = 1 \ldots T$ **do**
    ▷ Select optimal intervention value per node j using GP-UCB	**4** $V_j^t \leftarrow \arg\max_v \mu_j^{t-1}(v) + \sqrt{\beta^t}\sigma_j^{t-1}(v)$
4 $V_j \leftarrow \arg\max_v \mathcal{I}(\Xi \cup \{(j,v)\})$	**5** $\hat{U}_j^t \leftarrow \mathcal{I}(\{(j,V_j^t)\})$
5 $U_j \leftarrow \mathcal{I}(\Xi \cup \{(j,V_j)\})$	**6** Update the GP to obtain μ_j^t and σ_j^t
6 $j^* \leftarrow \arg\max_j U_j$	**7** $\{(t_i,j_i)\}_{i\in\{1,\ldots,\mathcal{B}\}} \leftarrow \mathcal{B}$ samples *without* replacement $\propto \exp(\hat{U}_j^t/\zeta)$
7 $v^* \leftarrow V_{j^*}$	**8** $\Xi \leftarrow \{(j_i, V_{j_i}^{t_i})\}_{i\in\{1,\ldots,\mathcal{B}\}}$
8 $\Xi \leftarrow \Xi \cup \{(j^*,v^*)\}$	**9** **return** Ξ
9 **return** Ξ	

4 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. BOECD was also explored in Greenewald et al. (2019) under the assumption that undirected edges of the graph always forms a tree. 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, Bengio et al., 2019). 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, Shanmugam et al., 2015, Squires et al., 2020, Kocaoglu et al., 2017). 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,

7
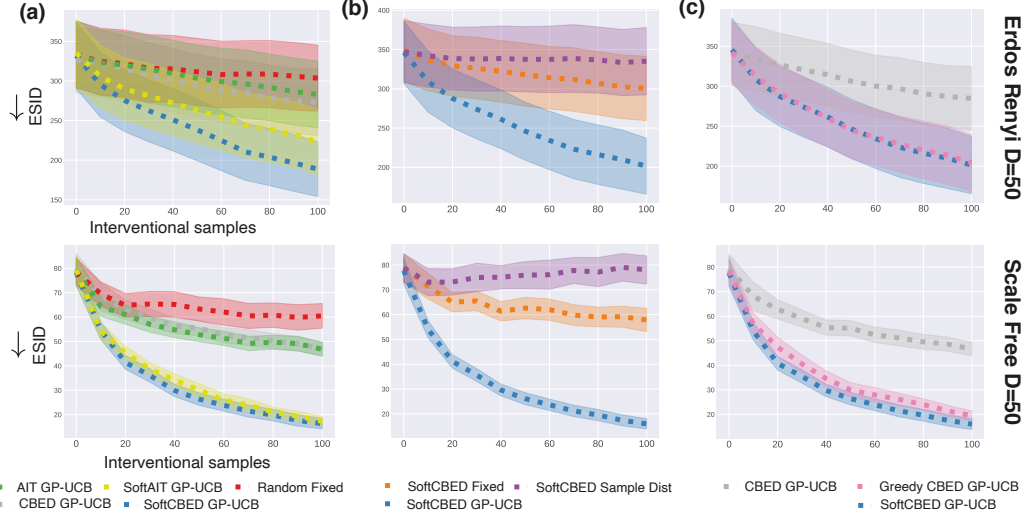
Figure 3: Results on the $\mathbb{E}$-**SID** $\downarrow$ metric (100 seeds, with standard error of the mean shaded) for 50 variables involving nonlinear functional relationships and additive Gaussian noise. (a) We show that `Soft-CBED` with GP-UCB value selection strategy significantly outperforms 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.

this work is not applicable in the setting where the full graph needs to be recovered. In Zhang et al. (2021), the authors propose an active learning method to the problem of identifying the interventions that push a dynamical causal network towards a desired state. A few approaches tackle the problem of actively acquiring interventional data to orient edges of a skeletal graph (Shanmugam et al., 2015, Squires et al., 2020, Kocaoglu et al., 2017). 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.

5 Experiments

We evaluate the performance of our method on synthetic and real-world causal experimental design problems and a range of baselines. We aim to investigate the following aspects empirically: (1) competitiveness of the overall proposed strategies of `Greedy-CBED` and `Soft-CBED` at scale (50 nodes) on synthetic datasets; (2) performance of the value acquisition strategy based on GP-UCB; and (3) performance of the proposed approach on a real-world inspired dataset.

5.1 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 B.6 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 3.2.

CBED / GreedyCBED / SoftCBED . These are the Monte Carlo estimates of MI, as described in section 3. 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 3.2 and Algorithm 2.

8

5.2 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.

5.3 Tasks

Synthetic Graphs.   In this setting, we generate Erdős–Rényi (Erdős and Rényi, 1959) (ER) and Scale-Free (SF) graphs (Barabási and Albert, 1999) of size 20 and 50. 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. 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. A set of ODEs and SDEs generates the dataset, simulating the reverse-engineered networks of single cells. We use `GeneNetWeaver` (Schaffter et al., 2011) to simulate the steady-state *wind-type* expression and single-gene *knockouts*. Refer to appendix D.2 for the exact settings.

5.4 Metrics

$\mathbb{E}$**-SHD**: Defined as the *expected structural hamming distance* between samples from the posterior model over graphs and the true graph $\mathbb{E}\text{-SHD} := \mathbb{E}_{\mathbf{g}\sim p(\mathcal{G}|\mathcal{D})}\big[\text{SHD}(\mathbf{g}, \tilde{\mathbf{g}})\big]$

$\mathbb{E}$**-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.

6 Results

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**. We evaluate these metrics as 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 I.

Table 1: 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

On the synthetic graphs (Figure 3(a)), we can see that for ER and 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. In addition, we found this trend to hold for $20D$ variables and linear models. Full results are presented in the appendix I.

9

Figure 4: Comparison of acquisition functions on DREAM dataset, for 50 dimensions and batch size 10 on $\mathbb{E}$**-SID**↓ metric (6 seeds, with standard error of the mean).

