Bayesian Analysis (0000) **00**, Number 0, pp. 1

Bayesian optimal experimental design for inferring causal structure

Michele Zemplenyi* and Jeffrey W. Miller*

Abstract. Inferring the causal structure of a system typically requires interventional data, rather than just observational data. Since interventional experiments can be costly, it is preferable to select interventions that yield the maximum amount of information about a system. We propose a novel Bayesian method for optimal experimental design by sequentially selecting interventions that minimize the expected posterior entropy as rapidly as possible. A key feature is that the method can be implemented by computing simple summaries of the current posterior, avoiding the computationally burdensome task of repeatedly performing posterior inference on hypothetical future datasets drawn from the posterior predictive. After deriving the method in a general setting, we apply it to the problem of inferring causal networks. We present a series of simulation studies, in which we find that the proposed method performs favorably compared to existing alternative methods. Finally, we apply the method to real data from two gene regulatory networks.

Keywords: Optimal experimental design, active learning, graphical models.

1 Introduction

Inferring the causal structure of a set of related variables is key to understanding how a system works. By interpreting directed edges as implying causal relationships, a causal network model extends standard (non-causal) graphical models by specifying the distribution of the data when one or more variables are manipulated (Pearl, 2000). A large body of work exists on learning the structure of graphical models from observational data, that is, data passively collected without any experimental intervention performed on the system under study; see Daly et al. (2011) for a comprehensive review. However, typically, observational data alone can only reveal the structure of a graphical model up to the Markov equivalence class containing the true data-generating graph (Verma and Pearl, 1991). To fully determine the structure of a causal network without making additional assumptions about specific functional model classes and error distributions, interventional data are needed to resolve the directionality of uncompelled edges (Peters et al., 2011). Interventional data result from experiments in which one or more nodes have been actively manipulated, for example, by activating or inhibiting the expression of a gene in a model organism (Sachs et al., 2005; Nagarajan et al., 2013).

Crucially, different intervention experiments yield different amounts of information about the causal structure. Thus, since experiments are often expensive and time-consuming, it is advantageous to select interventions that provide the maximum amount

*Harvard T.H. Chan School of Public Health, 677 Huntington Ave, Boston, MA 02115 mzemplenyi@g.harvard.edu jwmiller@hsph.harvard.edu

 DOI: 0000

of information. Optimal experimental design (OED) methods, also referred to as active learning algorithms, attempt to optimize this experiment selection process by providing a means of evaluating which experiments should be performed next given the current state of knowledge. From the Bayesian perspective, a naive approach would be as follows: for each candidate experiment, generate hypothetical datasets from the posterior predictive, perform posterior inference on each dataset, and compute a functional of the posterior that summarizes the amount of information gained. Averaging over many datasets would yield an estimate of the posterior expected amount of information gain for each candidate experiment. However, this naive approach would involve an inordinate amount of computation.

In this article, we develop a novel Bayesian OED technique that is principled and computationally tractable. Roughly speaking, we consider the asymptotic information gain that each experiment would yield in the limit of infinitely many replicates, as a proxy for the expected gain from finitely many replicates. Under fairly general conditions, in this limit, the posterior is simply obtained by restricting the current posterior to a subset of the parameter space. Thus, it turns out that the reduction in entropy can be easily computed using samples from the current posterior, without generating or performing inference on any hypothetical datasets. This leads to a vast reduction in the computational burden required to select experiments.

Based on this principle, we introduce a class of entropy-based criteria for determining the optimal intervention to perform in the next experiment. After the selected experiment is performed and new experimental data is obtained, we update the posterior on graphs and use it to select the next experiment. To sample from the posterior distribution over graphs, we employ an existing Markov chain Monte Carlo (MCMC) algorithm with efficient dynamic programming-based proposals (Eaton and Murphy, 2007a). By iterating between experimentation and analysis in this cyclical fashion, we focus the data collection efforts in a way that reduces posterior uncertainty as rapidly as possible. We compare our method to other active learning and structure learning approaches in the context of several simulated data sets, as well as real data from a Perturb-seq dataset (Dixit et al., 2016) and the Sachs cell-signaling network, a commonly studied benchmark in the causal network literature (Sachs et al., 2005; Eaton and Murphy, 2007b; Cho et al., 2016; Ness et al., 2018).

The article is organized as follows. In Section 2, we derive our general criterion for selecting optimal experiments. In Section 3, we apply our general criterion to causal network models. In Section 4, we lay out the overall proposed framework, along with implementation details about the entropy-based criteria and the MCMC algorithm. Section 5 discusses related previous work on OED and active learning methods. In Section 6, we present a collection of simulation studies. Section 7 contains an application to the Perturb-seq dataset generated by Dixit et al. (2016). We conclude with a brief discussion of our findings and directions for further research. Additional results, including an extensive analysis of the Sachs network, can be found in the Appendix.

2 General criterion

In this section, we derive our general criterion for selecting optimal experiments. Readers less familiar with the optimal experimental design literature may find the intuition provided in Appendix A helpful before proceeding.

2.1 Preliminaries

Suppose $(\theta, \nu) \sim \pi$, $X_1, \ldots, X_N | \theta, \nu \sim P_{\theta,\nu}$ i.i.d., and $f(\theta)$ is a function of θ with the following three properties.

Condition 2.1.

(a) $\theta \perp\!\!\!\perp X_{1:N} \mid f(\theta)$,

(b) $f(\theta)$ *is identifiable, in the sense that there is a function* g *such that* $g(P_{\theta,\nu}) = f(\theta)$ *almost surely, and*

(c) $f(\theta)$ *can only take one of finitely many values.*

In an experimentation context, θ is a parameter of interest, ν is a nuisance parameter, π is the prior, $f(\theta)$ is the answer to a research question f (for example, is a certain hypothesis true), and $X_{1:N} = (X_1, \ldots, X_N)$ are data from N replicates of an experiment performed to obtain information about $f(\theta)$. The nuisance parameter ν is sometimes needed for the i.i.d. assumption to hold. The interpretation of Condition 2.1(a) is that if we know the true answer to the research question f, then the experiment provides no additional information about θ. Condition 2.1(b) means that the answer $f(\theta)$ is uniquely determined by the distribution of X_n, and thus, in the limit as $N \to \infty$, $f(\theta)$ can be recovered from $X_1, \ldots, X_N$.

For the causal network models that we consider in subsequent sections, θ is a directed acyclic graph and f is a set-valued function corresponding to an equivalence class of graphs. First, however, we consider a general model and any f satisfying Condition 2.1.

2.2 Approximate information gain

The entropy of a random variable Y is defined as $H(Y) := -\int p(y) \log p(y) d\mu(y)$ where $p(y)$ is the density of Y with respect to some dominating measure μ, or more succinctly, $H(Y) = -\mathrm{E}(\log p(Y))$. Similarly, $H(Y|Z) = -\mathrm{E}(\log p(Y|Z))$, where the expectation is over the joint distribution of Y and Z; thus, unlike the conditional expectation $\mathrm{E}(Y|Z)$, the conditional entropy $H(Y|Z)$ is not a random variable.

By standard properties of entropy, since $f(\theta)$ is a function of θ, the posterior entropy is

$$H(\theta \mid X_{1:N}) = H(\theta \mid f(\theta), X_{1:N}) + H(f(\theta) \mid X_{1:N}).$$

By Condition 2.1(a), $H(\theta \mid f(\theta), X_{1:N}) = H(\theta \mid f(\theta))$. Further, $H(\theta \mid f(\theta)) = H(\theta) - H(f(\theta))$ again using that $f(\theta)$ is a function of θ. By Conditions 2.1(b) and 2.1(c), we

have $H(f(\theta) \mid X_{1:N}) \to 0$ as $N \to \infty$, because the posterior on $f(\theta)$ is guaranteed to concentrate at a single value (Doob, 1949; Miller, 2018); see Lemma B.1 for details. Thus, we have the following result.

Theorem 2.2. *If $(\theta, \nu) \sim \pi$, $X_1, \ldots, X_N | \theta, \nu \sim P_{\theta,\nu}$ i.i.d., and $f(\theta)$ satisfies Condition 2.1, then*

$$H(\theta \mid X_{1:N}) \xrightarrow[N\to\infty]{} H(\theta) - H(f(\theta)). \tag{1}$$

In other words, when N is sufficiently large, the difference between the prior entropy $H(\theta)$ and the posterior entropy $H(\theta|X_{1:N})$ is approximately equal to $H(f(\theta))$. Put another way, the information gained—in terms of the reduction in entropy—is approximately equal to the entropy of the answer $f(\theta)$ *under the prior*. Thus, to estimate the information to be gained by a particular question f, we need only work with the prior — not the posterior $\theta|X_{1:N}$ for a yet unobserved dataset $X_{1:N}$.

2.3 Selection of experiments using approximate information gain

To apply this result to select experiments, suppose that instead of $\pi(\theta, \nu)$ being the prior, $\pi(\theta, \nu)$ is the current posterior given all the data from any previous experiments. Let $\mathcal{E}$ denote a set of possible experiments. For each experiment $e \in \mathcal{E}$, let $X^e_1, \ldots, X^e_N | \theta, \nu \sim P^e_{\theta,\nu}$ i.i.d. be hypothetical random data from N replicates of experiment e. Suppose $f_e(\theta)$ satisfies Condition 2.1 above.

Then $p(\theta \mid X^e_{1:N}) \propto p(X^e_{1:N} \mid \theta)\, \pi(\theta)$ is the posterior distribution of θ given the new data $X^e_{1:N}$ as well as data from any previous experiments. The expected posterior entropy of θ after experiment e is then

$$H(\theta \mid X^e_{1:N}) = \int H(\theta \mid X^e_{1:N} = x^e_{1:N})\, p(x^e_{1:N})\, dx^e_{1:N} \tag{2}$$

where $p(x^e_{1:N})$ is the posterior predictive distribution for experiment e given the data from previous experiments. We would like to choose e to minimize the expected posterior entropy $H(\theta \mid X^e_{1:N})$. However, approximating $H(\theta \mid X^e_{1:N})$ via Monte Carlo is computationally intensive, since for every e, it would typically involve (i) simulating T hypothetical datasets $x^{e,1}_{1:N}, \ldots, x^{e,T}_{1:N}$ from the posterior predictive $p(x^e_{1:N})$, (ii) approximating the resulting new posteriors, $p(\theta \mid X^e_{1:N} = x^{e,t}_{1:N})$ for $t = 1, \ldots, T$, for example, by running T MCMC chains, and (iii) approximating the entropy $H(\theta \mid X^e_{1:N} = x^{e,t}_{1:N})$ for $t = 1, \ldots, T$, in order to form a Monte Carlo approximation $H(\theta \mid X^e_{1:N}) \approx \frac{1}{T} \sum_{t=1}^{T} \widehat{H}(\theta \mid X^e_{1:N} = x^{e,t}_{1:N})$ using Equation 2.

In contrast, the computation is vastly simplified using the approximation in Equation 1. First, note that by Equation 1,

$$H(\theta \mid X^e_{1:N}) \approx H(\theta) - H(f_e(\theta)). \tag{3}$$

Since $H(\theta)$ does not depend on e, this implies that minimizing $H(\theta \mid X^e_{1:N})$ is approximately equivalent to maximizing $H(f_e(\theta))$. Further, since $H(f_e(\theta))$ depends only on π

and f_e (and not on $P^e_{\theta,\nu}$ or $X^e_{1:N}$), it is often relatively easy to approximate $H(f_e(\theta))$ using posterior samples $\theta_1, \ldots, \theta_T$. Specifically, we can generate a single set of samples $\theta_1, \ldots, \theta_T$ from the current posterior π, and then for each potential experiment $e \in \mathcal{E}$, compute

$$H(f_e(\theta)) = -\sum_y p(f_e(\theta) = y) \log p(f_e(\theta) = y) \approx -\sum_y \hat{p}_e(y) \log \hat{p}_e(y) \tag{4}$$

where $\hat{p}_e(y) := \frac{1}{T} \sum_{t=1}^T \mathbb{1}\big(f_e(\theta_t) = y\big)$ and $\mathbb{1}(\cdot)$ is the indicator function. In Equation 4, the sum is over all values y in the range of f_e. Thus, our proposed method of choosing e is as follows.

1. Generate samples $\theta_1, \ldots, \theta_T$ from the current posterior π.
2. Compute $\hat{p}_e(y) = \frac{1}{T} \sum_{t=1}^T \mathbb{1}\big(f_e(\theta_t) = y\big)$ for each candidate experiment e.
3. Select the experiment e with the largest value of $\hat{h}_e := -\sum_y \hat{p}_e(y) \log \hat{p}_e(y)$.

Note that, equivalently, $\hat{h}_e = -\sum_{A \in \mathcal{A}_e} \hat{\pi}(A) \log \hat{\pi}(A)$ where $\hat{\pi} = \frac{1}{T} \sum_{t=1}^T \delta_{\theta_t}$ and $\mathcal{A}_e$ is the partition of θ-space induced by f_e, since $\hat{p}_e(y) = \hat{\pi}(A)$ where $A = \{\theta : f_e(\theta) = y\}$. Therefore, we can interpret $\hat{h}_e$ as an approximation to the current posterior entropy of the partition induced by f_e.

It is important to note that although our criterion is motivated by the asymptotics as the number of replicates goes to ∞, it accounts for finite sample uncertainty due to the fact that the posterior π quantifies our uncertainty in θ based on finitely many previous experiments and finitely many replicates of each previous experiment. Also, a further advantage of our approach is that $f(\theta)$ often takes a small number of values, such as two for a binary function, and thus, $H(f(\theta))$ is often much easier to estimate from samples than $H(\theta \mid X^e_{1:N})$ or even $H(\theta)$.

3 Criterion for causal network models

In this section, we apply our general criterion to the setting of causal network models. First, we define the model we will use and provide some intuition for partitions of graph space that are informed by interventional experiments.

3.1 Causal network models

We use the standard causal network model specification, which we review here. Note that it is common to refer to these models as "Bayesian networks" (Pearl, 2000), but we avoid this term because the Bayesian aspect of our methodology comes from the use of posterior distributions to quantify uncertainty, rather than from features inherent to the model itself.

All graphical models use nodes to represent random variables and edges to represent the probabilistic relationships among nodes. In contrast to traditional graphical models,

which only specify the joint distribution in the observational setting, a causal network model also specifies the joint distribution when one or more nodes are manipulated. To represent this causal structure, it is standard to use a directed acyclic graph (DAG) along with the conditional probability distribution (CPD) of each node given the values of its parent nodes. The directed edges in this structure represent cause and effect relationships between parent and child nodes. We refer to a DAG topology along with all the CPDs as a *causal network model*. Our assumptions implicitly entail the causal Markov, causal sufficiency, and causal faithfulness conditions, as well as the assumption of no selection bias, all of which are common in the causal structure learning literature.

