Near-Optimal Multi-Perturbation Experimental Design for Causal Structure Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Causal structure learning is a key problem in many domains. Causal structures can be learnt by performing experiments on the system of interest. We address the largely unexplored problem of designing experiments that *simultaneously intervene on multiple variables*. While potentially more informative than the commonly considered single-variable interventions, selecting such interventions is algorithmically much more challenging, due to the doubly-exponential combinatorial search space over sets of composite interventions. In this paper, we develop efficient algorithms for optimizing different objective functions quantifying the informativeness of experiments. By establishing novel submodularity properties of these objectives, we provide approximation guarantees for our algorithms. Our algorithms empirically perform superior to both random interventions and algorithms that only select single-variable interventions.

1 Introduction

The problem of finding the causal relationships between a set of variables is ubiquitous throughout the sciences. For example, scientists are interested in reconstructing gene regulatory networks (GRNs) of biological cells [10]. Directed Acyclic Graphs (DAGs) are a natural way to represent causal structures, with a directed edge from variable X to Y representing X being a direct cause of Y [31].

Learning the causal structure of a set of variables is fundamentally difficult. With only observational data, in general we can only identify the true DAG up to a set of DAGs called its *Markov Equivalence Class (MEC)* [33]. Empirically, for sparse DAGs the size of the MEC grows exponentially in the number of nodes [15]. Identifiability can be improved by intervening on variables, meaning one perturbs a subset of the variables and then observes more samples from the system [7, 14, 38]. There exist various inference algorithms for learning causal structures from a combination of observational and interventional data [14, 34, 38, 26, 32].

Performing experiments is often expensive, however. Thus, we are interested in learning as much about the causal structure as possible given some constraints on the interventions. In this work, we focus on the *batched setting*, where several interventions are performed in parallel. This is a natural setting in scientific domains like reconstructing GRNs. Existing works propose meaningful objective functions for this batched causal structure learning problem and then give algorithms that have provable guarantees [38, 12]. However, these works focus on the setting where only a *single* random variable is perturbed per intervention. It is an open question as to whether there exist efficient algorithms for the *multiple-perturbation* setting, where more than one variable is perturbed in each intervention.

For the task of reconstructing GRNs, it is now possible for experimenters to perturb multiple genes in a single cell [2, 6]. Figures 1 b) and c) illustrate a specific example where a two-node intervention completely identifies a DAG in half as many interventions as single-node interventions. In general,

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

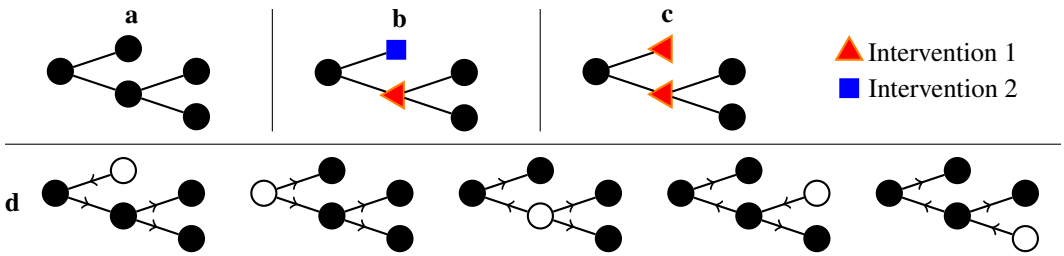

Figure 1: **a)** We illustrate the MEC of a tree graph on 5 nodes. **b)** Two single-node interventions are required to fully identify the true DAG. **c)** Only one two-node intervention is required to fully identify the true DAG. **d)** This MEC contains 5 DAGs, each corresponding to a different root node (marked white). This is a property particular to tree MECs.

it is possible for a set of q-node interventions to orient *up to q-times more* edges in a DAG than single-node interventions (see the supplementary material for a more general example). While multi-perturbation interventions can be more informative, designing them is algorithmically challenging because it leads to an *exponentially* larger search space: any algorithm must now select a set of sets.

Our main contribution is to provide efficient algorithms for different objective functions with accompanying performance bounds. We demonstrate empirically on both synthetic and GRN graphs that our algorithms result in greater identifiability than existing approaches that do not make use of multiple perturbations [12, 38], as well as a random strategy.

We begin by introducing the notation and the objective functions considered in this work in Section 2, before reviewing related work in Section 3. In Section 4 we present our algorithms along with proofs of their performance guarantees. Finally, in Section 5 we demonstrate the superior empirical performance of our method over existing baselines on both synthetic networks and on data generated from models of real GRNs.

2 Background and Problem Statement

Causal DAGs Consider a causal DAG $G = ([p], E)$ where $[p] := \{1, ..., p\}$ is a set of nodes and E is a set of directed edges. Let $(i, j) \in E$ iff there is an edge from node i to node j. Each node i is associated with a random variable X_i. In the GRN example, X_i would be the measurement of the gene expression level for gene i. An edge from $i \to j$ would represent gene i having a causal effect on the expression of gene j. The functional dependence of a random variable on its parents can be described by a structural equation model (SEM).

The probability distribution over $X = (X_1, \ldots, X_p)$ is related to G by the Markov property, meaning each variable X_i is conditionally independent of its non-descendants given its parents [31]. From conditional independence tests one can determine the MEC of G, a set of DAGs with the same conditional independancies between variables. All members of the MEC share the same undirected skeleton and colliders [33]. A collider is a pair of edges $(i, k), (j, k) \in E$ such that $(i, j), (j, i) \notin E$. The *essential graph* of G, $\text{Ess}(G)$, is a partially directed graph, with directed edges where all members of the MEC share the same edge direction, and with undirected edges otherwise [4]. $\text{Ess}(G)$ uniquely represents the MEC of G. These MECs can be large, so we seek to perform interventions on the nodes to reduce the MEC to a smaller set of possible DAGs.

Interventions We use the term *intervention* to refer to a set $I \subset [p]$ of perturbation targets (variables). We assume all interventions are *hard interventions*, meaning intervening on a set I removes the incoming edges to the random variables $X_I := (X_i)_{i \in I}$ and sets their joint distribution to some interventional distribution $\mathcal{P}^I$ [8]. In the GRN reconstruction example this corresponds to, for example, running an experiment where we knockout all genes in set I. Some of our results extend easily to the alternative model of *soft* interventions [23], as we discuss in the supplementary material.

We use $\mathcal{I} = 2^{[p]}$ to refer to the set of all possible interventions. Our goal will be to select a batch ξ of interventions, where ξ is a multiset of some $I \in \mathcal{I}$. For practical reasons, we typically have *constraints* on the number of interventions, i.e., $|\xi| \leq m$ and on the number of variables involved in each intervention $|I| \leq q, \forall I \in \xi$. Namely, there are at most m interventions per batch and each

intervention contains at most q nodes. A constraint on the number of perturbations per intervention is natural in reconstructing GRNs, since perturbing too many genes in one cell will leave it unlikely to survive. We refer to the set of ξs satisfying these constraints as $C_{m,q}$. The observational distribution (no intervention) is given by $\xi = \emptyset$.

For any set of interventions ξ and DAG G, there is a set of ξ-Markov equivalent DAGs. These are the set of DAGs that have the same set of conditional independencies under all $I \in \xi$ and under the observational distribution. This set of DAGs is no larger than the MEC of G and can similarly be characterized by an essential graph $\text{Ess}^{\xi}(G)$ [14].

We will always assume that there exist no unobserved common causes of any pair of nodes in G. We also assume that the distribution of the random variables satisfies faithfulness with respect to G [31].

Choosing optimal interventions We seek to maximize an objective function F that quantifies our certainty in the true DAG. In general our goal is to determine

$$\arg\max_{\xi \in C_{m,q}} F(\xi).$$

A natural choice for F is given by Agrawal et al. [3]. They assume that there exist parameters θ that determine the functional relationships between random variables. For example, this could be the coefficients in a linear model. Given existing data D, we try to choose ξ that maximizes

Objective 1 (Mutual Information (MI)).

$$\text{F}_{\text{MI}}(\xi) = \mathbb{E}_{G|D}\mathbb{E}_{y|G,\hat{\theta},\xi}\left[\tilde{U}_{M.I}(y, \xi; D)\right], \tag{1}$$

where y is the set of samples from the interventions, $\hat{\theta}$ is the current estimate of the parameters given D and G, and $\tilde{U}_{M.I}(y, \xi; D)$ is the mutual information between the posterior over G and the samples y. Each intervention produces one sample in y. The use of mutual information means the objective aims to, in expectation over all observed samples and true DAGs, minimize the entropy of the posterior distribution over DAGs. There already exist a number of algorithms for determining the posterior over DAGs from observational or experimental data [38, 34, 14].

Infinite sample objectives Finding algorithms that optimize the MI objective is difficult because we have to account for noisy observations and limited samples. To remove this complexity, we study the limiting case of *infinitely many samples* per unique intervention. The constraints given by $C_{m,q}$ still stipulate that there can be only m unique interventions, but each intervention can be performed with an infinite number of samples. We also assume that an essential graph is already known (i.e., we have infinite observational samples and infinite samples for any experiments performed so far). For objectives with infinite samples per intervention, we treat ξ as a set of interventions, not a multiset, since there is no change in objective value for choosing an intervention twice. Consider ξ' to be the set of interventions contained in our dataset before our current batch. In this setting, maximizing Objective 1 reduces to maximizing

Objective 2 (Mutual info. inf. samples (MI-∞)).

$$F_{\infty}(\xi) = -\frac{1}{|\mathcal{G}|}\sum_{G \in \mathcal{G}} \log_2 |\text{Ess}^{\xi \cup \xi'}(G)|, \tag{2}$$

where $\text{Ess}^{\xi \cup \xi'}(G)$ refers to the updated essential graph after performing interventions in ξ. The objective aims to, on average across possible true DAGs, minimize the $\log$ of the essential graph size after performing the interventions. The derivation of this objective is given in the supplementary material.

Ghassami et al. [12] study a different objective in the infinite-sample setting. The objective seeks to, on average across possible G given the current essential graph, orient as many edges as possible. Let $R(\xi, G)$ be the set of edges oriented by ξ if the true DAG is G.

Objective 3 (Edge-orientation (EO)).

$$F_{\text{EO}}(\xi) = \frac{1}{|\mathcal{G}|}\sum_{G \in \mathcal{G}} |R(\xi, G)|. \tag{3}$$

The function R is computed as follows. Firstly, $\forall I \in \xi$, orient undirected edge $i - j$ in $\text{Ess}^{\xi'}(G)$ if $i \in I$ but $j \notin I$ or vice-versa. Secondly, execute the *Meek Rules* [24], which allow inferring additional edge orientations (discussed in the supplementary material) on the resulting partially directed graph. Finally, output the set of all edges oriented. Agrawal et al. [3] show that this objective is not consistent; however, this is because they fix $\mathcal{G}$ to be the MEC instead of using the most up-to-date essential graph for each batch. In the supplementary material, we show that the version of the objective we work with is indeed consistent.

Below, we provide algorithms with near-optimality guarantees for Objectives 2 and 3, while motivating a practical algorithm for Objective 1.

3 Related Work

Causality has been widely studied in machine learning [28, 29]. Here we focus on prior research that is most relevant to our work.

