Verification and search algorithms for causal DAGs

Davin Choo*
National University of Singapore
davin@u.nus.edu

Kirankumar Shiragur*
Stanford University
shiragur@stanford.edu

Arnab Bhattacharyya
National University of Singapore
arnabb@nus.edu.sg

Abstract

We study two problems related to recovering causal graphs from interventional data: (i) *verification*, where the task is to check if a purported causal graph is correct, and (ii) *search*, where the task is to recover the correct causal graph. For both, we wish to minimize the number of interventions performed. For the first problem, we give a characterization of a minimal sized set of atomic interventions that is necessary and sufficient to check the correctness of a claimed causal graph. Our characterization uses the notion of *covered edges*, which enables us to obtain simple proofs and also easily reason about earlier known results. We also generalize our results to the settings of bounded size interventions and node-dependent interventional costs. For all the above settings, we provide the first known provable algorithms for efficiently computing (near)-optimal verifying sets on general graphs. For the second problem, we give a simple adaptive algorithm based on graph separators that produces an atomic intervention set which fully orients any essential graph while using $\mathcal{O}(\log n)$ times the optimal number of interventions needed to *verify* (verifying size) the underlying DAG on n vertices. This approximation is tight as *any* search algorithm on an essential line graph has worst case approximation ratio of $\Omega(\log n)$ with respect to the verifying size. With bounded size interventions, each of size $\leq k$, our algorithm gives an $\mathcal{O}(\log n \cdot \log k)$ factor approximation. Our result is the first known algorithm that gives a non-trivial approximation guarantee to the verifying size on general unweighted graphs and with bounded size interventions.

1 Introduction

Causal inference has long been an important concept in various fields such as philosophy [Rei56, Woo05, ES07], medicine/biology/genetics [KWJ+04, SC17, RHT+17, POS+18], and econometrics [Hoo90, RW06]. Recently, there has also been a growing interest in the machine learning community to use causal inference techniques to improve generalizability to novel testing environments (e.g. see [GUA+16, LKC17, ABGLP19, Sch22] and references therein). Under the assumption of causal sufficiency, where there is no unobserved confounders or selection bias, causal inference using observational data has been extensively studied and many algorithms such as PC [SGSH00] and GES [Chi02] have been proposed. These algorithms typically recover a causal graph up to its Markov equivalence class and it is known to be fundamentally impossible to learn causal relationships solely based on observational data. To overcome this issue, one either adds data modeling assumptions (e.g. see [SHHK06, PB14, MPJ+16]) or performs interventions to obtain interventional

*Equal contribution

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

data (see Section 2.1 for a literature review). In our work, we study the causal discovery problem via interventions. As interventions often correspond to real-world experimental trials, they can be costly and it is of practical importance to minimize the number or cost of interventions.

In this work, we consider *ideal interventions* (i.e. hard interventions with infinite samples[2]) to recover causal graphs from its Markov equivalence class – the *Markov equivalence class* (MEC) of G, denoted by $[G]$, is the set of graphs that encode the same conditional distributions and it is known that any MEC $[G]$ can be represented by a unique partially oriented *essential graph* $\mathcal{E}(G)$. While ideal interventions may not always be possible in practice, they serve as a first step to understand the verification and search problems. Such interventions give us a clean graph-theoretic way to reason and identify arc directions in the causal graph $G = (V, E)$. With these interventions, it is known that intervening on a set $S \subseteq V$ allows us to infer the edge orientation of any edge cut by S and $V \setminus S$ [Ebe07, HEH13, HLV14, SKDV15, KDV17].

Using the ideal interventions, we solve the verification and search problems. The *search problem* is the traditional question of finding a minimum set of interventions to recover the underlying ground truth causal graph. As for the *verification problem*, consider the following scenario: Suppose we have performed a observational study to obtain a MEC and consulted an expert about the identity of the ground truth causal graph. The question of verification involves testing if the expert is correct using the minimal number of interventional studies.

Definition 1 (Search problem)**.** Given the essential graph $\mathcal{E}(G^*)$ of an unknown causal graph G^*, use the minimal number of interventions to fully recover the ground truth causal graph G^*.

Definition 2 (Verification problem)**.** Given the essential graph $\mathcal{E}(G^*)$ of an unknown causal graph G^* and an expert's graph $G \in [G^*]$, use the minimal number of interventions to verify $G \stackrel{?}{=} G^*$.

To solve both these problems, we compute *verifying sets* (see Definition 4). A verifying set is a collection of interventions that fully orients the essential graph. Note that the minimum size/cost verifying set serves as a natural lower bound for both the search and verification problems. One of our key contributions is to efficiently compute these verifying sets for a given causal graph.

Contributions We study the problems of verification and search for a causal graph from its MEC using ideal interventions. In our work, we make standard assumptions such as the Markov assumption, the faithfulness assumption, and causal sufficiency [SGSH00].

1. **Verification**: We provide the first known efficient algorithms for computing minimal sized atomic verifying sets and near-optimal bounded size verifying sets (that use at most one more intervention than optimal) on general graphs. When vertices have additive interventional costs according to a weight function $w : V \to \mathbb{R}$, we give efficient computation of verifying sets $\mathcal{I}$ that minimizes $\alpha \cdot w(\mathcal{I}) + \beta \cdot |\mathcal{I}|$. Our atomic verifying sets have optimal cost and bounded size verifying sets incur a total additive cost of at most 2β more than optimal. To achieve these results, we prove properties about covered edges and give a characterization of verifying sets as a separation of unoriented covered edges in the given essential graph. Using our covered edge perspective, we show that the universal lower bounds of [SMG+20, PSS22] are *not* tight and give a simple proof recovering the verification upper bound of [PSS22].
2. **Adaptive search**: We consider adaptive search algorithms which produce a *sequence* of interventions one-at-a-time, possibly using information gained from the outcomes of earlier chosen interventions. Building upon the lower bound of [SMG+20], we establish a stronger (but not computable) lower bound to verify a causal graph. This further implies a lower bound on the minimum interventions needed by *any* adaptive search algorithm. We also provide an adaptive search algorithm (based on graph separators for chordal graphs) and use our lower bound to prove that our approach uses at most a logarithmic multiplicative factor more interventions than a minimum sized verifying set of the true underlying causal graph.

Outline Section 2 introduces notation and preliminary notions, as well as discuss some known results on ideal interventions. We give our results in Section 3 and provide an overview of the techniques used in Section 4. We discuss our experimental results in Section 5. Full proofs, side discussions, and source code/scripts are given in the appendix.

[2]We do not consider sample complexity issues in this work.

2 Preliminaries and related work

For any set A, we denote its powerset by 2^A. We write $\{1, \ldots, n\}$ as $[n]$ and hide absolute constant multiplicative factors in n using asymptotic notations $\mathcal{O}(\cdot)$, $\Omega(\cdot)$, and $\Theta(\cdot)$. Throughout, we use G^* to denote the (unknown) ground truth DAG and we only know its essential graph $\mathcal{E}(G^*)$. We say that a randomized algorithm for a graph on n vertices succeeds with high probability in n when the failure probability decays quickly in the form of n^{-c}, for some constant $c > 1$.

Graph notions

We study partially directed graphs $G = (V, E, A)$ on $|V| = n$ vertices with unoriented edges E and oriented arcs A. Any possible edge between two distinct vertices $u, v \in V$ is either undirected, oriented in one direction, or absent in the graph. We write $u \sim v$ if these vertices are connected (either through an unoriented edge or an arc) in the graph and $u \not\sim v$ to indicate the absence of any edge/arc connection. If $(u, v) \in A$ or $(v, u) \in A$, we write $u \to v$ or $u \leftarrow v$ respectively. When $E = \emptyset$, we say that the graph is fully oriented. For any graph G, we use $V(G), E(G), A(G)$ to denote its vertices, unoriented edges and oriented arcs.

For any subset of vertices $V' \subseteq V$, $G[V']$ denotes the vertex-induced subgraph on V'. Similarly, we define $G[A']$ and $G[E']$ as arc-induced and edge-induced subgraphs of G for $A' \subseteq A$ and $E' \subseteq E$ respectively. For an undirected graph G, $\omega(G)$ refers to the size of its maximum clique and $\chi(G)$ refers to its chromatic number. For any vertex $v \in V$ in a directed graph, $Pa(v) \subseteq V$ denotes the parent set of v and $pa(v)$ denotes the vector of values taken by v's parents.

The *skeleton* of a graph G refers to the graph $G' = (V, E \cup A, \emptyset)$ where all arcs are made unoriented. A *v-structure* refers to three distinct vertices $u, v, w \in V$ such that $u \to v \leftarrow w$ and $u \not\sim w$. A simple cycle is a sequence of $k \geq 3$ vertices where $v_1 \sim v_2 \sim \ldots \sim v_k \sim v_1$. The cycle is directed if at least one of the edges is directed and all directed arcs are in the same direction along the cycle. A partially directed graph is a *chain graph* if it contains no directed cycle. In the undirected graph G' obtained by removing all arcs from a chain graph G, each connected component in G' is called a *chain component*. We use $CC(G)$ to denote the set of chain components in G. Note that the vertices of these chain components form a partition of V.

Directed acyclic graphs (DAGs), a special case of chain graphs where *all* edges are directed, are commonly used as graphical causal models [Pea09] where vertices represents random variables and the joint probability density f factorizes according to the Markov property: $f(v_1, \ldots, v_n) = \prod_{i=1}^{n} f(v_i \mid pa(v))$. We can associate a *valid permutation / topological ordering* $\pi : V \to [n]$ to any (partially oriented) DAG such that oriented arcs (u, v) satisfy $\pi(u) < \pi(v)$ and any unoriented arc $\{u, v\}$ can be oriented as $u \to v$ whenever $\pi(u) < \pi(v)$ without forming directed cycles. While there may be multiple valid permutations, we often only care that there exists at least one such permutation. For any DAG G, we denote its *Markov equivalence class* (MEC) by $[G]$ and *essential graph* by $\mathcal{E}(G)$. It is known that two graphs are Markov equivalent if and only if they have the same skeleton and v-structures [VP90, AMP97].

A *clique* is a graph where $u \sim v$ for any pair of vertices $u, v \in V$. A *maximal clique* is an vertex-induced subgraph of a graph that is a clique and ceases to be one if we add any other vertex to the subgraph. If all edges in the clique are oriented in an acyclic manner, then there is a unique valid permutation π that respects this orientation. We denote $v = \text{argmax}_{u \in V} \pi(u)$ as the *sink* of the clique.

Interventions, verifying sets, and additive vertex intervention costs

An *intervention* $S \subseteq V$ is an experiment where the experimenter forcefully sets each variable $s \in S$ to some value, independent of the underlying causal structure. An intervention is called an *atomic intervention* if $|S| = 1$ and called a *bounded size intervention* if $|S| \leq k$ for some size upper bound k. One can view observational data as a special case where $S = \emptyset$. Interventions affect the joint distribution of the variables and are formally captured by Pearl's do-calulus [Pea09]. An *intervention set* $\mathcal{I} \subseteq 2^V$ is a collection of interventions and $\cup_{S \in \mathcal{I}} S$ is the union of all intervened vertices.

In this work, we study ideal interventions. Graphically speaking, an ideal intervention S on G induces an interventional graph G_S where all incoming arcs to vertices $v \in S$ are removed [EGS12] and it is known that intervening on a set $S \subseteq V$ allows us to infer the edge orientation of any edge cut by S and $V \setminus S$ [Ebe07, HEH13, HLV14, SKDV15, KDV17]. For ideal interventions, an $\mathcal{I}$-essential

graph $\mathcal{E}_{\mathcal{I}}(G)$ of G is the essential graph representing the Markov equivalence class of graphs whose interventional graphs for each intervention is Markov equivalent to G_S for any intervention $S \in \mathcal{I}$.

There are several known properties about $\mathcal{I}$-essential graph properties [HB12, HB14, SMG+20]: Every $\mathcal{I}$-essential graph is a chain graph with chordal[3] chain components. This includes the case of $S = \emptyset$. Orientations in one chain component do not affect orientations in other components. In other words, to fully orient any essential graph $\mathcal{E}(G^*)$, it is necessary and sufficient to orient every chain component in $\mathcal{E}(G^*)$ independently. More formally, we have[4]

Lemma 3 (Modified lemma 1 of [HB14])**.** *Let $\mathcal{I} \subseteq 2^V$ be an intervention set. Consider the $\mathcal{I}$-essential graph $\mathcal{E}_{\mathcal{I}}(G^*)$ of some DAG G^* and let $H \in CC(\mathcal{E}_{\mathcal{I}}(G^*))$ be one of its chain components. Then, for any additional interventional set $\mathcal{I}' \subseteq 2^V$ such that $\mathcal{I} \cap \mathcal{I}' = \emptyset$, we have*

$$\mathcal{E}_{\mathcal{I}\cup\mathcal{I}'}(G^*)[V(H)] = \mathcal{E}_{\{S\cap V(H)\,:\,S\in\mathcal{I}'\}}(G^*[V(H)]).$$

As a consequence of Lemma 3, one may assume without loss of generality that $CC(\mathcal{E}(G))$ is a single connected component and then generalize results by summing across all connected components.

A *verifying set* $\mathcal{I}$ for a DAG $G \in [G^*]$ is an intervention set that fully orients G from $\mathcal{E}(G^*)$, possibly with repeated applications of Meek rules (see Appendix A). In other words, for any graph $G = (V, E)$ and any verifying set $\mathcal{I}$ of G, we have $\mathcal{E}_{\mathcal{I}}(G)[V'] = G[V']$ for *any* subset of vertices $V' \subseteq V$. Furthermore, if $\mathcal{I}$ is a verifying set for G, then $\mathcal{I} \cup S$ is also a verifying set for G for any additional intervention $S \subseteq V$. While DAGs may have multiple verifying sets in general, we are often interested in finding one with minimum size or cost.