In a causal network model, the graph topology and the CPDs can be viewed as specifying an algorithm for generating data under manipulation of the nodes. Here, we assume interventions that assign some subset of nodes to values that may be fixed or random, but are independent of all other nodes. Thus, when intervening on node i, we effectively sever all incoming edges to node i. The data generating process under such an intervention can be described as follows: each manipulated node is set to its assigned value and each non-manipulated node is drawn from its CPD, proceeding in an ordering of the nodes that ensures parents are drawn before their children. In the observational setting (that is, when no nodes are manipulated), this reduces to the usual graphical model specification; that is, the joint distribution of the nodes $\mathcal{V} = (X_1, \ldots, X_V)$ factors as $p(X_1, \ldots, X_V \mid \beta, G) = \prod_{i=1}^{V} p(X_i \mid X_{\mathrm{pa}(i)}, \beta_i, G)$ where G is the graph topology and β_i contains the parameters of the CPD for X_i. However, the algorithmic nature of the causal network also specifies that, when some subset of nodes S is independently manipulated, the joint distribution is $p^*(X_1, \ldots, X_V \mid \beta, G) = \prod_{i=1}^{V} p(X_i \mid X_{\mathrm{pa}(i)}, \beta_i, G)^{\mathbb{1}(i \notin S)} p^*(X_i \mid \beta_i^*)^{\mathbb{1}(i \in S)}$ where $p^*(X_i \mid \beta_i^*)$ is the distribution of X_i when intervening on node i. Here, we define $\beta := (\beta_1, \ldots, \beta_V, \beta_1^*, \ldots, \beta_V^*)$.

For a data set $D = \big((X_{1,n}, \ldots, X_{V,n}) : n = 1, \ldots, N\big)$ consisting of N samples of the V nodes under interventions on subsets $S_1, \ldots, S_N$, respectively, the marginal likelihood is

$$p^*(D|G) = \int p^*(D|\beta, G)\, p(\beta|G)\, d\beta \tag{5}$$

$$= \int \Big(\prod_{n=1}^{N} p^*(X_{1,n}, \ldots, X_{V,n} \mid \beta, G) \Big)\, p(\beta|G)\, d\beta \tag{6}$$

$$= p_O^*(D|G) p_S^*(D) \tag{7}$$

where

$$p_O^*(D|G) = \prod_{i=1}^{V} \int \big(\prod_{n=1}^{N} p(X_{i,n} \mid X_{\mathrm{pa}(i),n},\, \beta_i,\, G)^{\mathbb{1}(i \notin S_n)} \big)\, p(\beta_i|G)\, d\beta_i \tag{8}$$

$$p_S^*(D) = \prod_{i=1}^{V} \int \Big(\prod_{n=1}^{N} p^*(X_{i,n} \mid \beta_i^*)^{\mathbb{1}(i \in S_n)} \Big)\, p(\beta_i^*)\, d\beta_i^*. \tag{9}$$

When $p(\beta_i|G)$ and $p(\beta_i^*)$ are conjugate priors, these integrals can be computed in closed form. Since $p_S^*(D)$ does not provide any information about G, it is often omitted, however, we include it for theoretical purposes.

In this paper, we consider both categorical CPDs with Dirichlet priors and Gaussian CPDs with Gaussian-Wishart priors for which the marginal likelihoods $p^*(D \mid G)$ can be computed in closed form. Specifically, in the categorical case we assume that the CPD of each node is

$$p(X_i = k \mid X_{\mathrm{pa}(i)} = j,\ \beta_i,\ G) = \beta_{ijk} \tag{10}$$

for $i \in \{1, \ldots, V\}$, $j \in \{1, \ldots, q_i\}$, and $k \in \{1, \ldots, r_i\}$. Here, j enumerates the possible joint states of $X_{\mathrm{pa}(i)}$, and we abuse notation slightly by writing $X_{\mathrm{pa}(i)} = j$ to mean that $X_{\mathrm{pa}(i)}$ takes the jth possible state. We use the BDeu Dirichlet prior, $\beta_{ij} \sim \mathrm{Dirichlet}(\boldsymbol{\alpha}_{ij})$ with $\alpha_{ijk} = 1/(r_i q_i)$, following standard practice in the categorical setting (Heckerman et al., 1995; Cooper and Yoo, 1999; Eaton and Murphy, 2007a). Similarly, for the interventions, we assume $p^*(X_i = k \mid \beta_i^*) = \beta_{ik}^*$ and for simplicity, $\beta_i^* \sim \mathrm{Dirichlet}(1/r_i, \ldots, 1/r_i)$.

The BDeu prior has the favorable property of likelihood equivalence, which we use in Section 4.2 (Buntine, 1991; Heckerman et al., 1995). For the Dirichlet-Categorical case with BDeu prior,

$$p_O^*(D \mid G) = \prod_{i=1}^{V} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})} \tag{11}$$

where Γ is the gamma function, $N_{ijk} = \sum_{n=1}^{N} \mathbb{1}(i \notin S_n,\ X_{\mathrm{pa}(i)} = j,\ X_{i,n} = k)$ is the number of samples in which node X_i is observed (not manipulated) to have state k when its parents have state j, $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$, and $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$.

In the Gaussian case, we assume that the CPD for each node is

$$p(x_i \mid x_{\mathrm{pa}(i)},\ \beta_i,\ G) = N\big(x_i \,\big|\, m_i + \textstyle\sum_{j \in \mathrm{pa}(i)} \gamma_{ij} x_j,\ \sigma_i^2\big) \tag{12}$$

where $N(x_i \mid \mu, \sigma^2)$ is the density of a Gaussian with mean μ and variance $\sigma^2 > 0$. The local parameters are $\beta_i = (m_i, \gamma_i, \sigma_i^2)$, where $\gamma_i = (\gamma_{ij} : j \in \mathrm{pa}(i))$. Following standard practice in the Gaussian case, we use the BGe Gaussian-Wishart prior; for details on this prior and marginal likelihood, see Geiger and Heckerman (2013, 1999, 1994).

The method we propose can be used with other CPDs as well, so long as the marginal likelihood can be computed or approximated in a computationally tractable way.

3.2 Intuition for partitions of graph space

In this section, we illustrate examples of partitions that intervention experiments induce over a space of graphs. For expository purposes, suppose the true graph G is known to be one of the four graphs shown in Figure 1; this example is inspired by an example from Pournara and Wernisch (2004). In general, the set of graphs under consideration, $\mathcal{G}$, would consist of all possible DAGs on nodes $\{A, B, C, D\}$ rather than just these four graphs.

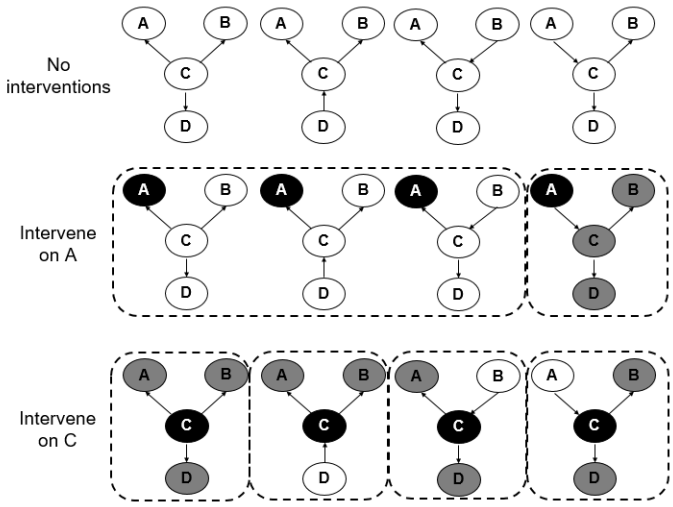

Figure 1: Top panel: Graphs in $\mathcal{G}$. Middle panel: Partition induced by intervening on A. The manipulated node is shown in black while descendants of the manipulated node are shaded grey. Bottom panel: Partition induced by intervening on C.

First, consider what features of a graph a single node intervention on node e could help reveal. For example, we can expect an intervention on A to have observable downstream effects on at least some of the descendants of A, but it would not affect any ancestors of A. Therefore, intervening on A should give us information about which nodes are descendants of A. Thus, one possible choice for $f_e(G)$ is the set of descendants of the manipulated node. Since $f_e(G)$ induces a partition of the set of graphs, we refer to it as a partition scheme.

To select which node to manipulate, we compare the information each candidate intervention is expected to yield with respect to a given partition scheme. Suppose A is manipulated. Figure 1 shows the partition of graphs according to the descendants of A; that is, G and G' are in the same part if $f_e(G) = f_e(G')$. In the first three graphs, A has no descendants, whereas in the last graph, $\{B, C, D\}$ are all descendants of A. Thus, as long as intervening on A has an effect on one or more descendants, we could distinguish whether $G \in \{G_1, G_2, G_3\}$ or $G = G_4$ after sufficiently many replicates of the intervention on A. Meanwhile, if node C is manipulated, a different partition over $\mathcal{G}$ is induced since C has a different pattern of descendant sets than A; see Figure 1.

In general, an intervention partitions the set of graphs into equivalence classes such that (i) the graphs in each class are indistinguishable with respect to this intervention (corresponding to Condition 2.1(a)), and (ii) graphs in different classes are distinguishable (corresponding to Condition 2.1(b)). For the Dirichlet-Categorical model that we use, the equivalence classes induced by the likelihood have an elegant graph-based characterization; see Section 4.2. However, we have also found several other partition schemes to be useful in practice; see Section 4.1.

After an intervention is performed, the generated data provide evidence to suggest which parts of the partition are compatible with the experimental data — specifically, parts that are more compatible with the data will have higher posterior mass. Roughly speaking, we would like to choose an experiment that narrows down the set of compatible graphs as much as possible. For instance, in the toy example in Figure 1, intervening on C is preferable to intervening on A, since C induces a finer partition of $\mathcal{G}$. However, in general it is also important to consider the posterior probability of the graphs given the data from any previous experiments, since there is no point in finely partitioning regions of the space with very low probability. To make this precise, we apply our general entropy-based criterion from Section 2 to the causal network setting, as described next.

3.3 Applying the experiment selection criterion to causal networks

Specializing from the general setting of Section 2 to the case of causal networks, we define the unknown parameter of interest to be $\theta := G$ and let $f_e(G)$ be a partition scheme, where here f_e is a set-valued function. Further, in the Dirichlet-Categorical case, we define $\nu := \beta$, that is, the nuisance parameter ν is the collection of CPD parameters β. Our goal is to perform experiments that make the posterior on graphs concentrate at the true graph as quickly as possible. We quantify concentration using the entropy of the posterior on graphs, $H(G)$, where G is distributed according to the posterior given all experiments so far. Thus, we wish to perform experiments that minimize $H(G)$.

In many cases, inferring the entire graph G is overly ambitious. For instance, if the number of nodes is even moderately large, the number of possible graphs is extremely large, making it infeasible to infer G completely. However, often, one only needs to infer a specific feature of G, such as whether node X_1 is an ancestor of node X_2, or whether X_3 mediates the effect of X_1 on X_2. In such cases, one can instead define θ to be a function of G, say, $\theta := \varphi(G)$ and focus on minimizing $H(\varphi(G))$ rather than $H(G)$.

To prioritize experiments, we apply the general criterion derived in Section 2. Specifically, given samples $G_1, \ldots, G_T$ from the current posterior, we choose the next experiment e to maximize

$$\hat{h}_e = -\sum_y \hat{p}_e(y) \log \hat{p}_e(y) \tag{13}$$

where $\hat{p}_e(y) = \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}(f_e(G_t) = y)$. Thus, we choose the intervention that maximizes the posterior entropy (under the current posterior) of the partition induced by f_e. If Condition 2.1 is satisfied, then this minimizes the approximate expected entropy of the new posterior given the additional data from the experiment. Meanwhile, if Condition 2.1 is not fully satisfied, then this approach is not guaranteed to reduce the entropy optimally, but is still a sensible way of choosing interventions that quickly reduce the entropy.

4 Practical implementation of the method

In this section, we provide practical details on implementing the criterion in Section 3.3, including specifics regarding partition schemes, equivalence classes of graphs, sampling from the posterior on graphs, and an overall algorithm.

4.1 Partition schemes

In the context of graphs, Condition 2.1(a) is that given $f_e(G)$, the graph G is conditionally independent of data from experiment e, and Condition 2.1(b) is that $f_e(G)$ is identifiable with respect to the distribution of the data under experiment e. Condition 2.1(c) is always satisfied since there are finitely many graphs G.

While in theory we use Condition 2.1 to justify the method, in practice Conditions 2.1(a) and 2.1(b) are not strictly necessary. Recall that the optimality theory is based on the expected information gain from asymptotically many replicates of the next selected experiment. Thus, in practice, it may be possible to obtain excellent performance using a partition scheme that violates Condition 2.1(a) or 2.1(b). Consequently, we define a variety of partition schemes here, and we empirically compare their performance in Section 6.

We consider experiments e that intervene on a single node, and for brevity we use e to denote the manipulated node. Consider the following partition schemes.

1. Markov equivalence class (MEC): $f_e(G)$ equals the Markov equivalence class of graphs when intervening on node $e \in \mathcal{V}$; see Section 4.2.
2. Child Set (CS): $f_e(G)$ equals the set of children of node $e \in \mathcal{V}$.
3. Descendant Set (DS): $f_e(G)$ equals the set of descendants of node $e \in \mathcal{V}$.
4. Parent Set (PS): $f_e(G)$ equals the set of parents of node $e \in \mathcal{V}$.

We also consider the following slightly different approach.

5. Pairwise Child (PWC): Maximize $\sum_{v \in \mathcal{V}} H(f_{e,v}(G))$ where $f_{e,v}(G) = \mathbb{1}((e, v) \in G)$ is the indicator of whether G has an edge from e to v.

4.2 Markov equivalence classes

Two DAGs are said to be *Markov equivalent* if they represent the same set of conditional independence relations. While all of the conditional independence relationships entailed by a graph can be computed using the d-separation algorithm (Pearl, 1988), the following elegant result provides a simpler way to determine whether two DAGs are Markov equivalent based on their topology.

Theorem 4.1 (Verma and Pearl (1991))**.** *Two DAGs G_1 and G_2 are Markov equivalent if and only if they have the same skeleton and the same v-stuctures.*

The *skeleton* of a graph is its topology ignoring edge directions. A *v-structure* is a triple of nodes (x, y, z) with topology $x \to y \leftarrow z$, where there is no edge connecting x and z. In general, observational data alone cannot distinguish between Markov equivalent graphs unless one assumes specific error distributions or functional model classes (Peters et al., 2011). A rich literature exists on Markov equivalence; see, for example, Andersson et al. (1997) and Chickering (1996).

While Markov equivalent graphs represent the same conditional independence relationships, they differ in the causal relationships they encode since it is the direction of arrows, not just the skeleton, that is important for causal interpretation. Interventions can help distinguish among graphs in the same Markov equivalence class. However, even after an intervention is performed, some graphs may still be indistinguishable. Hauser and Bühlmann (2012a) consider performing a sequence of interventions, and they provide a generalization of Theorem 4.1 that characterizes the equivalence classes of graphs that are indistinguishable with respect to the whole sequence of interventions.

For our approach, however, we only need to consider the partition induced by a single candidate intervention (rather than the whole sequence of interventions), since the information from previous interventions is already represented in the posterior distribution. Thus, for our approach, a natural choice of partition scheme is to define $f_e(G)$ to be the Markov equivalence class of G^e, where G^e is the DAG obtained from G by removing all edges from $\text{pa}(e)$ to e. This is referred to as the "MEC" scheme in Section 4.1.