Agrawal et al. [3] and Ghassami et al. [12] give near-optimal greedy algorithms for Objectives 1 and 3 respectively. However, these algorithms only apply to *single-perturbation* interventions. Both of these works use the submodularity of the two objectives. In this paper we address the exponentially large search space that arises when designing multi-perturbation interventions, a strictly harder problem.

Much existing work in experimental design for causal DAGs is focused on identifying the graph uniquely, while minimizing some cost associated with doing experiments [8, 16, 30, 19, 21]. When the MEC is large and the number of experiments is small, identifying the entire graph will be infeasible, and instead one must select interventions that optimize some objective.

Lindgren et al. [21] show NP-hardness for selecting an intervention set of at most m interventions, with minimum number of perturbations, that completely identifies the true DAG. This, however, does not directly imply a hardness result for our problem.

Gamella and Heinze-Deml [9] propose an approach to experimental design for causal structure learning based on invariant causal prediction. While our approach has guarantees for objectives relating to either the whole graph or functions of the oriented edges, their work is specific to the problem of learning the direct causes of one variable.

Acharya et al. [1] consider *testing* between two candidate causal models. However, the setting differs from ours: they assume the underlying DAG is known but allow for unobserved confounding variables.

4 Greedy Algorithms for Experiment Design

All of our algorithms follow the same general strategy. Like in previous works on single-perturbation experimental design [3, 12], we greedily add interventions to our intervention set. We add I maximizing $F(\xi \cup \{I\})$ where ξ is the current set of interventions. This greedy selection is justified because our objectives are submodular, a property we define formally later. For single-perturbation experimental design, this is algorithmically simple since there are only p possible interventions. However, for multi-perturbation interventions even selecting greedily is *intractable* at scale since we have $\binom{p}{q}$ possible interventions. Therefore we provide ways to find an intervention that is approximately greedy, i.e, an intervention with marginal improvement in objective that is close to that of the greedy intervention.

A further challenge with the greedy approach is that it involves evaluating the objective, which for our objectives is a potentially *exponential sum* over members of an essential graph. Each of the two algorithms we give has a different strategy for overcoming this.

In Section 4.1, we present DOUBLE GREEDY CONTINUOUS (DGC) for optimizing Objective 3, the edge-orientation objective. For greedily selecting interventions to maximize an exponential sum, we employ the stochastic continuous optimization technique of Hassani et al. [13].

In Section 4.2, we present SEPARATING SYSTEM GREEDY (SSG) for optimizing Objective 2, MI-∞. To greedily select interventions, we use the construction of *separating systems* (SS) [35, 30, 21], to create a smaller set of interventions to search over. Collectively, the interventions in the SS fully orient the graph. To handle tractably evaluating the objective, we use the idea of Ghassami et al.

[12] and Agrawal et al. [3] to optimize an approximation of the objective constructed using a limited sample of DAGs.

To give near-optimality guarantees for these algorithms, we will use two properties of the objectives: monotonicity and submodularity.

Definition 1. A set function $F : 2^V \to \mathbb{R}$ is *monotonically increasing* if for all sets $I_1 \subseteq I_2 \subseteq V$ we have $F(I_1) \leq F(I_2)$.

Definition 2. A set function $F : 2^V \to \mathbb{R}$ is *submodular* if for all sets $I_1 \subseteq I_2 \subseteq V$ and all $v \in V \setminus I_2$ we have $F(I_1 \cup \{v\}) - F(I_1) \geq F(I_2 \cup \{v\}) - F(I_2)$.

Submodularity is a natural diminishing returns property, and many strategies have been studied for optimizing submodular objectives [20]. In both the above definitions, V is called the *groundset*, the set that we can choose elements from. In the single-perturbation problem, the groundset is just $[p]$, whereas in our case it is all subsets of up to q nodes.

We show that DGC achieves an objective value within a constant factor of the optimal intervention set on Objective 3. SSG does not achieve a constant-factor guarantee, but for both infinite sample objectives we obtain a lower bound on its performance.

All of our algorithms run in polynomial time; however, they assume access to a uniform sampler across all DAGs in the essential graph. This exists for sampling from the MEC [37] but not for essential graphs given existing interventions. In practice, we find that an efficient non-uniform sampler [12] can be used to achieve strong empirical performance.

4.1 Optimizing the Edge-orientation Objective

In the following, we develop an algorithm for maximizing Objective 3. In fact, the algorithm we provide has a near-optimality guarantee for a more general form of F_{EO}, namely

$$F_{\text{EO}}(\xi) = \sum_{G \in \mathcal{G}} a(G) \sum_{e \in G} w(e) \mathbb{1}(e \in R(\xi, G)),$$

where $\forall e, w(e) \geq 0$ and $\forall G, a(G) \geq 0$. The weights $a(G)$ can be thought of as corresponding to having a non-uniform prior over the DAGs in the essential graph, whilst the weights $w(e)$ can be thought of as assigning priority to the orienting of certain edges. The inner sum above is a *weighted coverage function* [20] over the set of edges.

We will first show that F_{EO} is monotone submodular over groundset $\mathcal{I}$. This generalizes a result by Ghassami et al. [12] who showed the same result for groundset $[p]$ (single perturbation interventions).

Lemma 1. *F_{EO} is monotone submodular over the groundset $\mathcal{I}$.*

Proof. All proofs are presented in the supplementary material unless otherwise stated. □

As mentioned, we cannot use a greedy search directly since the groundset $\mathcal{I}$ is too large. Instead, we develop an algorithm for selecting an intervention with near-maximal utility compared to the greedy choice. In particular, our strategy is to prove a submodularity result over the function F with modified domain. Consider the set function $F^{\xi}_{\text{EO}}(I) = F_{\text{EO}}(\xi \cup \{I\})$ for fixed ξ.

Lemma 2. *F^{ξ}_{EO} is non-monotone submodular over the groundset $[p]$.*

The Non-monotone Stochastic Continuous Greedy (NMSCG) algorithm of Mokhtari, Hassani, and Karbasi [25] can therefore be used as a subroutine to select, in expectation, an approximately greedy intervention to add to an existing intervention set. The algorithm uses a stochastic gradient-based method to optimize a continuous relaxation of our objective, and then rounds the solution to obtain a set of interventions. The continuous relaxation of F^{ξ}_{EO} is the multilinear extension

$$f^{\xi}_{\text{EO}}(x) = \sum_{I \in \mathcal{I}} F^{\xi}_{\text{EO}}(I) \prod_{i \in I} x_i \prod_{i \notin I} (1 - x_i)$$

with constraints $\sum_i x_i \leq q$, $0 \leq x_i \leq 1$ for all nodes i. The multilinear extension can be thought of as computing the expectation of $F^{\xi}_{\text{EO}}(I)$, when input x is a vector of independent probabilities

such that x_i is the probability of including node i in the intervention. The sum over $\mathcal{I}$ in f^{ξ}_{EO} and the sum over DAGs in F^{ξ}_{EO} make computing the gradient of this objective intractable. Therefore, we compute an unbiased stochastic approximation of the gradient $\nabla f^{\xi}_{\text{EO}}(x)$ by uniformly sampling a DAG G from $\mathcal{G}$ and intervention I from the distribution specified by x. Define

$$\hat{f}^{\xi}_{\text{EO}}(I) = |R(\xi \cup \{I\}, G)|.$$

Mokhtari, Hassani, and Karbasi [25] show that an unbiased estimate of the gradient of $f(x)$ can be computed by sampling G and I to approximate

$$\frac{\partial}{\partial x_i} f^{\xi}_{\text{EO}}(x) = \mathop{\mathbb{E}}_{G,I|x}\left[\hat{f}^{\xi}_{\text{EO}}(I; I_i \leftarrow 1) - \hat{f}^{\xi}_{\text{EO}}(I; I_i \leftarrow 0)\right],$$

where $I_i \leftarrow 0$ means that if $i \in I$, remove it. The use of a stochastic gradient means that F_{EO} can be efficiently optimized despite it being a possibly exponential sum over $\mathcal{G}$.

After several gradient updates we obtain a vector of probabilities x that approximately maximizes f_{EO}. To obtain an intervention I one uses a ROUND function, for example pipage rounding which, on submodular functions, has the guarantee that $\mathbb{E}[F^{\xi}_{\text{EO}}(\text{ROUND}(x))] = f^{\xi}_{\text{EO}}(x)$ [5].

Theorem 1 (Mokhtari, Hassani, and Karbasi [25])**.** *Let I^* be the maximizer of F^{ξ}_{EO}.* NMSCG *with pipage rounding, after $\mathcal{O}\left(p^{5/2}/\epsilon^3\right)$ applications of the Meek rules, achieves a solution I such that*

$$\mathbb{E}\left[F^{\xi}_{EO}(I)\right] \geq \frac{1}{e} F^{\xi}_{EO}(I^*) - \epsilon.$$

The original result measures runtime in terms of the number of evaluations of a single term in the outer sum in F^{ϵ}_{EO}. In our case this corresponds to one application of the Meek rules, which can be computed in time polynomial in p [24]. Note that the subroutine can be modified to stabilize gradient updates by the approximation of a Hessian, in which case the same guarantee can be achieved in $\mathcal{O}\left(p^{3/2}/\epsilon^2\right)$ [13]. We use this version of NMSCG for the experiments.

Algorithm 1 DOUBLE GREEDY CONTINUOUS(DGC)

Input: essential graph $\mathcal{G}$, constraints $C_{m,q}$, objective F_{EO}
Init $\xi \leftarrow \emptyset$
while $|\xi| \leq m$ **do**
  $I \leftarrow \text{ROUND}(\text{NMSCG}(F^{\xi}_{\text{EO}}, q))$
  $\xi \leftarrow \xi \cup \{I\}$
end while
Output: a set ξ of interventions

Algorithm 2 SEPARATING SYSTEM GREEDY(SSG)

Input: essential graph $\mathcal{G}$, constraints $C_{m,q}$, objective $\tilde{F}_{\infty}$
Init $\mathcal{I} \leftarrow \emptyset$
$\mathcal{S} \leftarrow \text{SEPARATE}(q, \mathcal{G})$
while $|\xi| \leq m$ **do**
  $I' \leftarrow \arg\max_I \tilde{F}_{\infty}(\mathcal{I} \cup \{I\})$
  $\xi \leftarrow \xi \cup \{I'\}$
end while
Output: a set ξ of interventions

Our main result now follows from the fact that selecting interventions approximately greedily will lead to an approximation guarantee due to lemma 1.

Theorem 2. *Let $\xi^* \in C_{N,b}$ be the maximizer of Objective 3.* DGC *will, after $\mathcal{O}\left(m^4 p^{5/2}/\epsilon^3\right)$ applications of the Meek rules, achieve a solution ξ such that*

$$\mathbb{E}[F_{EO}(\xi)] \geq \left(1 - \frac{1}{e^{1/e}}\right) F_{EO}(\xi^*) - \epsilon.$$

We highlight this is a constant-factor guarantee with respect to the optimal batch of interventions. Our bound requires compute that is low order in p with no dependence on q. Without even accounting for computing the possibly exponential sum in the objective, merely enumerating all possible interventions for fixed q is $\mathcal{O}(p^q)$.