Definition 4 (Minimum size/cost verifying set)**.** Let w be a weight function on intervention sets. An intervention set $\mathcal{I}$ is called a verifying set for a DAG G^* if $\mathcal{E}_{\mathcal{I}}(G^*) = G^*$. $\mathcal{I}$ is a *minimum size (resp. cost) verifying set* if $\mathcal{E}_{\mathcal{I}'}(G^*) \neq G^*$ for any $|\mathcal{I}'| < |\mathcal{I}|$ (resp. for any $w(\mathcal{I}') < w(\mathcal{I})$).

When restricting to interventions of size at most k, the *minimum verification number* $\nu_k(G)$ of G denotes the size of the minimum size verifying set for any DAG $G \in [G^*]$. That is, any revealed arc directions when performing interventions on $\mathcal{E}(G^*)$ respects G. $\nu_1(G)$ denotes the case where we restrict to atomic interventions. One of the goals of this work is to characterize $\nu(G)$ given an essential graph $\mathcal{E}(G^*)$ and some $G \in [G^*]$.

Covered edges

Covered edges are special arcs in a causal graph where the endpoints of $u \sim v$ share the same set of parents in $V \setminus \{u, v\}$. These edges are crucial in causal discovery because their orientation can be reversed and they still yield the same conditional independencies. See Fig. 1 for an illustration. Note that one can compute all covered edges of a given DAG G in polynomial time.

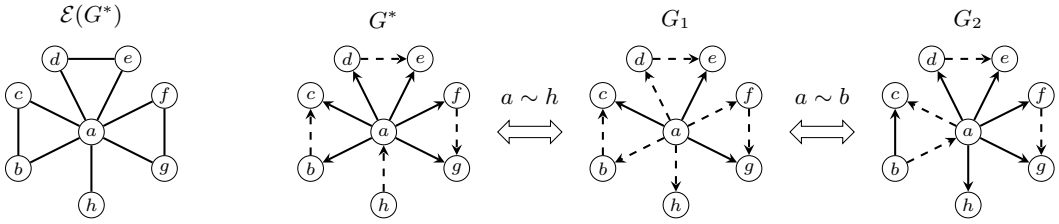


Figure 1: A DAG G^* with its essential graph $\mathcal{E}(G^*)$ on the left. G_1 and G_2 are two other DAGs that belong to the same Markov equivalence class $[G^*]$. Dashed arcs are covered edges in each DAG. One can perform a sequence of covered edge reversals to transform between the DAGs (see Lemma 7). Note that the sizes of the minimum vertex cover of the covered edges may differ across DAGs.

[3] A chordal graph is a graph where every cycle of length at least 4 has a chord, which is an edge that is not part of the cycle but connects two vertices of the cycle. See [BP93] for more properties.

[4] Lemma 1 of [HB14] actually considers a *single* additional intervention, but a closer look at their proof shows that the statement can be strengthened to allow for *multiple* additional interventions. For completeness, we provide the proof of this strengthened version in Appendix B. Note that we can drop the $\emptyset$ intervention in the statement since essential graphs are defined with the observational data provided.

Definition 5 (Covered edge)**.** An edge $u \sim v$ is a covered edge if $Pa(u) \setminus \{v\} = Pa(v) \setminus \{u\}$.

Definition 6 (Covered edge reversal)**.** A covered edge reversal means that we replace $u \to v$ with $v \to u$, for some covered edge $u \to v$, while keeping all other arcs unchanged.

Lemma 7 ([Chi95])**.** *If G and G' belong in the same MEC if and only if there exists a sequence of covered edge reversals to transform between them.*

Definition 8 (Separation of covered edges)**.** We say that an intervention $S \subseteq V$ *separates* a covered edge $u \sim v$ if $|\{u, v\} \cap S| = 1$. That is, *exactly* one of the endpoints is intervened by S. We say that an intervention set $\mathcal{I}$ separates a covered edge $u \sim v$ if there exists $S \in \mathcal{I}$ that separates $u \sim v$.

Relationship between searching and verification

Recall Definition 1 and Definition 2. The verification number is a useful analytical tool for the search problem as $\nu_k(G^*)$ is a lower bound on the number of interventions used by an optimal search algorithm. Furthermore, since the search problem needs to fully orient $\mathcal{E}(G^*)$ regardless of which DAG is the ground truth, any search algorithm given $\mathcal{E}(G^*)$ requires *at least* $\min_{G \in [G^*]} \nu_k(G)$ interventions, even if it is adaptive and randomized. In fact, the *strongest possible universal lower bound* guarantee one can prove must be *at most* $\min_{G \in [G^*]} \nu_k(G)$ and the *strongest possible universal upper bound* guarantee one can prove must be *at least* $\max_{G \in [G^*]} \nu_k(G)$. Note that if the search algorithm is *non-adaptive*, then it trivially needs *at least* $\max_{G \in [G^*]} \nu_k(G)$ interventions.

Example Consider the graph G^* in Fig. 1 where the essential graph $\mathcal{E}(G^*)$ representing the MEC $[G^*]$ is the standing windmill[5]. Now, consider only atomic interventions. We will later show that the minimum verification number of a DAG is the size of the minimum vertex cover of its covered edges (see Theorem 11). One can check that $\nu_1(G^*) = \nu_1(G_1) = 4$ while $\nu_1(G_2) = 3$. In fact, we actually show that $\min_{G \in [G^*]} \nu_1(G) = 3$ and $\max_{G \in [G^*]} \nu_1(G) = 4$ in Appendix C. Thus, any search algorithm using only atomic interventions on $\mathcal{E}(G^*)$ needs at least 3 atomic interventions.

2.1 Related work

Table 1 and Table 2 in Appendix D summarize[6] the existing upper (sufficient) and lower (worst case necessary) bounds on the size ($|\mathcal{I}|$, or $\mathbb{E}[|\mathcal{I}|]$ for randomized algorithms) of intervention sets that fully orient a given essential graph. These lower bounds are "worst case" in the sense that there exists a graph, typically a clique, which requires the stated number of interventions. Observe that there are settings where adaptivity[7] and randomization strictly improves the number of required interventions.

Separating systems [HEH13] drew connections between causal discovery via interventions and the concept of separating systems from the combinatorics literature. This was extended by [SKDV15] to the bounded size and adaptive settings. An (n, k)-separating system is a Boolean matrix with n columns where each row has at most k ones, indicating which vertex is to be intervened upon. Using their proposed separating system construction based on "label indexing", [SKDV15] showed that roughly $\frac{n}{k} \log_{\frac{n}{k}} n$ interventions is sufficient to fully an essential graph G with bounded size interventions. On cliques (i.e. worst case lower bound), [SKDV15] showed that the bound is tight while only roughly $\frac{\chi(\mathcal{E}(G))}{k} \log_{\frac{\chi(\mathcal{E}(G))}{k}} \chi(\mathcal{E}(G))$ interventions are necessary for general graphs[8], even if the interventions are chosen adaptively or in a randomized fashion.

Universal bounds for minimum sized atomic interventions Beyond worst case lower bounds, recent works have studied universal bounds for orienting essential graphs $\mathcal{E}(G^*)$ using atomic interventions [SMG+20, PSS22]. These universal bounds depend on graph parameters of $\mathcal{E}(G)$ beyond the number of nodes n. [SMG+20] showed that search algorithms must use at least $\sum_{H \in CC(\mathcal{E}(G^*))} \lfloor \frac{\omega(H)}{2} \rfloor$ interventions, where H is a chain component of $\mathcal{E}(G^*)$ and the summation across chain components is a consequence of Lemma 3. They also introduced a graph concept called directed clique trees

[5] To be precise, it is the Wd(3,3) windmill graph with an additional edge from the center.

[6] Some known results are discussed in further detail below instead of being summarized in the table format.

[7] Given an essential graph $\mathcal{E}(G^*)$, non-adaptive algorithms decide a *set* of interventions without looking at the outcomes of the interventions. Meanwhile, adaptive algorithms can provide a *sequence* of interventions one-at-a-time, possibly using any information gained from the outcomes of earlier chosen interventions.

[8] Note that there is a slight gap between $\frac{\chi(\mathcal{E}(G))}{k} \log_{\frac{\chi(\mathcal{E}(G))}{k}} \chi(\mathcal{E}(G))$ and $\frac{n}{k} \log_{\frac{n}{k}} n$ on general graphs.

and designed an adaptive, deterministic algorithm. On intersection-incomparable chordal graphs, their algorithm outputs an intervention set of size $\mathcal{O}(\log_2(\max_{H \in CC(\mathcal{E}(G^*))} \omega(H)) \cdot \nu_1(G^*))$. More recently, [PSS22] introduced the notion of clique-block shared-parents orderings and showed that any search algorithm for an essential graph $\mathcal{E}(G^*)$ with r maximal cliques requires at least $\lceil \frac{n-r}{2} \rceil$ interventions and $\nu_1(G) \leq n - r$ for any $G \in [G^*]$.

Non-atomic interventions The randomized algorithm of [HLV14] fully orients an essential graph using $\mathcal{O}(\log(\log(n)))$ unbounded interventions in expectation. Building upon this, [SKDV15] shows that $\mathcal{O}(\frac{n}{k} \log(\log(k)))$ bounded sized interventions (each involving at most k nodes) suffice.

Additive vertex costs [KDV17, GSKB18, LKDV18] studied the *non-adaptive* search setting where vertices may have different intervention costs and intervention costs accumulate additively. [GSKB18] studied the problem of maximizing number of oriented edges given a budget of atomic interventions while [KDV17, LKDV18] studied the problem of finding a minimum cost (bounded size) intervention set that fully orients the essential graph. [LKDV18] showed that computing the minimum cost intervention set is NP-hard and gave search algorithms with constant approximation factors.

Other related work [HLV14, KSSU19] showed that Erdős-Rényi graphs can be easily oriented.

3 Results

3.1 Verification

Our core contribution for the verification problem is deriving an interesting connection between the covered edges and verifying sets. We show importance of this connection by using it to derive several novel results on finding optimal verifying sets in various settings such as bounded size interventions and when vertices have varying interventional costs. For detailed proofs, see Appendix E.

Theorem 9. *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. An intervention set $\mathcal{I}$ is a verifying set for G if and only if $\mathcal{I}$ is a set that separates every covered edge of G that is unoriented in $\mathcal{E}(G^*)$.*

Together with Lemma 7 (any undirected edge in $\mathcal{E}(G^*)$ is a covered edge for *some* $G \in [G^*]$), Theorem 9 implies a simple alternative proof for an earlier known result that characterizes non-adaptive search algorithms via separating systems [HEH13, SKDV15]: any non-adaptive search algorithm, which has *no* knowledge of G^*, should separate *every* undirected edge in $\mathcal{E}(G^*)$.

Another immediate application of Theorem 9 is the following result for the verification problem.

Corollary 10. *Given an essential graph $\mathcal{E}(G^*)$ of an unknown ground truth DAG G^* and a causal DAG $G \in [G^*]$, we can test if $G \stackrel{?}{=} G^*$ by intervening on any verifying set of G. Furthermore, in the worst case,* any *algorithm that correctly resolves $G \stackrel{?}{=} G^*$ needs at least $\nu(G)$ interventions.*

The above corollary provides a solution to the verification problem in terms of verifying sets and in the following we give efficient algorithms for computing these verifying sets that are optimal in the atomic setting and are near-optimal in the case of bounded size.

Theorem 11. *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. An atomic intervention set $\mathcal{I}$ is a minimal sized verifying set for G if and only if $\mathcal{I}$ is a minimum vertex cover of unoriented covered edges of G. A minimal sized atomic verifying set can be computed in polynomial time.*

Our result provides the first efficient algorithm for computing minimum sized atomic verifying set for general graphs. Previously, efficient algorithms for computing minimum sized atomic verifying sets were only known for simple graphs such as cliques and trees. For general graphs, only a brute force algorithm is known [SMG+20, Appendix F] which takes exponential time in the worst case[9]. In contrast to optimal algorithms, [PSS22] provides an efficient algorithm that returns a verifying set of size at most 2 times that of the optimum.

Theorem 12. *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. If $\nu_1(G) = \ell$, then $\nu_k(G) \geq \lceil \frac{\ell}{k} \rceil$ and there exists a polynomial time algo. to compute a bounded size intervention set $\mathcal{I}$ of size $|\mathcal{I}| \leq \lceil \frac{\ell}{k} \rceil + 1$.*

[9] Theorem 11 also provides a rigorous justification to the observation of [SMG+20] that "In general, the size of an [atomic verifying set] cannot be calculated from just its essential graph". This is because essential graphs could imply minimum vertex covers of different sizes (see Fig. 1).

To the best of our knowledge, our work provides the first known efficient algorithm for computing near-optimal bounded sized verifying sets for general graphs. Furthermore, we note that, for every k, there exists a family of graphs where the optimum solution requires at least $\lceil \frac{\ell}{k} \rceil + 1$ bounded size interventions. Thus, our upper bound is tight in the worst case (see Fig. 6 in Appendix E.5).