Following the notation of Sections 2 and 3, we write $P_{G,\beta}$ for the distribution of $(X_1, \ldots, X_V)$ given graph G and CPD parameters $\beta = (\beta_1, \ldots, \beta_V, \beta_1^*, \ldots, \beta_V^*)$. Define β^e to be a modified copy of β in which β_e^* takes the place of β_e and β_{e1} takes the place of β_e^*. Thus, when intervening on node e, the distribution can be written as P_{G^e,β^e}.

Consider a sequence of interventions in which a single node is manipulated at a time. For instance, suppose we have performed N_k replicates intervening on node i_k for $k = 1, \ldots, K$, and we are considering intervening on node i' for the next set of N' replicates. The joint model is then

$$\begin{aligned} &(G, \beta) \sim \pi \\ &X_1^e, \ldots, X_N^e \mid G, \beta \text{ i.i.d.} \sim P_{G^e,\beta^e} \text{ for } (e, N) \in \{(i_1, N_1), \ldots, (i_K, N_K), (i', N')\} \end{aligned} \tag{14}$$

where $P_{G,\beta}$ is the categorical model, $\pi(\beta|G)$ is the BDeu-based prior defined in Section 3.1, and $\pi(G)$ is an arbitrary prior on DAGs.

Theorem 4.2. *Under the joint model in Equation 14, if $f_e(G)$ is the Markov equivalence class of G^e then*

$$X_{1:N'}^{i'} \perp\!\!\!\perp G \mid f_{i'}(G), f_{i_1}(G), \ldots, f_{i_K}(G).$$

Theorem 4.3. *Assume the joint model in Equation 14, and let $D = (X_{1:N_1}^{i_1}, \ldots, X_{1:N_K}^{i_K})$ denote the data observed so far. If $f_e(G)$ is the Markov equivalence class of G^e then there is a function g such that $g(P_{G^{i'},\beta^{i'}}) = f_{i'}(G)$ almost surely when $(G, \beta) \sim p(G, \beta \mid D)$.*

Now, to employ Theorem 2.2, observe that under the model in Equation 14, if we condition on D then we obtain the following model:

$$(G, \beta) \sim p(G, \beta \mid D)$$

$$X_1^{i'}, \ldots, X_N^{i'} | G, \beta \text{ i.i.d. } \sim P_{G^{i'},\beta^{i'}}.$$

This follows the form of the abstract model in Section 2, with appropriate notational substitutions. As above, let $f_e(G)$ be the Markov equivalence class of G^e. Condition 2.1(a) is that $X_{1:N'}^{i'} \perp\!\!\!\perp G \mid f_{i'}(G)$ in this conditional model given D, or equivalently, $X_{1:N'}^{i'} \perp\!\!\!\perp G \mid f_{i'}(G), D$ under the joint model in Equation 14. By Theorem 4.2, $X_{1:N'}^{i'} \perp\!\!\!\perp G \mid f_{i'}(G), f_{i_1}(G), \ldots, f_{i_K}(G)$, so we can expect that $X_{1:N'}^{i'} \perp\!\!\!\perp G \mid f_{i'}(G), D$ holds approximately when $N_1, \ldots, N_K$ are sufficiently large, since $D = (X_{1:N_1}^{i_1}, \ldots, X_{1:N_K}^{i_K})$ and $X_{1:N_k}^{i_k}$ pertains to $f_{i_k}(G)$. Condition 2.1(b) is that there exists g such that $g(P_{G^{i'},\beta^{i'}}) = f_{i'}(G)$ almost surely when $(G, \beta) \sim p(G, \beta \mid D)$, which is precisely what Theorem 4.3 shows. Finally, Condition 2.1(c) is that $f_{i'}(G)$ takes finitely many values, which is true since there are only finitely many graphs G on V nodes.

Therefore, Theorem 2.2 indicates that selecting the next intervention using the strategy in Section 2.3 with $f_e(G)$ chosen to be the Markov equivalence class of G^e is a natural choice to optimally reduce entropy, under the asymptotic approximation that the number of replicates in each experiment is sufficiently large.

4.3 Sampling from the posterior distribution on graphs

A large body of work exists on MCMC methods for sampling from $p(G|D)$. This is a challenging task since the number of DAGs increases super-exponentially with the number of nodes and the posterior on graphs is often highly multi-modal. Some have proposed searching the space of graphs using local proposals that add, delete, or reverse edges at random (Madigan et al., 1995) and others have improved chain mixing by sampling over the space of node orderings (Friedman and Koller, 2003; Ellis and Wong, 2006). We use a clever MCMC algorithm developed by Eaton and Murphy (2007a) that uses dynamic programming (DP) to construct proposals. This method explores the space of DAGs using a Metropolis-Hastings algorithm with a proposal distribution that is a mixture of local moves (edge deletions, additions, or reversals) and a global move that proposes a new graph in which an edge exists between two nodes with probability equal to the exact marginal posterior edge probability, computed using DP.

Key to the DP algorithm's ability to compute exact marginal posterior edge probabilities is the assumption of a "modular prior" over structures. Rather than directly specifying a prior over DAGs, a modular prior requires specifying a prior over node orderings and a prior that gives weight to sets of parents (and not to their relative order). Together these terms define a joint prior over graphs and orders. Defining the prior in this way allows the contribution to the marginal likelihood for nodes with the same parent sets to be cached and re-used for efficient exact computation, regardless of the orderings of the parents; see Koivisto (2006) for details of the DP algorithm and see Koivisto and Sood (2004), Friedman and Koller (2003), and Ellis and Wong (2006) for further discussion of priors on orderings and graphs.

A modular prior tends to favor graphs that are consistent with more orderings, such as fully disconnected graphs and tree structures. In fact, the modular prior favors tree structures over chains even if the two structures are Markov equivalent. For instance,

tree structure $1 \leftarrow 2 \rightarrow 3$ has higher prior probability than the chain $1 \rightarrow 2 \rightarrow 3$ under a modular prior since the tree structure is consistent with two node orderings (Eaton and Murphy, 2007a). While one may want to use a uniform prior over DAGs in the absence of prior knowledge, Ellis and Wong (2006) and Eaton and Murphy (2007a) show how a uniform prior over orderings and flat prior over parent sets together encode a highly nonuniform prior over DAGs. The hybrid MCMC-DP approach that we use (Eaton and Murphy, 2007a) overcomes this limitation of the DP algorithm. With MCMC-DP, we can use an arbitrary prior on graphs and draw valid samples from $p(G|D)$, while benefiting from a fast, data-driven proposal distribution to help traverse the DAG space. We implemented our method in MATLAB (version 2017a) and we use the BDAGL package (Eaton and Murphy, 2007a) to sample from $p(G|D)$ using the MCMC-DP algorithm. Source code is available online at https://github.com/mzemplenyi/OED-graphical-models.

4.4 Overall algorithm

The inputs to our proposed algorithm are (i) a set of candidate experiments $\mathcal{E}$, (ii) a mechanism for generating i.i.d. samples $(X_1, \ldots, X_V)$ from $P_{e,0}$, the true distribution under experiment $e \in \mathcal{E}$, and (iii) a partition scheme $f_e(G)$ for each experiment $e \in \mathcal{E}$. For the first experiment, we generate observational data by not intervening on any nodes. Each subsequent experiment sets a single node from $\mathcal{E}$ to a fixed value. The algorithm proceeds as follows. Let D denote the collection of data from the experiments so far.

1. Obtain posterior samples from $\pi(G) = p(G|D)$ and approximate the posterior entropy, $H(G)$.
2. Check the stop criteria. Stop the algorithm if either:
   (a) $H(G)$ falls below a given entropy tolerance threshold, or
   (b) the maximum number of allowed experiments has been reached.

   Otherwise, continue.
3. For each $e \in \mathcal{E}$, enumerate the partition *over the sampled graphs* induced by e and calculate $\hat{h}_e$, the approximate posterior entropy over the partition (Equation 13).
4. Select the experiment e that maximizes $\hat{h}_e$ as the next intervention experiment to perform.
5. Generate data for experiment e: draw N i.i.d. samples from $P_{e,0}$.
6. Combine the new data with the existing data and repeat from the beginning.

For computational efficiency, note that in step 3, it is only necessary to consider those parts A in the partition $\mathcal{A}_e$ that contain one or more posterior samples. Since it is not necessary to consider all parts in the partition, this can provide a computational advantage in cases where there are an intractable number of parts.

5 Previous work

Previously proposed methods for learning causal network models from observational and interventional data in the non-OED setting typically fall into one of two categories: (1) constraint-based methods, such as the PC-algorithm (Spirtes et al., 2001), that test for conditional independence constraints in the data and select models that match those constraints, and (2) score-based methods, wherein the space of structures is searched for ones that that are most supported by the data, as quantified by scores such as the Bayesian marginal likelihood or BIC. More recently, OED methods, also referred to as active learning methods, have been developed for both approaches.

He and Geng (2008), Eberhardt (2008), and Hauser and Bühlmann (2012b) propose constraint-based active learning methods that first use observational data or prior knowledge to construct a partially directed acyclic graph (PDAG), also called an essential chain graph. They then use graph-theoretic results to select the interventions required to orient all edges in the essential chain graph, using variations on criteria that generally seek to minimize the number of undirected edges in the post-intervention equivalence class of graphs. These algorithms take as a starting point a known observational essential graph, which would require infinite observational data in principle, and in practice is estimated from finite observational data. However, they often do not perform as well in finite sample settings where estimation errors introduced in the initial chain graph can lead to an incorrectly estimated DAG. Hauser and Bühlmann (2012a) demonstrate that, in the finite sample setting, estimation errors can be reduced by using interventional data to refine not only the directionality of uncompelled edges in the chain graph (as done by He and Geng), but also the skeleton of the chain graph.

Tong and Koller (2001) and Murphy (2001) develop score-based active learning methods for Bayesian networks. They use MCMC to sample from the space of node orderings (Tong and Koller) or graphs (Murphy) along with a decision-theoretic framework to select interventions. Both methods are computationally expensive since they involve computing the predictive density for each sampled graph subject to each possible intervention. Other methods forgo the predictive sampling step and instead use selection criteria with lower computational burdens. Pournara and Wernisch (2004) consider the equivalence classes of high-scoring networks (determined by a greedy hill-climbing search) and select interventions that tend to partition transition sequence equivalence classes into smaller and smaller subclasses. Li and Leong (2009) propose a 'non-symmetrical entropy' criterion closely related to Tong and Koller's loss function, but use the DP algorithm rather than MCMC to calculate edge probabilities between nodes. Cho et al. (2016) adapt the active learning framework of Murphy (2001) to the Gaussian Bayesian network setting. Ness et al. (2018) use a Bayesian framework that allows one to directly encode prior causal knowledge about each edge, which then induces a prior over graphs. Their method, bninfo, takes a set of highly scoring PDAGs and returns the minimally sized batch of interventions that is expected to correctly orient the greatest number of edges; this method shares elements of both the score-based and constraint-based methods.

The previous work that is most similar to ours is the method of Almudevar and Salzman (2005), who explore the performance of an entropy-based criterion that uses

the idea of partitioning the space of graphs to minimize the entropy on the posterior on graphs. While our method is based on a similar theoretical justification as that of Almudevar and Salzman (2005), we generalize the approach to a large class of partition schemes and we improve performance by using a more efficient MCMC procedure. Further, in contrast to the limited simulation study of Almudevar and Salzman (2005), we provide a more extensive set of empirical results on a wide variety of networks, we compare with other leading algorithms, and we make our software publicly available.

For a comprehensive review of optimal experimental design methods for other models, including Boolean networks and differential equation models, see Sverchkov and Craven (2017).

6 Simulation results

In this section we first evaluate the performance of our OED method under various partition schemes. Then we compare our method to other causal structure learning methods in both discrete and Gaussian settings. Throughout, we evaluate performance using the following metrics:

1. Mean Hamming distance: After each experiment, we calculate the posterior probability of an edge between nodes X_i and X_j via:

$$p(i \to j \mid D) = \sum_{G \in \mathcal{S}_{i \to j}} p(G|D) \tag{15}$$

   where $\mathcal{S}_{i \to j}$ is the set of graphs containing the edge $i \to j$. We then construct the median probability graph, defined as the graph containing only those edges for which $p(i \to j \mid D) \geq 0.5$ (Castelletti et al., 2018; Peterson et al., 2015). The Hamming distance between the median probability graph and the ground truth network is equal to the number of false detected edges (false positives) and missing edges (false negatives) in the median probability graph. We then take the average of this distance over all simulations.

2. Mean true positive rate (TPR): After each experiment, we calculate the proportion of correctly detected edges present in the median probability graph among the edges in the ground truth network. We then find the average of this proportion over all simulations.

3. Mean true negative rate (TNR): After each experiment, we calculate the proportion of correctly detected non-edges present in the median probability graph among the non-edges in the ground truth network. We then find the average of this proportion over all simulations.

4. Mean false discovery rate (FDR): After each experiment, we calculate the proportion of false positive edges out of the sum of the false positive and true positive edges present in the median probability graph. We then find the average of this proportion over all simulations.

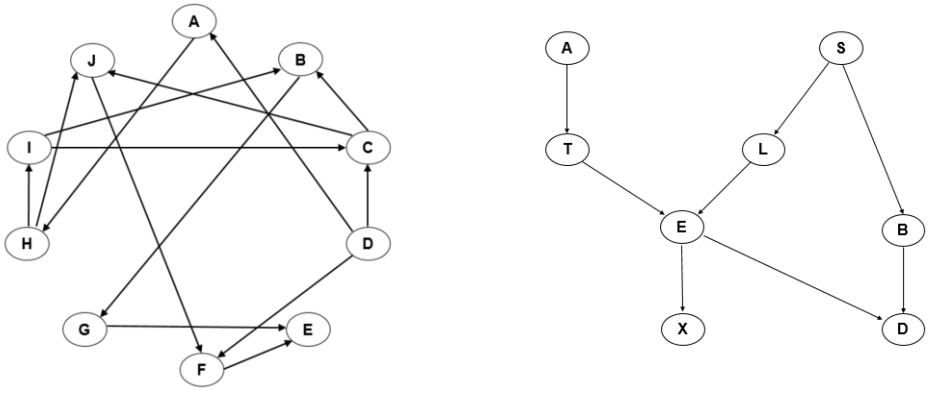

Figure 2: Left: Structure of ten-node network. Right: Structure of Asia network as defined by the Bayesian Network Repository available in the *bnlearn* R package (Scutari, 2010).

For all figures, error bars represent the standard error of the mean over 50 simulations.

6.1 Comparison of partition schemes

Our first simulation study compared the five different ways, defined in Section 4.1, to partition graphs sampled from the posterior $p(G|D)$. We randomly generated a discrete graph with 10 binary nodes and generated observational and intervention samples from this ground truth network; see Figure 2 for the graph's structure.

For each partition scheme, we ran $n_{sims} = 50$ simulations wherein each simulation consisted of a series of $n_{exp} = 7$ experiments. For the first experiment, we generated $n_{obs} = 1000$ observational samples to form the initial dataset D. For each of the six subsequent experiments, we generated $n_{intv} = 1000$ additional intervention samples and appended them to D, where the manipulated node was selected via our entropy-based selection criterion based on the partition scheme under consideration. After generating data for each experiment, we used MCMC-DP to draw 250,000 posterior samples from $p(G|D)$, discarded the first 150,000 samples as burn-in, and used the remaining 100,000 samples for posterior inference. We used a uniform prior over graphs and did not allow a node to be manipulated more than once. In addition to the five partition schemes, we also evaluated a "random learner" that randomly selected the next node to be manipulated, rather than using an entropy criterion.