Next, we examine the importance of having a value selection strategy for active causal discovery. We use the MI estimator in Equation 4; 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 3(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 3(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 1. 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 4). 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).

7 Summary and Conclusions

This paper studies the problem of efficiently selecting the Bayes optimal experiments to discover causal models. Our proposed framework simultaneously answers the questions of *where* and *how* to intervene in a batched setting. We present a Bayesian optimization strategy to acquire interventional targets and values. Further, we propose two different batching strategies: one based on greedy selection and the other based on soft top-k selection. The proposed methodology for selecting intervention target-value pairs in a batched setting provides superior performance over the state-of-the-art for causal models up to $50D$ variables. We validate this using synthetic datasets and using real-world inspired datasets of single-cell regulatory networks, showing the potential impact on areas like biology and other experimental sciences.

Acknowledgments and Disclosure of Funding

We would like to thank Nino Scherrer, Tom Rainforth, Desi R. Ivanova and all anonymous reviewers for sharing their valuable feedback and insights. Panagiotis Tigas is supported by the UK EPSRC CDT in Autonomous Intelligent Machines and Systems (grant reference EP/L015897/1). We are grateful for compute from the Berzelius Cluster and the Swedish National Supercomputer Centre.

10

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.

Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. A meta-transfer objective for learning to disentangle causal mechanisms. *arXiv preprint arXiv:1901.10912*, 2019.

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

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

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, Eli Bingham, Paul Horsfall, Yee Whye Teh, Tom Rainforth, and Noah Goodman. Variational bayesian optimal experimental design. *arXiv preprint arXiv:1903.05480*, 2019.

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

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

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.

Kristjan Greenewald, Dmitriy Katz, Karthikeyan Shanmugam, Sara Magliacane, Murat Kocaoglu, Enric Boix Adsera, and Guy Bresler. Sample efficient active learning of causal trees. *Advances in Neural Information Processing Systems*, 32, 2019.

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.

11

Desi R Ivanova, Adam Foster, Steven Kleinegesse, Michael U Gutmann, and Thomas Rainforth. Implicit deep adaptive design: policy-based experimental design without likelihoods. *Advances in Neural Information Processing Systems*, 34:25785–25798, 2021.

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.

Steven Kleinegesse and Michael U Gutmann. Efficient bayesian experimental design for implicit models. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 476–485. PMLR, 2019.

Murat Kocaoglu, Alex Dimakis, and Sriram Vishwanath. Cost-optimal learning of causal graphs. In *International Conference on Machine Learning*, pages 1875–1884. PMLR, 2017.

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.

Vincent Lim, Ellen Novoseller, Jeffrey Ichnowski, Huang Huang, and Ken Goldberg. Policy-based bayesian experimental design for non-differentiable implicit models. *arXiv preprint arXiv:2203.04272*, 2022.

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

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *arXiv preprint arXiv:1608.04471*, 2016.

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

Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning latent permutations with gumbel-sinkhorn networks. *arXiv preprint arXiv:1802.08665*, 2018.

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, Joris Mooij, Dominik Janzing, and Bernhard Schölkopf. Identifiability of causal graphs using functional models. *arXiv preprint arXiv:1202.3757*, 2012.

12

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.

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

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.

Mark Schmidt, Alexandru Niculescu-Mizil, Kevin Murphy, et al. Learning graphical model structure using l1-regularization paths. In *AAAI*, volume 7, pages 1278–1283, 2007.

Karthikeyan Shanmugam, Murat Kocaoglu, Alexandros G Dimakis, and Sriram Vishwanath. Learning causal graphs with small interventions. *Advances in Neural Information Processing Systems*, 28, 2015.

Chandler Squires, Sara Magliacane, Kristjan Greenewald, Dmitriy Katz, Murat Kocaoglu, and Karthikeyan Shanmugam. Active structure learning of causal dags via directed clique trees. *Advances in Neural Information Processing Systems*, 33:21500–21511, 2020.

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.

Jiaqi Zhang, Chandler Squires, and Caroline Uhler. Matching a desired causal state via shift interventions. *Advances in Neural Information Processing Systems*, 34:19923–19934, 2021.

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.

Checklist

1. For all authors...

    (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]

    (b) Did you describe the limitations of your work? [Yes] See end of Section 1. Our limitations arise from the fact that the proposed methodology and conclusion hold when the assumptions laid out are satisfied.

    (c) Did you discuss any potential negative societal impacts of your work? [Yes] Negative societal impact is discussed in appendix A and will be added to the extra page of the camera ready after the reviewing process.

    (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...

    (a) Did you state the full set of assumptions of all theoretical results? [Yes] The assumptions are laid out in Introduction as well as in Theorem 3.1.

    (b) Did you include complete proofs of all theoretical results? [Yes] The proofs are in Appendix.

3. If you ran experiments...

    (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]

    (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix D.1, D.2 and H

    (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Figure 3 and 4.

    (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Please check appendix K for details.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

    (a) If your work uses existing assets, did you cite the creators? [Yes]

    (b) Did you mention the license of the assets? [Yes] Please check appendix L for details.

    (c) Did you include any new assets either in the supplemental material or as a URL? [No]

    (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] Data are simulated.

    (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] Data are simulated.

5. If you used crowdsourcing or conducted research with human subjects...

    (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

    (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

    (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

14

A Potential negative societal impacts

Causal Experimental Design has the potential to impact several sectors; *healthcare, biology, mechanical and material engineering, computational advertisement,* to name a few. As any AI powered field, it can have negative societal impact when being used by malicious actors.

B Theoretical Results

B.1 Deriving the Mutual Information over Outcomes

In the following lemma, we derive the mutual information over outcomes given in (4).