4.2 Optimizing the Mutual Information Objective

We now consider an algorithm for maximizing Objective 2. First, we note that computing the sum over $\mathcal{G}$ and the size of $\xi \cup \xi'$-essential graphs is computationally intractable. The computation of $|\text{Ess}^{\xi \cup \xi'}(G)|$ makes F_∞ a nested sum over DAGs. Like Agrawal et al. [3], we optimize a computationally tractable approximation to F_∞. First, we uniformly sample a multiset of DAGs $\tilde{\mathcal{G}}$ from $\mathcal{G}$ to construct our approximate objective

$$\tilde{F}_\infty(\xi) = -\frac{1}{|\tilde{\mathcal{G}}|} \sum_{G \in \tilde{\mathcal{G}}} \log_2 |\tilde{\text{Ess}}^{\xi \cup \xi'}(G)|,$$

where $\tilde{\text{Ess}}^{\xi \cup \xi'}$ is the submultiset of $\tilde{\mathcal{G}}$ consisting of elements in $\text{Ess}^{\xi \cup \xi'}$. A submodularity result similar to lemma 1 can also be proven for $\tilde{F}_\infty$.

Lemma 3. *$\tilde{F}_\infty$ is monotone submodular.*

We first show that an approach similar to that used by DGC does not so easily give a near-optimal guarantee. Similarly to above, define $\tilde{F}^{\xi}_\infty(I) = \tilde{F}_\infty(\xi \cup \{I\})$ for fixed ξ.

Proposition 1. *There exists $\mathcal{G}, \tilde{\mathcal{G}}$ such that $\tilde{F}^{\xi}_\infty$ is not submodular.*

Hence we cannot use existing algorithms for submodular optimization to construct near-greedy interventions. We instead take a different approach. Suppose we can reduce $\mathcal{I}$ to some set of interventions $\mathcal{S}$ much smaller than $\mathcal{I}$, such that $\tilde{F}_\infty(\mathcal{S})$ has the maximum possible objective value. Due to lemma 3, we can obtain a guarantee by greedily selecting m interventions from $\mathcal{S}$. A method for constructing the set $\mathcal{S}$ comes from *separating system* constructions.

Definition 3. A q-sparse $\mathcal{G}$-separating system of size N is a set of interventions $\mathcal{S} = \{S_1, S_2 ..., S_N\}$ such that $|S_i| \leq q$ and for every undirected edge $(i, j) \in \mathcal{G}$ there is an element $S \in \mathcal{S}$ such that exactly one of i, j is in S [30].

A separating system of $\mathcal{G}$ completely identifies the true DAG, and hence obtains the maximum possible objective value 0 without necessarily satisfying the constraints $C_{m,q}$. As an example, in Figure 1(b,c) we see 1 and 2–sparse separating systems respectively.

We will make use of an algorithm SEPARATE$(q, \mathcal{G})$ which efficiently constructs a q-sparse separating system of $\mathcal{G}$. Wegener [35] and Shanmugam et al. [30] give construction methods that are agnostic to the structure of $\mathcal{G}$ (it will identify any DAG with p nodes). Lindgren et al. [21] give a construction method that depends on the structure of $\mathcal{G}$.

Since the separating system obtains the maximum objective value, we can greedily select m interventions from this set and obtain a lower bound on the objective value due to submodularity (lemma 3).

Theorem 3. *For $q \leq \lfloor p/2 \rfloor, m \leq |\mathcal{S}|$,* SSG *outputs $\xi \in C_{m,q}$ with objective value*

$$\tilde{F}_\infty(\xi) \geq (1 - \frac{m}{\lceil p/q \rceil \lceil \log p \rceil}) \tilde{F}_\infty(\emptyset)$$

in $\mathcal{O}(m|\tilde{\mathcal{G}}| \frac{p}{q} \log p)$ applications of the Meek rules, when using SEPARATE *as in Shanmugam et al. [30].*

Increasing q does not necessarily increase $\tilde{F}_\infty(\xi)$. However, the bound we give becomes more favourable as q increases because the upper bound on $|\mathcal{S}|$ decreases. In practice, we run SSG $\forall q' \leq q$ and pick the intervention set with the highest objective value on $\tilde{F}_\infty$.

Note that SSG can also be used with a similar guarantee for an analogous approximation of F_{EO}.

5 Experiments

To evaluate our algorithms we consider three settings. Firstly, randomly generated DAGs, using infinite samples per intervention. Secondly, randomly generated DAGs with linear SEMs, using finite samples per intervention. Finally, subnetworks of GRN models, using infinite samples per intervention. Full details on all of our experiments can be found in the supplementary material.

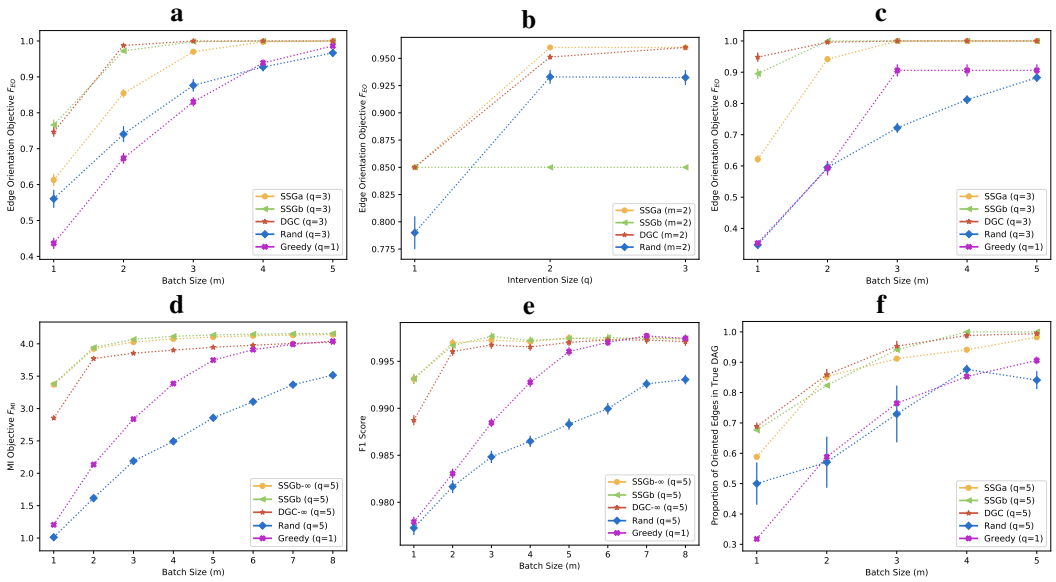

Figure 2: a–c and f give infinite sample experiments and d–e give finite samples. **a)** Our algorithms ($q = 3$) orient more edges than random interventions ($q = 3$) and greedily chosen $q = 1$ interventions for $p = 40$, ER(0.1) graphs. **b)** On a fully connected graph (p=5, m=2), SSG-B does not improve as q increases. **c)** On a $p = 20$ forest of 3 disconnected star graphs, SSG-A cannot orient the full graph as quickly as our alternative approaches ($q = 3$). **d)** For finite-samples, $p = 40$, ER(0.1) graphs, the proposed methods ($q = 5$) achieve greater objective value than greedy $q = 1$ interventions. Optimizing the finite sample objective directly yields slightly greater objective value than the infinite sample approximations. **e)** The F1 scores for predicting the presence of each edge correspond well with the objective in d). **f)** For a $p = 50$ yeast subnetwork from the DREAM3 challenge, our algorithms ($q = 5$) orient more edges than random or $q = 1$ greedy algorithms with the same batch size.

Infinite samples We evaluate our algorithms using Objective 3. We consider selecting a batch of experiments where only the MEC is currently known. We vary the type of random graph and the constraint set $C_{m,q}$. The following methods are compared:

- RAND: a baseline that for m interventions, independently and uniformly at random selects q nodes from those adjacent to at least one undirected edge;
- GREEDY: greedily selects a single-perturbation intervention as in Ghassami et al. [12];
- DGC: our stochastic optimization approach;
- SSG-A: our greedy approach using the graph agnostic separating systems of Shanmugam et al. [30];
- SSG-B: as above, using the graph-dependent separating system constructor of Lindgren et al. [21].

Since there are infinite samples per intervention, the exact SEM used to generate data is not relevant. We plot the mean proportion of edges identified and error bars of 1 standard deviation over 100 repeats. Noise between repeats is due to randomness in the graph structure and in the algorithms themselves.

In Figure 2 a) we display the results for Erdös-Renyí random graphs with edge density 0.1 (ER(0.1)) and 40 nodes. To prevent trivial graphs and large runtimes, we only consider graphs with MEC sizes in the range $[20, 200]$. The observations given here were also found for denser Erdös-Renyí graphs and tree graphs, in addition to graphs with less nodes.

For all constraint values, all the algorithms improve greatly over RAND and GREEDY. SSG-B outperforms SSG-A, likely because the graph-sensitive separating system construction tends to return a groundset of more effective interventions.

SSG-B achieves similar objective value to DGC. However, DGC behaves most robustly when the graph structure is chosen adversarially. For example, consider Figure 2 b). Here we plot the proportion of identified edges on a $p = 5$ fully connected graph. On this graph, the separating system construction of Lindgren et al. [21] will always return the set of all single-node interventions.

Therefore, its performance does not improve with q, whilst DGC's does. An adversarial example for SSG-A is constructed in Figure 2 c): an MEC that consists of 3 disconnected star graphs with 7, 7 and 6 nodes. In this case, DGC and SSG-B can orient most of the graph in a single intervention, whereas SSG-A will likely not contain such an intervention in the separating system it constructs.

Finite samples We use linear SEMs, with weights generated uniformly in $[-1, -0.25] \cup [0.25, 1]$. Measurement noise is given by the standard normal distribution. The underlying DAGs are generated in the same way as the infinite sample experiments. Before experiment selection, we obtain 800 observational samples of the system. 3 samples are obtained for each intervention selected by our algorithms. Each perturbation fixes the value of a node to 5. We approximate Objective 1 using the methods of Agrawal et al. [3]. In particular, an initial distribution over DAGs is estimated by bootstrapping the observational data and using the techniques of Yang, Katcoff, and Uhler [38] to infer DAGs. Each DAG in the distribution is weighted proportionally to the likelihood of the observational data given the DAG and the maximum likelihood estimate of the linear model weights. The posterior over DAGs after interventions is computed by re-weighting the existing set of DAGs based on the likelihood of the combined observational and interventional data. To ensure the distribution has support near the true DAG, we include all members of the true DAG's MEC in the initial distribution.

With finite samples, our methods do not have guarantees but can be adapted into practical algorithms:

- GREEDY: greedily optimize Objective 1 with single perturbation interventions (Agrawal et al. [3]);
- DGC-∞ : optimizes Objective 3, with the summation over DAGs being a weighted sum over the DAGs in the initial distribution;
- SSG-B: optimizes F_{MI}, greedily selecting from the separating system of Lindgren et al. [21];
- SSG-B-∞: approximates the objective using Objective 2 and optimizes with SSG.

For each algorithm, we record F_{MI} of the selected interventions, over 200 repeats. In Figure 2 d) the objective values obtained by each algorithm are shown for varying batch size and $q = 5$. DGC-∞ performs worse than SSG-B and SSG-B-∞, perhaps because it is optimizing F_{EO} which is not totally aligned with F_{MI}. Between SSG-B and SSG-B-∞, there was a small benefit to directly optimizing F_{MI} as opposed to its infinite sample approximation F_{∞}. Accounting for finite samples may lead to greater improvements when there is heteroscedastic noise or a wider range of weights.