Figure 3 shows the mean Hamming distance and mean TPR for the five entropy-based methods and the random learner on the 10-node network. FDR and TNR results can be found in Appendix E. Each of the entropy-based methods performed better than the random learner according to all metrics. For this network, the entropy-based methods all performed similarly. We found this to be the case in simulations using other randomly generated networks as well.

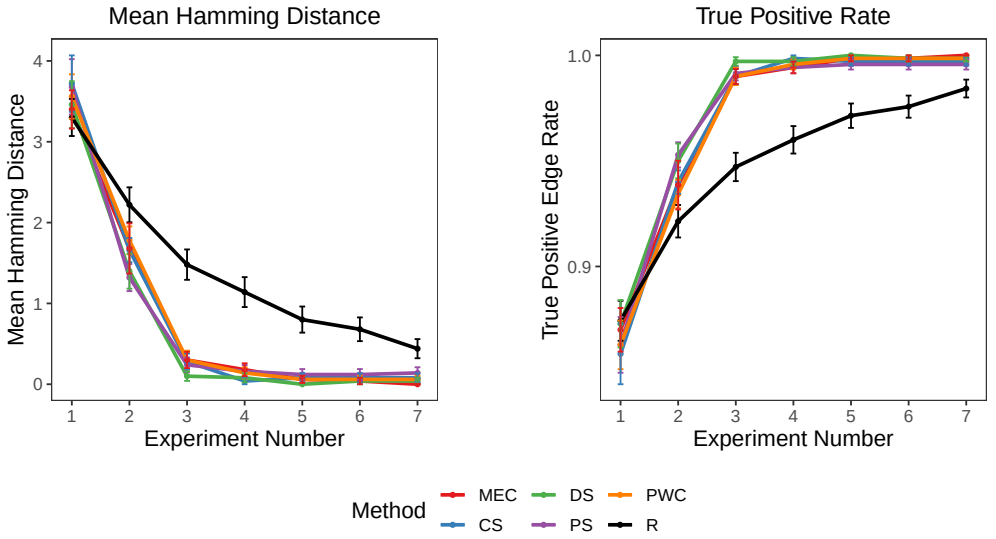

Figure 3: Mean Hamming distance and mean TPR for five entropy-based OED methods (each using a different partition scheme) and the random learner on a ten-node binary network. Settings: $n_{sim} = 50$, $n_{exp} = 7$, $n_{obs} = 1000$, $n_{intv} = 1000$. MEC = Markov equivalence class; CS = child set; DS = descendant set; PS = parent set; PWC = pairwise child; R = random learner. See Appendix E for accompanying graphs of false discovery rate and true negative rate.

The benefit of an entropy-driven experimental design method over randomly selected interventions is evident from Figure 3. To fall within a given mean Hamming distance from the true graph required far fewer interventions using the OED methods compared to the random learner.

Sensitivity analysis varying the intervention sample size

Since the justification of our proposed criterion is based on the asymptotics as n_{intv} goes to infinity, we ran a sensitivity analysis varying n_{intv} to assess performance in small sample size settings. Using the ten-node binary network, the MEC partition scheme, and the random learner as a baseline, we ran simulations for $n_{intv} \in \{20, 50, 200, 1000\}$ (Figure 4). As expected, the network is learned less accurately with fewer intervention samples, but even when using the fewest intervention samples ($n_{intv} = 20$), our approach provides significant improvement over the random baseline. Thus, while the optimality theory underlying the method is based on the expected information gain from asymptotically many replicates of the next selected experiment, this sensitivity analysis suggests that our approach performs well even when a practically reasonable number of replicates of the next experiment are actually done.

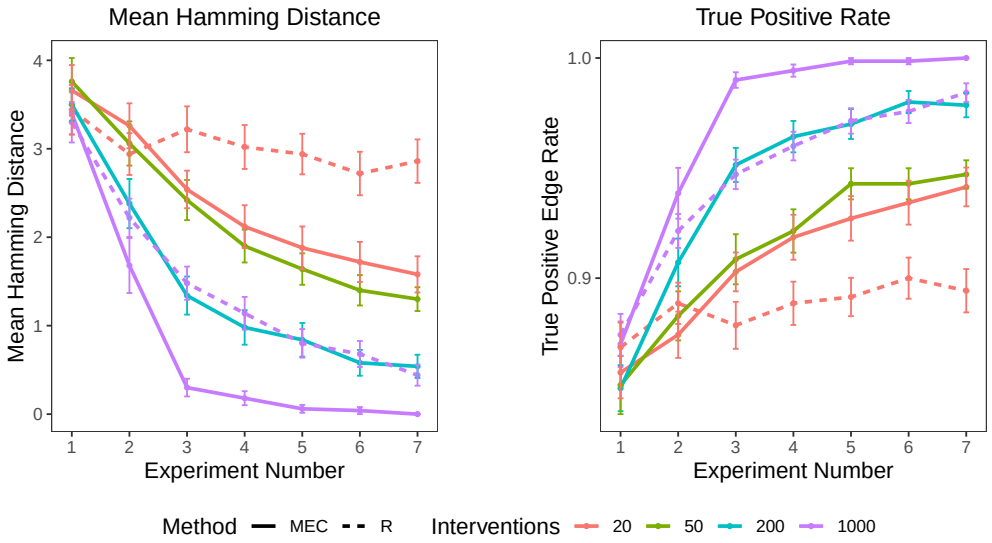

Figure 4: Sensitivity analysis varying the number of intervention samples (n_{intv}) on the ten-node binary network. Mean Hamming distance and mean TPR are displayed for the Markov equivalence class (MEC) partition method. The random learner (R) is included for comparison at the highest and lowest n_{intv} levels. Settings: $n_{sim} = 50$, $n_{exp} = 7$, $n_{obs} = 1000$, $n_{intv} \in \{20, 50, 200, 1000\}$. Curves for the random learner at $n_{intv} \in \{50, 200\}$ are omitted here for the sake of visual clarity, but can be found in Appendix C.

6.2 Comparison with other methods

We compared our OED algorithm to two other OED methods, as well as the state-of-the-art greedy equivalence search (GES) (Chickering, 2002) and greedy interventional equivalence search (GIES) (Hauser and Bühlmann, 2012a) structure learning methods.

The first OED method is proposed by Li and Leong (2009) and also uses an entropy-based criterion. In fact, what Li and Leong refer to as their non-symmetrical entropy criterion for selecting interventions is equivalent to the pairwise child (PWC) entropy criterion described in Section 4.1. The difference between the methods, however, is that Li and Leong (2009) do not use MCMC to sample from the posterior on graphs. Rather, they evaluate their criterion using exact edge probabilities computed using a DP algorithm (Koivisto, 2006); see Section 4.3 for details on the DP algorithm and its assumptions. We refer to the method of Li and Leong as "DP" in subsequent figures.

The second OED method, "bninfo" (Ness et al., 2018), evaluates the expected causal information gain of candidate interventions and outputs a minimally sized batch of interventions expected to maximize that gain. Ness et al. (2018) define causal information gain as the increase in correctly oriented edges in the causal network. Their algorithm constructs the recommended batch of interventions one node at a time, in descending order of expected causal information gain. In order to compare our method with bninfo, for a given causal network, we took the sequence of interventions that bninfo recommended and used MCMC-DP to sample from the posterior distribution on graphs between each recommended intervention. This allowed us to construct the median probability graph and calculate the Hamming distance and TPR after each intervention that bninfo recommended. Note that Ness et al. (2018) provide a way to encode prior knowledge on each edge in the graph, but to facilitate comparison with the other methods, we use a uniform prior on the graph topology. Ness et al. (2018) implemented bninfo for discrete settings so we do not include bninfo in our analysis of networks with continuous data.

While the GES and GIES algorithms are not OED methods, they can serve as helpful comparisons for structure learning. GES is a score-based algorithm that estimates the Markov equivalence class of a DAG using observational data, while GIES estimates the inverventional Markov equivalence class of a DAG using both observational and interventional data. They are both implemented in the 'pcalg' R package (Kalisch et al., 2012) for Gaussian settings so we do not include them for comparison in our analysis of discrete data networks.

Asia network

We first assessed performance of the various methods on the Asia network, a commonly used network with eight binary nodes first described by Lauritzen and Spiegelhalter (1988) (Figure 2). We used the conditional probability table provided by the *bnlearn* R package to generate observational and interventional data (Scutari, 2010).

Figure 5 compares the performance of our method (using the MEC partition scheme), DP, bninfo, and the random learner. For this simulation study we used the following

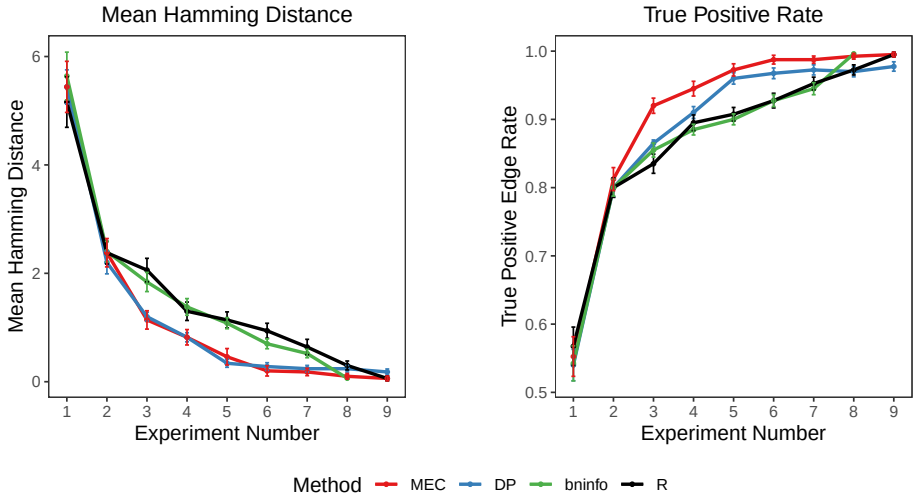

Figure 5: Mean Hamming distance and mean TPR on the 8-node Asia network with the following settings: $n_{sim} = 50$, $n_{exp} = 9$, $n_{obs} = 300$, $n_{intv} = 300$. MEC = our method with MEC partition scheme; DP = dynamic programming method of Li and Leong (2009); bninfo = method of Ness et al. (2018); R = random learner. See Appendix E for accompanying graphs of false discovery rate and true negative rate.

settings: $n_{sim} = 50$, $n_{exp} = 9$, $n_{obs} = 300$, $n_{intv} = 300$. For the MCMC methods, we drew 250,000 samples, discarded the first 150,000 samples, and used the remaining 100,000 samples for inference. For the sake of clarity, we omitted the other partition schemes since they performed similarly to the MEC partition scheme.

After the first intervention experiment, the four methods performed nearly identically. The methods then diverged at experiment 3, with the random learner and bninfo lagging behind MEC and DP. Note that the maximum batch size of interventions recommended by bninfo across the simulations consisted of seven nodes, so the bninfo results in Figure 5 only extend to the eighth experiment (the observational experiment followed by seven interventions). For the other methods, by the ninth experiment, all 8 nodes had been manipulated since we did not allow for repeat interventions. Thus, it makes sense that the MEC and random learner results align by the last experiment; they have each sampled the same data, albeit in different orders. Even though the DP method had also sampled intervention data for all nodes by the ninth experiment, it does not converge with the other methods because the DP method uses a different prior over graphs (a non-uniform prior induced by its modular joint prior over node ordering and parent sets as described in Section 4.3).

Gaussian setting

Figure 6 shows MEC, DP, GES, GIES, and random learner performance on a randomly generated 11-node Gaussian causal network. The GES algorithm of Chickering (2002) performs structure learning solely using observational data, so its results are limited to the first experiment. The GIES algorithm proposed by Hauser and Bühlmann (2012a) extends GES to perform structure learning from a mix of observational and interventional data, but does not recommend the optimal next intervention to perform as OED methods do. Thus, to compare the structure learning abilities of GIES vs MEC over the course of a sequence of experiments, we first ran MEC to obtain a recommended sequence of interventions, and then provided the same set of interventional data to GIES at each step.

Across the six experiments, MEC, DP, and the random learner all had a lower Hamming distance than GIES, including at the initial experiment when only observational data was used to infer the network structure. GIES's higher Hamming distance was driven primarily by a high false discovery rate; whereas in terms of TPR, GIES outperformed MEC, DP, and the random learner. On balance, MEC (our method) appears to have the best performance overall, when considering the three metrics as a whole.

Effect of network topology on inference

Next, we explored the effect of network topology on performance of MEC, PWC, DP, bninfo, and the random learner. Here, as in Section 6.1, "PWC" refers to using our OED method with the PWC partition scheme. We include PWC for direct comparison with the DP method in order to illustrate how two methods that employ the same entropy criterion for selecting interventions—but differ in how posterior edge probabilities are calculated (see Section 4.3)—may perform differently depending on network topology. We considered two 8-node networks, one with a chain structure and one with a tree structure (Figure 7). For the chain network, the DP method performed worse than all methods, including PWC and the random learner, whereas for the tree network, all OED methods performed equally well. The poor DP performance for the chain network is a result of the non-uniform prior over graphs that the DP algorithm uses to achieve its computational efficiency, as described in Section 4.3. The full results for the chain and tree networks can be found in Appendix D.