Beyond minimal sized interventions, a natural and much broader setting in causal inference is one where different vertices have varying intervention costs, e.g. in a smoking study, it is easier to modify a subject's diet than to force the subject to smoke (or stop smoking). Formally, one can define a weight function on the vertices $w : V \to \mathbb{R}$ which overloads to $w(S) = \sum_{v \in S} w(v)$ on interventions and $w(\mathcal{I}) = \sum_{S \in \mathcal{I}} S$ on intervention sets. Such an additive cost structure has been studied by [KDV17, GSKB18]. Consider an essential graph which is a star graph on n nodes where the leaves have cost 1 and the root has cost significantly larger than n. For atomic verifying sets, we see that the *minimum cost* verifying set is to intervene on the leaves while the *minimum size* verifying set is to simply intervene on the root. Since one may be more preferred over the other, depending on the actual real-life situation, we propose to find a verifying set $\mathcal{I}$ which minimizes

$$\alpha \cdot w(\mathcal{I}) + \beta \cdot |\mathcal{I}| \qquad \text{where } \alpha, \beta \geq 0 \tag{1}$$

so as to explicitly trade-off between the cost and size of the intervention set. This objective also naturally allows the constraint of bounded size interventions by restricting $|S| \leq k$ for all $S \in \mathcal{I}$. Note that the earlier results on minimum size verifying sets correspond to $\alpha = 0$ and $\beta = 1$. Interestingly, the techniques we developed in the minimum size setting generalizes to this broader setting. Furthermore, our framework of covered edges allow us to get simple proofs.

Theorem 13. *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. An atomic verifying set $\mathcal{I}$ for G that minimizes Eq. (1) can be computed in polynomial time.*

Theorem 14. *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. Suppose the optimal bounded size intervention set that minimizes Eq. (1) costs OPT. Then, there exists a polynomial time algorithm that computes a bounded size intervention set with total cost $OPT + 2\beta$.*

3.2 Adaptive search

Here, we study the unweighted search problem in causal inference where one wishes to fully orient an essential graph obtained from observational data while minimizing the number of interventions used. Formally, we give an algorithm (Algorithm 1) that fully orients $\mathcal{E}(G^*)$ using at most a logarithmic multiplicative factor more interventions than $\nu_k(G^*)$, the number of (bounded size) interventions needed to *verify* the ground truth G^*. Since *any* search algorithm will incur at least $\nu_k(G^*)$ interventions, our result implies that search is (almost, up to $\log n$ multiplicative approximation) as easy as the verification. For detailed proofs, see Appendix F.

Theorem 15. *Fix an essential graph $\mathcal{E}(G^*)$ with an unknown underlying ground truth DAG G^*. Given $k = 1$, Algorithm 1 runs in polynomial time and computes an atomic intervention set $\mathcal{I}$ in a deterministic and adaptive manner such that $\mathcal{E}_{\mathcal{I}}(G^*) = G^*$ and $|\mathcal{I}| \in \mathcal{O}(\log(n) \cdot \nu_1(G^*))$.*

Theorem 16. *Fix an essential graph $\mathcal{E}(G^*)$ with an unknown underlying ground truth DAG G^*. Given $k > 1$, Algorithm 1 runs in polynomial time and computes a bounded size intervention set $\mathcal{I}$ in a deterministic and adaptive manner such that $\mathcal{E}_{\mathcal{I}}(G^*) = G^*$ and $|\mathcal{I}| \in \mathcal{O}(\log(n) \cdot \log(k) \cdot \nu_k(G^*))$.*

These results are the first competitive results that holds for using atomic or bounded size interventions on *general graphs*. The only previously known result of $\mathcal{O}(\log_2(\max_{H \in CC(\mathcal{E}(G^*))} \omega(H)) \cdot \nu_1(G^*))$ by [SMG+20] was an algorithm based on directed clique trees with provable guarantees only for atomic interventions on intersection-incomparable chordal graphs. To obtain our results, we are *not* simply improving the analysis of [SMG+20]. Instead, we developed a new algorithmic approach that is based on graph separators which is a much simpler concept than directed clique trees.

The approximation of $\mathcal{O}(\log n)$ to $\nu_1(G^*)$ is the tightest one can hope for atomic interventions in general. For instance, consider the case where $\mathcal{E}(G^*)$ is an undirected line graph on n vertices. Then, *any* adaptive algorithm needs $\Omega(\log n)$ atomic interventions in the worst case[10] while $\nu_1(G^*) = 1$. The line graph also provides a clear distinction between adaptive and non-adaptive search algorithms since *any* non-adaptive algorithm needs $\Omega(n)$ atomic interventions to separate all the edges in $\mathcal{E}(G^*)$.

[10] The lower bound reasoning is similar to the lower bound for binary search.

4 Overview of techniques

4.1 Verification

Our results for the verification problem are broadly divided into two categories: (I) Connection between the covered edges and verification set (Theorem 9, Corollary 10); (II) Efficient computation of verification sets under various settings using the connection established.

For the first type of results, we show that any intervention set is a verifying set if and only if it separates every unoriented covered edge of $G \in \mathcal{E}(G^*)$. For necessity, we show that all four Meek rules (which are known to be consistent and complete) will *not* orient any unoriented covered edge of G that is *not* separated by any intervention. Our proof is simple due to the usage of covered edges. For sufficiency, we show that *every* unoriented non-covered edge of G will be oriented by Meek rules if all covered edges are separated. We prove this using a subtle induction over a valid topological ordering of the vertices π of G^*: Let V_i be the first i smallest vertices in π, for $i = 1, 2, \ldots, n$. Consider subgraph $\mathcal{E}(G^*)[V_i]$ induced by V_i with v_i being the last vertex in the ordering of V_i. By induction, it suffices to show that all non-covered $u \to v_i$ edges are oriented for $u \in V_{i-1}$. To show this, we perform case analysis to argue that either $u \to v_i$ is part of v-structure (i.e. is already oriented in $\mathcal{E}(G^*)$) or Meek rule R2 will orient it.

The previous result only establishes equivalence between the verifying sets and the intervention sets that separate unoriented covered edges. For efficient computation of optimal verifying sets, we prove several additional properties of covered edges, which may be of independent interest.

Lemma 17 (Properties of covered edges)**.**

1. *Let H be the edge-subgraph induced by covered edges of a DAG G. Then, every vertex in H has at most one incoming edge and thus H is a forest of directed trees.*
2. *If a DAG G is a clique on $n \geq 3$ vertices $v_1, v_2, \ldots, v_n$ with $\pi(v_1) < \pi(v_2) < \ldots < \pi(v_n)$, then $v_1 \to v_2, \ldots, v_{n-1} \to v_n$ are the covered edges of G.*
3. *If $u \to v$ is a covered edge in a DAG G, then u* cannot *be a sink of any maximal clique of G.*

The fact that the edge-induced subgraph of the covered edges is a forest enables us to use standard dynamic programming techniques to compute (weighted) minimum vertex covers for the unoriented covered edges of G, corresponding to minimum size atomic verifying sets (Theorem 11 and Theorem 13). For bounded size verifying sets, we exploit the fact that trees are bipartite and so we can divide the minimum vertex covers into two partitions. Since vertices within each partite are non-adjacent, we can group them into larger interventions without affecting the overall number of separated edges, giving us the guarantees in Theorem 12 and Theorem 14.

Through the lens of covered edges, we see that existing universal bounds of [SMG+20, PSS22] are *not* tight[11]. Consider the case where the essential graph $\mathcal{E}(G^*)$ is the standing windmill graph given in Fig. 1. The graph $\mathcal{E}(G^*)$ has $n = 8$ nodes, $r = 4$ maximal cliques and the largest maximal clique is size 3. The lower bound of [SMG+20] yields $\sum_{H \in CC(\mathcal{E}(G^*))} \lfloor \frac{\omega(H)}{2} \rfloor = \lfloor \frac{3}{2} \rfloor = 1$ while lower bound of [PSS22] yields $\lceil \frac{n-r}{2} \rceil = \lceil \frac{8-4}{2} \rceil = 2$. Meanwhile, we show $\min_{G \in [G^*]} \nu_1(G) = 3$ in Appendix C.

We can also recover the $\nu_1(G^*) \leq n - r$ bound of [PSS22] with a short proof using covered edges.

Lemma 18. *For any essential graph $\mathcal{E}(G^*)$ on n vertices with r maximal cliques, there exists an atomic verifying set of size at most $n - r$.*

Proof. By Theorem 11, it suffices to find a vertex cover of the unoriented covered edges of $\mathcal{E}(G^*)$. By Lemma 17, any covered edge $u \to v$ cannot have u as a sink of any maximal clique. So, the set of all vertices without the r sink vertices is a vertex cover of the covered edges in $\mathcal{E}(G^*)$. □

4.2 Search

Here we provide the description and proof overview of the search results. Our search algorithm (Algorithm 1) relies on graph separators that we formally define next. Existence and efficient

[11] The non-tightness of the universal lower bound of [PSS22] was known and verified via random graph experiments.

computation of graph separators are well studied [LT79, GHT84, GRE84, AST90, KR10, WN11] and are commonly used in divide-and-conquer graph algorithms and as analysis tools.

Definition 19 (α-separator and α-clique separator)**.** Let A, B, C be a partition of the vertices V of a graph $G = (V, E)$. We say that C is an *α-separator* if no edge joins a vertex in A with a vertex in B and $|A|, |B| \leq \alpha \cdot |V|$. We call C is an *α-clique separator* if it is an *α-separator* and a clique.

Algorithm 1 Search algorithm via graph separators.

Input: Essential graph $\mathcal{E}(G^*)$, intervention size k. **Output**: A fully oriented graph $G \in [G^*]$.
Initialize $i = 0$ and $\mathcal{I}_0 = \emptyset$.
while $\mathcal{E}_{\mathcal{I}_i}(G^*)$ still has undirected edges **do**
  For each $H \in CC(\mathcal{E}_{\mathcal{I}_i}(G^*))$ of size $|H| \geq 2$, find a 1/2-clique separator K_H using Theorem 20. Define $Q = \{K_H\}_{H \in CC(\mathcal{E}_{\mathcal{I}_i}(G^*)), |H| \geq 2}$ as the union of clique separator nodes.
  if $k = 1$ or $|Q| = 1$ **then** Define $C_i = Q$ as an atomic intervention set.
  else Define $k' = \min\{k, |Q|/2\}$, $a = \lceil |Q|/k' \rceil \geq 2$, and $\ell = \lceil \log_a n \rceil$. Compute labelling scheme of [SKDV15, Lemma 1] on Q with $(|Q|, k', a)$, and define $C_i = \{S_{x,y}\}_{x \in [\ell], y \in [a]}$, where $S_{x,y} \subseteq Q$ is the subset of vertices whose x^{th} letter in the label is y.
  Update $i \leftarrow i + 1$, intervene on C_i to obtain $\mathcal{E}_{\mathcal{I}_i}(G^*)$, and update $\mathcal{I}_i \leftarrow \mathcal{I}_{i-1} \cup C_i$.
end while

We first give the proof strategy for Theorem 15, the atomic case where $k = 1$. We divide the analysis into two steps. In the first step, we show that the algorithm terminates in $\mathcal{O}(\log n)$ iterations. In the second step, we argue that the number of interventions performed in each iteration is in $\mathcal{O}(\nu_1(G))$.

The analysis of the first step relies on the following result of [GRE84].

Theorem 20 ([GRE84], instantiated for unweighted graphs)**.** *Let $G = (V, E)$ be a chordal graph with $|V| \geq 2$ and p vertices in its largest clique. There exists a $1/2$-clique-separator C of size $|C| \leq p - 1$. The clique C can be computed in $\mathcal{O}(|E|)$ time.*

At each iteration i of the algorithm, we intervene on a $1/2$-clique separators of each connected chain component. Note that the size of each connected chain compoenent at the end of iteration i is at most $n/2^i$. Therefore, after $\mathcal{O}(\log n)$ iterations, each connected chain component will contain at most 1 vertex, implying that all the edges have been oriented.

To bound the number of interventions used in each iteration, we prove a stronger universal lower bound that is built upon the lower bound of [SMG+20]. While *not* computable[12], it is a very powerful lower bound for analysis: In Fig. 7 of Appendix G, we give an example where $\nu_1(G^*) \approx n$ while the lower bound of [SMG+20] on $CC(\mathcal{E}(G^*))$ is a constant. Meanwhile, there exists a set of atomic interventions $\mathcal{I}$ such that applying [SMG+20] on $CC(\mathcal{E}_{\mathcal{I}}(G^*))$ yields a much stronger $\Omega(n)$ bound.

Lemma 21. *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. Then,*

$$\nu_1(G) \geq \max_{\mathcal{I} \subseteq V} \sum_{H \in CC(\mathcal{E}_{\mathcal{I}}(G^*))} \left\lfloor \frac{\omega(H)}{2} \right\rfloor$$

As we intervene on cliques in each connected component at every iteration, Lemma 21 shows that we use at most $2 \cdot \nu_1(G^*)$ interventions per iteration. Therefore, the total interventions used by Algorithm 1 is in $\mathcal{O}(\log(n) \cdot \nu_1(G^*))$.

For bounded size interventions (Theorem 16), we follow the same strategy as above but we modify how we orient the edges within and cut by the $1/2$-clique separators. More specifically, we compute a separating system for the union of clique separator nodes based on the labeling scheme of [SKDV15] and perform case analysis to argue that we use $\mathcal{O}(\log(k) \cdot \nu_k(G^*))$ bounded size interventions per iteration.

Lemma 22 (Lemma 1 of [SKDV15])**.** *Let (n, k, a) be parameters where $k \leq n/2$. There is a polynomial time labeling scheme that produces distinct ℓ length labels for all elements in $[n]$ using letters from the integer alphabet $\{0\} \cup [a]$ where $\ell = \lceil \log_a n \rceil$. Further, in every digit (or position), any integer letter is used at most $\lceil n/a \rceil$ times. This labelling scheme is a separating system: for any $i, j \in [n]$, there exists some digit $d \in [\ell]$ where the labels of i and j differ.*

[12] It involves a maximization over all possible atomic interventions *and* we do not know the $\mathcal{E}_{\mathcal{I}}(G^*)$'s.