Performance on F_{MI} corresponds closely with performance on a downstream task as shown in Figure 2 e). For each algorithm, we compute the posterior over DAGs given the selected interventions. Then, we independently predict the presence of each edge in the true DAG. Figure 2 e) plots the average F1 score for each algorithm. In finite samples, our approaches outperform both RAND and GREEDY.

DREAM 3 networks We evaluate our algorithms under infinite samples using subgraphs of GRN models. From the "DREAM 3 In Silico Network" challenge [22], we use the 5 subgraphs with $p = 50$ nodes. Here, we present the results for "Yeast1", which is the graph requiring the most interventions for our methods to orient. Results for the other graphs are in the supplementary material.

We compare the same algorithms as considered in the other infinite sample experiments. In Figure 2 f) we record the proportion of unknown edges that were oriented by each algorithm. Our methods and RAND all have intervention sizes of 5. For each method we perform 5 repeats. On Yeast1, our methods all perform similarly and outperform RAND and GREEDY. This is found for the other DREAM 3 graphs too, except on one subgraph where GREEDY performs similarly to our methods.

6 Conclusions

We presented near-optimal algorithms for causal structure learning through multi-perturbation interventions. Our results make novel use of submodularity properties and separating systems to search over a doubly exponential domain. Empirically, we demonstrated that these algorithms yield significant improvements over random interventions and state-of-the-art single-perturbation algorithms. These methods are particularly relevant in genomics applications, where causal graphs are large but multiple genes can be intervened upon simultaneously.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We present and discuss adversarial cases for some of our algorithms in Section 5 and discuss how our theoretical setup simplifies the more general setting in Section 2.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] In the first section of the supplementary material.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] In Section 2 and Section 4.
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Attached as a zip in the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In Section 5 and in the supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] In the figures in Section 4 and supplementary material.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] In the supplementary material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] Dream3, the source of some networks used in our experiments, is cited. Python packages used are included in the environment.yml file of the code.
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

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

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

References

[1] Jayadev Acharya et al. "Learning and Testing Causal Models with Interventions". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper/2018/file/78631a4bb5303be54fa1cfdcb958c00a-Paper.pdf.

[2] Britt Adamson et al. "A multiplexed single-cell CRISPR screening platform enables systematic dissection of the unfolded protein response". In: *Cell* 167.7 (2016), pp. 1867–1882.

[3] Raj Agrawal et al. "ABCD-strategy: Budgeted experimental design for targeted causal structure discovery". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. 2019, pp. 3400–3409.

[4] Steen A Andersson, David Madigan, Michael D Perlman, et al. "A characterization of Markov equivalence classes for acyclic digraphs". In: *The Annals of Statistics* 25.2 (1997), pp. 505–541.

[5] Gruia Calinescu et al. "Maximizing a monotone submodular function subject to a matroid constraint". In: *SIAM Journal on Computing* 40.6 (2011), pp. 1740–1766.

[6] Atray Dixit et al. "Perturb-Seq: dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens". In: *Cell* 167.7 (2016), pp. 1853–1866.

[7] F. Eberhardt and R. Scheines. "Interventions and causal inference". In: *Philosophy of Science* 74.5 (2007), pp. 981–995.

[8] Frederick Eberhardt, Clark Glymour, and Richard Scheines. "On the number of experiments sufficient and in the worst case necessary to identify all causal relations among N variables". In: *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*. 2005, pp. 178–184.

[9] Juan L Gamella and Christina Heinze-Deml. "Active Invariant Causal Prediction: Experiment Selection through Stability". In: *arXiv preprint arXiv:2006.05690* (2020).

[10] Timothy S Gardner et al. "Inferring genetic networks and identifying compound mode of action via expression profiling". In: *Science* 301.5629 (2003), pp. 102–105.

[11] AmirEmad Ghassami, Saber Salehkaleybar, and Negar Kiyavash. "Interventional Experiment Design for Causal Structure Learning". In: *arXiv preprint arXiv:1910.05651* (2019).

[12] AmirEmad Ghassami et al. "Budgeted experiment design for causal structure learning". In: *International Conference on Machine Learning*. 2018, pp. 1724–1733.

[13] Hamed Hassani et al. "Stochastic conditional gradient++". In: *arXiv preprint arXiv:1902.06992* (2019).

[14] Alain Hauser and Peter Bühlmann. "Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs". In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 2409–2464.

[15] Yangbo He, Jinzhu Jia, and Bin Yu. "Counting and exploring sizes of Markov equivalence classes of directed acyclic graphs". In: *The Journal of Machine Learning Research* 16.1 (2015), pp. 2589–2609.

[16] Antti Hyttinen, Frederick Eberhardt, and Patrik O Hoyer. "Experiment selection for causal discovery". In: *The Journal of Machine Learning Research* 14.1 (2013), pp. 3041–3071.

[17] Seiya Imoto et al. "Analysis of gene networks for drug target discovery and validation". In: *Target Discovery and Validation Reviews and Protocols*. Springer, 2007, pp. 33–56.

[18] Richard M Karp. "Reducibility among combinatorial problems". In: *Complexity of computer computations*. Springer, 1972, pp. 85–103.

[19] Murat Kocaoglu, Alex Dimakis, and Sriram Vishwanath. "Cost-Optimal Learning of Causal Graphs". In: *International Conference on Machine Learning*. 2017, pp. 1875–1884.

[20] Andreas Krause and Daniel Golovin. "Submodular Function Maximization". In: *Tractability: Practical Approaches to Hard Problems*. Cambridge University Press, 2014, pp. 71–104.

[21] Erik Lindgren et al. "Experimental design for cost-aware learning of causal graphs". In: *Advances in Neural Information Processing Systems*. 2018, pp. 5279–5289.

[22] Daniel Marbach et al. "Generating realistic in silico gene networks for performance assessment of reverse engineering methods". In: *Journal of computational biology* 16.2 (2009), pp. 229–239.

[23] Florian Markowetz, Steffen Grossmann, and Rainer Spang. "Probabilistic soft interventions in conditional Gaussian networks". In: *Tenth International Workshop on Artificial Intelligence and Statistics*. Society for Artificial Intelligence and Statistics. 2005, pp. 214–221.

[24] Christopher Meek. "Casual Inference and Causal Explanation with Background Knowledge". In: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann. 1995, pp. 403–410.

[25] Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. "Stochastic conditional gradient methods: From convex minimization to submodular maximization". In: *Journal of Machine Learning Research* 21.105 (2020), pp. 1–49.

[26] Joris M. Mooij, Sara Magliacane, and Tom Claassen. "Joint causal inference from multiple contexts". In: *Journal of Machine Learning Research* 21 (2020), pp. 1–108.

[27] Christos H Papadimitriou and Mihalis Yannakakis. "Optimization, approximation, and complexity classes". In: *Journal of computer and system sciences* 43.3 (1991), pp. 425–440.

[28] Judea Pearl. *Causality*. Cambridge university press, 2009.

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

[30] Karthikeyan Shanmugam et al. "Learning causal graphs with small interventions". In: *Advances in Neural Information Processing Systems* 28 (2015), pp. 3195–3203.

[31] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, prediction, and search*. Lecture Notes in Statistics. New York: Springer, 1993. DOI: `10.1007/978-1-4612-2748-9`. URL: `https://cds.cern.ch/record/1607850`.

[32] Chandler Squires, Yuhao Wang, and Caroline Uhler. "Permutation-based causal structure learning with unknown intervention targets". In: *The Thirty-Sixth Conference on Uncertainty in Artificial Intelligence*. 2020.

[33] Thomas Verma and Judea Pearl. "Equivalence and synthesis of causal models". In: *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*. Elsevierence Inc. 1990, pp. 255–270.

[34] Yuhao Wang et al. "Permutation-based causal inference algorithms with interventions". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5822–5831.

[35] Ingo Wegener. "On separating systems whose elements are sets of at most k elements". In: *Discrete Mathematics* 28.2 (1979), pp. 219–222.

[36] Dominic JA Welsh and Martin B Powell. "An upper bound for the chromatic number of a graph and its application to timetabling problems". In: *The Computer Journal* 10.1 (1967), pp. 85–86.

[37] Marcel Wienöbst, Max Bannach, and Maciej Liśkiewicz. *Polynomial-Time Algorithms for Counting and Sampling Markov Equivalent DAGs*. 2020. arXiv: `2012.09679 [cs.LG]`.

[38] Karren Yang, Abigail Katcoff, and Caroline Uhler. "Characterizing and Learning Equivalence Classes of Causal DAGs under Interventions". In: *International Conference on Machine Learning*. 2018, pp. 5541–5550.

Appendix

A.1 Potential Negative Societal Impacts

We propose algorithms for experimental design when learning causal structures. The most obvious application is in designing scientific experiments to learn gene regulatory networks. It is possible that the ability to gather increasingly detailed information about gene networks could be used by malicious actors. However, to the authors' knowledge there are no examples of such use with existing related methods. On the other hand, there is longstanding scientific interest in learning such networks and potentially beneficial applications in drug design [17].

A.2 The Power of Multi-Perturbation Interventions

We can construct a more general example than the one given in Figure 1, to demonstrate that the optimal q-node intervention can learn up to q-times more edges than the optimal single node intervention. Consider a graph with MEC that is a forest of undirected star graphs, each with an equal number of nodes. The optimal single-node intervention can intervene on the center node in one of these stars, and entirely orient that star. The optimal q-node intervention can achieve this for q of the stars.

This example illustrates the upper bound on the number of additional edges that can be oriented by the optimal q-node intervention compared to the optimal single node intervention. It can be seen that this is a tight upper bound from the submodularity of F_{EO}^{ξ} in lemma 2.

A.3 Deriving F_∞

Agrawal et al. [3] derive this infinite sample objective for the case of having gathered only infinite observational data before doing experiments. For the case of having infinite samples from observational data and some interventions, we follow a similar argument.

For evaluating $U_{M.I}(y, \xi, D) = H(G \mid D) - H(G \mid D, y, \xi)$ where H is the entropy, we only consider the second term. The first term does not depend on the interventions we do so is irrelevant for optimizing the objective.

Given D consisting of existing interventions $\xi^{'}$ and observational data, the true DAG is already recovered up to it's ξ'-MEC. After obtaining infinite samples from each intervention in intervention set ξ, we recover the true DAG up to its $\xi' \cup \xi$-MEC. Therefore $H(G \mid D, y, \xi) = \log_2(|Ess^{\xi \cup \xi'}(G)|)$ when the true DAG is G. $Ess^{\xi \cup \xi'}(G)$ is the essential graph obtained after interventions ξ. Our prior distribution over DAGs is uniform over the ξ'-MEC of the true DAG. Averaging over these possibilities, the final objective is given by Objective 2.

A.4 Background on the Meek Rules

After performing interventions, the Meek rules can be used to orient additional edges. The rules are given in Figure 3. The Meek rules are continually applied until none of the left side patterns appear in the partially directed graph.

A.5 F_{EO} is consistent

We use the same definition of *budgeted batch consistency* introduced in Agrawal et al. [3].