7 Application to gene expression data

OED methods for graphical models are often developed with the goal of inferring biological networks, such as gene regulatory networks or cell-signaling networks. Especially in light of recent advances in the precision of gene-editing technologies, the ability to iterate between experimentation and analysis by adaptively selecting experiments is a promising avenue for reconstructing biological networks. We applied our method to two datasets generated from biological networks: the Sachs dataset (Sachs et al., 2005) and a subset of the Perturb-seq dataset from Dixit et al. (2016). We provide the Perturb-seq results here, and we have placed the Sachs network results in Appendix F. The Sachs

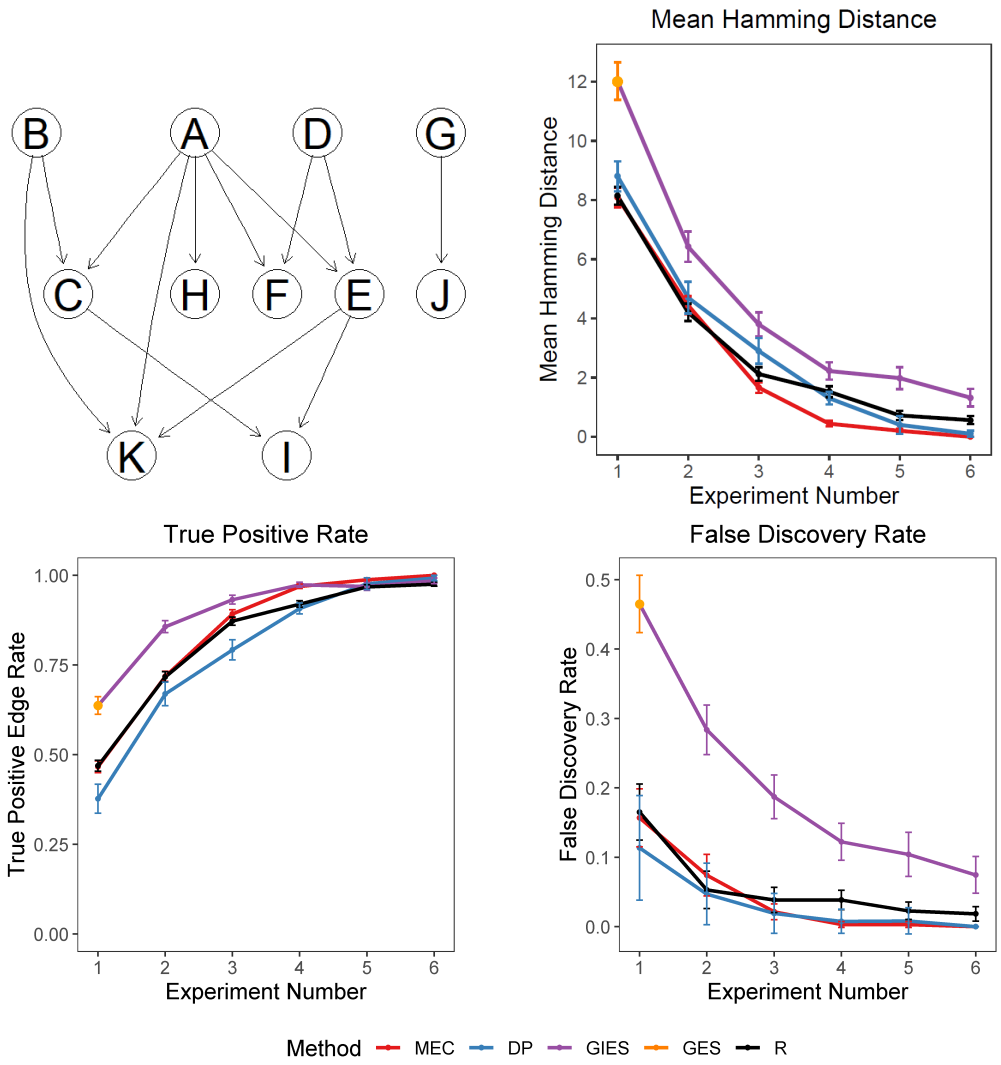

Figure 6: Comparison of the MEC, DP, GIES, GES, and random learner methods on a randomly generated 11-node Gaussian causal network. Settings: n_{sim} = 50, n_{exp} = 6, n_{obs} = 50, n_{intv} = 50. MEC = our method with MEC partition scheme; DP = dynamic programming method; GIES = Greedy Interventional Equivalence Search; GES = Greedy Equivalence Search; R = random learner.

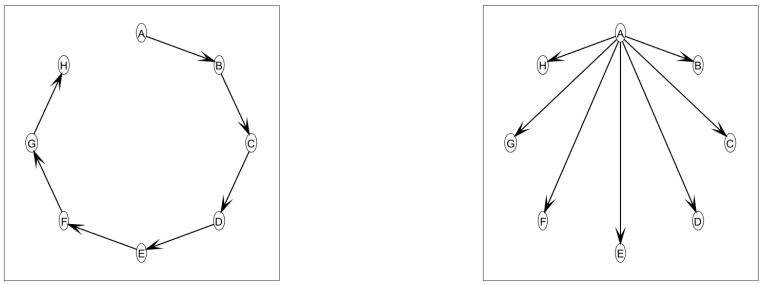

Figure 7: Topology for the 8-node chain structure (left) and 8-node tree structure (right). The results for the chain and tree networks can be found in Appendix D.

network presents particular challenges—likely due to cycles existing in the true network that violate the causal Markov assumption—leading to difficulties for many structure learning methods including our own (Cho et al., 2016; Ness et al., 2018; Sverchkov and Craven, 2017; Wang et al., 2017).

Here, we present the results of applying our method, as well as the DP, GES, and GIES methods, to the Perturb-seq genetic screening data. We focused our analysis on a network composed of 14 transcription factors highlighted in Figure 4B of Dixit et al. (2016) that are involved in regulating various cell programs including antiviral response, T-cell activation, and mitochondrial function. Wang et al. (2017) also studied this set of transcription factors in their work introducing the interventional greedy sparsest permutation (IGSP) structure learning algorithm. To facilitate comparison with the results of the IGSP algorithm, we followed the example of Wang et al. (2017) and restricted our analysis to 992 observational samples and 13,435 interventional samples collected under 8 gene interventions for the 14 measured transcription factors.

While in some well-studied cases there is a consensus network that can serve as a benchmark, there is not yet an established benchmark network for these 14 transcription factors. Thus, since we cannot calculate performance metrics (such as Hamming distance) relative to a consensus network, we instead benchmark each method against itself. More precisely, for each method, we define the benchmark graph to be the median probability graph estimated by that method when run on all of the observational and interventional data. Performance metrics are computed relative to these method-specific benchmarks. This approach allows us to visualize the relative rates of information accumulation for each method, even in the absence of a common consensus network.

Figure 8 shows that the MEC and DP methods perform comparably on all diagnostics and both significantly outperform the random learner in terms of Hamming distance, TPR, and FDR. As in Figure 6, we ran the GIES algorithm using the sequence of interventions recommended by MEC. This sequence of interventions, particularly the first intervention, results in a rapid improvement in GIES's performance relative to its benchmark. We exclude GES from the comparison since, for this setting in which each

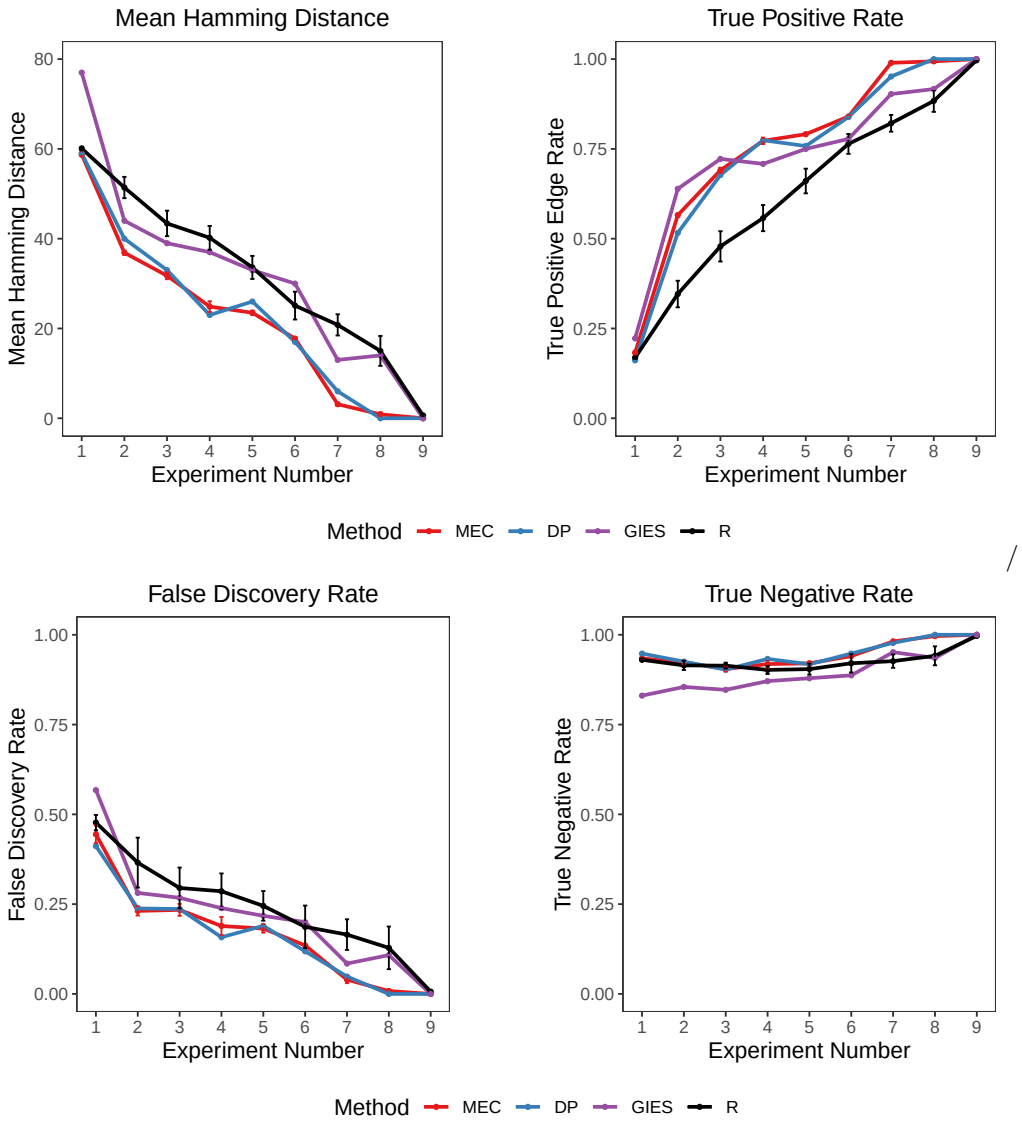

Figure 8: Diagnostics comparing MEC, DP, and GIES performance on the 14-gene Perturb-seq dataset relative to the random learner. The median probability graph attained by each method after all observational and interventional data has been used serves as the benchmark network for each method.

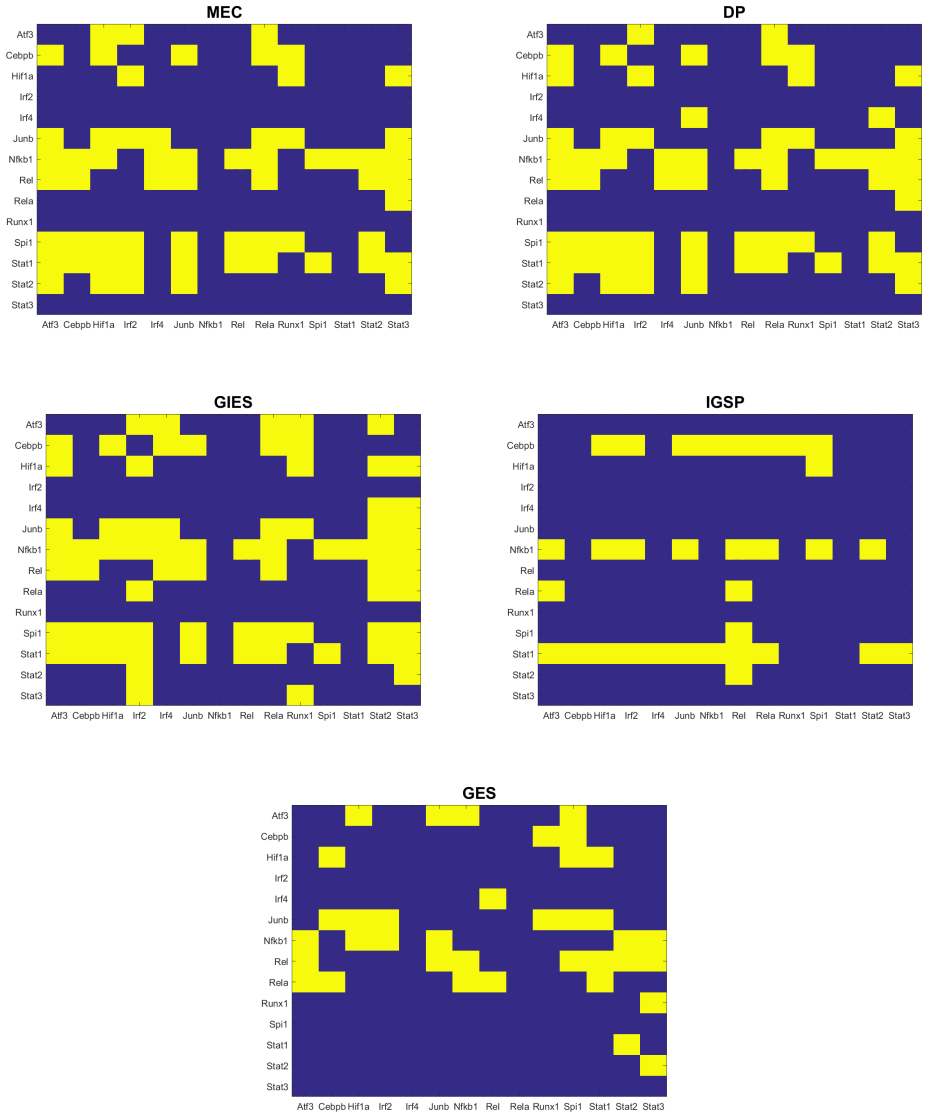

Figure 9: Adjacency matrices estimated by the MEC, DP, GIES, IGSP, and GES methods on the 14-gene Perturb-seq dataset. Yellow entries indicate edges. Rows are parent nodes and columns are child nodes; for example, the MEC-estimated adjacency matrix contains edges from parent node Atf3 to child nodes Hif1a, Irf2, and Rela.

method serves as its own benchmark, the GES algorithm would show no improvement. For example, GES would arrive at its own benchmark and have a Hamming distance of 0 after using all observational data for structure learning from the first experiment.

Figure 9 compares the adjacency matrices estimated by the MEC, DP, GIES, and IGSP methods using all observational and interventional samples, as well as the GES adjacency matrix estimated from all observational data. Visually, MEC and DP are most similar in terms of their estimated network structures. GIES also exhibits strong similarity to MEC and DP, particularly with respect to the parent-child relationships of transcription factors Junb, Nfkb1, Rel, Spi1, and Stat1. Comparing the GIES and GES adjacency matrices illustrates how augmenting observational data with interventional data can dramatically change the estimated network.

The network estimated by IGSP (adapted here from the results shown in Wang et al., 2017) is much more sparse than the networks estimated by MEC, DP, and GIES. IGSP suggests a network primarily composed of three disconnected tree structures with Cebpb, Nfkb1, and Stat1 as parent nodes each regulating many child nodes.

8 Conclusion

We presented a novel Bayesian OED methodology for optimizing the experiment selection process in a computationally tractable way. The core of the method is a criterion for selecting the experiment that is expected to yield the greatest reduction in posterior entropy. We found that the method efficiently infers causal relationships in networks with various topologies, with the greatest gains in information coming from the first few optimally chosen interventions. We provided a theoretical justification for using Markov equivalence classes as the choice of partition in our method, and in simulations, we found that this entropy criterion generally performs well empirically.

Currently, our method is limited to networks with less than 20 nodes due to the computational limits of the DP-based MCMC proposals. Scaling up posterior inference methods to work on larger networks is an area for future work.

The difficulty that many OED and active learning methods, including our own, have in inferring the Sachs network (see Appendix F) suggests that additional research is needed on ways of relaxing the acyclicity assumption and being more robust to model misspecification in general. Additionally, the OED and active learning fields would benefit from additional data sets similar in nature to the Perturb-seq and Sachs datasets with a mix of observational and intervention data. These will be helpful for evaluating and comparing OED methods. As recent advances in gene-editing technologies make targeted interventions more feasible, we expect these types of data sets will become more widely available, and the demand for OED methods in the biological sciences will grow in tandem.

Supplementary Material

Appendix A: Intuition for General Criterion. Provides intuition for the general criterion in the context of the game "Twenty Questions."

Appendix B: Theory. Proofs of Theorem 4.2 and Theorem 4.3.

Appendix C: Sensitivity analysis varying the intervention sample size. Full results for the sensitivity analysis on the ten-node binary network.