5 Experiments and implementation

We implement our verification algorithm and test its correctness on some well-known graphs such as cliques and trees, for which we know the exact verification number. In addition, we implement Algorithm 1 and compare its performance with other known atomic search algorithms [HG08, HB14, SKDV15, SMG+20] via the experimental setup of [SMG+20]: on synthetic graphs of varying sizes, we compare the runtime and total number of interventions performed compared to the verification number of the underlying DAG. In Appendix H, we provide the full experimental details and results of running various search algorithms (including ours) on different graphs. Qualitatively, our algorithm is competitive with the state-of-the-art search algorithms while being $\sim$10x faster in some experiments. Fig. 2 shows a subset of these results. We also investigated the impact of different k values on the performance of our search algorithm in Appendix H. The implementations, along with entire experimental setup, are available at https://github.com/cxjdavin/verification-and-search-algorithms-for-causal-DAGs.

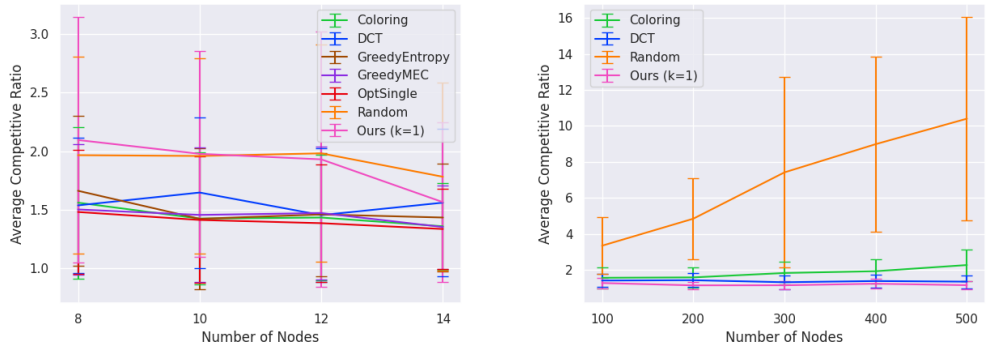


Figure 2: The figures show the average competitive ratios with respect to synthetic graphs of different node sizes. See Appendix H for details about how the synthetic graphs are generated and for details on the algorithms benchmarked. We also show the *maximum* competitive ratios in Appendix H.

Conclusion, limitations, and societal impact Learning causal relationships is of fundamental importance to science and society in general. This is especially important when one wishes to correctly predict effects of making changes to a system for downstream tasks such as designing fair algorithms. In this work, we gave a complete understanding of the verification problem and an improved search algorithm under the standard causal inference assumptions (see Section 1). However, if our assumptions are violated by the data, then wrong causal conclusions may be drawn and possibly lead to unintended downstream consequences. Hence, it is of great interest to remove/weaken these assumptions while maintaining strong theoretical guarantees. A crucial limitation of this work is that we study an idealized setting with hard interventions and infinite samples while soft interventions may be more realistic in certain real-life scenarios (e.g. effects from parental vertices are not completely removed but only altered) and sample complexities play a crucial role when one has limited experimental budget (e.g. see [KJSB19] and [ABDK18] respectively).

Acknowledgments and Disclosure of Funding

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-PhD/2021-08-013). KS was supported by a Stanford Data Science Scholarship, a Dantzig-Lieberman Research Fellowship and a Simons-Berkeley Research Fellowship. AB is partially supported by an NUS Startup Grant (R-252-000-A33-133) and an NRF Fellowship for AI (NRFFAI1-2019-0002). We would like to thank Themis Gouleakis, Dimitrios Myrisiotis, and Chandler Squires for valuable feedback and discussions. Part of this work was done while the authors were visiting the Simons Institute for the Theory of Computing.

References

[ABDK18] Jayadev Acharya, Arnab Bhattacharyya, Constantinos Daskalakis, and Saravanan Kandasamy. Learning and Testing Causal Models with Interventions. *Advances in Neural Information Processing Systems*, 31, 2018.

[ABGLP19] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[AMP97] Steen A. Andersson, David Madigan, and Michael D. Perlman. A characterization of Markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541, 1997.

[AST90] Noga Alon, Paul Seymour, and Robin Thomas. A separator theorem for nonplanar graphs. *Journal of the American Mathematical Society*, 3(4):801–808, 1990.

[BP93] Jean R. S. Blair and Barry W. Peyton. An introduction to chordal graphs and clique trees. In *Graph theory and sparse matrix computation*, pages 1–29. Springer, 1993.

[Chi95] David Maxwell Chickering. A Transformational Characterization of Equivalent Bayesian Network Structures. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, page 87–98, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[Chi02] David Maxwell Chickering. Optimal Structure Identification with Greedy Search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

[Ebe07] Frederick Eberhardt. Causation and Intervention. *Unpublished doctoral dissertation, Carnegie Mellon University*, page 93, 2007.

[Ebe10] Frederick Eberhardt. Causal Discovery as a Game. In *Causality: Objectives and Assessment*, pages 87–96. PMLR, 2010.

[EGS06] Frederick Eberhardt, Clark Glymour, and Richard Scheines. N-1 Experiments Suffice to Determine the Causal Relations Among N Variables. In *Innovations in machine learning*, pages 97–112. Springer, 2006.

[EGS12] 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. *arXiv preprint arXiv:1207.1389*, 2012.

[ES07] Frederick Eberhardt and Richard Scheines. Interventions and Causal Inference. *Philosophy of science*, 74(5):981–995, 2007.

[GHT84] John R. Gilbert, Joan P. Hutchinson, and Robert Endre Tarjan. A Separator Theorem for Graphs of Bounded Genus. *Journal of Algorithms*, 5(3):391–407, 1984.

[GKS+19] Kristjan Greenewald, Dmitriy Katz, Karthikeyan Shanmugam, Sara Magliacane, Murat Kocaoglu, Enric Boix-Adserà, and Guy Bresler. Sample Efficient Active Learning of Causal Trees. *Advances in Neural Information Processing Systems*, 32, 2019.

[GRE84] John R. Gilbert, Donald J. Rose, and Anders Edenbrandt. A Separator Theorem for Chordal Graphs. *SIAM Journal on Algebraic Discrete Methods*, 5(3):306–313, 1984.

[GSKB18] AmirEmad Ghassami, Saber Salehkaleybar, Negar Kiyavash, and Elias Bareinboim. Budgeted Experiment Design for Causal Structure Learning. In *International Conference on Machine Learning*, pages 1724–1733. PMLR, 2018.

[GUA+16] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[HB12] Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs. *The Journal of Machine Learning Research*, 13(1):2409–2464, 2012.

[HB14] Alain Hauser and Peter Bühlmann. Two optimal strategies for active learning of causal models from interventional data. *International Journal of Approximate Reasoning*, 55(4):926–939, 2014.

[HEH13] Antti Hyttinen, Frederick Eberhardt, and Patrik O. Hoyer. Experiment Selection for Causal Discovery. *Journal of Machine Learning Research*, 14:3041–3071, 2013.

[HG08] Yang-Bo He and Zhi Geng. Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research*, 9(Nov):2523–2547, 2008.

[HLV14] Huining Hu, Zhentao Li, and Adrian Vetta. Randomized Experimental Design for Causal Graph Discovery. *Advances in neural information processing systems*, 27, 2014.

[Hoo90] Kevin D Hoover. The logic of causal inference: Econometrics and the Conditional Analysis of Causation. *Economics & Philosophy*, 6(2):207–234, 1990.

[KDV17] Murat Kocaoglu, Alex Dimakis, and Sriram Vishwanath. Cost-Optimal Learning of Causal Graphs. In *International Conference on Machine Learning*, pages 1875–1884. PMLR, 2017.

[KF09] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[KJSB19] Murat Kocaoglu, Amin Jaber, Karthikeyan Shanmugam, and Elias Bareinboim. Characterization and Learning of Causal Graphs with Latent Variables from Soft Interventions. *Advances in Neural Information Processing Systems*, 32, 2019.

[KR10] Ken-ichi Kawarabayashi and Bruce Reed. A separator theorem in minor-closed classes. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 153–162. IEEE, 2010.

[KSSU19] Dmitriy Katz, Karthikeyan Shanmugam, Chandler Squires, and Caroline Uhler. Size of Interventional Markov Equivalence Classes in Random DAG Models. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3234–3243. PMLR, 2019.

[KWJ+04] Ross D. King, Kenneth E. Whelan, Ffion M. Jones, Philip G. K. Reiser, Christopher H. Bryant, Stephen H. Muggleton, Douglas B. Kell, and Stephen G. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427(6971):247–252, 2004.

[LKC17] Gilles Louppe, Michael Kagan, and Kyle Cranmer. Learning to pivot with adversarial networks. *Advances in neural information processing systems*, 30, 2017.

[LKDV18] Erik M. Lindgren, Murat Kocaoglu, Alexandros G. Dimakis, and Sriram Vishwanath. Experimental Design for Cost-Aware Learning of Causal Graphs. *Advances in Neural Information Processing Systems*, 31, 2018.

[LT79] Richard J Lipton and Robert Endre Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979.

[Mee95] Christopher Meek. Causal Inference and Causal Explanation with Background Knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, page 403–410, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[MPJ+16] Joris M. Mooij, Jonas Peters, Dominik Janzing, Jakob Zscheischler, and Bernhard Schölkopf. Distinguishing Cause from Effect Using Observational Data: Methods and Benchmarks. *J. Mach. Learn. Res.*, 17(1):1103–1204, Jan 2016.

[PB14] Jonas Peters and Peter Bühlmann. Identifiability of Gaussian structural equation models with equal error variances. *Biometrika*, 101(1):219–228, 2014.

[Pea09] Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009.

[POS+18] Jean-Baptiste Pingault, Paul F O'reilly, Tabea Schoeler, George B Ploubidis, Frühling Rijsdijk, and Frank Dudbridge. Using genetic data to strengthen causal inference in observational research. *Nature Reviews Genetics*, 19(9):566–580, 2018.

[PSS22] Vibhor Porwal, Piyush Srivastava, and Gaurav Sinha. Almost Optimal Universal Lower Bound for Learning Causal DAGs with Atomic Interventions. In *The 25th International Conference on Artificial Intelligence and Statistics*, 2022.

[Rei56] Hans Reichenbach. *The direction of time*, volume 65. Univ of California Press, 1956.

[RHT+17] Maya Rotmensch, Yoni Halpern, Abdulhakim Tlimat, Steven Horng, and David Sontag. Learning a Health Knowledge Graph from Electronic Medical Records. *Scientific reports*, 7(1):1–11, 2017.

[RW06] Donald B Rubin and Richard P Waterman. Estimating the Causal Effects of Marketing Interventions Using Propensity Score Methodology. *Statistical Science*, pages 206–222, 2006.

[SC17] Yuriy Sverchkov and Mark Craven. A review of active learning approaches to experimental design for uncovering biological networks. *PLoS computational biology*, 13(6):e1005466, 2017.

[Sch22] Bernhard Schölkopf. *Causality for Machine Learning*, page 765–804. Association for Computing Machinery, New York, NY, USA, 1 edition, 2022.

[SGSH00] Peter Spirtes, Clark N. Glymour, Richard Scheines, and David Heckerman. *Causation, Prediction, and Search*. MIT press, 2000.

[SHHK06] Shohei Shimizu, Patrik O. Hoyer, Aapo Hyvärinen, and Antti Kerminen. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(10), 2006.

[SKDV15] Karthikeyan Shanmugam, Murat Kocaoglu, Alexandros G. Dimakis, and Sriram Vishwanath. Learning Causal Graphs with Small Interventions. *Advances in Neural Information Processing Systems*, 28, 2015.

[SMG+20] Chandler Squires, Sara Magliacane, Kristjan Greenewald, Dmitriy Katz, Murat Kocaoglu, and Karthikeyan Shanmugam. Active Structure Learning of Causal DAGs via Directed Clique Trees. *Advances in Neural Information Processing Systems*, 33:21500–21511, 2020.

[VP90] Thomas Verma and Judea Pearl. Equivalence and Synthesis of Causal Models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, UAI '90, page 255–270, USA, 1990. Elsevier Science Inc.

[WBL21] Marcel Wienöbst, Max Bannach, and Maciej Liśkiewicz. Extendability of causal graphical models: Algorithms and computational complexity. In Cassio de Campos and Marloes H. Maathuis, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 1248–1257. PMLR, 27–30 Jul 2021.

[WN11] Christian Wulff-Nilsen. Separator theorems for minor-free and shallow minor-free graphs with applications. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 37–46. IEEE, 2011.

[Woo05] James Woodward. *Making Things Happen: A theory of Causal Explanation*. Oxford university press, 2005.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 5.
   (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] See Section 1.
   (b) Did you include complete proofs of all theoretical results? [Yes] We provide full proofs in the appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Appendix H and `https://github.com/cxjdavin/verification-and-search-algorithms-for-causal-DAGs`.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A] There is no training involved.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Appendix H.
   (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] See Appendix H.
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] We based our evaluation on the experimental framework of [SMG+20] (`https://github.com/csquires/dct-policy`). Proper credit was given.
   (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? [N/A]
   (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]

A Meek rules

Meek rules are a set of 4 edge orientation rules that are sound and complete with respect to any given set of arcs that has a consistent DAG extension [Mee95]. Given any edge orientation information, one can always repeatedly apply Meek rules till a fixed point to maximize the number of oriented arcs.