Definition 4 (Agrawal et al. [3])**.** Assume our goal is to identify the true DAG G^*. Let us have constraints for the bth batch ($0 \leq b \leq B$) of experiments $C_{m,q}^{b}$ ($q \geq 1$). Objective F is budgeted batch consistent if maximizing it in every batch implies

$$\mathbb{P}(G \mid D_B) \to \mathbb{1}(G = G^*)$$

asymptotically as $m, B \to \infty$, where D_B is the combined data obtained from all batches of experiments and the original dataset.

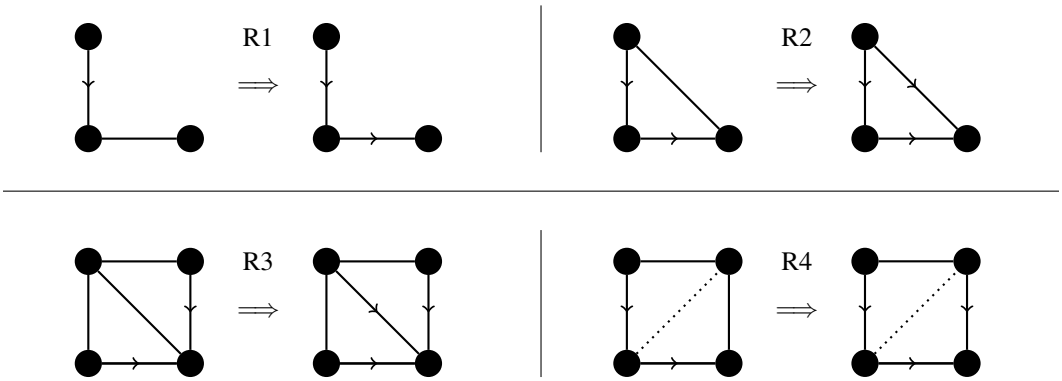

Figure 3: The 4 Meek rules. When a pattern on the left of the implication occurs, edges are oriented according to the pattern on the right of the implication. A dashed line means the direction of the edge may or may not be already identified. The name of each Meek rule is given above the implication sign. R0 refers to the step of orienting edges before Meek rules are applied.

We show that our definition of F_{EO} satisfies budgeted batch consistency. Then we explain how the slight difference in the definition of F_{EO} given in Agrawal et al. [3] leads to the authors concluding that it is not budgeted batch consistent.

Proposition 2. *F_{EO} is budgeted batch consistent.*

Proof. Since each intervention offers an infinite number of samples, we can reason directly about the subsequent orienting of edges due to the obtained samples. Let the set of interventions after b batches be ξ_b. We simply need to prove that as $k \to \infty$, ξ_b will identify the orientation of every edge in the true DAG. Equivalently, we show that if ξ_b has not oriented every edge (the current essential graph has size greater than 1), ξ_{b+1} will orient additional edges in the true DAG. There are only a finite number of edges to be oriented, so this means ξ_∞ will fully identify the true DAG.

If ξ_b does not orient every edge in the true DAG, Objective 3 will have a maximum in the next batch of greater than 0. Moreover, to obtain objective greater than 0, the interventions selected in batch $b+1$ must orient edges not oriented by ξ_b. This is because after each batch, the objective is updated to be with reference to the essential graph of G under interventions in ξ_b. To orient unidentified edges in any of the possible DAGs (those in the current ξ_b-MEC), we must orient at least one edge in the true DAG after obtaining samples. This can be seen from the first step of computing function R. One can also see that if an edge can be oriented, it can always be oriented by selecting a single unique intervention of size 1 and thus the constraints $C_{m,q}$ can always be satisfied. □

The key difference between this argument and the one given by Agrawal et al. [3] is that they maintain a static objective function between batches, so the same set of interventions is selected every round.

A.6 Proof of Lemma 1

We'll work with the notation of $\xi_1 \subset \xi_2$ are sets of interventions, and I is an intervention. We'll write $R(\xi, G) = M(A(\xi, G), G)$. Here, A carries out the first step of orienting edges based on one of the nodes in that edge being intervened on (we refer to this as R0). M carries out the Meek rules given the orientations of edges in A. Note that R, M, A implicitly depend on the interventions carried out in previous batches ξ', since this determines what edges might already be oriented in G (whether we are orienting the MEC or some essential graph).

We'll first show that function R has a monotonicity-like property: adding an intervention only adds to the set of oriented edges.

Proposition 3. *Monotonicity-like property of R: $R(\xi_1, G) \subseteq R(\xi_2, G)$ for all G.*

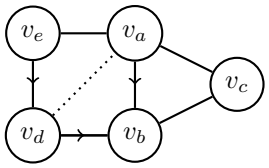

Figure 4: When an edge $v_a \to v_b$ is identified by R4

Proof. The same argument is given in Ghassami et al. [12]. By the definition of A, $A(\xi_1, G) \subseteq A(\xi_2, G)$. The Meek rules are sound and order-independent [24], and therefore $M(A(\xi_1, G)) \subseteq M(A(\xi_2, G))$ □

From this we can see that F_{EO} is also monotonic.

To prove lemma 1, we swill first prove some propositions regarding the marginal change in R when adding a new intervention.

Proposition 4. *Consider vertices v_a, v_b, v_c. Consider $\bar{G}$, the partially directed graph obtained after doing intervention set ξ and then applying the Meek rules exhaustively. If $v_a \to v_b \in \bar{G}$, and $v_b - v_c \in \bar{G}$, then $v_a \to v_c \in \bar{G}$.*

Proof. If $v_a \to v_b$, and $V_b - V_c$, we must have that v_a and v_c are adjacent, else $v_b \to v_c$ by R1. We cannot have $v_c \to v_a$ since this would identify $v_a - v_b$ by R2. Hence we have either that $v_a - v_c$ or $v_a \to v_c$.

Suppose for contradiction that, after applying all Meek rules, for some nodes v_a, v_b, v_c we have $v_a \to v_b$, $v_b - v_c$ and $v_a - v_c$. We will gain a contradiction by an infinite descent. Any DAG can be associated with some permutation of its nodes that specifies a topological ordering, with the lowest ranked node being the root. Suppose that B is the lowest ranked node in the topological ordering given by the true DAG (closest to the root) such that this pattern holds. For all ways in which $v_a \to v_b$ could have been identified, we will show that either in fact $v_a \to v_c$ or find a vertex lower than v_b in the topological ordering that fits into an identical pattern.

Some cases are covered in Meek [24] when proving a similar result (lemma 1).

Suppose $v_a \to v_b$ is known by being a collider (identified before any interventions take place). This is handled in Meek [24].

Suppose $v_a \to v_b$ is known by R1, R2, R3. These cases are all handled by Meek [24].

Suppose $v_a \to v_b$ is learnt directly by an intervention (rule R0). If it is identfied in this way, there must exist an intervention I such that exactly one of $v_a, v_b \in I$. However, in either case regardless of whether $v_c \in I$ or not, we identify one of $v_a - v_c$ or $v_b - v_c$ by R0. Hence the pattern cannot occur if $v_a \to v_b$ is identified by R0.

Suppose $v_a \to v_b$ is learnt directly by R4 as in Figure 4. Consider the pattern given in Figure 4 where v_e and v_d are the other edges in the R4 pattern. Now if $v_a - v_d$ is undirected, then AED is the same pattern as v_a, v_b, v_c but with $v_d < v_b$ in the topological ordering (a contradiction). If $v_a \to v_d$, then v_a, v_d, v_b gives the same setup as if we discovered $v_a \to v_b$ through R2. Similarly, if $v_d \to v_a$, then $v_e \to v_a$ by R2 and then we have the same setup as if we discovered $v_a \to v_b$ by R1. An alternative R4 pattern can also orient $v_a \to v_b$, however it involves node v_c being part of the pattern. In this case, we must have oriented $v_c \to v_b$, a contradiction. □

Proposition 4 allows us to prove two propositions more directly related to our final result.

Proposition 5. $R(\xi_2 \cup \{I\}) \setminus R(\xi_2) \subseteq R(\xi_1 \cup \{I\}) \setminus R(\xi_1)$

Proof. Here we'll take R to also include all edges oriented before the interventions, since this doesn't change the outcome of the set difference operation above. We also drop G from the notation since we work with a fixed true graph. This is just done out of convenience for the proof.

Take edge $e \in R(\xi_2 \cup \{I\})$ and $e \notin R(\xi_2)$. By the monotonicity-like property, $R(\xi_1) \subseteq R(\xi_2)$, so $e \notin R(\xi_1)$. Thus we just need to prove that for all such e, we have $e \in R(\xi_1 \cup \{I\})$.

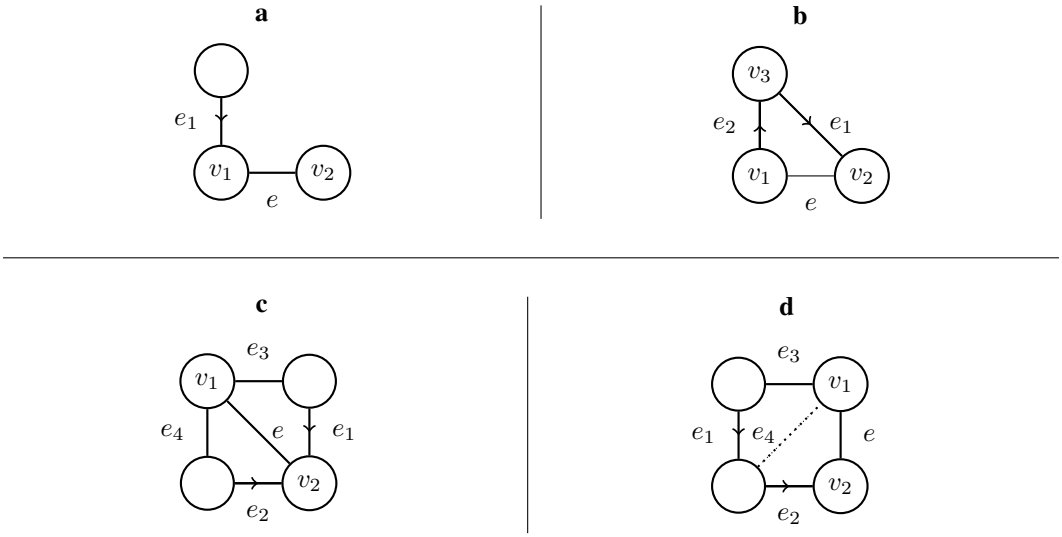

Figure 5: The Meek rules with labels on edges and nodes. **a)** R1, **b)** R2, **c)** R3, **d)** R4.

Assume for contradiction there is some nonempty set $E^\dagger$ of edges such that $\forall e \in E^\dagger$, $e \in R(\xi_2 \cup \{I\})$ and $e \notin R(\xi_2)$, but $e \notin R(\xi_1 \cup \{I\})$. We can specify an ordering over these edges. Order the edges (written $v_1 \to v_2$) such that: the edges are in increasing order upon the position of v_2 in the topological ordering of graph G (lower is closer to the root). Settle all ties by decreasing order on the position of v_1 in the topological ordering. We will now show that if there exists some e that is the lowest ordered element in $E^\dagger$, we can always either create a contradiction or find some alternative edge in $E^\dagger$ with lower position in the ordering (itself a contradiction).

If $e = v_1 \to v_2$ is discovered in $R(\xi_2 \cup \{I\})$ by R0, we know I must intervene on one of v_1, or v_2 but not the other. Hence $e \in R(\xi_1 \cup \{I\})$.