Appendix D: Effect of network topology on inference. Full results for the chain and tree networks.

Appendix E: Additional diagnostics. Figures showing the false discovery rate and true negative rate for analyzed networks.

Appendix F: Application to the Sachs network. Includes an application of our OED method to the Sachs network, a discussion of violation of model assumptions, and a simulation study using the Sachs network.

Appendix A: Intuition for General Criterion

In this section, we provide an intuitive illustration of the basic idea of our method in terms of the popular game Twenty Questions. In this game one person, the "answerer," thinks of an object. The other player, the "questioner," then asks a sequence of "yes" or "no" questions with the goal of guessing the answerer's object using fewer than twenty questions.

At the beginning of the game, the questioner has a prior over objects, representing the probability that the answerer has selected a given object. A question such as, "Is the object living?" partitions the objects into two parts: living and non-living. The subsequent answer provides information that allows the questioner to eliminate the objects in one of the parts and update their posterior beliefs accordingly. If the prior is uniform, then the most efficient strategy is to select questions that partition the set of remaining objects roughly in half (Figure 10). More generally, if the prior is not uniform, then it is most efficient to split the posterior probability roughly in half at each step.

The following observations about Twenty Questions are helpful to consider when discussing the causal network setting in Section 3. First, there are many possible ways of partitioning a space into two parts of equal size (or equal posterior probability, more generally). Different questions partition the space along different features of the objects. Second, we might relax the restriction to yes/no questions and also allow questions such as, "Is the object a vegetable, animal, or mineral?" Such questions partition the space of objects into more than two parts. In choosing what question to ask next, the questioner is implicitly considering the informativeness of the partition induced by their question.

In the context of this Twenty Questions example, the setup in Section 2.1 has the following intepretation: θ represents the object selected by the answerer, π is the prior

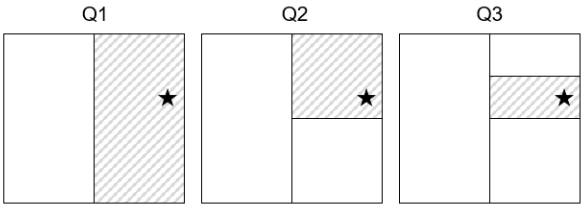

Figure 10: Schematic illustration of sequential partitioning by a series of questions, Q1-Q3. The answer to each question indicates the part (shaded region) containing the object of interest (denoted by a star.)

distribution, $f(\theta)$ is the true answer to question f, and $X_1, \ldots, X_N$ represent noisy answers to question f. Note that $X_1, \ldots, X_N$ all pertain to the same question about θ, not N different questions. Condition 2.1(a) means that once we know the true answer to the question, the noisy answers provide no additional information about θ. Condition 2.1(b) means that the answer $f(\theta)$ is uniquely determined by the distribution of X_n, and thus, in the limit as $N \to \infty$, $f(\theta)$ can be recovered from $X_1, \ldots, X_N$.

Appendix B: Theory

Lemma B.1. *Suppose* $(\theta, \nu) \sim \pi$, $X_1, \ldots, X_N | \theta, \nu \sim P_{\theta,\nu}$ *i.i.d., and* $f(\theta)$ *satisfies Conditions 2.1(b) and 2.1(c). Then* $H(f(\theta) \mid X_{1:N}) \to 0$ *as* $N \to \infty$.

Proof. Since $g(P_{\theta,\nu}) = f(\theta)$ a.s. under the prior, then the same also holds a.s. under the posterior. Thus, for any value y in the range of f,

$$p(f(\theta) = y \mid X_{1:N}) = p(g(P_{\theta,\nu}) = y \mid X_{1:N}) = \mathrm{E}(\mathbb{1}(g(P_{\theta,\nu}) = y) \mid X_{1:N})$$
$$\xrightarrow[N\to\infty]{\text{a.s.}} \mathbb{1}(g(P_{\theta,\nu}) = y) \overset{\text{a.s.}}{=} \mathbb{1}(f(\theta) = y)$$

where $\mathbb{1}(\cdot)$ is the indicator function. Here, the limiting value is a random variable in which $(\theta, \nu) \sim \pi$, whereas (θ, ν) is integrated out in the probabilities/expectations. Thus, since the range of f is finite,

$$-\sum_y p(f(\theta) = y \mid X_{1:N}) \log p(f(\theta) = y \mid X_{1:N}) \xrightarrow[N\to\infty]{\text{a.s.}} 0,$$

with the convention that $0 \log 0 = 0$. Since the entropy of a random variable on a finite set is bounded, then by the dominated convergence theorem,

$$H(f(\theta) \mid X_{1:N}) = \mathrm{E}\Big(-\sum_y p(f(\theta) = y \mid X_{1:N}) \log p(f(\theta) = y \mid X_{1:N}) \Big) \xrightarrow[N\to\infty]{} 0.$$

This completes the proof. □

Lemma B.2. *Let X and θ be random variables with joint density $p(x, \theta)$. Suppose $f(\theta)$ is a discrete random variable such that: for any θ, θ', if $f(\theta) = f(\theta')$ then for all x, $p(x|\theta) = p(x|\theta')$. Then $X \perp\!\!\!\perp \theta \mid f(\theta)$.*

Proof. Let $Y = f(\theta)$. Let y be any value such that $p(y) > 0$, and define $A = \{\theta : f(\theta) = y\}$. Then for all $\theta, \theta' \in A$, we have $p(x|\theta, y) = p(x|\theta) = p(x|\theta')$ by assumption. Hence,

$$\begin{aligned} p(x|y) &= \int p(x|\theta, y)\, p(\theta|y) \lambda(d\theta) = \int_A p(x|\theta, y)\, p(\theta|y) \lambda(d\theta) \\ &= \int_A p(x|\theta')\, p(\theta|y) \lambda(d\theta) = p(x|\theta') = p(x \mid \theta, y) \end{aligned}$$

where $\theta, \theta' \in A$, and $\lambda(d\theta)$ is the dominating measure for $p(\theta|y)$. Thus, $p(x|y)p(\theta|y) = p(x|\theta, y)p(\theta|y) = p(x, \theta|y)$. □

Proof of Theorem 4.2. For notational brevity, denote $e = i'$ and $N = N'$, and define $f(G) = (f_e(G), f_{i_1}(G), \ldots, f_{i_K}(G))$. Suppose we can show that for all G_1 and G_2, if $f_e(G_1) = f_e(G_2)$ then $X^e_{1:N}|G_1$ is equal in distribution to $X^e_{1:N}|G_2$. Then the result will follow by Lemma B.2, since if $f(G_1) = f(G_2)$, then in particular, $f_e(G_1) = f_e(G_2)$.

We show that if $f_e(G_1) = f_e(G_2)$ then $X^e_{1:N}|G_1 \stackrel{\mathrm{d}}{=} X^e_{1:N}|G_2$. First observe that the assumed prior factors as $\pi(\beta|G) = \prod_{i=1}^V p(\beta_i|G)p(\beta^*_i)$, and therefore, since node e has no parents in G^e,

$$p(X^e_{1:N} = x_{1:N} \mid G) = p(X_{1:N} = x_{1:N} \mid G^e) \frac{p(X^e_{e,1:N} = x_{e,1:N} \mid G)}{p(X_{e,1:N} = x_{e,1:N} \mid G^e)} \tag{16}$$

where $x_{e,1:N}$ denotes $(x_{e,n} : n = 1, \ldots, N)$. Since e has no parents in G^e, $p(X_{e,1:N} = x_{e,1:N} \mid G^e)$ does not depend on G. Similarly, since

$$p(X^e_{e,1:N} = x_{e,1:N} \mid G) = \int \Big(\prod_{n=1}^N p^*(x_{e,n} \mid \beta^*_e) \Big) p(\beta^*_e) d\beta^*_e,$$

this does not depend on G either.

By Theorem 5 of Heckerman et al. (1995), the BDeu metric is likelihood equivalent, which implies that for any G_1, G_2 such that $f_e(G_1) = f_e(G_2)$, we have $p(X_{1:N} = x_{1:N} \mid G^e_1) = p(X_{1:N} = x_{1:N} \mid G^e_2)$. Therefore, applying these invariance properties to Equation 16, we see that if $f_e(G_1) = f_e(G_2)$ then $p(X^e_{1:N} = x_{1:N} \mid G_1) = p(X^e_{1:N} = x_{1:N} \mid G_2)$. This completes the proof. □

Proof of Theorem 4.3. For notational brevity, denote $e = i'$ and $N = N'$. A distribution P is said to be *faithful* to a graph G if the set of conditional independence relations that are true for P are all and only those implied by G. More precisely, given a distribution P on $(X_1, \ldots, X_V)$, define $g(P) = (\mathbb{1}(X_A \perp\!\!\!\perp_P X_B \mid X_C) : A, B, C \subseteq \{1, \ldots, V\})$, that is, $g(P)$ is a binary vector indicating which conditional independence properties hold under

P. Meanwhile, given a DAG G on $\{1,\ldots,V\}$, define $f(G) = (\mathbb{1}(X_A \perp\!\!\!\perp_G X_B \mid X_C) : A, B, C \subseteq \{1,\ldots,V\}\}$, that is, $f(G)$ is a binary vector indicating which conditional independence properties are implied by G according to the d-separation criterion. Then P is faithful to G if and only if $g(P) = f(G)$.

Let $B(G)$ be the support of the prior $\pi(\beta|G)$. Let λ_G denote the dominating measure of $\pi(\beta|G)$ on $B(G)$. (Colloquially, one might refer to λ_G as "Lebesgue measure on $B(G)$", but technically there are sum-to-one constraints on the probability vectors, so technically it is Lebesgue measure on a lower-dimensional subspace.) By Theorem 7 of Meek (1995), for any G, the set $\{\beta \in B(G) : P_{G,\beta}$ is not faithful to $G\}$ has measure zero under λ_G. In particular, $\{\beta^e \in B(G^e) : P_{G^e,\beta^e}$ is not faithful to $G^e\}$ has measure zero under λ_{G^e}. Let π^e denote the distribution of (G^e, β^e) when $(G, \beta) \sim p(G, \beta \mid D)$. Suppose we can show that $\pi^e(\beta^e|G^e)$ has a density with respect to λ_{G^e}. Then it follows that, almost surely under π^e, P_{G^e,β^e} is faithful to G^e. In other words, $g(P_{G^e,\beta^e}) = f(G^e)$ almost surely when $(G, \beta) \sim p(G, \beta \mid D)$. The conclusion of the theorem follows since, by construction, there is a one-to-one mapping between $f(G^e)$ and $f_e(G)$.

To complete the proof, we need to show that $\pi^e(\beta^e|G^e)$ has a density with respect to λ_{G^e}, or in mathematical notation, $\pi^e(\beta^e|G^e) \ll \lambda_{G^e}$. To see this, first observe that $\pi(\beta|G) \ll \lambda_G$, and thus, $p(\beta|G, D) \ll \lambda_G$.

Next, we argue that $p(\beta^e|G, D) \ll \lambda_{G^e}$. Recall that β^e is a function of β that is obtained by copying β and then (i) putting β^*_e in place of β_e, and (ii) putting β_{e1} in place of β^*_e. Let $A^e \subseteq B(G^e)$ such that $\lambda_{G^e}(A^e) = 0$, and define $A = \{\beta \in B(G) : \beta^e \in A^e\}$. Then $\lambda_G(A) = 0$, since λ_G is the product of identical measures (colloquially, "Lebesgue measure on the probability simplex") for each β_{ij} and each β^*_i. Hence, $p(\beta^e \in A^e \mid G, D) = p(\beta \in A \mid G, D) = 0$. This implies that $p(\beta^e|G, D) \ll \lambda_{G^e}$.

Letting H be a function of G defined by $H = G^e$, we have

$$\begin{aligned} p(\beta^e|G^e, D) = p(\beta^e|H, D) &= \sum_G p(\beta^e|G, H, D)p(G|H, D) \\ &= \sum_{G\,:\,G^e=H} p(\beta^e|G, D)p(G|H, D), \end{aligned}$$

and thus, $p(\beta^e|G^e, D) \ll \lambda_{G^e}$. Since $\pi^e(\beta^e|G^e)$ is just another way of writing $p(\beta^e|G^e, D)$, then $\pi^e(\beta^e|G^e) \ll \lambda_{G^e}$, as claimed. □

Appendix C: Sensitivity analysis varying the intervention sample size

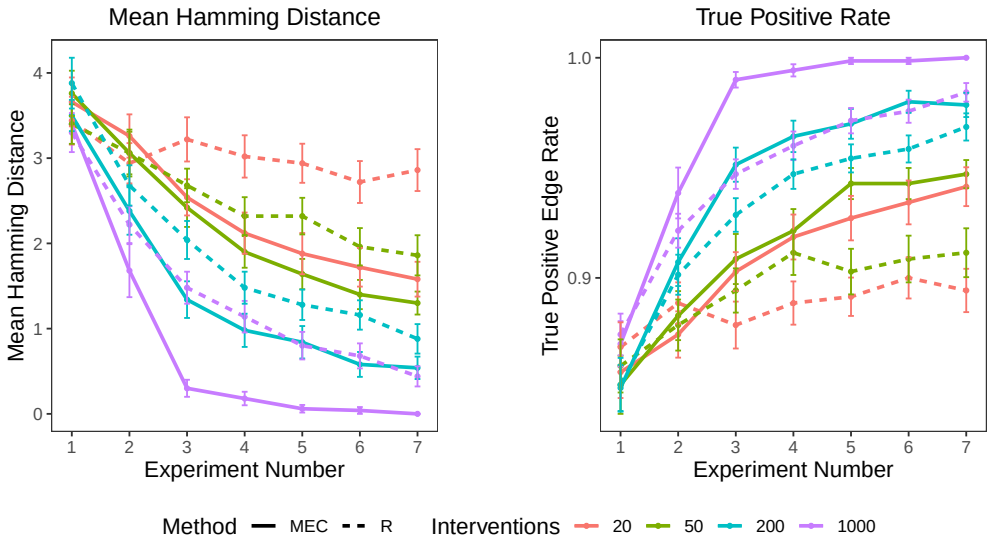

Figure 11: Sensitivity analysis varying the number of intervention samples (n_{intv}) on the ten-node binary network. Mean Hamming distance and mean TPR are displayed for the Markov equivalence class (MEC) partition method. The random learner (R) is included for comparison at the highest and lowest n_{intv} levels. Settings: $n_{sim} = 50$, $n_{exp} = 7$, $n_{obs} = 1000$, $n_{intv} \in \{20, 50, 200, 1000\}$.

Appendix D: Effect of network topology on inference

This section contains full details and results for the experiments described in Section 6.2 studying the effect of network topology on performance of each method.

Figure 12 shows the results for the chain network (Figure 7), using the settings $n_{sim} = 50$, $n_{exp} = 7$, $n_{obs} = 1000$, $n_{intv} = 1000$. The DP algorithm performed worse than the other methods, including the random learner, until the fourth experiment. Interestingly, DP performed worse than PWC, even though the two use the same entropy criterion. This can be explained by the fact that, as described in Section 4.3, the non-uniform prior over graphs that the DP algorithm uses in order to achieve its computational efficiency puts less mass on chain structures relative to other structures. Meanwhile, the hybrid MCMC-DP approach we used in the PWC method does not have such constraints on its prior over structures, thus, it performed better in this setting since a uniform prior on graphs was used.