Lemma B.1.

$$
\begin{aligned}
&\mathrm{I}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) \\
&= - \underset{p(\mathbf{y}\mid\{(j,v)\},\mathcal{D})}{\mathbb{E}} \left[\log \left(\underset{p(\boldsymbol{\phi}\mid\{(j,v)\},\mathcal{D})}{\mathbb{E}} [p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\})] \right) \right] + \underset{p(\boldsymbol{\phi}\mid\mathcal{D})}{\mathbb{E}} \left[\underset{p(\mathbf{y}\mid\boldsymbol{\phi},\{(j,v)\})}{\mathbb{E}} [\log (p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}))] \right]
\end{aligned}
\tag{5}
$$

Proof.

$$
\begin{aligned}
\mathrm{I}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) &= \mathrm{H}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D}) - \mathrm{H}(\mathbf{Y} \mid \boldsymbol{\Phi}, \{(j,v)\}, \mathcal{D}) & \text{(6a)} \\
&= \mathrm{H}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D}) - \underset{p(\boldsymbol{\phi}\mid\mathcal{D})}{\mathbb{E}} [\mathrm{H}(\mathbf{Y} \mid \boldsymbol{\phi}, \{(j,v)\})] & \text{(6b)}
\end{aligned}
$$

$$
= - \underset{p(\mathbf{y}\mid\{(j,v)\},\mathcal{D})}{\mathbb{E}} [\log (p(\mathbf{y} \mid \{(j,v)\}, \mathcal{D}))] + \underset{p(\boldsymbol{\phi}\mid\mathcal{D})}{\mathbb{E}} \left[\underset{p(\mathbf{y}\mid\boldsymbol{\phi},\{(j,v)\})}{\mathbb{E}} [\log (p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}))] \right]
\tag{6c}
$$

$$
= - \underset{p(\mathbf{y}\mid\{(j,v)\},\mathcal{D})}{\mathbb{E}} \left[\log \left(\int_{\boldsymbol{\phi}} p(\mathbf{y}, \boldsymbol{\phi} \mid \{(j,v)\}, \mathcal{D}) d\boldsymbol{\phi} \right) \right] + \underset{p(\boldsymbol{\phi}\mid\mathcal{D})}{\mathbb{E}} \left[\underset{p(\mathbf{y}\mid\boldsymbol{\phi},\{(j,v)\})}{\mathbb{E}} [\log (p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}))] \right]
\tag{6d}
$$

$$
\begin{aligned}
&= - \underset{p(\mathbf{y}\mid\{(j,v)\},\mathcal{D})}{\mathbb{E}} \left[\log \left(\int_{\boldsymbol{\phi}} p(\boldsymbol{\phi} \mid \{(j,v)\}, \mathcal{D}) p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}, \mathcal{D}) d\boldsymbol{\phi} \right) \right] \\
&\quad + \underset{p(\boldsymbol{\phi},\mid\mathcal{D})}{\mathbb{E}} \left[\underset{p(\mathbf{y}\mid\boldsymbol{\phi},\{(j,v)\})}{\mathbb{E}} [\log (p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}))] \right]
\end{aligned}
\tag{6e}
$$

$$
= - \underset{p(\mathbf{y}\mid\{(j,v)\},\mathcal{D})}{\mathbb{E}} \left[\log \left(\underset{p(\boldsymbol{\phi}\mid\{(j,v)\},\mathcal{D})}{\mathbb{E}} [p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\})] \right) \right] + \underset{p(\boldsymbol{\phi}\mid\mathcal{D})}{\mathbb{E}} \left[\underset{p(\mathbf{y}\mid\boldsymbol{\phi},\{(j,v)\})}{\mathbb{E}} [\log (p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}))] \right] .
\tag{6f}
$$

$\square$

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

$$
\widehat{\mathrm{I}}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) = \widehat{\mathrm{H}}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D}) - \widehat{\mathrm{H}}(\mathbf{Y} \mid \boldsymbol{\Phi}, \{(j,v)\}, \mathcal{D})
\tag{7}
$$

15

Algorithm 3: Mutual Information Computation

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

$\triangleright$Sample from the posterior

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

$\triangleright$Sample from mutilated SCMs

2 $\{\widehat{\mathbf{y}}_{i,j,k} \sim p(\mathbf{y} \mid \widehat{\boldsymbol{\phi}}_i, \{(j, v)\})\}_{k=1}^m$

3 **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 B.1. *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:*

$$-\frac{1}{c_o \times m} \sum_{i=1}^{c_o} \sum_{k=1}^m \log \left(\frac{1}{c_{in}} \sum_{l=1}^{c_{in}} p(\widehat{\mathbf{y}}_{i,k} \mid \widehat{\boldsymbol{\phi}}_l, \{(j, v)\}) \right), \qquad (8)$$

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 ith 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 lth 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 $\mathrm{H}(\mathbf{Y} \mid \{(j, v)\}, \mathcal{D})$:

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

$$\leq -\mathop{\mathbb{E}}_{p(\mathbf{y} \mid \{(j,v)\}, \mathcal{D})} \left[\mathop{\mathbb{E}}_{p(\boldsymbol{\phi} \mid \mathcal{D})} [\log (p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j, v)\}))] \right],$$

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

Definition B.2. *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,* $\mathbb{E}_{p(\mathbf{y} \mid \{(j,v)\}, \mathcal{D})} \left[\mathbb{E}_{p(\boldsymbol{\phi}, \mid \mathcal{D})} [\log (p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j, v)\}))] \right]$*, is given by:*

$$-\frac{1}{c_o \times c_{in} \times m} \sum_{i=1}^{c_o} \sum_{k=1}^m \sum_{l=1}^{c_{in}} \log \left(p(\widehat{\mathbf{y}}_{i,k} \mid \widehat{\boldsymbol{\phi}}_l, \{(j, v)\}) \right), \qquad (9)$$

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

Definition B.3. *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:*

$$-\frac{1}{c_o \times m} \sum_{i=1}^{c_o} \sum_{k=1}^m \log \left(p(\widehat{\mathbf{y}}_{i,k} \mid \widehat{\boldsymbol{\phi}}_i, \{(j, v)\}) \right), \qquad (10)$$