The diagram in Figure A.6 is given for following the other cases. Notation refers to the names of nodes and edges given on the diagrams.

If $e = v_1 \to v_2$ is discovered in $R(\xi_2 \cup \{I\})$ by R1, we must have $e_1 \in R(\xi_2 \cup \{I\})$. We must also have that $e_1 \notin R(\xi_2)$ else $e \in R(\xi_2)$ by R1. The same must be true of e' not being in $R(\xi_1 \cup \{I\})$, so $e_1 \in E^\dagger$. But e_1 will have lower ordering since v_1 is below v_2 in the topological ordering.

If $e = v_1 \to v_2$ is discovered in $R(\xi_2 \cup \{I\})$ by R2, we must have some pair $e_1, e_2 \in R(\xi_2 \cup \{I\})$ such that $e_1 = v_3 \to v_2$ and $e_2 = v_1 \to v_3$. One of these edges cannot be in $R(\xi_2)$. In fact, neither can be in $R(\xi_2)$ due to proposition 4. This is because assuming only one is identified, based on proposition 4, another edge is either identified which leads to orienting e or another edge is incorrectly oriented in the true DAG. The same holds for $R(\xi_1 \cup \{I\})$, but then we've found an edge $e_2 \in E^\dagger$ that is lower in the ordering than e.

If $e = v_1 \to v_2$ is discovered in $R(\xi_2 \cup \{I\})$ by R3, we must have $e_1, e_2 \in R(\xi_2 \cup \{I\})$ and $e_3, e_4 \in G$ but not necessarily in $R(\xi_2 \cup \{I\})$. e_1 and e_2 form an unshielded collider and are identified before intervening, so $e \in R(\xi_1 \cup \{I\})$ by R3.

If $e = v_1 \to v_2$ is discovered in $R(\xi_2 \cup \{I\})$ by R4, we must have $e_1, e_2 \in R(\xi_2 \cup \{I\})$ and $e_3, e_4 \in G$ but not necessarily in $R(\xi_2 \cup \{I\})$. At least one of $e_1, e_2 \notin R(\xi_2)$. Suppose only e_1 is in, then e_2 is in by $R1$. Therefore e_1 is not in $R(\xi_2)$ or $R(\xi_1 \cup \{I\})$, but this is lower in the ordering than e. □

Proposition 6. $R(\xi_1) \setminus R(\xi_1 \cup \{I\}) \subseteq R(\xi_2) \setminus R(\xi_2 \cup \{I\})$

Proof. Follows by monotonicity of R, both sides are the empty set. □

We can rewrite $F_{EO} = \sum_{G \in \mathcal{G}} g(R(\xi, G))$ where g is the weighted coverage function. g is a monotonic function of the set of edges in R.

$$\begin{aligned} g(R(\xi_1 \cup \{I\})) - g(R(\xi_1)) &\overset{\text{(i)}}{=} g(R(\xi_1 \cup \{I\}) \setminus R(\xi_1)) \\ &\quad - g(R(\xi_1) \setminus R(\xi_1 \cup \{I\})) \\ &\overset{\text{(ii)}}{\geq} g(R(\xi_2 \cup \{I\}) \setminus R(\xi_2)) \\ &\quad - g(R(\xi_2) \setminus R(\xi_2 \cup \{I\})) \\ &= g(R(\xi_2 \cup \{I\})) - g(R(\xi_2)). \end{aligned}$$

Step (i) is a property of the weighted coverage function. Step (ii) comes from propositions 5 and 6 and the monotonicity of g. This shows that definition 2 holds for $g(R(\xi, G))$ as a function of ξ for all G. Since the sum of submodular functions is submodular, this implies lemma 1.

A.7 Proof of Lemma 2

We can see that a monotonicity property like proposition 3 does not hold in this case. The intervention $[p]$, for example, orients no edges. Nevertheless, we follow the same approach to prove submodularity of F_{EO}^{ξ}. For this we consider $I_1 \subset I_2$ and consider adding node $v \notin I_2$ to these interventions.

For notational simplicity, $R(I, G)$ will now denote all edges oriented after intervention set ξ (fixed) and intervention I on true graph G. We will drop G from the notation in cases where the graph is fixed.

Proposition 7. $R(I_2 \cup \{v\}) \setminus R(I_2) \subseteq R(I_1 \cup \{v\}) \setminus R(I_1)$

Proof. We want to show two things. First we want to show that if $e \in R(I_2 \cup \{v\})$ and $e \notin R(I_2)$, then $e \in R(I_1 \cup \{v\})$. This is shown with an identical technique to the one in proposition 5. This is because the Meek rules are the same in both cases. The only difference is the case when $e = v_1 \to v_2$ is discovered by R0. In this case, v must be either v_1 or v_2 and neither of v_1 or v_2 can be in I_2. However therefore neither are in I_1 and hence $e \in R(I_1 \cup \{v\})$.

The second thing we need to show is that if $e \in R(I_2 \cup \{v\})$ and $e \notin R(I_2)$, then $e \notin R(I_1)$. Suppose for contradiction that there exists some such $e \in R(I_1)$. We'll proceed in two steps. First we'll show that in order to avoid a contradiction, we must have that I_2 intervenes on both vertices in e. Second we'll show that if we intervene on both vertices in e for I_2, we cannot have that $e \in R(I_2 \cup \{v\})$ and $e \notin R(I_2)$.

We can represent the identification of edge e in $R(I_1)$ by a directed tree diagram. The root in the diagram is e, and the children of each node in the diagram are the directed edges involved in the Meek rule that identifies the parent edge. Each node can have either one or two children (since each Meek Rule depends on up to two specific edges being directed). Leaf nodes must have been identified by R0. Since $e \in R(I_1)$ and $e \notin R(I_2)$, there must be leaf nodes in the diagram that are not identified by R0 using intervention I_2. Since $I_1 \subset I_2$, this means that for these leaf edges, I_2 must contain both nodes. However we now show that, for the structure of all Meek rules, in our tree diagram if a child edge has both vertices intervened on then the parent is identified anyway, unless both vertices of the parent are intervened on. This is shown pictorially in Figure 6. R4 case 1 requires some extra explanation. e_1 is identified and must be oriented towards the node in e to prevent identifying e by R2. Hence e_2 must direct into e to prevent a cycle. Then e is learnt by R1. Given all this, we can see that to prevent identification of e by I_2, there must be a path from a leaf edge to the root e where all the edges in the path have both vertices in the edge intervened on in I_2. Hence e has both vertices contained in I_2.

Now consider a different tree diagram for how e is oriented in $R(I_2 \cup \{v\})$. Clearly v must be included as a vertex in at least one of the leaf edges, else the same pattern could allow us to orient e using I_2. We show that if both vertices of an edge are intervened on, to prevent the identification of both child edges and hence the edge itself by the Meek rules, it must be that both children have both of their vertices intervened on in I_2. This is shown in Figure 7.

In Figure 7, **R1**, **R2** and **R3** are clear. For **R4 case 1**, e_2 must orient right to left (or get e by R1). Also, e_1 must go towards the top right or we get e by R2. Thus e_3 must point upwards by R2 which

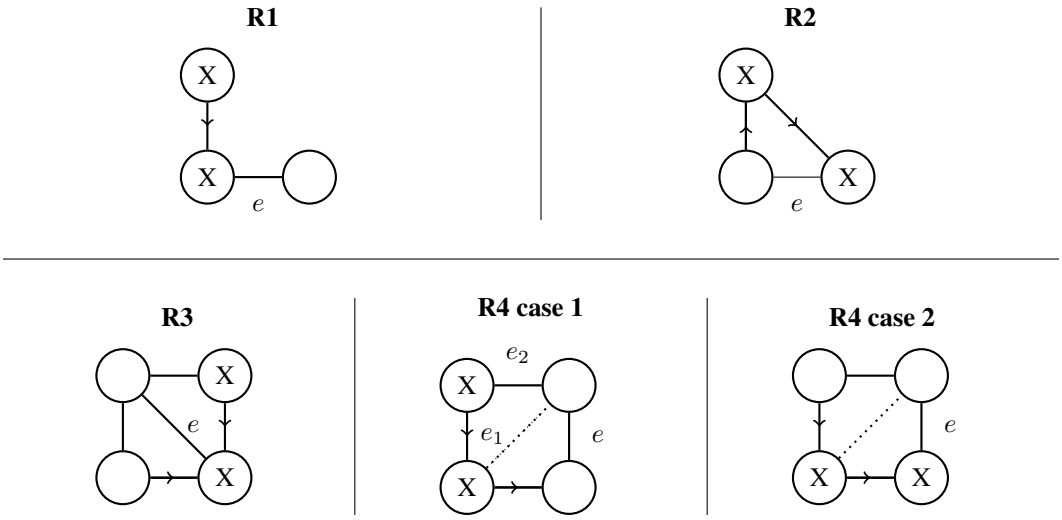

Figure 6: The pattern of each Meek rule as in the first part of proposition 7. An X denotes that an intervention occured at that node. In our tree representation, if a child edge has both vertices intervened on then the parent is identified anyway, unless both vertices of the parent are intervened on.

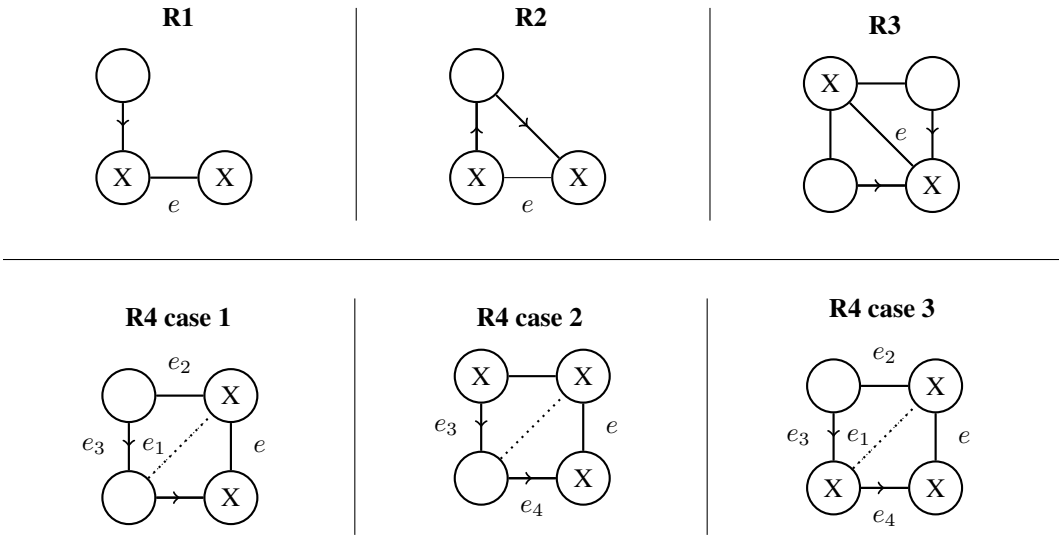

Figure 7: The pattern of each Meek rule as in the second part of the proof of proposition 7. An X denotes that an intervention occured at that node. In our tree diagrams, if both vertices of an edge are intervened on, to prevent the identification of both child edges (and hence the edge itself by the corresponding Meek rule), it must be that both children have both of their vertices intervened on.

is a contradiction, since we know in the R4 pattern e_3 points downwards. Intervening on more edges to avoid this leads us to cases 2 and 3. For **R4 case 2**, we learn e_3 and e_4 and hence orient e. For **R4 case 3**, e_2 must point left else we get e by R1. We know e_3 points down and it is oriented by R0. Then, e_4 goes left to right by R1. E_1 then points towards the bottom left by R_2 and hence e is oriented by R2.