Figure 13 shows the results for the tree network (Figure 7), using the settings $n_{sim} = 50$, $n_{exp} = 8$, $n_{obs} = 200$, $n_{intv} = 200$. MEC outperformed the other methods according

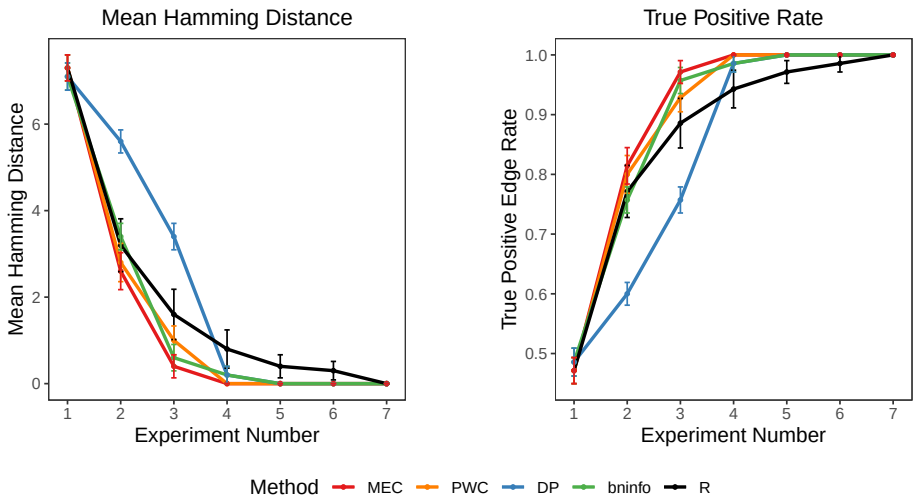

Figure 12: Mean Hamming distance and mean TPR for the 8-node chain structure. MEC = our method with MEC partition scheme; PWC = our method with pairwise child partition scheme; DP = dynamic programming method of Li and Leong (2009); R = random learner.

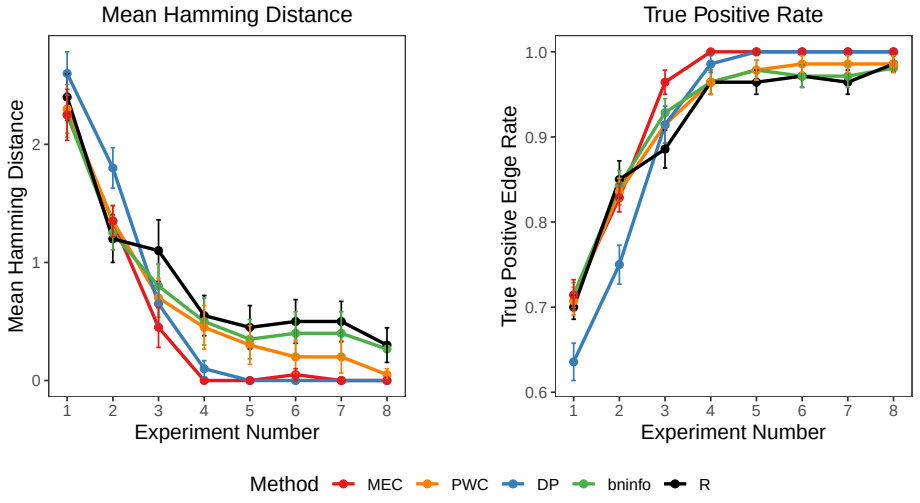

Figure 13: Mean Hamming distance and mean TPR for the 8-node tree structure. MEC = our method with MEC partition scheme; PWC = our method with pairwise child partition scheme; DP = dynamic programming method of Li and Leong (2009); R = random learner.

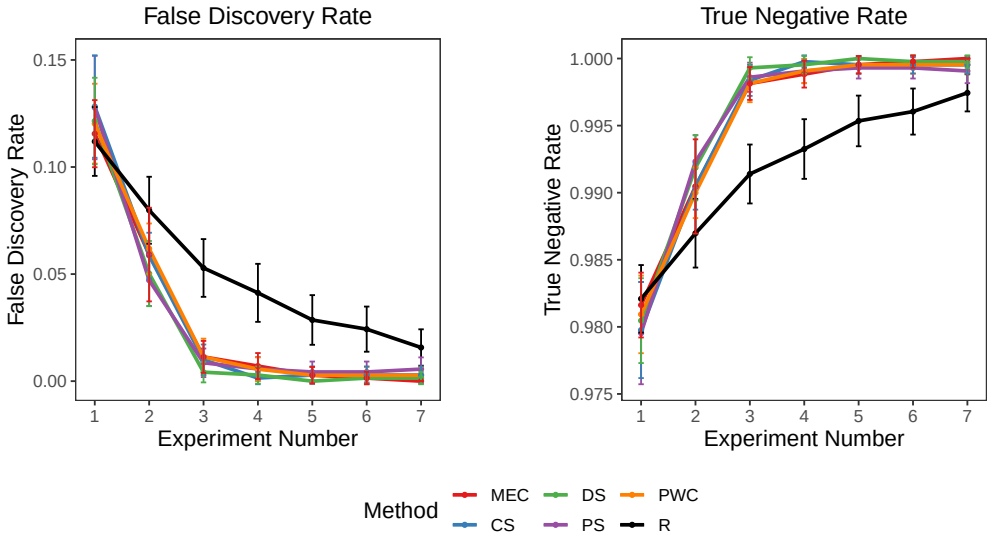

Figure 14: False discovery rate and true negative rate for the simulated ten-node binary network. These results accompany Figure 3.

to both mean Hamming distance and TPR, but all methods performed well, falling within a mean Hamming distance of one from the true network by the third experiment. While the DP method initially had a higher mean Hamming distance and lower TPR than the other methods, by the fourth experiment DP outperformed the PWC method. This illustrates that the DP method can work better than PWC when the true graph is more probable under its implicitly assumed prior.

Appendix E: Additional diagnostics

Figures 14 and 15 show plots of the false discovery rate (FDR) and true negative rate (TNR) for the simulated ten-node binary network, accompanying the results in Section 6.1.

Appendix F: Application to the Sachs network

In this section we apply our OED method, along with the DP and bninfo methods, to the published human T-cell signaling data collected by Sachs et al. (2005). We then explore the performance of the methods on a simulated data set based on the Sachs network.

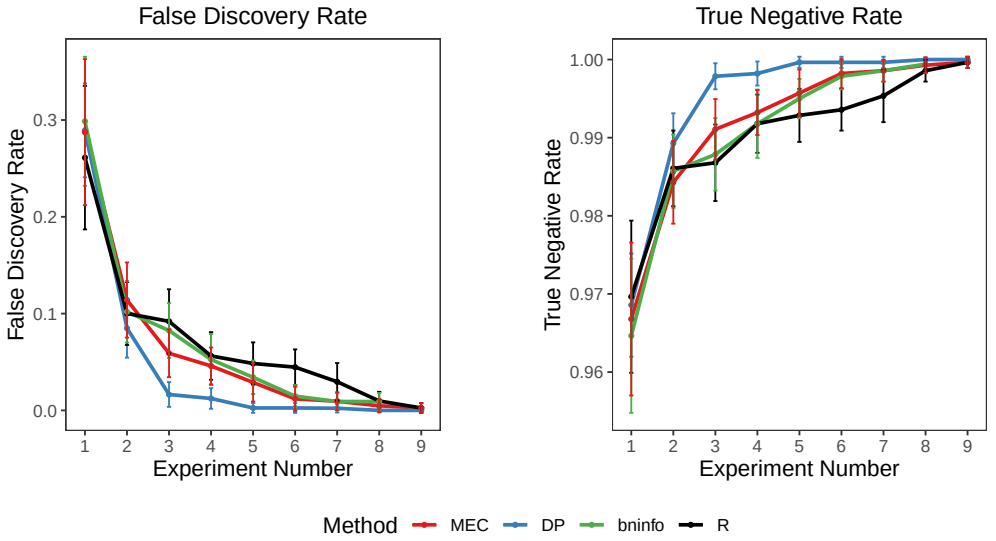

Figure 15: False discovery rate and true negative rate for the Asia network. These results accompany Figure 5.

F.1 Analysis on real experimental data from the Sachs network

The Sachs data set consists of concentration levels measured via flow cytometry for 11 proteins involved in activating the immune system. The true network describing the relationship between these proteins is unknown. Many network inference studies have explored this data set, but what they define as the benchmark graph often varies by 1-2 edges; this is likely because the biologists' consensus network is complex and contains a bidirectional relationship that would induce a cycle (Figure 16, left panel). Here, we use the benchmark network provided by Scutari (2010) (Figure 16, right panel) as well as the discretized data set with three levels (low, medium, high) available via the *bnlearn* package. A portion of the data, 1800 samples, was gathered under no targeted interventions and the remaining 3600 samples were collected after activating or inhibiting five signaling proteins: Mek12, Pip2, Akt, PKA, and PKC. (Mek12, Pip2, and Akt were each inhibited in 600 samples, PKA was activated in 600 samples, and PKC was inhibited in 600 samples and activated in an additional 600 samples.)

We compared how well the MEC, DP, bninfo, and random intervention methods inferred the cell-signaling network over a series of six experiments. For all methods, the first experiment used the 1800 observational samples. For subsequent experiments, each method then chose from the set of five candidate interventions performed by Sachs et al. (2005). Figure 17 summarizes the results. While all methods end closer to the benchmark structure after accumulating data from the five interventions, no method performs particularly well. Curiously, the mean Hamming distance initially gets worse after the first couple of experiments before getting better. See Figure 19 (Appendix B)

for a comparison of the benchmark adjacency matrix to the matrices estimated by the MEC, DP, and bninfo methods.

To determine the upper and lower bounds on performance for this data, we also considered all 120 possible permutations of the sequence of five manipulated nodes. Figure 17 shows the results for the best- and worst-performing fixed sequences in dashed lines, as determined by mean Hamming distance averaged over the six experiments. By "fixed" sequence, we mean the sequence of manipulated nodes was prespecified before the first experiment as opposed to adaptively or randomly chosen over the course of the experiments. Even the fixed sequence with the lowest mean Hamming distance over the six experiments (Figure 17 orange dashed line) initially moves further from the benchmark network after the first intervention.

There are several reasons why the methods perform differently on this data set than they did in simulation studies. First, if the true biological network consists of cycles, then the directed acyclic graphical models assumed by each of the algorithms would be misspecified. The consensus network determined by biologists suggests a short cycle among PIP3 $\rightarrow$ PLCγ $\rightarrow$ PIP2, so model misspecification is a concern. Mooij and Heskes (2013) identified another reason why the model might be misspecified: Sachs et al. (2005) used an experimental intervention that changed the *activity* of the target proteins rather than directly intervening on the abundance of the protein. An intervention that affects the underlying topology of the network in ways other than only removing arrows into the manipulated protein differs from the type of edge-breaking intervention that our model assumes. Even if the acyclicity and edge-breaking intervention assumptions were not violated, the Hamming distance at the end of the six experiments was likely high because we were limited to interventions on five candidate proteins rather than all 11 proteins. Our results would also change if we used a different discretization of the data than the one provided by Sachs et al. (2005). However, given that Cho et al. (2016) used a continuous version of the Sachs data set for their Gaussian Bayesian network and encountered similar difficulties in network reconstruction, we do not believe our results would improve substantially if we used a different discretization of the data.

We note that the performance of the bninfo method reported here differs from that published by Ness et al. (2018) for the Sachs network. This is likely due in large part to differences in the data sets used in our analyses. Ness et al. (2018) used 11,672 observational samples (they refer to this as "historic data"), whereas we used only the 1800 observational samples provided in the *bnlearn* R package since Ness et al. (2018) were unable to provide us with access to the larger data set they used in their analysis.

F.2 Analysis on simulated data from the Sachs benchmark network

To understand whether the poor performance seen on the Sachs data was due to misspecification, we tried simulating data from the benchmark network to see how the methods perform when the model assumptions hold. Figure 18 shows the results of a simulation study comparing the same methods as in Figure 17, but using simulated data generated from the benchmark network in Figure 17, using a conditional probability table estimated from the Sachs data and available in the *bnlearn* R package. The MEC

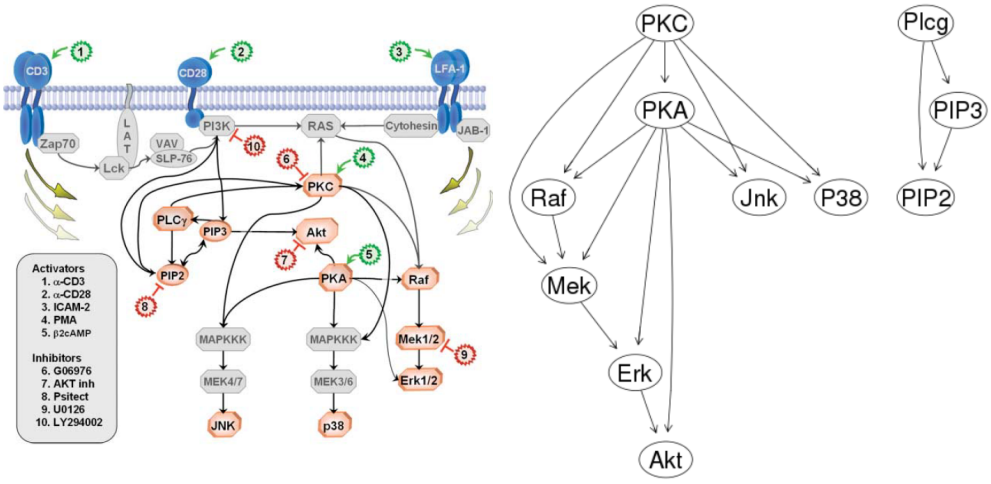

Figure 16: Left: Signaling network diagram taken from Sachs et al. (2005). Reprinted with permission from AAAS. Right: Network structure from the Bayesian Network Repository available in the *bnlearn* R package (Scutari (2010)) used here as the benchmark network.

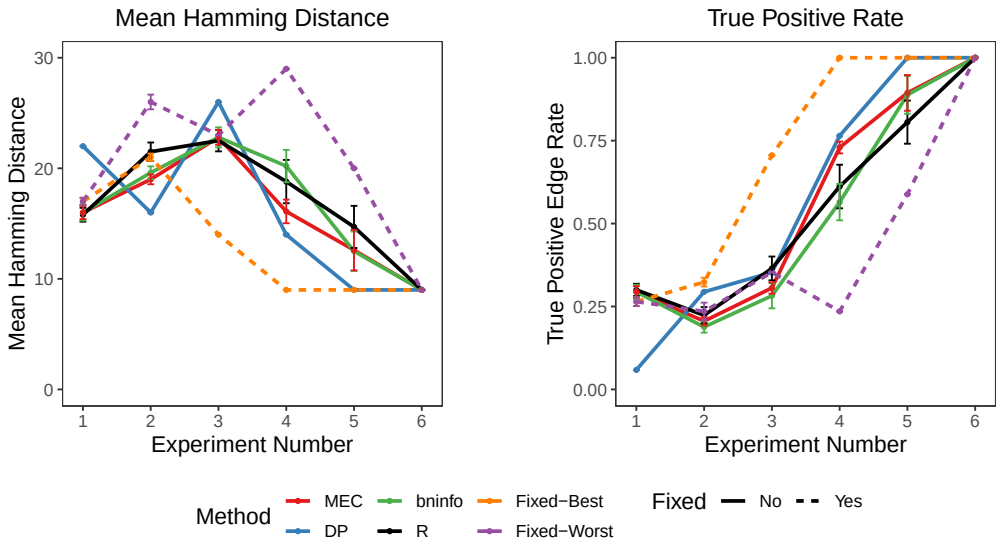

Figure 17: Mean Hamming distance and true positive rate on the cell- signaling data from the 11-node Sachs network data. MEC = our method with MEC partition scheme; DP = dynamic programming method of Li and Leong (2009); R = random learner. Dashed lines represent the best-case and worst-case fixed sequence of interventions.