Definition 23 (Consistent extension)**.** A set of arcs is said to have a *consistent DAG extension* π for a graph G if there exists a permutation on the vertices such that (i) every edge $\{u, v\}$ in G is oriented $u \to v$ whenever $\pi(u) < \pi(v)$, (ii) there is no directed cycle, (iii) all the given arcs are present.

Definition 24 (The four Meek rules [Mee95], see Fig. 3 for an illustration)**.**

R1 Edge $\{a, b\} \in E$ is oriented as $a \to b$ if $\exists\, c \in V$ such that $c \to a$ and $c \not\sim b$.

R2 Edge $\{a, b\} \in E$ is oriented as $a \to b$ if $\exists\, c \in V$ such that $a \to c \to b$.

R3 Edge $\{a, b\} \in E$ is oriented as $a \to b$ if $\exists\, c, d \in V$ such that $d \sim a \sim c$, $d \to b \leftarrow c$, and $c \not\sim d$.

R4 Edge $\{a, b\} \in E$ is oriented as $a \to b$ if $\exists\, c, d \in V$ such that $d \sim a \sim c$, $d \to c \to b$, and $b \not\sim d$.

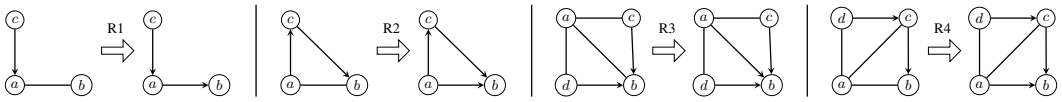


Figure 3: An illustration of the four Meek rules

There exists an algorithm [WBL21, Algorithm 2] that runs in $\mathcal{O}(d \cdot |E \cup A|)$ time and computes the closure under Meek rules, where d is the degeneracy of the graph skeleton[13].

B Proof of Lemma 3

Lemma 1 of [HB14] actually considers a *single* additional intervention, but a closer look at their proof shows that the statement can be strengthened to allow for *multiple* additional interventions. In fact, the proof below will almost mimic the proof of [HB14, Lemma 1] except for some minor changes[14]. Note that we can drop the $\emptyset$ intervention in the statement since essential graphs are defined with the observational data provided. The proof relies on the definition of *strongly protected edges* and a characterization of $\mathcal{I}$-essential graphs from [HB12].

Definition 25 (Strong protection; Definition 14 of [HB12])**.** Let $G = (V, E, A)$ be a (partially oriented) DAG and $\mathcal{I} \subseteq 2^V$ be an intervention set. An arc $a \to b$ is *strongly $\mathcal{I}$-protected* in G if there is some intervention $S \in \mathcal{I}$ such that $|S \cap \{a, b\}| = 1$, or the arc $a \to b$ occurs in at least one of the following four configurations as an induced subgraph of G (see Fig. 4):

1. There exists $c \in V$ such that $c \to a \to b$ and $c \not\sim b$.
2. There exists $c \in V$ such that $a \to b \leftarrow c$ and $c \not\sim a$.
3. There exists $c \in V$ such that $a \to c \to b$ and $a \to b$.
4. There exists $c, d \in V$ such that $a \sim c \to b$, $a \sim d \to b$, and $a \to b$.

Theorem 26 (Characterization of $\mathcal{I}$-essential graphs; Theorem 18 of [HB12])**.** *A (partially oriented) DAG $\mathcal{E}(G)$ is an $\mathcal{I}$-essential graph of G if and only if*

1. *G is a chain graph.*

[13] A d-degenerate graph is an undirected graph in which every subgraph has a vertex of degree at most d. Note that the degeneracy of a graph is typically smaller than the maximum degree of the graph.

[14] [HB14] considered whether the additional intervention $S \subseteq V$ separates a particular edge. In our proof, we change that argument to whether *some* intervention $S \in \mathcal{I}'$ separates that same edge. To argue that two graphs G and H are the same, one can show that $G \subseteq H$ and $H \subseteq G$. They only proved "one direction" and claim that the other holds by similar arguments. For completeness, we state *exactly* what are changes needed.

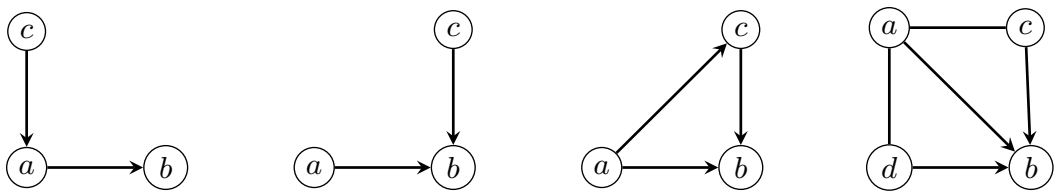


Figure 4: An illustration of the four configurations of strongly protected arc $a \to b$

2. *For each chain component $H \in CC(G)$, $G[V(H)]$ is chordal.*
3. *G has no induced subgraph of the form $a \to b \sim c$.*
4. *G has no undirected edge $a \sim b$ whenever $\exists S \in \mathcal{I}$ such that $|S \cap \{a, b\}| = 1$.*
5. *Every arc $a \to b$ in G is strongly $\mathcal{I}$-protected.*

For simplicity, we say that an intervention $S \subseteq V$ *separates* an edge $a \sim b$ if $|S \cap \{a, b\}| = 1$ and that an intervention set $\mathcal{I} \subseteq 2^V$ *separates* an edge $a \sim b$ if it has an intervention that separates it.

Lemma 3 (Modified lemma 1 of [HB14])**.** *Let $\mathcal{I} \subseteq 2^V$ be an intervention set. Consider the $\mathcal{I}$-essential graph $\mathcal{E}_{\mathcal{I}}(G^*)$ of some DAG G^* and let $H \in CC(\mathcal{E}_{\mathcal{I}}(G^*))$ be one of its chain components. Then, for any additional interventional set $\mathcal{I}' \subseteq 2^V$ such that $\mathcal{I} \cap \mathcal{I}' = \emptyset$, we have*

$$\mathcal{E}_{\mathcal{I} \cup \mathcal{I}'}(G^*)[V(H)] = \mathcal{E}_{\{S \cap V(H) \,:\, S \in \mathcal{I}'\}}(G^*[V(H)]).$$

Proof. To shorten notation, define $G = \mathcal{E}_{\mathcal{I} \cup \mathcal{I}'}(G^*)[V(H)]$ and $G' = \mathcal{E}_{\{S \cap V(H) \,:\, S \in \mathcal{I}'\}}(G^*[V(H)])$. Since $G = (V, E, A)$ and $G' = (V', E', A')$ share the same skeleton (i.e. $V = V'$ and $E \cup A = E' \cup A'$) and must respect the same underlying DAG directions of G^*, it suffices to argue that $A \subseteq A'$ and $A' \subseteq A$ (i.e. they share the same set of directed arcs). Let π be the topological ordering of the ground truth DAG G^*.

Direction 1 ($A \subseteq A'$): Suppose there are directed arcs in G that are undirected in G'. Let $a \to b$ be one such arc where $\pi(b)$ is *minimized*.

By property 5 of Theorem 26, $a \to b$ is strongly $(\mathcal{I} \cup \mathcal{I}')$-protected in G. If $\{S \cap V(H) \,:\, S \in \mathcal{I}'\}$ separates $a \sim b$, then $a \to b$ must be oriented in G'. Otherwise, let us consider the 4 configurations given by Definition 25:

1. In G, there exists $c \in V$ such that $c \to a \to b$ and $c \not\sim b$. By minimality of $\pi(b)$, $c \to a$ must be oriented in G'. Thus, Meek rule R1 will orient $a \to c$.
2. In G, there exists $c \in V$ such that $a \to b \leftarrow c$ and $c \not\sim a$. This is a v-structure in G^* and so $a \to b$ would also be oriented in G'.
3. In G, there exists $c \in V$ such that $a \to c \to b$ and $a \to b$. By minimality of $\pi(b)$, $a \to c$ must be oriented in G'. Then, if $a \to b$ is *not* directed in G', we will have a directed cycle of the form $a \to c \sim b \sim a$ (regardless of whether the edge $b \sim c$ is directed). By property 1 of Theorem 26, G' is a chain graph and *cannot* have such a directed cycle. Therefore, $a \to b$ must be oriented in G'.
4. In G, there exists $c, d \in V$ such that $a \sim c \to b$, $a \sim d \to b$, and $a \to b$. We cannot have $c \to a \leftarrow d$ otherwise such a v-struct will prevent this configuration from occurring. Without loss of generality, $a \to c$. Then, we can apply the argument of the third configuration on the subgraph induced by $\{a, b, c\}$ to conclude that $a \to b$ is also oriented in G'.

Direction 2 ($A' \subseteq A$): Repeat the *exact* same argument but perform the following 2 swaps:

1. Swap the roles of G and G'
2. Swap the roles of $(\mathcal{I} \cup \mathcal{I}')$ and $\{S \cap V(H) \,:\, S \in \mathcal{I}'\}$ □

C Further analysis of the standing windmill essential graph

In this section, we show that *all* DAGs in the standing windmill essential graph requires at least 3 and at most 4 atomic interventions.

By Theorem 11, we know that the optimal number of atomic interventions needed to verify any graph is the size of the minimum vertex cover of its oriented edges. To explore the space of DAGs in the essential graph, we will perform covered edge reversals (as justified by Lemma 7).

Consider the DAG G^* with MEC $[G^*]$ and the standing windmill essential graph $\mathcal{E}(G^*)$ in Fig. 5. Starting from G^*, if we fix the arc direction $h \to a$, then reversing any arc (possibly multiple times) from the set $\{b \sim c, d \sim e, f \sim g\}$ does *not* change the covered edge status of any edge (i.e. the covered edges remain exactly the same 4 edges) and thus the size of the minimum vertex cover remains unchanged. Meanwhile, reversing $a \sim h$ in G^* yields the graph G_1. Fixing the arc direction $a \to h$, we observe that the three sets of edges $\{a \sim b, a \sim c, b \sim c\}$, $\{a \sim d, a \sim e, d \sim e\}$, and $\{a \sim f, a \sim g, f \sim g\}$ are symmetric. Furthermore, if we flip one of the edges from $\{a \sim b, a \sim d, a \sim f\}$ from G_1 (or $\{a \sim c, a \sim d, a \sim f\}$ from G_4), then all other two $a \to \cdot$ arcs are no longer covered edges. So, it suffices to study what happens when we only reverse arc directions in one of these sets: $\{a \sim b, a \sim c, b \sim c\}$, $\{a \sim d, a \sim e, d \sim e\}$, and $\{a \sim f, a \sim g, f \sim g\}$. The graphs G_1 to G_6 illustrate all possible cases when we fix $a \to h$ and only reverse edges in the set $\{a \sim b, a \sim c, b \sim c\}$. We see that $\nu_1(G^*) = \nu_1(G_1) = \nu_1(G_4) = 4$ and $\nu_1(G_2) = \nu_1(G_3) = \nu_1(G_5) = \nu_1(G_6) = 3$. Thus, we can conclude that $\min_{G \in [G^*]} \nu_1(G) = 3$ and $\max_{G \in [G^*]} \nu_1(G) = 4$.

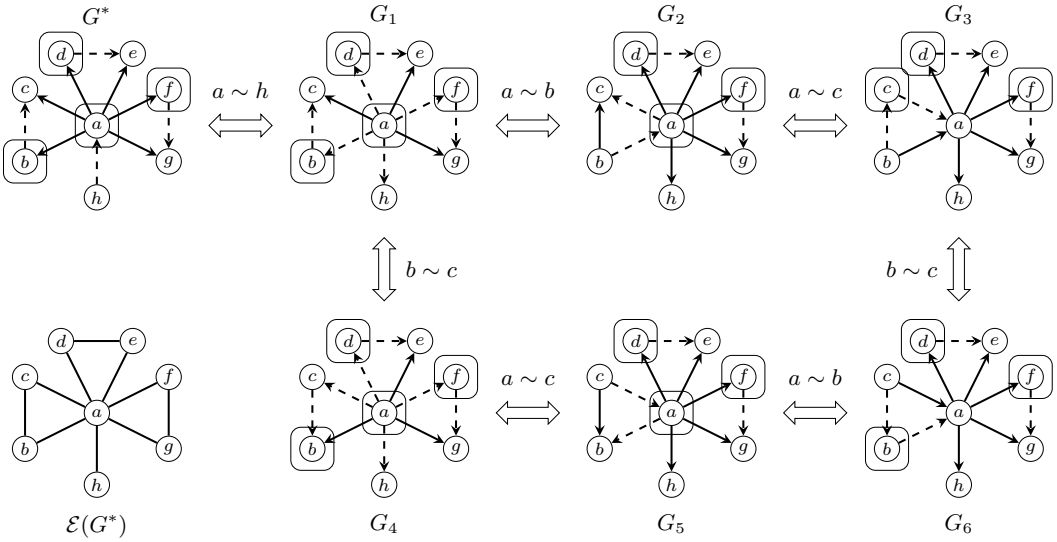


Figure 5: A DAG G^* with its essential graph $\mathcal{E}(G^*)$ and some of the graphs $G \in [G^*]$. In each DAG, dashed arcs are covered edges and the boxed vertices represent a minimum vertex cover.

D Upper and (worst case) lower bounds for fully orienting a DAG from its essential graph using ideal interventions

Let us briefly distinguish the various problem settings before summarizing the state of the art results.

Intervention size Since interventions are expensive, natural restrictions on the size of any intervention $S \in \mathcal{I}$ has been studied. Bounded size interventions enforce that an upper bound of $|S| \leq k$ always while unbounded size interventions allow k to be as large as $n/2$. Note that it does not make sense to intervene on a set S with $|S| > n/2$ since intervening on $\overline{S}$ yields the same information while being a strictly smaller interventional set. Atomic interventions are a special case where $k = 1$.