Hence, there must be some path from the root e to a leaf edge such that all members of the path have both of their vertices intervened on in I_2 and hence $I_2 \cup \{v\}$. If $v \in e$, we have a contradiction since $v \notin I_2$. If $v \notin e$, then the leaf edge in our tree representation containing v is not identified using R0

with $I_2 + v$ either. Hence, this tree cannot possibly represent the sequence of Meek rules that lead to orienting e. Thus if $e \in R(I_2 \cup \{v\})$ and $e \notin R(I_2)$, then $e \notin R(I_1)$. □

Proposition 8. $R(I_1) \setminus R(I_1 \cup \{v\}) \subseteq R(I_2) \setminus R(I_2 \cup \{v\})$

Proof. As in the proof of lemma 1, we can write $R(I) = M(A(I))$, where A returns edges oriented directly by the intervention and M returns these in addition to any oriented due to Meek rules.

By symmetry in the definition of A, we can see that $A(I^C) = A(I)$ and hence $R(I^C) = R(I)$. Take some edge $e \in R(I_1) \setminus R(I_1 \cup \{v\})$. Then by symmetry we have $e \in R(I_1^C) \setminus R(I_1^C \setminus \{v\})$. Then since $I_1^C \setminus \{v\} \supseteq I_2 - \{v\}$, by proposition 7 we must have $e \in R(I_2^C) \setminus R(I_2^C \setminus \{v\})$. Again by symmetry we then have $e \in R(I_2) \setminus R(I_2 \cup \{v\})$. □

We can conclude that F_{EO}^{ξ} is submodular in an identical way to how we did in proving lemma 1: by combining propositions 7 and 8.

A.8 Proof of Theorem 2

At each iteration of selecting an intervention, Theorem 1 lower bounds how close the marginal gain compared to the greedy intervention is. Let ξ be the set of interventions DGC selects. Let ξ^* be the optimal batch of interventions, and ${I^*}_i$ be the ith intervention in this set. Due to lemma 1 (monotonicity), ξ^* contains exactly m interventions. $\{\xi_i\}_{i \geq 0}$ is the entire intervention set after each greedy selection. Define marginal improvement $\Delta(I|\xi) = F_{\text{EO}}(I \cup \xi) - F_{\text{EO}}(\xi)$. The following holds for all i:

$$
\begin{aligned}
F_{\text{EO}}(\xi^*) &\leq F_{\text{EO}}(\xi^* \cup \xi_i) \\
&\leq F_{\text{EO}}(\xi_i) + \sum_{j=1}^{k} \Delta({I^*}_j \mid \xi_i \cup \{{I^*}_1, ..., {I^*}_{j-1}\}) \\
&\leq F_{\text{EO}}(\xi_i) + \sum_{I \in \xi^*} \Delta(I \mid \xi_i) \\
&\leq F_{\text{EO}}(\xi_i) + \sum_{I \in \xi^*} e\,\mathbb{E}\left[F_{\text{EO}}(\xi_{i+1}) - F_{\text{EO}}(\xi_i)\right] + e\epsilon \\
&\leq F_{\text{EO}}(\xi_i) + em\,\mathbb{E}\left[F_{\text{EO}}(\xi_{i+1}) - F_{\text{EO}}(\xi_i)\right] + em\epsilon
\end{aligned}
$$

The first line is due to monotonicity. The second is a telescoping sum. The third is due to submodularity (lemma 1). The fourth is a result of Theorem 1, since what we write is lower bounded by the greedy choice, which by definition has greater marginal improvement than any other intervention. The expectation here is over noise in selecting the $(i+1)$th intervention. The final line just notes that there are m elements in the sum.

Now define $\delta_i = F_{\text{EO}}(\xi^*) - F_{\text{EO}}(\xi_i)$. We rearrange the above to get

$$\delta_i \leq em(\delta_i - \mathbb{E}[\delta_{i+1}] + \epsilon),$$

where again the expectation is over selection of the $(i+1)$th intervention.

Now we telescope this inequality, subbing in $i + 1 = m$ to obtain our final result.

$$
\begin{aligned}
\mathbb{E}[\delta_m] &\leq \left(1 - \frac{1}{em}\right)\delta_{m-1} + \epsilon \\
&\leq \left(1 - \frac{1}{em}\right)^m \delta_0 + \sum_{j=0}^{m} \left(1 - \frac{1}{em}\right)^j \epsilon \\
&\leq e^{-1/e}\delta_0 + \sum_{j=0}^{m} e^{-\frac{j}{em}}\epsilon
\end{aligned}
$$

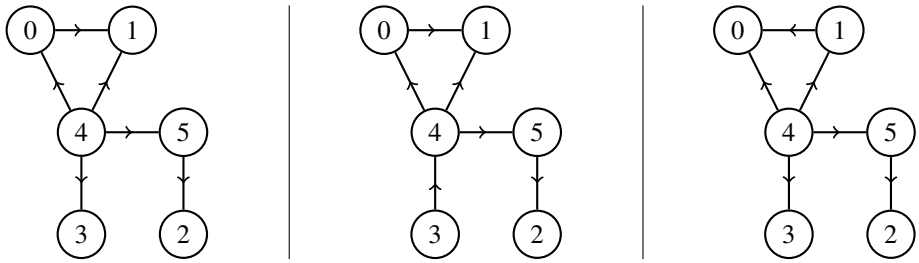

Figure 8: The 3 DAGS in $\tilde{\mathcal{G}}$ for the counterexample in proving Proposition 1.

The third line uses $1 - x \leq e^{-x}$. The final term on the right hand side of the last line reduces to $\sigma\epsilon$, where

$$\sigma = \frac{e^{-1/e} - e^{1/(em)}}{1 - e^{1/(em)}}.$$

By noting that $\delta_0 = F_{\text{EO}}(\xi^*)$, we get

$$\mathbb{E}[F_{\text{EO}}(\xi)] \geq \left(1 - e^{-\frac{1}{e}}\right) F_{\text{EO}}(\xi^*) - \epsilon\sigma.$$

Given Theorem 1 and that we select m greedy interventions, this requires $\mathcal{O}\left(mp^{5/2}/\epsilon^3\right)$ calls to the Meek rules. We remove the dependence on m in σ into the runtime by noting that $\sigma = \mathcal{O}(m)$ and hence that if we do $\mathcal{O}\left(m^4p^{5/2}/\epsilon^3\right)$ calls to the Meek rules, we get

$$\mathbb{E}[F_{\text{EO}}(\xi)] \geq \left(1 - e^{-\frac{1}{e}}\right) F_{\text{EO}}(\xi^*) - \epsilon.$$

A.9 Proof of Lemma 3

Agrawal et al. [3] prove monotone submodularity of F_{MI}. $\tilde{F}_\infty$ is the special case of this objective in the limit of infinite samples per intervention.

A.10 Proof of Proposition 1

To show that $\tilde{F}_\infty^\xi$ is in general not submodular, we need to give a specific example of a constraint set, and set of DAGs $\tilde{\mathcal{G}}$. We let $m = 1, k = 4$. The set $\tilde{\mathcal{G}}$ is given in Figure 8. We consider the set of interventions before carrying out experiments, ξ' to be the empty set. To break the definition of submodularity as given in definition 2 we need to define an existing intervention to add, I_2, and a subset of this, I_1. For this example we choose $I_2 = [1, 2, 3]$ with nodes numbered as in Figure 8. $I_1 = [1, 2]$. We also need to choose a perturbation to add to each intervention, and we choose node 0.

The computation of the objective is carried out in *proposition.py* in the accompanying code.

Note that for the special case $\tilde{\mathcal{G}} = \mathcal{G}$, we have not constructed a counterexample. In fact, in this case we suspect that $\tilde{F}_\infty^\xi$ is submodular, but don't have a proof. If true, this may suggest that an algorithm similar to DGC could be used to maximize Objective 2 directly without approximating it. An additional difficulty for this objective, however, is the second potentially exponential sum required to compute essential graph sizes embedded within the logarithm.

A.11 Proof of Theorem 3

We know that $\tilde{F}_\infty(\mathcal{S}) = 0$, because the graph is then fully identified, meaning $|\tilde{\text{Ess}}^{\xi \cup \xi'}(G)|$ is 1 for all G. We also know that $\min(\tilde{F}_\infty) = \tilde{F}_\infty(\emptyset)$. The submodularity of $\tilde{F}_\infty$ over groundset $\mathcal{S} \subset \mathcal{I}$

(lemma 3) along with this boundedness of the function is used to get the final bound.

Say we greedily select m members of $\mathcal{S}$ to construct ξ_m. We prove by induction that $Q(m) = \left(\tilde{F}_\infty(\xi_m) \geq \left(1 - \frac{m}{|\mathcal{S}|}\right)\tilde{F}_\infty(\emptyset)\right)$ is true for all m where $0 \leq m \leq |S|$.

The base case $Q(0)$ is trivial. Now assume that $Q(m)$ is true. Since $\tilde{F}_\infty$ is submodular over groundset $\mathcal{S}$, it satisfies the diminishing returns property of definition 2. Therefore it must be the case that $\exists I \in \mathcal{S} \setminus \xi_m$ such that $\tilde{F}_\infty(\{I\} \cup \xi_m) \geq \frac{1}{|\mathcal{S}|-m}\tilde{F}_\infty(\xi_m)$. Note that if this was not the case, because of submodularity, adding all of the remaining interventions in S in sequence would give $\tilde{F}_\infty(\mathcal{S}) < 0$ which would be a contradiction. Therefore

$$
\begin{aligned}
\tilde{F}_\infty(\xi_{m+1}) &\geq \frac{1}{|\mathcal{S}| - m}\tilde{F}_\infty(\xi_m) \\
&\geq \frac{1}{|\mathcal{S}| - m}\left(1 - \frac{m}{|\mathcal{S}|}\right)\tilde{F}_\infty(\emptyset) \\
&= \left(1 - \frac{m+1}{|\mathcal{S}|}\right)\tilde{F}_\infty(\emptyset)
\end{aligned}
$$

which completes the induction. The final result follows by applying the lower-bound on $|\mathcal{S}|$ given in Shanmugam et al. [30]. A bound for the graph-sensitive separating system in Lindgren et al. [21] can also be obtained by plugging in their lower-bound on $|\mathcal{S}|$.

The runtime can be seen by observing that for each intervention, we compare $|\mathcal{S}| = \mathcal{O}(\frac{p}{q}\log p)$ possible interventions. For each we compute the Meek rules $\mathcal{O}(|\tilde{\mathcal{G}}|)$ times to compute $\tilde{F}_\infty$. Thus, runtime is $\mathcal{O}(m|\tilde{\mathcal{G}}|\frac{p}{q}\log p)$ calls to the Meek rules. The construction of the separating system itself is efficient compared to the computation of the Meek rules [30].