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 ith of* c_o *graphs* $\widehat{\boldsymbol{\phi}}_i \sim p(\boldsymbol{\phi} \mid \mathcal{D})$ *augmented by intervention* $\{(j, v)\}$*.*

16

B.3 Monte Carlo Estimator of the Batch Mutual Information

While Equation 4 pertains to MI for a single design, we present here the MI estimator for the batch design.

$$
\begin{aligned}
\mathrm{I}(\mathbf{Y};\boldsymbol{\Phi}\mid \boldsymbol{\Xi},\mathcal{D}) &= \sum_{\{(j,v)\}\in\boldsymbol{\Xi}} \mathrm{I}(\mathbf{Y};\boldsymbol{\Phi}\mid \{(j,v)\},\mathcal{D}) \\
&= \sum_{\{(j,v)\}\in\boldsymbol{\Xi}} \mathrm{H}(\mathbf{Y}\mid \{(j,v)\},\mathcal{D}) - \mathrm{H}(\mathbf{Y}\mid \boldsymbol{\Phi},\{(j,v)\},\mathcal{D}) \\
&= -\sum_{\{(j,v)\}\in\boldsymbol{\Xi}} \mathop{\mathbb{E}}_{p(\mathbf{y}\mid\{(j,v)\},\mathcal{D})}\left[\log\left(\mathop{\mathbb{E}}_{p(\boldsymbol{\phi}\mid\mathcal{D})}[p(\mathbf{y}\mid\boldsymbol{\phi},\{(j,v)\})]\right)\right] + \mathop{\mathbb{E}}_{p(\boldsymbol{\phi}\mid\mathcal{D})}\left[\mathop{\mathbb{E}}_{p(\mathbf{y}\mid\boldsymbol{\phi},\{(j,v)\})}[\log\left(p(\mathbf{y}\mid\boldsymbol{\phi},\{(j,v)\})\right)]\right]
\end{aligned}
\tag{11}
$$

B.4 Mutual Information Submodularity and Monotonicity Proofs

Theorem B.2. $\mathbf{I}(Y;\omega\mid X)$ *is submodular.*

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

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

□

Theorem B.3. $\mathbf{I}(Y;\omega\mid X)$ *is non-decreasing.*

Proof.

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

□

17

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 (3) can be written as:

$$I(\mathbf{Y}; \mathbf{\Phi} \mid \{(j,v)\}, \mathcal{D}) = H(\mathbf{\Phi} \mid \{(j,v)\}, \mathcal{D}) - H(\mathbf{\Phi} \mid \mathbf{Y}, \{(j,v)\}, \mathcal{D}) \qquad (12)$$

where $H(\cdot)$ 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 $\mathbf{\Phi}$.

$$H(\mathbf{\Phi} \mid \mathbf{Y}, \{(j,v)\}, \mathcal{D})$$

$$= - \mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\}, \mathcal{D})} \left[\mathop{\mathbb{E}}_{p(\boldsymbol{\phi}|\mathbf{y}, \{(j,v)\}, \mathcal{D})} \left[\log p(\boldsymbol{\phi} \mid \mathbf{y}, \{(j,v)\}, \mathcal{D}) \right] \right] \qquad (13)$$

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 B.4. *The weighted importance sampling estimate of entropy over SCM* (12) *with weights* $w(\boldsymbol{\phi})$ *is given by*

$$\widehat{I}_{WIS} = \frac{1}{c_o} \sum_{i=1}^{c_o} \mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\}, \mathcal{D})} \left[\log w(\widehat{\boldsymbol{\phi}}_i) \right] - \mathop{\mathbb{E}}_{p(\mathbf{y}|\{(j,v)\}, \mathcal{D})} \left[\log \left[\mathop{\mathbb{E}}_{p(\boldsymbol{\phi}|\mathcal{D}, \{(j,v)\})} w(\boldsymbol{\phi}) \right] \right] \qquad (14)$$

B.5.2 Entropy Over Outcomes.

We can instead consider an alternative factorisation of (3) which would not require importance sampling and also compute entropies in the lower dimensional space of experimental outcomes, as given in Equation 4.

Definition B.5. *The Monte Carlo estimate of entropy over outcomes* (4) *is given by*

$$\widehat{I}_{MC} = \frac{1}{c_o \times m} \sum_{i=1}^{c_o} \sum_{k=1}^{m} \log \left(p(\widehat{\mathbf{y}}_{i,k} \mid \widehat{\boldsymbol{\phi}}_i, \{(j,v)\}) \right) - \frac{1}{c_o \times m} \sum_{i=1}^{c_o} \sum_{k=1}^{m} \log \left(\frac{1}{c_{in}} \sum_{l=1}^{c_{in}} p(\widehat{\mathbf{y}}_{i,k} \mid \widehat{\boldsymbol{\phi}}_l, \{(j,v)\}) \right) \qquad (15)$$

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

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

Theorem B.1. *Let* $\widehat{I}_{WIS}$ (14) *be the weighted importance sampling estimate of entropy over SCM* (12) *with weights* $w(\boldsymbol{\phi})$ *and* $\widehat{I}_{MC}$ (15) *be the Monte Carlo estimate of entropy over outcomes* (4). *Then,* $\widehat{I}_{WIS} = \widehat{I}_{MC}$ *if* $w(\boldsymbol{\phi}) = p(\mathbf{y} \mid \boldsymbol{\phi}, \{(j,v)\}, \mathcal{D})$.