Adaptivity A passive/non-adaptive/simultaneous algorithm is one which, given an essential graph $\mathcal{E}(G^*)$, decides a *set* of interventions without looking at the outcomes of the interventions. Meanwhile, active/adaptive algorithms can provide a *sequence* of interventions one-at-a-time, possibly using any information gained from the outcomes of earlier chosen interventions.

Determinism An algorithm is deterministic if it always produces the same output given the same input. Meanwhile, randomized algorithms produces an output from a distribution. Analyses of randomized algorithms typically involve probabilistic arguments and their performance is measured in expectation with probabilistic success[15]. The ability to use random bits (e.g. outcome of coin flips) is very powerful and may allow one to circumvent known deterministic lower bounds.

Special graph classes Two graph classes of particular interest are cliques and trees. If $CC(\mathcal{E}(G^*))$ is a clique, then all $\binom{n}{2}$ edges are present and fully orienting the clique is equivalent to finding the unique valid permutation on the vertices. As such, cliques are often used to prove worst case lower bounds. Meanwhile, if $CC(\mathcal{E}(G^*))$ is a tree, then there must be a unique root (else there will be v-structures) and it suffices[16] to intervene on the root node to fully orient the tree.

Table 1 and Table 2 summarize some existing upper (sufficient) and lower (worst case necessary) bounds on the size ($|\mathcal{I}|$, or $\mathbb{E}[|\mathcal{I}|]$ for randomized algorithms) of intervention sets that fully orient a given essential graph. These lower bounds are "worst case" in the sense that there exists a graph, typically a clique, which requires the stated number of interventions. Observe that there are settings where adaptivity and randomization strictly improves the number of required interventions.

Size	Adaptive	Randomized	Graph	Upper bound	Reference
1	✗	✗	General	$n-1$	[EGS06]
1	✗	✓	General	$\frac{2}{3}n - \frac{1}{3}$ for $n > 3$	[Ebe10]
1	✓	✗	Tree	$\mathcal{O}(\log n)$	[SKDV15]
1	✓	✗	Tree	$\lceil \log n \rceil$	[GKS+19]
$\leq k$	✗	✗	General	$(\frac{n}{k} - 1) + \frac{n}{2k} \log_2 k$	[EGS12]
$\leq k$	✓	✗	Tree	$\lceil \log_{k+1} n \rceil$	[GKS+19]
$\leq k$	✓	✓	Clique	$\mathcal{O}(\frac{n}{k} \log \log k)$	[SKDV15]
∞	✗	✗	General	$\log_2 n$	[EGS12]
∞	✗	✗	General	$\lceil \log_2(\omega(\mathcal{E}(G)) \rceil$	[HB14]
∞	✗	✓	General	$\mathcal{O}(\log \log n)$	[HLV14]

Table 1: Upper bounds on the size ($|\mathcal{I}|$, or $\mathbb{E}[|\mathcal{I}|]$ for randomized algorithms) of the intervention set sufficient to fully orient a given essential graph $\mathcal{E}(G)$. The first three columns indicate the setting which the algorithm operates in terms of intervention size, adaptivity, and randomness. The fourth column indicate whether the algorithm is for special graph classes. Roughly speaking, the algorithm has *more power as we move down the rows* since it can use larger intervention sets, be adaptive, utilize randomization, and possibly only work on special graph classes.

E Verification

E.1 Properties of covered edges

Lemma 17 (Properties of covered edges)**.**

1. *Let H be the edge-subgraph induced by covered edges of a DAG G. Then, every vertex in H has at most one incoming edge and thus H is a forest of directed trees.*
2. *If a DAG G is a clique on $n \geq 3$ vertices $v_1, v_2, \ldots, v_n$ with $\pi(v_1) < \pi(v_2) < \ldots < \pi(v_n)$, then $v_1 \to v_2, \ldots, v_{n-1} \to v_n$ are the covered edges of G.*

[15]Typically, they will be shown to succeed with high probability in n: as the size of the graph n increases, the failure probability decays quickly in the form of n^{-c} for some constant $c > 1$.

[16]This will later be obvious through the lenses of covered edges: all covered edges are incident to the root.

Size	Adaptive	Randomized	Lower bound	Reference
1	✗	✓	$\frac{2}{3}n - \frac{1}{3}$ for $n > 3$	[Ebe10]
1	✓	✗	$n - 1$	[EGS06]
$\leq k$	✗	✗	$(\frac{n}{k} - 1) + \frac{n}{2k} \log_2 k$	[EGS12]
$\leq k$	✓	✓	$\frac{n}{2k}$	[SKDV15]
∞	✗	✗	$\log_2 n$	[EGS12]
∞	✗	✓	$\Omega(\log \log n)$	[HLV14]
∞	✓	✓	$\lceil \log_2(\omega(\mathcal{E}(G)) \rceil$	[HB14]

Table 2: Lower bounds on the size ($|\mathcal{I}|$, or $\mathbb{E}[|\mathcal{I}|]$ for randomized algorithms) of the intervention set necessary to fully orient a given essential graph $\mathcal{E}(G)$. The first three columns indicate the setting which the algorithm operates in terms of intervention size, adaptivity, and randomness. Roughly speaking, the setting becomes *easier as we move down the rows* so the lower bounds are *stronger as we move down the rows*. On cliques, [SKDV15] also showed that $\geq n/2$ vertices must be intervened.

3. *If $u \to v$ is a covered edge in a DAG G, then u* cannot *be a sink of any maximal clique of G.*

Proof.

1. Suppose, for a contradiction, that there exists some vertex w with two incoming covered edges $u \to w \leftarrow v$. For $u \to w$ to be covered, we must have $v \to u$. Similarly, for $v \to w$ to be covered, we must have $u \to v$. However, we cannot simultaneously have both $u \to v$ and $v \to u$, as it would lead to a contradiction as G is a DAG. Furthermore, since G is acyclic, it implies that H must also be acyclic. Therefore H is a forest of directed trees.

2. Let $A = \{v_1 \to v_2, v_2 \to v_3, \ldots, v_{n-1} \to v_n\}$ be the set of arcs of interest. For any arc $v_i \to v_{i+1} \in A$, one can check that they share the same parents by the topological ordering π. Consider an arbitrary arc $v_i \to v_j \notin A$. Since $v_i \to v_j \notin A$, there exists $v_k \in V$ such that $\pi(v_i) < \pi(v_k) < \pi(v_j)$. Then, since G is a clique, we must have $v_i \to v_k \to v_j$ and so $v_i \to v_j$ *cannot* be covered since $v_k \in Pa(v_j) \setminus \{v_i\}$ but $v_k \notin Pa(v_i) \setminus \{v_j\}$.

3. Suppose, for a contradiction, that u is a sink of some maximal clique K_h of size h and $u \to v$ is a covered edge. Then, we must have $Pa(v) \setminus \{u\} = Pa(u) \setminus v$. However, that means that $V(K_h) \cup \{v\}$ is a clique of size $h + 1$. Thus, K_h was not a maximal clique. Contradiction.

□

E.2 Characterization via separation of covered edges

Lemma 27 (Necessary)**.** *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. If $\mathcal{I} \subseteq 2^V$ is a verifying set, then $\mathcal{I}$ separates all unoriented covered edge $u \sim v$ of G.*

Proof. Let $u \to v$ be an arbitrary unoriented covered edge in $\mathcal{E}(G^*)$ and $\mathcal{I}$ be an intervention set where u and v are *never* separated by any $S \in \mathcal{I}$. Then, interventions will not orient $u \to v$ and we can only possibly orient it via Meek rules. We check that all four Meek rules will *not* orient $u \to v$:

(R1) For R1 to trigger, we need to have $w \to u \to v$ and $w \not\sim v$ for some vertex $w \in V \setminus \{u, v\}$. However, such a vertex w will imply that $u \to v$ is *not* a covered edge.

(R2) For R2 to trigger, we need to have $u \to w \to v$ for some $w \in V \setminus \{u, v\}$. However, such a vertex w will imply that $u \to v$ is *not* a covered edge.

(R3) For R3 to trigger, we must have $w \sim u \sim x$, $w \to v \leftarrow x$, and $w \not\sim x$ for some $w, x \in V \setminus \{u, v\}$. Since $u \to v$ is a covered edge, we must have $w \to u \leftarrow x$. This implies that both $w \to u \leftarrow x$ appear as v-structures in $\mathcal{E}(G^*)$ and thus R3 will not trigger.

(R4) For R4 to trigger, we must have $w \sim u \sim x$, $w \to x \to v$, and $w \not\sim v$ for some $w, x \in V \setminus \{u, v\}$. Since $u \to v$ is covered, we must have $x \to u$. To avoid directed cycles, it must be the case that $w \to u$. However, this implies that $u \to v$ is *not* covered since $w \to u$ while $w \not\sim v$.

Therefore, $\mathcal{I}$ *cannot* be a verifying set if u and v are *never* separated by any $S \in \mathcal{I}$. □

Lemma 28 (Sufficient)**.** *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. If $\mathcal{I} \subseteq 2^V$ is an intervention set that separates every unoriented covered edge $u \sim v$ of G, then $\mathcal{I}$ is a verifying set.*

Proof. Let $\mathcal{I}$ be an arbitrary intervention set such that every unoriented covered edge $u \sim v$ of G has an set $S \in \mathcal{I}$ that separates u and v. Fix an arbitrary valid vertex permutation $\pi : V \to [n]$ of G. For any $i \in [n]$, define $V_i = \{\pi^{-1}(1), \ldots, \pi^{-1}(i)\} \subseteq V$ as the i smallest vertices according to π's ordering. We argue that any unoriented edges in $\mathcal{E}(G^*)[V_i]$ will be oriented by $\mathcal{I}$ by performing induction on i.

Base case ($i = 1$): There are no edges in $G[V_1]$ so $\mathcal{E}(G^*)[V_1]$ is trivially fully oriented.

Inductive case ($i > 1$): Suppose $v = \pi^{-1}(i)$. By induction hypothesis, $\mathcal{E}(G^*)[V_{i-1}]$ is fully oriented so any unoriented edge in $\mathcal{E}(G^*)[V_i]$ must have the form $u \to v$, where $\pi(u) < \pi(v)$. For any $u \to v$ is an unoriented covered edge in $\mathcal{E}(G^*)[V_i]$, there will be an intervention $S \in \mathcal{I}$ that separates u and v (or both), and hence orient $u \to v$.

Suppose, for a contradiction, that there exists unoriented edges in $\mathcal{E}(G^*)[V_i]$ that are *not* covered edges. Let $u \to v$ be the unoriented edge where $\pi(u)$ is *maximized*. Then, one of the two cases must occur:

Case 1 ($u \to v$ and $\exists w \in V_i$ such that $w \to u$ and $w \not\to v$) Since $\pi(v) > \pi(w)$, we must have $w \not\sim v$. By induction, $w \to u$ will be oriented. So, R1 orients $u \to v$.

Case 2 ($u \to v$ and $\exists w \in V_i$ such that $w \to v$ and $w \not\to u$) If $w \not\sim u$, then $u \to v \leftarrow w$ is a v-structure and $u \to v$ would have been oriented. If $w \sim u$, then we must have $u \to w$ and $\pi(u) < \pi(w)$. By induction, $u \to w$ will be oriented. Since $\pi(u) < \pi(w)$ and $\pi(u)$ is maximized out of all possible unoriented edges in $\mathcal{E}(G^*)[V_i]$ involving v, $w \to v$ must be an oriented edge and will be oriented by $\mathcal{I}$. So, R2 orients $u \to v$.

In either case, $u \to v$ will be oriented. Contradiction. □

Combining Lemma 27 and Lemma 28 gives the following characterization of verifying sets.

Theorem 9. *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. An intervention set $\mathcal{I}$ is a verifying set for G if and only if $\mathcal{I}$ is a set that separates every covered edge of G that is unoriented in $\mathcal{E}(G^*)$.*

E.3 Solving the verification problem

Corollary 10. *Given an essential graph $\mathcal{E}(G^*)$ of an unknown ground truth DAG G^* and a causal DAG $G \in [G^*]$, we can test if $G \stackrel{?}{=} G^*$ by intervening on any verifying set of G. Furthermore, in the worst case,* any *algorithm that correctly resolves $G \stackrel{?}{=} G^*$ needs at least $\nu(G)$ interventions.*

Proof. Using Theorem 9, we know that the minimal verifying set for G is the smallest possible set of interventions $\mathcal{I}$ such that *all* covered edges of G is separated by some intervention $S \in \mathcal{I}$. If the graph is fully oriented after intervening on all $S \in \mathcal{I}$, then it must be the case that $G = G^*$. Otherwise, we will either detect that some edge orientation disagrees with G or there remains some unoriented edge at the end of all our interventions. In the first case, we trivially conclude that $G \neq G^*$. In the second case, Theorem 9 tells us that any such unoriented edge must be an unoriented covered edge of G^* (which was not an unoriented cover edge of G^A) and so we can also conclude that $G \neq G^*$.