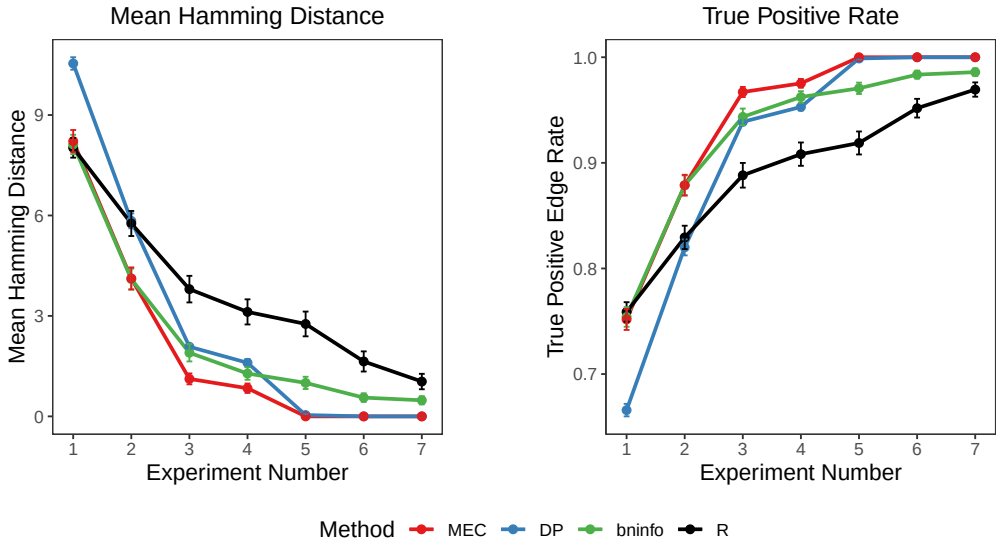

Figure 18: Mean Hamming distance and true positive rate on the cell-signaling data from the simulated 11-node Sachs network data. MEC = our method with MEC partition scheme; DP = dynamic programming method of Li and Leong (2009); R = random learner.

method performed well and fell within a Hamming distance of one from the benchmark network by the fourth experiment, on average. Bninfo also initially performed well, but then plateaued sooner than the other OED methods, failing to reach a Hamming distance of zero or TPR of 1 by the seventh experiment. The higher mean Hamming distance of the DP method for the first two experiments arose from a combination of both a lower true positive rate and higher false negative rate than the other methods. This is likely because the DP prior over graphs tends to encourage sparsity, but the ground truth network contains nodes like PKC and PKA with five and six children, respectively.

F.3 Adjacency matrices across methods for Sachs network

The adjacency matrices in Figure 19 for the MEC, DP, and bninfo methods all converge to the same DAG after experiment six. Note, however, that this estimated DAG differs from the structure of the benchmark DAG shown in the top left panel of Figure 19. The DAGs differ by a Hamming distance of nine, which upon further inspection is due to the estimated DAGs including the following false positive edges: raf $\rightarrow$ akt, mek $\rightarrow$ akt, mek $\rightarrow$ jnk, pkc $\rightarrow$ plcg, pkc $\rightarrow$ pip3, pkc $\rightarrow$ erk, p38 $\rightarrow$ plcg, jnk $\rightarrow$ plcg, jnk $\rightarrow$ p38.

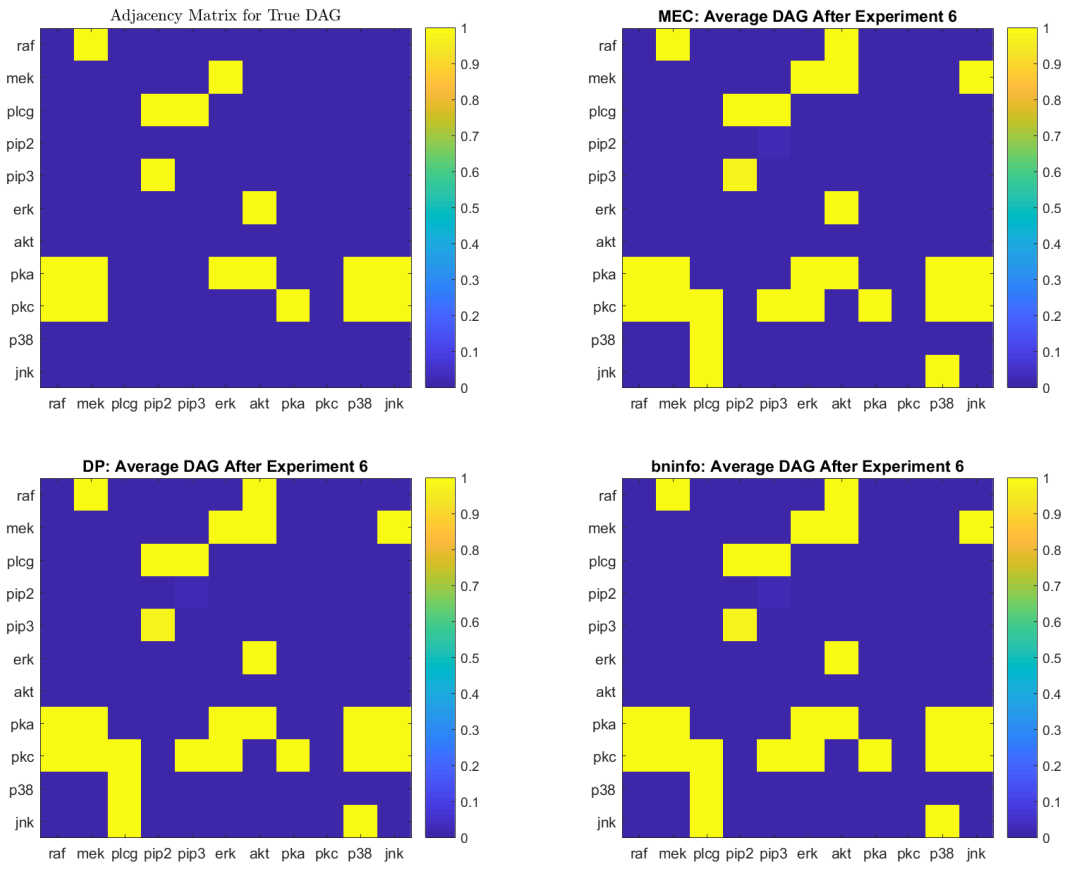

Figure 19: Adjacency matrices after the sixth experiment for the Markov equivalence class (MEC), dynamic programming (DP), and bninfo methods on the Sachs data. Top left: benchmark DAG provided by Scutari (2010). Rows denote parent nodes and columns denote child nodes. Yellow indicates presence of a directed edge while blue indicates absence of an edge.

References

Almudevar, A. and Salzman, P. (2005). "Using a Bayesian posterior density in the design of perturbation experiments for network reconstruction." In *Proceedings of the IEEE Symposium on Computational Intelligence in Bioinformatics & Computational Biology*, 1–7. 14, 15

Andersson, S. A., Madigan, D., and Perlman, M. (1997). "A characterization of Markov equivalence classes for acylic digraphs." *The Annals of Statistics*, 25(2): 505 – 541. 11

Buntine, W. (1991). "Theory refinement on Bayesian networks." In Kaufman, M. (ed.), *Proceedings of Seventh Conference on Uncertainty in Artificial Intelligence*, 52–60. 7

Castelletti, F., Consonni, G., Vedova, M. L. D., and Peluso, S. (2018). "Learning Markov equivalence classes of directed acyclic graphs: an objective Bayes approach." *Bayesian Analysis*, 13(4). 15

Chickering, D. (1996). "Learning equivalence classes of Bayesian network structures." In Horvitz, E. and Jensen, F. (eds.), *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*, 150–157. Morgan Kaufman. 11

Chickering, D. M. (2002). "Optimal structure identification with greedy search." *Journal of Machine Learning Research*, 3(Nov): 507–554. 19, 21

Cho, H., Berger, B., and Peng, J. (2016). "Reconstructing causal biological networks through active learning." *PLoS ONE*, 11(3): 1–15. 2, 14, 23, 35

Cooper, G. and Yoo, C. (1999). "Causal discovery from a mixture of experimental and observational data." In Horvitz, E. and Jensen, F. (eds.), *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, 116–125. 7

Daly, R., Shen, Q., and Aitken, S. (2011). "Learning Bayesian networks: Approaches and issues." *The Knowledge Engineering Review*, 26(2): 99–157. 1

Dixit, A., Parnas, O., Li, B., Chen, J., Fulco, C. P., Jerby-Arnon, L., Marjanovic, N. D., Dionne, D., Burks, T., Raychowdhury, R., et al. (2016). "Perturb-Seq: dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens." *Cell*, 167(7): 1853–1866. 2, 21, 23

Doob, J. L. (1949). "Application of the theory of martingales." In *Actes du Colloque International Le Calcul des Probabilités et ses applications (Lyon, 28 Juin – 3 Juillet, 1948)*, 23–27. Paris CNRS. 4

Eaton, D. and Murphy, K. (2007a). "Bayesian structure learning using dynamic programming and MCMC." In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*, 101–108. 2, 7, 12, 13

— (2007b). "Exact Bayesian structure learning from uncertain interventions." In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, 107–114. 2

Eberhardt, F. (2008). "Almost optimal intervention sets for causal discovery." In *Uncertainty in Artifical Intelligence*, 161–168. 14

Ellis, B. and Wong, W. (2006). "Sampling Bayesian Networks quickly." In *Interface*. 12, 13

Friedman, N. and Koller, D. (2003). "Being Bayesian about network structure: a Bayesian approach to structure discovery in Bayesian networks." *Machine Learning*, 50: 95–126. 12

Geiger, D. and Heckerman, D. (1994). "Learning Gaussian Networks." In *UAI*. 7

— (1999). "Parameter Priors for Directed Acyclic Graphical Models and the Characterization of Several Probability Distributions." *ArXiv*, abs/2105.03248. 7

— (2013). "Learning Gaussian Networks."
URL https://arxiv.org/abs/1302.6808 7

Hauser, A. and Bühlmann, P. (2012a). "Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs." *Journal of Machine Learning Research*, 13: 2409–2464. 11, 14, 19, 21

— (2012b). "Two optimal strategies for active learning of causal models from interventions." In *Sixth European Workshop on Probabilistic Graphical Models*, 123–130. 14

He, Y.-B. and Geng, Z. (2008). "Active learning of causal networks with intervention experiments and optimal designs." *Journal of Machine Learning Research*, 9: 2523–2547. 14

Heckerman, D., Geiger, D., and Chickering, D. (1995). "Learning Bayesian networks: The combination of knowledge and statistical data." *Machine Learning*, 20: 197 – 243. 7, 29

Kalisch, M., Mächler, M., Colombo, D., Maathuis, M. H., and Bühlmann, P. (2012). "Causal Inference Using Graphical Models with the R Package pcalg." *Journal of Statistical Software*, 47(11): 1–26. 19

Koivisto, M. (2006). "Advances in exact Bayesian structure discovery in Bayesian networks." In Press, A. (ed.), *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, 241–248. 12, 19

Koivisto, M. and Sood, K. (2004). "Exact Bayesian structure discovery in Bayesian networks." *Journal of Machine Learning Research*, 5: 549–573. 12

Lauritzen, S. and Spiegelhalter, D. (1988). "Local Computation with Probabilities on Graphical Structures and their Application to Expert Systems (with discussion)." *Journal of the Royal Statistical Society: Series B*, 50(2): 157–224. 19

Li, G. and Leong, T.-Y. (2009). "Active learning for causal Bayesian network structure with non-symmetrical entropy." In *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, 290–301. Springer-Verlag. 14, 19, 20, 32, 36, 37

Madigan, D., York, J., and Allard, D. (1995). "Bayesian Graphical Models for Discrete Data." *International Statistical Review*, 63(2): 215–232. 12

Meek, C. (1995). "Strong completeness and faithfulness in Bayesian networks." In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, 411–418. 30

Miller, J. W. (2018). "A detailed treatment of Doob's theorem." *arXiv preprint arXiv:1801.03122*. 4

Mooij, J. M. and Heskes, T. (2013). "Cyclic causal discovery from continuous equilibrium data." In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence*. 35

Murphy, K. P. (2001). "Active Learning of Causal Bayes Net Structure." Technical report, Department of Computer Science, University of California, Berkeley. 14

Nagarajan, M., Scutari, M., and Lebre, S. (2013). *Bayesian Networks in R with Applications in Systems Biology*. USA: Springer. 1

Ness, R. O., Sachs, K., Mallick, P., and Vitek, O. (2018). "A Bayesian active learning experimental design for inferring signaling networks." *Journal of Computational Biology*, 25(7): 709–725. 2, 14, 19, 20, 23, 35

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems*. San Mateo: Morgan and Kaufman. 10

— (2000). *Causality: models, reasoning, and inference*. New York: Cambridge University Press. 1, 5

Peters, J., Mooij, J., Janzing, D., and Scholkopf, B. (2011). "Identifiability of causal graphs using functional models." In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, 589–598. UAI 2011. 1, 11

Peterson, C., Stingo, F. C., and Vannucci, M. (2015). "Bayesian inference of multiple Gaussian graphical models." *Journal of the American Statistical Association*, 110. 15

Pournara, I. and Wernisch, L. (2004). "Reconstruction of gene networks using Bayesian learning and manipulation experiments." *Bioinformatics*, 20(17): 2934–2942. 7, 14

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D., and Nolan, G. (2005). "Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data." *Science*, 308. 1, 2, 21, 33, 34, 35, 36

Scutari, M. (2010). "Learning Bayesian Networks with the bnlearn R Package." *Journal of Statistical Software*, 35(3): 1–22. 16, 19, 34, 36, 38

Spirtes, P., Glymour, C., and Scheines, R. (2001). *Causation, Prediction, and Search*. MIT Press. 14

Sverchkov, Y. and Craven, M. (2017). "A review of active learning approaches to experimental design for uncovering biological networks." *PLoS Computational Biology*, 13(6): 1–26. 15, 23

Tong, S. and Koller, D. (2001). "Active learning for structure in Bayesian networks." In *International Joint Conference on Artificial Intelligence*, 863–869. 14

Verma, T. and Pearl, J. (1991). "Equivalence and Synthesis in Causal Models." *Uncertainty in Artificial Intelligence*, 6. 1, 10

Wang, Y., Solus, L., Yang, K., and Uhler, C. (2017). "Permutation-based causal inference algorithms with interventions." *Advances in Neural Information Processing Systems*, 30. 23, 26