Proof. Consider the entropy over SCM:

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

18

$$I(\mathbf{Y}; \mathbf{\Phi} \mid \{(j,v)\}, \mathcal{D}) = H(\mathbf{\Phi} \mid \{(j,v)\}, \mathcal{D}) + \mathbb{E}_{p(\mathbf{y}\mid\{(j,v)\},\mathcal{D})} \left[\mathbb{E}_{p(\phi\mid\mathbf{y},\{(j,v)\},\mathcal{D})} \left[\log p(\phi \mid \mathbf{y}, \{(j,v)\}, \mathcal{D}) \right] \right]$$
(16)

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

$$p(\phi \mid \mathbf{y}, \{(j,v)\}, \mathcal{D}) = \frac{w(\phi) p(\phi \mid \mathcal{D}, \{(j,v)\})}{\mathbb{E}_{p(\phi\mid\mathcal{D},\{(j,v)\})}\left[w(\phi)\right]}$$
(17)

Let $\{\widehat{\phi}_i \sim p(\phi \mid \mathcal{D})\}_{i=1}^{c_o}$, using (17) in (16),

$$\widehat{I}_{\text{WIS}}(\mathbf{Y}; \mathbf{\Phi} \mid \{(j,v)\}, \mathcal{D}) = H(\mathbf{\Phi} \mid \{(j,v)\}, \mathcal{D}) + \frac{1}{c_o}\sum_{i=1}^{c_o} \mathbb{E}_{p(\mathbf{y}\mid\{(j,v)\},\mathcal{D})} \log\left[\frac{w(\widehat{\phi}_i) p(\widehat{\phi}_i \mid \mathcal{D}, \{(j,v)\})}{\mathbb{E}_{p(\phi\mid\mathcal{D},\{(j,v)\})}\left[w(\phi)\right]} \right]$$
(18a)

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

$$\widehat{I}_{\text{WIS}}(\mathbf{Y}; \mathbf{\Phi} \mid \{(j,v)\}, \mathcal{D}) = \frac{1}{c_o}\sum_{i=1}^{c_o} \left[-\log p(\widehat{\phi}_i \mid \mathcal{D}, \{(j,v)\}) + \mathbb{E}_{p(\mathbf{y}\mid\{(j,v)\},\mathcal{D})} \log\left(\frac{w(\widehat{\phi}_i) p(\widehat{\phi}_i \mid \mathcal{D}, \{(j,v)\})}{\mathbb{E}_{p(\phi\mid\mathcal{D},\{(j,v)\})}\left[w(\phi)\right]} \right) \right]$$
(18b)

Focusing on the second term,

$$\mathbb{E}_{p(\mathbf{y}\mid\{(j,v)\},\mathcal{D})} \log\left[\frac{w(\widehat{\phi}_i) p(\widehat{\phi}_i \mid \mathcal{D}, \{(j,v)\})}{\mathbb{E}_{p(\phi\mid\mathcal{D},\{(j,v)\})}\left[w(\phi)\right]} \right]$$

$$= \mathbb{E}_{p(\mathbf{y}\mid\{(j,v)\},\mathcal{D})} \left[\log w(\widehat{\phi}_i) \right] + \log p(\widehat{\phi}_i \mid \mathcal{D}, \{(j,v)\}) - \mathbb{E}_{p(\mathbf{y}\mid\{(j,v)\},\mathcal{D})} \left[\log\left[\mathbb{E}_{p(\phi\mid\mathcal{D},\{(j,v)\})} w(\phi) \right] \right]$$
(19)

Plugging the above result back in (18b) and noticing that second term in the above equation cancels with first term in (18b), we get:

$$\widehat{I}_{\text{WIS}} = \frac{1}{c_o}\sum_{i=1}^{c_o} \mathbb{E}_{p(\mathbf{y}\mid\{(j,v)\},\mathcal{D})} \left[\log w(\widehat{\phi}_i) \right] - \mathbb{E}_{p(\mathbf{y}\mid\{(j,v)\},\mathcal{D})} \left[\log\left[\mathbb{E}_{p(\phi\mid\mathcal{D},\{(j,v)\})} w(\phi) \right] \right]$$
(20)

$\widehat{I}_{\text{MC}}$ is given by (8)+(10). We can notice that $\widehat{I}_{\text{WIS}} = \widehat{I}_{\text{MC}}$ if $w(\phi) = p(\mathbf{y} \mid \phi, \{(j,v)\}, \mathcal{D})$ and approximating remaining expectations in the above equation with Monte Carlo samples. □

B.6 Information Theoretic Interpretation of Neural Causal Models with Active Interventions

Here we provide the proof for Theorem 3.1.

Definition B.6. *The discrepancy score for a target j in AIT (Scherrer et al., 2021) is given by:*

$$D_j \equiv \frac{VBG_j}{VWG_j} \equiv \frac{\sum_k \left(\widehat{\mu}_k^j - \widehat{\mu}^j \right)^2}{\sum_i \sum_k \left(\widehat{\mathbf{y}}_{i,j,k} - \widehat{\mu}_k^j \right)^2}$$
(21)

where $\widehat{\mathbf{y}}_{i,j,k}$ is the interventional sample from a hypothetical intervention on node j on a graph i sampled from the model. $\widehat{\mu}_i^j$ is the sample mean over samples k in $\widehat{\mathbf{y}}_{i,j,k}$ and $\widehat{\mu}^j$ is the mean over all graphs and samples.

We restate Theorem 3.1 for the sake of completeness.

Theorem 3.1. *Let $\mathbf{Y}$ be a Gaussian random variable. Then the discrepancy score of Scherrer et al. (2021) is a Monte Carlo estimate of an approximation to mutual information (Eq. (4)). See Appendix B.6 for proof.*

19

Proof. Mutual Information over outcomes is given by

$$\mathrm{I}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) = \mathrm{H}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D}) - \mathrm{H}(\mathbf{Y} \mid \boldsymbol{\Phi}, \{(j,v)\}, \mathcal{D}) \tag{22a}$$

$$= \mathrm{H}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D}) - \underset{p(\boldsymbol{\phi}\mid\mathcal{D})}{\mathbb{E}} [\mathrm{H}(\mathbf{Y} \mid \boldsymbol{\phi}, \{(j,v)\})] \tag{22b}$$