Suppose, for a contradiction, that some algorithm $\mathcal{A}$ uses strictly less than $\nu(G)$ interventions to verify a graph G. Then, there exists at least one covered edge $u \to v$ in G is not separated by the interventions used. Define $G'_1 = G$ and G'_2 as G with this covered edge reversed (i.e. $v \to u$ instead). Note that G'_2 is also a DAG in the same MEC due to Lemma 7. We see that $\mathcal{A}$ *cannot* distinguish

between G_1' and G_2' and thus cannot correctly output $G = G'$ or $G \neq G'$ respectively. This is a contradiction, i.e. at least $\nu(G)$ interventions are needed in the worst case. □

E.4 Efficient optimal verification via atomic interventions

Theorem 11. *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. An atomic intervention set $\mathcal{I}$ is a minimal sized verifying set for G if and only if $\mathcal{I}$ is a minimum vertex cover of unoriented covered edges of G. A minimal sized atomic verifying set can be computed in polynomial time.*

Proof. For $|S| = 1$, we see that $\mathcal{I}$ separates every unoriented covered edge in $\mathcal{E}(G)$ if and only if the set $\cup_{S \in \mathcal{I}} S$ is a vertex cover of the unoriented covered edges in $\mathcal{E}(G)$. Lemma 17 tells us that the edge-induced subgraph on covered edges of G is a forest. Thus, one can perform the standard dynamic programming algorithm to compute the minimum vertex cover on each tree. □

E.5 Efficient near-optimal verification via bounded size interventions

We first prove a simple lower bound on the minimum number of non-atomic bounded size interventions (i.e. $|S| \leq k$) needed for verification and then show how to adapt a minimal atomic verifying set to obtain a near-optimal bounded size verifying set.

Lemma 29. *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. Suppose $\mathcal{I}$ is an arbitrary bounded size intervention set. Intervening on vertices in $\cup_{S \in \mathcal{I}} S$ one at a time, in an atomic fashion, can only increase the number of separated covered edges of G.*

Proof. Consider an arbitrary covered edge $u \sim v$ that was seprated by some intervention $S \in \mathcal{I}$. This means that $|\{u, v\} \cap S| = 1$. Without loss of generality, suppose $u \in S$. Then, when we intervene on u in an atomic fashion, we would also separate the edge $u \sim v$. □

Lemma 30. *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. If $\nu_1(G) = \ell$, then $\nu_k(G) \geq \lceil \frac{\ell}{k} \rceil$.*

Proof. A bounded size intervention set of size strictly less than $\lceil \frac{\ell}{k} \rceil$ involves strictly less than ℓ vertices. By Lemma 29, such an intervention set cannot be a verifying set. □

Lemma 31. *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. If $\nu_1(G) = \ell$, then there exists a polynomial time algorithm that computes a bounded size verifying set $\mathcal{I}$ of size $|\mathcal{I}| \leq \lceil \frac{\ell}{k} \rceil + 1$.*

Proof. Consider any atomic verifying set $\mathcal{I}$ of G of size ℓ. By Lemma 17, the edge-induced subgraph on covered edges of G is a forest and is thus 2-colorable.

Split the vertices in $\mathcal{I}$ into partitions according to the 2-coloring. By construction, vertices belonging in the same partite will *not* be adjacent and thus choosing them together to be in an intervention S will *not* reduce the number of separated covered edges. Now, form interventions of size k by greedily picking vertices in $\mathcal{I}$ within the same partite. For the remaining unpicked vertices (strictly less than k of them), we form a new intervention with them. Repeat the same process for the other partite.

This greedy process forms groups of size k and at most 2 groups of sizes, one from each partite. Suppose that we formed z groups of size k in total and two "leftover groups" of sizes x and y, where $0 \leq x, y < k$. Then, $\ell = z \cdot k + x + y$, $\frac{\ell}{k} = z + \frac{x+y}{k}$, and we formed at most $z + 2$ groups. If $0 \leq x + y < k$, then $\lceil \frac{\ell}{k} \rceil = z + 1$. Otherwise, if $k \leq x + y < 2k$, then $\lceil \frac{\ell}{k} \rceil = z + 2$. In either case, we use at most $\lceil \frac{\ell}{k} \rceil + 1$ interventions, each of size $\leq k$.

One can compute a bounded size intervention set efficiently because the following procedures can all be run in polynomial time: (i) checking if each edge is a covered edge; (ii) computing a minimum vertex cover on a tree; (iii) 2-coloring a tree; (iv) greedily grouping vertices into sizes $\leq k$. □

Theorem 12 follows by combining Lemma 30 and Lemma 31.

Theorem 12. *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. If $\nu_1(G) = \ell$, then $\nu_k(G) \geq \lceil \frac{\ell}{k} \rceil$ and there exists a polynomial time algo. to compute a bounded size intervention set $\mathcal{I}$ of size $|\mathcal{I}| \leq \lceil \frac{\ell}{k} \rceil + 1$.*

Observe that there exists graphs and values k such that the optimal bounded size verifying set requires at least $\lceil \frac{\ell}{k} \rceil + 1$, and thus our upper bound is tight in the worst case: Fig. 6 shows there exists a family of graphs (and values k) such that the optimal bounded size verifying set requires $\lceil \frac{\ell}{k} \rceil + 1$. However, we do not have a proof that Theorem 12 is optimal (or counter example that it is not).

Conjecture 32. The construction of bounded size verifying set given in Theorem 12 is optimal.

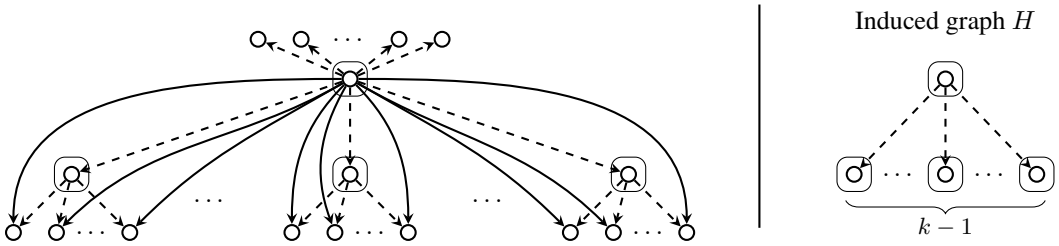


Figure 6: A DAG with its covered edges given in dashed arcs. The edge-induced subgraph of the covered edges is a tree and the minimum vertex cover is all the non-leaf vertices (the boxed vertices) of size ℓ. Denote the graph induced by the boxed vertices by H. Now consider the star graph H on $\ell = k$ nodes with $k-1$ leaves. All the leaf nodes can be put in the same intervention without affecting the separation of any covered edges. However, including the root with any of the leaf nodes in a same intervention will cause covered edges to be unseparated. Thus, using bounded size interventions of size at most k, verifying such a DAG requires at least $\lceil \frac{\ell}{k} \rceil + 1 = 2$ interventions.

E.6 Generalization to minimum cost verifying sets with additive cost structures

Consider an essential graph which is a star graph on n nodes where the leaves have cost 1 and the root has cost significantly larger than n. For atomic verifying sets, we see that the *minimum cost* verifying set to intervene on the leaves one at a time while the *minimum size* verifying set is to simply intervene on the root. Since one may be more preferred over the other, depending on the actual real-life situation, we propose to find a verifying set $\mathcal{I}$ which minimizes

$$\alpha \cdot w(\mathcal{I}) + \beta \cdot |\mathcal{I}| \qquad \text{where } \alpha, \beta \geq 0 \tag{1}$$

so as to explicitly trade-off between the cost and size of the intervention set. This objective also naturally allows the constraint of bounded size interventions by restricting $|S| \leq k$ for all $S \in \mathcal{I}$.

Theorem 13. *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. An atomic verifying set $\mathcal{I}$ for G that minimizes Eq. (1) can be computed in polynomial time.*

Proof. By Theorem 9 and accounting for Eq. (1), we need to compute a *weighted* minimum vertex cover in the edge-induced subgraph on covered edges of G. Efficiency is implied by Lemma 17. □

For bounded size interventions, we show that the ideas in Appendix E.5 translate naturally to give a near-optimal minimal generalized cost verifying set. To prove our lower bound, we first consider an optimal atomic verifying set for a slightly different objective from Eq. (1).

Lemma 33. *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. Let $\mathcal{I}_A$ be an atomic verifying set for G that minimizes $\alpha \cdot w(\mathcal{I}_A) + \frac{\beta}{k} \cdot |\mathcal{I}_A|$ and $\mathcal{I}_B$ be a bounded size verifying set for G that minimizes Eq. (1). Then, $\alpha \cdot w(\mathcal{I}_A) + \frac{\beta}{k} \cdot |\mathcal{I}_A| \leq \alpha \cdot w(\mathcal{I}_B) + \beta \cdot |\mathcal{I}_B|$.*

Proof. Let $\mathcal{I} = \sum_{S \in \mathcal{I}_B} S$ be the atomic verifying set derived from $\mathcal{I}_B$ by treating each vertex as an atomic intervention. Clearly, $w(\mathcal{I}_B) \geq w(\mathcal{I})$ and $k \cdot |\mathcal{I}_B| \geq |\mathcal{I}|$. So,

$$\alpha \cdot w(\mathcal{I}_B) + \beta \cdot |\mathcal{I}_B| \geq \alpha \cdot w(\mathcal{I}) + \frac{\beta}{k} \cdot |\mathcal{I}| \geq \alpha \cdot w(\mathcal{I}_A) + \frac{\beta}{k} \cdot |\mathcal{I}_A|$$

since $\mathcal{I}_A = \text{argmin}_{\text{atomic verifying set } \mathcal{I}'} \left\{ \alpha \cdot w(\mathcal{I}') + \frac{\beta}{k} \cdot |\mathcal{I}'| \right\}$. □

Theorem 14. *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. Suppose the optimal bounded size intervention set that minimizes Eq. (1) costs OPT. Then, there exists a polynomial time algorithm that computes a bounded size intervention set with total cost $OPT + 2\beta$.*

Proof. Let $\mathcal{I}_A$ be an atomic verifying set for G that minimizes $\alpha \cdot w(\mathcal{I}_A) + \frac{\beta}{k} \cdot |\mathcal{I}_A|$ and $\mathcal{I}_B$ be a bounded size verifying set for G that minimizes Eq. (1). Using the polynomial time greedy algorithm in Lemma 31, we construct bounded size intervention set $\mathcal{I}$ by greedily grouping together atomic interventions from $\mathcal{I}_A$. Clearly, $w(\mathcal{I}) = w(\mathcal{I}_A)$ and $|\mathcal{I}| \leq \lceil \frac{|\mathcal{I}_A|}{k} \rceil + 1$. So,

$$\alpha \cdot w(\mathcal{I}) + \beta \cdot |\mathcal{I}| \leq \alpha \cdot w(\mathcal{I}_A) + \beta \cdot \left(\left\lceil \frac{|\mathcal{I}_A|}{k} \right\rceil + 1 \right) \leq \alpha \cdot w(\mathcal{I}_B) + \beta \cdot |\mathcal{I}_B| + 2\beta = OPT + 2\beta$$

where the second inequality is due to Lemma 33. □

F Search

We begin by proving a strengthened version of [SMG+20]'s lower bound.

Note that we will be discussing only atomic interventions in Lemma 21, so notation such as $\mathcal{I} \cap V(H)$ makes sense for sets $\mathcal{I}, V(H) \subseteq V$.

Definition 34 (Moral DAG, Definition 3 of [SMG+20])**.** A graph G is a *moral DAG* if its essential graph only has a single chain component. That is, after removing directed edges, there is only one single connected component remaining.

Lemma 35 (Lemma 6 of [SMG+20])**.** *Let G be a moral DAG. Then, $\nu_1(G) \geq \lfloor \frac{\omega(\textit{skeleton}(G))}{2} \rfloor$.*

Lemma 21. *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. Then,*

$$\nu_1(G) \geq \max_{\mathcal{I} \subseteq V} \sum_{H \in CC(\mathcal{E}_{\mathcal{I}}(G^*))} \left\lfloor \frac{\omega(H)}{2} \right\rfloor$$

Proof. Consider be an arbitrary set of atomic interventions $\mathcal{I} \subseteq V$ and the resulting $\mathcal{I}$-essential graph $\mathcal{E}_{\mathcal{I}}(G^*)$. Let $H \in CC(\mathcal{E}_{\mathcal{I}}(G^*))$ be an arbitrary chain component.

Let $\mathcal{I}' \subseteq V$ be an arbitrary atomic verifying set of G. Then, $\mathcal{E}_{\mathcal{I}'}(G^*) = G$ and thus $\mathcal{E}_{\mathcal{I}'}(G^*)[V(H)] = G[V(H)]$. Then,

$$\mathcal{E}_{(\mathcal{I}' \setminus \mathcal{I}) \cap V(H)}(G[V(H)]) = \mathcal{E}_{\mathcal{I} \cup (\mathcal{I}' \setminus \mathcal{I})}(G)[V(H)] = \mathcal{E}_{\mathcal{I}'}(G)[V(H)] = G[V(H)]$$

where the first equality is due to Lemma 3 and the last equality is because $\mathcal{I}'$ is a verifying set of G. So, $(\mathcal{I}' \setminus \mathcal{I}) \cap V(H)$ is a verifying set for $G[V(H)]$, and so is $\mathcal{I}' \cap V(H)$. Thus, by minimality of ν_1, we have $\nu_1(G[V(H)]) \leq |\mathcal{I}' \cap V(H)|$ for *any* atomic verifying set $\mathcal{I}' \subseteq V$ of G.

Since $H \in CC(\mathcal{E}_{\mathcal{I}}(G^*))$, the graph $G[V(H)]$ is a moral DAG. Since H is a subgraph of $G[V(H)]$, $\omega(H) \leq \omega(G[V(H)])$. Thus, by Lemma 35, we have $\nu_1(G[V(H)]) \geq \lfloor \frac{\omega(G[V(H)])}{2} \rfloor \geq \lfloor \frac{\omega(H)}{2} \rfloor$.