A.12 Extension to Soft Interventions

Our results can also be used to develop algorithms for the soft intervention setting. A hard intervention makes the value of a variable independent of its parents. However, a soft intervention modifies a variable's value whilst maintaining the dependence on its parents. A simple example of a soft intervention is adding a constant value to the intervened node. In a GRN, a soft intervention might correspond to a gene knockdown, where a gene's expression is reduced but not set to 0.

Definition 5 (Soft intervention)**.** A soft intervention I on nodes X_T for all $i \in I$, adds the intervention variables W_i as an extra direct cause of X_i.

The soft intervention setting is greatly simplified by the following result in Ghassami, Salehkaleybar, and Kiyavash [11].

Lemma 4 (Ghassami, Salehkaleybar, and Kiyavash [11])**.** *In the infinite sample setting, a set of k soft interventions of size 1 is equivalent to targeting the same k nodes in a single soft intervention. By equivalent, we mean it identifies the same information regarding the true DAG.*

Therefore, the problem of selecting up to m interventions with size at most q is equivalent to selecting mq single node interventions in the hard intervention case. In Ghassami et al. [12], the authors show that for this setting a greedy algorithm can achieve a constant factor guarantee for the objective F_{EO}, where the constant factor will be $1 - \frac{1}{e}$ instead of $1 - \frac{1}{e^{1/e}}$. An almost identical analysis to that done in Theorem 2 can show that this guarantee is also achieved for optimizing $\tilde{F}_\infty$.

A.13 Further Experiment Details

All algorithms related to SSG make use of lazy evaluation for speed-up, as in Ghassami et al. [12]. In greedy selection, lazy evaluation skips interventions which, based on previous evaluations and

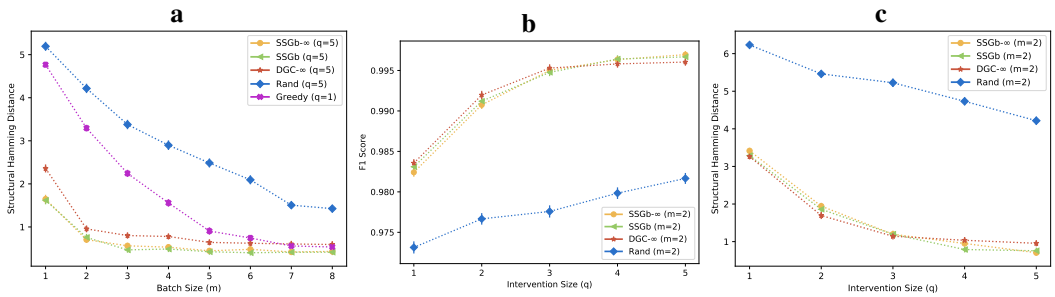

Figure 9: **a)** We measure performance on the finite sample task (same as in Figure 2(e)) with fixed intervention size ($q = 5$) using the SHD metric. The rank ordered performance of methods matches the outcome when we used the F1 metric. **b**, **c)** Here we consider the same experiment as (a) but plot F1 and SHD score respectively for changing intervention size with fixed batch size ($m = 2$). In both cases our algorithms clearly outperform a random approach.

submodularity of the objective, could not be the greedy option. This results in no change in the selected interventions but reduces the number of necessary comparisons.

When constructing graph agnostic separating systems according to the method of Shanmugam et al. [30], we compute them exactly as specified since the algorithm is fast. For the graph sensitive construction method of Lindgren et al. [21], we use approximate algorithms for constructing a minimal vertex covering and a minimal graph-coloring, since both subroutines are NP-hard [18]. For graph-coloring, we use the Welsh-Powell algorithm which is near-optimal for graphs of bounded degree [36]. For vertex cover, we use a 2-approximation algorithm based on greedily finding a maximal matching [27].

For infinite sample experiments, when approximating the objective for use in our algorithms we use a multiset of 40 DAGs uniformly sampled from the MEC with replacement. However, for evaluation we use all DAGs in the MEC.

For DGC, when using Pipage rounding, we round 10 times and select the intervention with greatest approximate objective value.

For finite-sample experiments, the approximate prior over DAGs consists of 100 DAGs uniformly sampled from the MEC of the true DAG, or the MEC itself if the MEC has size less than 100. In the latter case, the rest of the 100 DAGs are given by bootstrapping the observational data and using the techniques of Yang, Katcoff, and Uhler [38] to infer DAGs. In evaluating the finite-sample objective, there is some variance since we observe samples with noise. For evaluating methods on Objective 1 as in Figure 2(d), we average over 10 repeats. For SSG-B, which makes use of the objective for greedily selecting interventions, we approximate the objective by 10 repeats also. In Figure 2(e) we plot F1 scores for predicting the presence of edges in the true DAG. We compute the probability of each directed edge being present by

$$\mathbb{P}(u \to v) = \sum_{G \in \tilde{\mathcal{G}}} \hat{\mathbb{P}}(G) \mathbb{1}((u \to v) \in G)$$

where $\tilde{\mathcal{G}}$ is the set of bootstrapped DAGs and $\hat{\mathbb{P}}$ is the posterior after collecting samples from interventions. A predicted graph is then estimated by thresholding the edge probabilities. We select the threshold that gives the graph with maximum F1 score, and then plot the F1 score.

Similarly, we compute the weighted average (across the posterior) of structural hamming distances (SHDs) between graphs in the set of bootstrapped DAGs and the true DAG. This gives results similar to those of when we plot F1 score, as shown in Figure 9. Methods selecting the most informative interventions will have lower mean SHD because they will decrease the posterior probability of graphs distant from the true DAG.

In Figure 10(a-b) we compare the average runtimes of our algorithms running on the same hardware in a computing cluster. We do this for both the infinite and finite sample case. We see that although the DGC algorithm is slower than alternative approach on infinite samples, in finite samples it has a

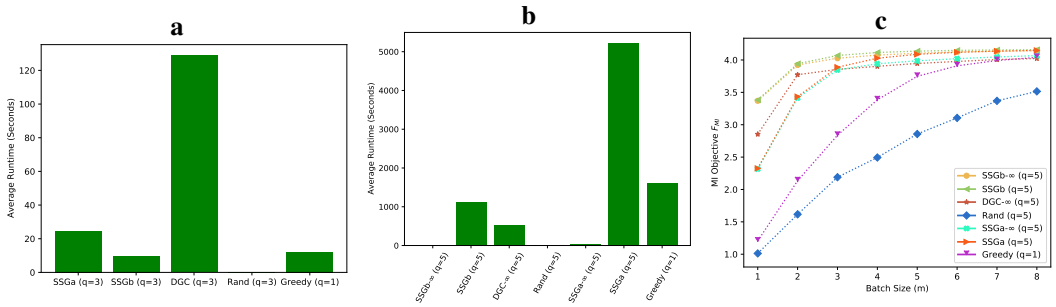

Figure 10: **a)** We demonstrate the runtime for selecting a batch of 5 interventions of size $q = 3$ with each algorithm in the infinite sample setting. This is on ER 0.1 graphs with $p = 40$ nodes. We see that DGC is slower than other approaches but not impractical, whilst out of the SSG methods, the graph-sensitive separating system construction results in faster runtimes. **b)** We show that for selecting 8 interventions of size $q = 5$, the infinite sample approximation methods we give result in substantially improved runtimes compared to methods that use the finite sample objective. This is again on the ER 0.1 graphs but in the finite sample setting. **c)** We replicate Figure 2(d) but include the SSG algorithms based on the graph agnostic separating system construction. This separating system construction does not perform as well as the other algorithms we propose.

faster runtime than approaches that use the true finite sample objective. We also note that SSG-B has faster runtimes than SSG-A, likely because the graph agnostic separating system construction returns larger separating systems. Whilst we aimed to maximize performance in terms of selecting the most informative experiments, these methods can be made faster at the expense of achieving lower objective values. For SSG we could take fewer gradient steps, and for DGC we could use a smaller set of sampled DAGs to approximate the objective.

In Figure 10(c) we include plot performance in finite samples of the graph agnostic separating system for SSG and see that its performance is lower than our alternative approaches.

A.14 Dream 3 Network Experiments

In Figure 11(a–d) we give the results of DREAM3 networks not included in the main paper. For each, we record the number of edges oriented in the true DAG by each method, averaging over 5 repeats. Again, we work in the infinite samples per intervention setting. In Figure 11(e–f) we give illustrations of the ground truth DAG associated with two of the DREAM3 networks (images generated using software from Marbach et al. [22]). We note that these networks differ from typical Erdös-Renyí random graphs in that they have a few nodes with many connections, and many nodes with sparser connections.

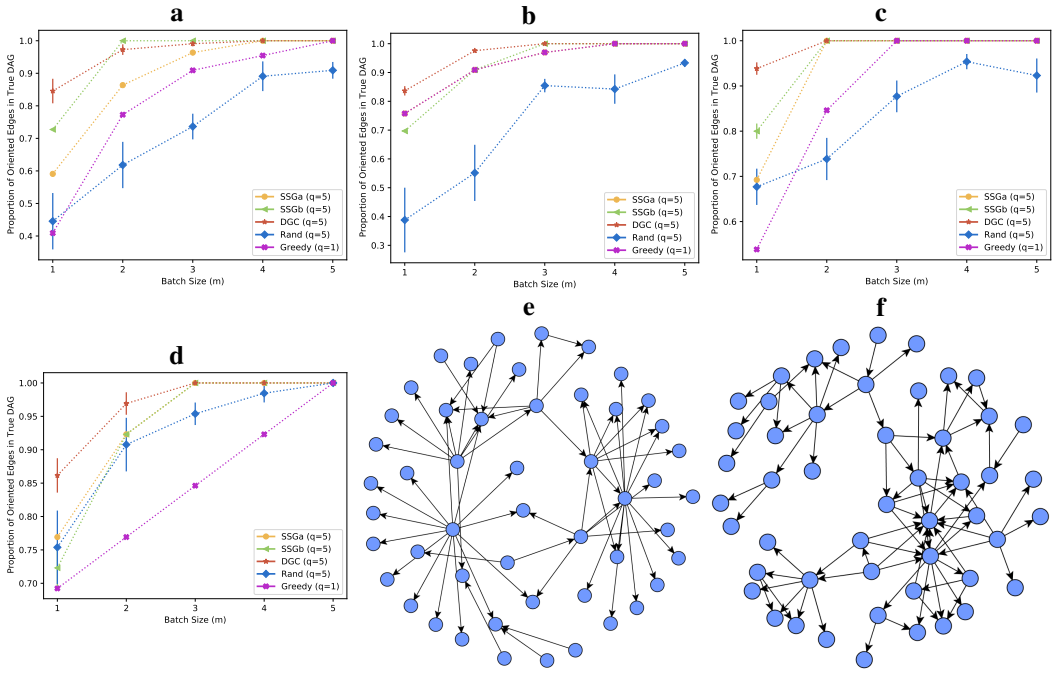

Figure 11: **a–d)** The results for experiments identical to those in Figure 2 for the $p = 50$ networks "Ecoli1", "Ecoli2", "Yeast2" and "Yeast3" from Marbach et al. [22] respectively. For an equal number of interventions and $q = 5$, our methods in general orient more edges than both random and single-perturbation greedy interventions. **e, f)** The ground truth network for "Ecoli1" and "Yeast1" respectively.