Making use of the fact that $\mathbf{Y}$ is Gaussian, its entropies are given by:

$$\tag{22c}$$

$$= \frac{1}{2} \log \left(2\pi \mathrm{Var}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D})\right) - \frac{1}{2} \underset{p(\boldsymbol{\phi}\mid\mathcal{D})}{\mathbb{E}} \left[\log \left(2\pi \mathrm{Var}(\mathbf{Y} \mid \boldsymbol{\phi}, \{(j,v)\})\right)\right] \tag{22d}$$

$$\geq \frac{1}{2} \log \left(2\pi \mathrm{Var}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D})\right) - \frac{1}{2} \log \left(\underset{p(\boldsymbol{\phi}\mid\mathcal{D})}{\mathbb{E}} \left[2\pi \mathrm{Var}(\mathbf{Y} \mid \boldsymbol{\phi}, \{(j,v)\})\right]\right) \tag{22e}$$

$$= \frac{1}{2} \log \left(\frac{\mathrm{Var}(\mathbf{Y} \mid \{(j,v)\}, \mathcal{D}))}{\mathbb{E}_{p(\boldsymbol{\phi}\mid\mathcal{D})}\left[\mathrm{Var}(\mathbf{Y} \mid \boldsymbol{\phi}, \{(j,v)\})\right]}\right) \tag{22f}$$

$$= \frac{1}{2} \log \left(\frac{\mathbb{E}_{p(\mathbf{y}\mid\{(j,v)\},\mathcal{D})}\left[\left(\mathbf{y} - \mathbb{E}_{p(\mathbf{y}\mid\{(j,v)\},\mathcal{D})}[\mathbf{y}]\right)^2\right]}{\mathbb{E}_{p(\boldsymbol{\phi}\mid\mathcal{D})}\left[\mathbb{E}_{p(\mathbf{y}\mid\boldsymbol{\phi},\{(j,v)\})}\left[\left(\mathbf{y} - \mathbb{E}_{p(\mathbf{y}\mid\boldsymbol{\phi},\{(j,v)\})}[\mathbf{y}]\right)^2\right]\right]}\right) \tag{22g}$$

$$= \frac{1}{2} \log \left(\frac{\mathbb{E}_{p(\boldsymbol{\phi}\mid\mathcal{D})}\left[\mathbb{E}_{p(\mathbf{y}\mid\boldsymbol{\phi},\{(j,v)\})}\left[\left(\mathbf{y} - \mu^k\right)^2\right]\right]}{\mathbb{E}_{p(\boldsymbol{\phi}\mid\mathcal{D})}\left[\mathbb{E}_{p(\mathbf{y}\mid\boldsymbol{\phi},\{(j,v)\})}\left[\left(\mathbf{y} - \mu_{\boldsymbol{\phi}}^k\right)^2\right]\right]}\right) \tag{22h}$$

$$\leq \frac{\mathbb{E}_{p(\boldsymbol{\phi}\mid\mathcal{D})}\left[\mathbb{E}_{p(\mathbf{y}\mid\boldsymbol{\phi},\{(j,v)\})}\left[\left(\mathbf{y} - \mu^k\right)^2\right]\right]}{\mathbb{E}_{p(\boldsymbol{\phi}\mid\mathcal{D})}\left[\mathbb{E}_{p(\mathbf{y}\mid\boldsymbol{\phi},\{(j,v)\})}\left[\left(\mathbf{y} - \mu_{\boldsymbol{\phi}}^k\right)^2\right]\right]} \tag{22i}$$

Since the bounds are in the opposite direction in (22e) and (22i), we cannot obtain a single common bound to MI but instead only a rough approximation given by (22i). We can now define a Monte Carlo estimate of the above *approximation*:

$$\widehat{\mathrm{I}}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) \equiv \frac{\sum_i \sum_k \left(\widehat{\mathbf{y}}_{i,j,k} - \widehat{\mu}^j\right)^2}{\sum_i \sum_k \left(\widehat{\mathbf{y}}_{i,j,k} - \widehat{\mu}_k^j\right)^2} \tag{23a}$$

Consider an m-sample Monte Carlo estimate over samples y, i.e $i = 1, \ldots, m$, then

$$\widehat{\mathrm{I}}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) = \frac{m^2 \sum_k \left(\sum_i \frac{\widehat{\mathbf{y}}_{i,j,k}}{m} - \frac{\widehat{\mu}^j}{m}\right)^2}{\sum_i \sum_k \left(\widehat{\mathbf{y}}_{i,j,k} - \widehat{\mu}_k^j\right)^2} \tag{23b}$$

$$= \frac{m^2 \sum_k \left(\widehat{\mu}_k^j - \frac{\widehat{\mu}^j}{m}\right)^2}{\sum_i \sum_k \left(\widehat{\mathbf{y}}_{i,j,k} - \widehat{\mu}_k^j\right)^2} \tag{23c}$$

For a single sample Monte-Carlo estimate $m = 1$, $\widehat{\mathrm{I}}(\mathbf{Y}; \boldsymbol{\Phi} \mid \{(j,v)\}, \mathcal{D}) = D_j$. $\qquad\square$

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:

20

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

C.2 DAG Bootstrap

The DAG bootstrap bootstraps observations and interventions to infer a different causal structure per bootstrap. We used GIES as the causal inference algorithm because of the adaptation of GES on interventional data as well. In our experiments, we used the pcalg R implementation `https://github.com/cran/pcalg/blob/master/R/gies.R` to discover 100 graphs. Each graph can be seen as a posterior sample from $p(\mathbf{G} \mid \mathcal{D})$. For each of the sampled graphs G_i we compute the appropriate θ_{MLE} under linear Gaussian assumption for the conditional distributions.

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.

Erdos-Renyi model:

We used `networkx`[3] 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[4] package to generate the graphs. We set the expected number of edges per vertex to 1.

For all the synthetic graph experiments, we used batch size of 10 and number of iterations of 10.

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`[5].

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 [6] of wild-type experiments. For the interventional data, we used the steady-state of

---

[3]`https://networkx.org/documentation/networkx-1.10/reference/generated/networkx.generators.random_graphs.fast_gnp_random_graph.html`

[4]`https://igraph.org/python/api/latest/igraph._igraph.GraphBase.html#Barabasi`

[5]`https://github.com/tschaffter/genenetweaver`

[6]Steady state is considered the result of the simulation of the SDE for maximum 2000 steps.

21

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 3: Settings of DREAM experiments for nodes 10 and 50.

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

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

For the Gaussian Process, we used the following hyperparameters. Matern kernel, `length scale` 1.0, `length scale` bounds (lower=$1e-5$, upper=$1e5$), `Nu` (smoothness of learned function) 2.5. Also added $1e-6$ to the diagonal of the kernel matrix.

F Related Work

Table 4: 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 (2001), Tong and Koller (2001)		✓			✓	
Agrawal et al. (2019)		✓	✓	✓	✓	
Scherrer et al. (2021)	✓			✓	✓	
Gamella and Heinze-Deml (2020)	✓			✓	✓	
von Kügelgen et al. (2019)	✓	✓		✓	✓	✓
Sussex et al. (2021)		✓		✓		
Ours	✓	✓	✓	✓	✓	✓

22

G   Mutual Information per value for two Variables graph
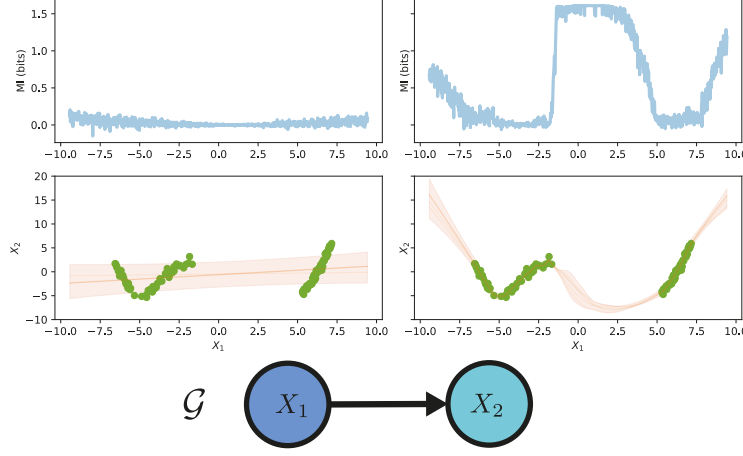
Figure 5: Estimation of the Mutual Information using two variables model $\mathcal{G}$. In green we represent the interventional data. We train an ensemble of a linear (left plot) and a non-linear (right plot) function approximator (NN) parametrizing a Gaussian Distribution. We can see that in both cases, MI is influenced by the value of intervention $\mathrm{do}(X_1 = x_1)$. In this experiment we used the BALD estimator of the MI.

H   metrics

$\mathbb{E}$-**SHD**: Defined as the *expected structural hamming distance* between samples from the posterior model over graphs and the true graph $\mathbb{E}\text{-SHD} := \mathbb{E}_{\mathbf{g} \sim p(\mathcal{G}|\mathcal{D})}\big[\mathrm{SHD}(\mathbf{g}, \tilde{\mathbf{g}})\big]$

$\mathbb{E}$-**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.

23

I 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 (↓)
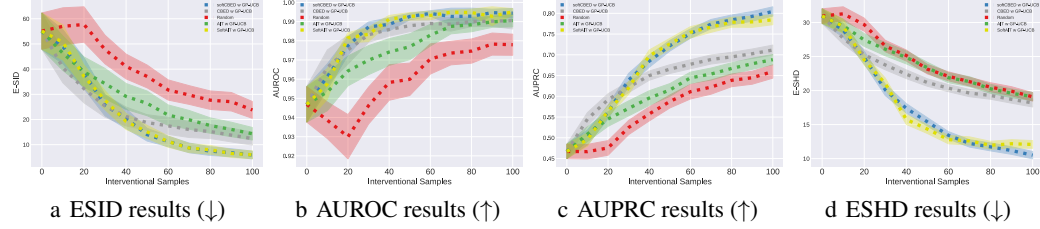
Figure 6: Results of Erdős–Rényi (Erdős and Rényi, 1959) linear SCMs with 20 variables. Experiments were performed with DAG Bootstrap as the underlying posterior model.

a ESID results (↓) b AUROC results (↑) c AUPRC results (↑) d ESHD results (↓)
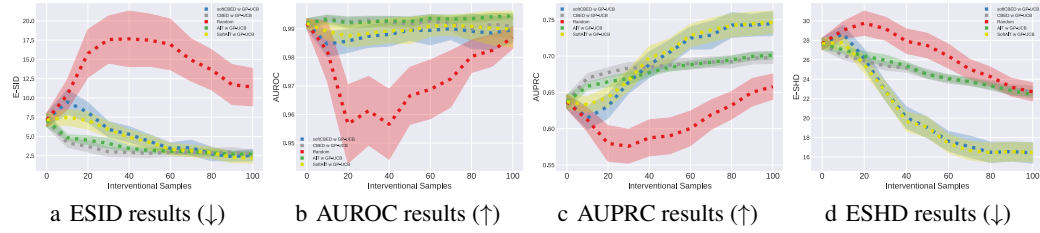
Figure 7: Results of scale-free linear SCMs with 20 variables. Experiments were performed with DAG Bootstrap as the underlying posterior model.

a ESID results (↓) b AUROC results (↑) c AUPRC results (↑) d ESHD results (↓)
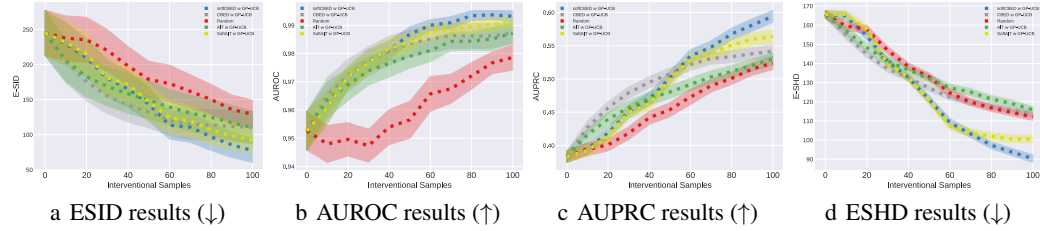
Figure 8: Results of Erdős–Rényi (Erdős and Rényi, 1959) linear SCMs with 50 variables. Experiments were performed with DAG Bootstrap as the underlying posterior model.

a ESID results (↓) b AUROC results (↑) c AUPRC results (↑) d ESHD results (↓)
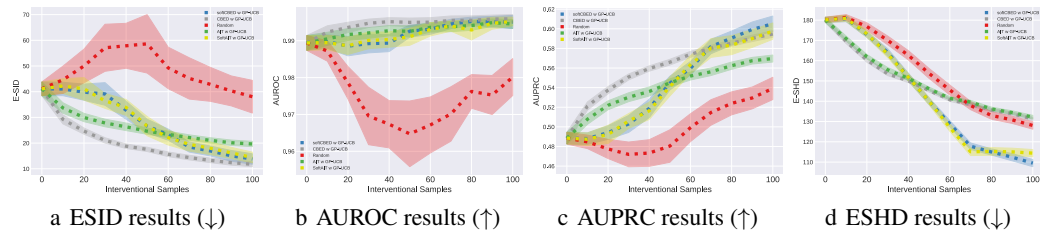
Figure 9: Results of scale-free linear SCMs with 50 variables. Experiments were performed with DAG Bootstrap as the underlying posterior model.

24

a ESID results (↓)   b AUROC results (↑)   c AUPRC results (↑)   d ESHD results (↓)
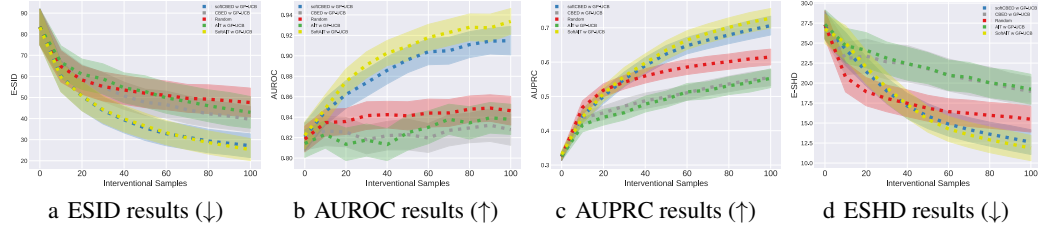
Figure 10: Results of Erdős–Rényi (Erdős and Rényi, 1959) nonlinear SCMs with 20 variables. Experiments were performed with DiBS as the underlying posterior model.

a ESID results (↓)   b AUROC results (↑)   c AUPRC results (↑)   d ESHD results (↓)
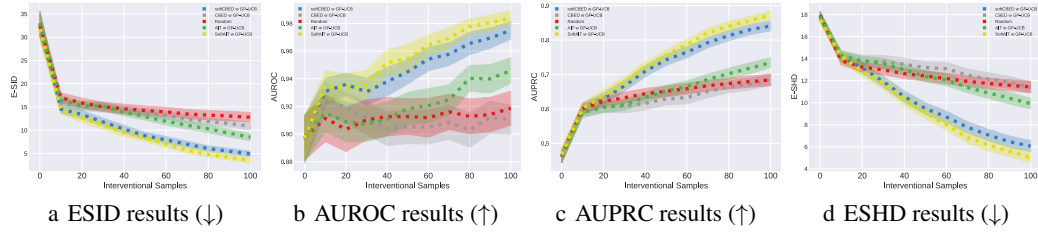
Figure 11: Results of scale-free nonlinear SCMs with 20 variables. Experiments were performed with DiBS as the underlying posterior model.

a ESID results (↓)   b AUROC results (↑)   c AUPRC results (↑)   d ESHD results (↓)
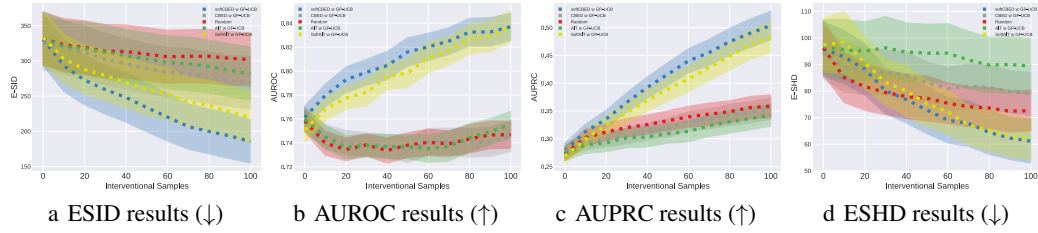
Figure 12: Results of Erdős–Rényi (Erdős and Rényi, 1959) nonlinear SCMs with 50 variables. Experiments were performed with DiBS as the underlying posterior model.

a ESID results (↓)   b AUROC results (↑)   c AUPRC results (↑)   d ESHD results (↓)
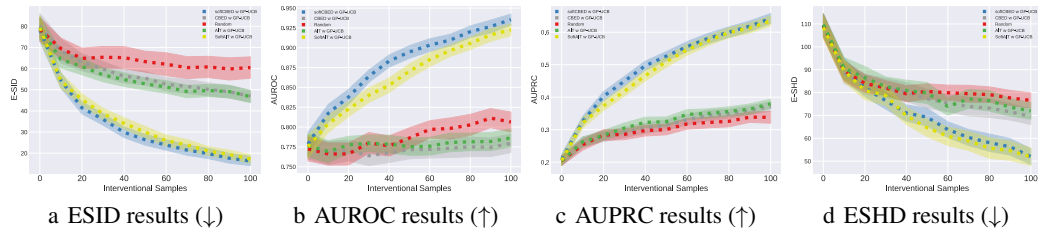
Figure 13: Results of scale-free nonlinear SCMs with 50 variables. Experiments were performed with DiBS as the underlying posterior model.

J   Code Dependencies

We are using the following dependencies.

25

Table 5: 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

K   Computation requirements

Table 6: 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 (hours)
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

L   License

We summarize the licenses on table 5.

26