Now, suppose $\mathcal{I}^*$ is a minimal size verifying set of G. Then,

$$\nu_1(G) = |\mathcal{I}^*| \geq \sum_{H \in CC(\mathcal{E}_{\mathcal{I}}(G^*))} |\mathcal{I}^* \cap V(H)| \geq \sum_{H \in CC(\mathcal{E}_{\mathcal{I}}(G^*))} \nu_1(G[V(H)]) \geq \sum_{H \in CC(\mathcal{E}_{\mathcal{I}}(G^*))} \left\lfloor \frac{\omega(H)}{2} \right\rfloor$$

The claim follows by taking the maximum over all possible atomic interventions $\mathcal{I} \subseteq V$. □

For convenience, we reproduce Algorithm 1 below.

Lemma 36. *Algorithm 1 terminates after at most $\mathcal{O}(\log n)$ iterations.*

Proof. Consider an arbitrary iteration i and chain component H with 1/2-clique separator K_H.

Edges within K_H will be fully oriented by Step 6. We now argue that any edge with exactly one endpoint in K_H will be oriented by Algorithm 1. For $k'_H = 1$, the algorithm intervenes on all nodes in K_H and thus such edges will be oriented trivially. For $k'_H > 1$, the additional $\leq |K_H|/k'_H$

Algorithm 1 Search algorithm via graph separators.

Input: Essential graph $\mathcal{E}(G^*)$, intervention size k. **Output**: A fully oriented graph $G \in [G^*]$.
Initialize $i = 0$ and $\mathcal{I}_0 = \emptyset$.
while $\mathcal{E}_{\mathcal{I}_i}(G^*)$ still has undirected edges **do**
  For each $H \in CC(\mathcal{E}_{\mathcal{I}_i}(G^*))$ of size $|H| \geq 2$, find a 1/2-clique separator K_H using Theorem 20. Define $Q = \{K_H\}_{H \in CC(\mathcal{E}_{\mathcal{I}_i}(G^*)), |H| \geq 2}$ as the union of clique separator nodes.
  if $k = 1$ or $|Q| = 1$ **then** Define $C_i = Q$ as an atomic intervention set.
  else Define $k' = \min\{k, |Q|/2\}$, $a = \lceil |Q|/k' \rceil \geq 2$, and $\ell = \lceil \log_a n \rceil$. Compute labelling scheme of [SKDV15, Lemma 1] on Q with $(|Q|, k', a)$, and define $C_i = \{S_{x,y}\}_{x \in [\ell], y \in [a]}$, where $S_{x,y} \subseteq Q$ is the subset of vertices whose x^{th} letter in the label is y.
  Update $i \leftarrow i + 1$, intervene on C_i to obtain $\mathcal{E}_{\mathcal{I}_i}(G^*)$, and update $\mathcal{I}_i \leftarrow \mathcal{I}_{i-1} \cup C_i$.
end while

interventions *after* the bounded size intervention strategy of [SKDV15] ensures that *every* edge with exactly one endpoint in K_H will be separated. Thus, after each iteration, the only remaining unoriented edges lie completely within the separated components that are of half the size.

Since the algorithm always recurse on graphs of size at least half the previous iteration, we see that $|H| \leq n/2^i$ for any $H \in CC(\mathcal{E}_{\mathcal{I}_i}(G^*))$. Thus, all chain components will become singletons after $\mathcal{O}(\log n)$ iterations and the algorithm terminates with a fully oriented graph. □

Theorem 15. *Fix an essential graph $\mathcal{E}(G^*)$ with an unknown underlying ground truth DAG G^*. Given $k = 1$, Algorithm 1 runs in polynomial time and computes an atomic intervention set $\mathcal{I}$ in a deterministic and adaptive manner such that $\mathcal{E}_{\mathcal{I}}(G^*) = G^*$ and $|\mathcal{I}| \in \mathcal{O}(\log(n) \cdot \nu_1(G^*))$.*

Proof. Algorithm 1 runs in polynomial time because 1/2-clique separators can be computed efficiently (see Theorem 20).

Fix an arbitrary iteration i of Algorithm 1 and let G_i be the partially oriented graph obtained after intervening on $\mathcal{I}_i$. By Lemma 21, $\sum_{H \in CC(\mathcal{E}_{\mathcal{I}_i}(G^*))} \lfloor \frac{\omega(H)}{2} \rfloor \leq \nu_1(G^*)$. By definition of ω, we always have $|K_H| \leq \omega(H)$. Thus, Algorithm 1 uses at most $2 \cdot \nu_1(G^*)$ interventions in each iteration.

By Lemma 36, there are $\mathcal{O}(\log n)$ iterations and so $\mathcal{O}(\log(n) \cdot \nu_1(G^*))$ atomic interventions are used by Algorithm 1. □

Lemma 37 (Lemma 1 of [SKDV15])**.** *Let (n, k, a) be parameters where $k \leq n/2$. There is a polynomial time labeling scheme that produces distinct ℓ length labels for all elements in $[n]$ using letters from the integer alphabet $\{0\} \cup [a]$ where $\ell = \lceil \log_a n \rceil$. Further, in every digit (or position), any integer letter is used at most $\lceil n/a \rceil$ times. This labelling scheme is a separating system: for any $i, j \in [n]$, there exists some digit $d \in [\ell]$ where the labels of i and j differ.*

Theorem 16. *Fix an essential graph $\mathcal{E}(G^*)$ with an unknown underlying ground truth DAG G^*. Given $k > 1$, Algorithm 1 runs in polynomial time and computes a bounded size intervention set $\mathcal{I}$ in a deterministic and adaptive manner such that $\mathcal{E}_{\mathcal{I}}(G^*) = G^*$ and $|\mathcal{I}| \in \mathcal{O}(\log(n) \cdot \log(k) \cdot \nu_k(G^*))$.*

Proof. Algorithm 1 runs in polynomial time because 1/2-clique separators can be computed efficiently (see Theorem 20). The label computation of [SKDV15, Lemma 1] also runs in polynomial time.

Fix an arbitrary iteration i of Algorithm 1 and let G_i be the partially oriented graph obtained after intervening on $\mathcal{I}_i$. Since $\mathcal{E}(G_i) \subseteq \mathcal{E}(G^*)$, we see that $\nu_1(G_i) \leq \nu_1(G^*)$. Algorithm 1 intervenes on

$$|C_i| \leq \left\lceil \frac{|Q|}{k'} \right\rceil \cdot \left\lceil \log_{\lceil \frac{|Q|}{k'} \rceil} |Q| \right\rceil$$

sets of bounded size interventions, where $k' = \min\{k, |Q|/2\} > 1$.

Since $\nu_1(G^*) \geq \sum_{H \in CC(\mathcal{E}(G^*))} \lfloor \omega(H)/2 \rfloor$, we know from Lemma 30 that

$$\nu_k(G^*) \geq \left\lceil \frac{\nu_1(G^*)}{k} \right\rceil \geq \left\lceil \frac{1}{k} \cdot \sum_{H \in CC(\mathcal{E}(G^*))} \left\lfloor \frac{\omega(H)}{2} \right\rfloor \right\rceil .$$

Since Q is the union of clique separator nodes, we have that $|Q| \leq \sum_{H \in CC(\mathcal{E}(G^*))} \omega(H)$ and so $\nu_k(G^*) \in \Omega(|Q|/k)$. Note that $\nu_k(G^*) \geq 1$ always.

Case 1: $k \leq |Q|/2$**.** Then, $k' = k$ and

$$|C_i| \leq \left\lceil \frac{|Q|}{k} \right\rceil \cdot \left\lceil \log_{\lceil \frac{|Q|}{k} \rceil} |Q| \right\rceil \leq \left\lceil \frac{|Q|}{k} \right\rceil \cdot \left\lceil \frac{\log |Q|}{\log \frac{|Q|}{k}} \right\rceil \leq \left\lceil \frac{|Q|}{k} \right\rceil \cdot \lceil \log(k) + 1 \rceil \in \mathcal{O}\left(\frac{|Q|}{k} \cdot \log(k) \right)$$

Case 2: $k \geq |Q|/2$**.** Then, $k' = |Q|/2$ and

$$|C_i| \leq 2 \cdot \lceil \log_2 |Q| \rceil \in \mathcal{O}(\log(k))$$

In either case, we see that $|C_i| \in \mathcal{O}\left(\nu_k(G^*) \cdot \log k\right)$. By Lemma 36, there are $\mathcal{O}(\log n)$ iterations and so $\mathcal{O}(\log(n) \cdot \log(k) \cdot \nu_k(G^*))$ bounded size interventions are used by Algorithm 1. □

G Example illustrating the power of Lemma 21

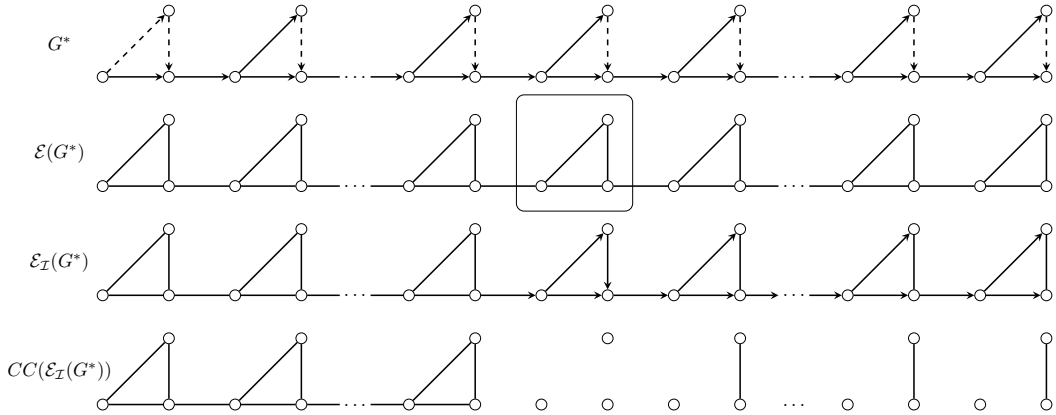


Figure 7: A DAG G^* where minimum vertex cover of the unoriented covered edges (dashed arcs) is much larger than the size of the maximal clique (triangle): $\nu_1(G^*) \approx n$ while the lower bound of [SMG+20] on $\mathcal{E}(G^*)$ is a constant. Let $\mathcal{I}$ be an atomic intervention set on the middle triangle, i.e. the three vertices boxed up in $\mathcal{E}(G^*)$. The partially directed graph $\mathcal{E}_{\mathcal{I}}(G^*)$ shows the learnt arc directions after intervening on $\mathcal{I}$ and applying Meek rules. Applying the lower bound of [SMG+20] on $CC(\mathcal{E}_{\mathcal{I}}(G^*))$ now gives a much stronger lower bound of $\approx n$ due to the single edge components.

H Experiments and implementation

Our code and entire experimental setup is available at https://github.com/cxjdavin/verification-and-search-algorithms-for-causal-DAGs.

H.1 Implementation details

Verification We implemented our verification algorithm and tested its correctness on some well-known graphs such as cliques and trees for which we know the exact verification number.

Search We use `FAST CHORDAL SEPARATOR` algorithm in [GRE84] to compute a chordal graph separator. This algorithm first computes a perfect elimination ordering of a given chordal graph and we use Eppstein's `LexBFS` implementation (https://www.ics.uci.edu/~eppstein/PADS/LexBFS.py) to compute such an ordering.

H.2 Experiments

We base our evaluation on the experimental framework of [SMG+20] (https://github.com/csquires/dct-policy) which empirically compares atomic intervention policies. The experiments are conducted on an Ubuntu server with two AMD EPYC 7532 CPU and 256GB DDR4 RAM.

H.2.1 Synthetic graph classes

The synthetic graphs are random connected DAGs whose essential graph is a single chain component (i.e. moral DAGs in [SMG+20]'s terminology). Below, we reproduce the synthetic graph generation procedure from [SMG+20, Section 5].

1. Erdős-Rényi styled graphs
   These graphs are parameterized by 2 parameters: n and density ρ. Generate a random ordering σ over n vertices. Then, set the in-degree of the nth vertex (i.e. last vertex in the ordering) in the order to be $X_n = \max\{1, \texttt{Binomial}(n-1, \rho)\}$, and sample X_n parents uniformly form the nodes earlier in the ordering. Finally, chordalize the graph by running the elimination algorithm of [KF09] with elimination ordering equal to the reverse of σ.
2. Tree-like graphs
   These graphs are parameterized by 4 parameters: n, degree d, $e_{\min}$, and $e_{\max}$. First, generate a complete directed d-ary tree on n nodes. Then, add $\texttt{Uniform}(e_{\min}, e_{\max})$ edges to the tree. Finally, compute a topological order of the graph by DFS and triangulate the graph using that order.

H.2.2 Algorithms benchmarked

The following algorithms perform *atomic interventions*. Our algorithm `separator` perform atomic interventions when given $k = 1$ and *bounded size interventions* when given $k > 1$.

`random`: A baseline algorithm that repeatedly picks a random non-dominated node (a node that is incident to some unoriented edge) from the interventional essential graph

`dct`: `DCT Policy` of [SMG+20]

`coloring`: `Coloring` of [SKDV15]

`opt_single`: `OptSingle` of [HB14]

`greedy_minmax`: `MinmaxMEC` of [HG08]

`greedy_entropy`: `MinmaxEntropy` of [HG08]

`separator`: Our Algorithm 1. It takes in a parameter k to serve as an upper bound on the number of vertices to use in an intervention.

H.2.3 Metrics measured

Each experiment produces 4 plots measuring "average competitive ratio", "maximum competitive ratio", "intervention count", and "time taken". For any fixed setting, 100 synthetic DAGs are generated as G^* for testing, so we include error bars for "average competitive ratio", "average intervention count", and "time taken" in the plots. For all metrics, "lower is better".

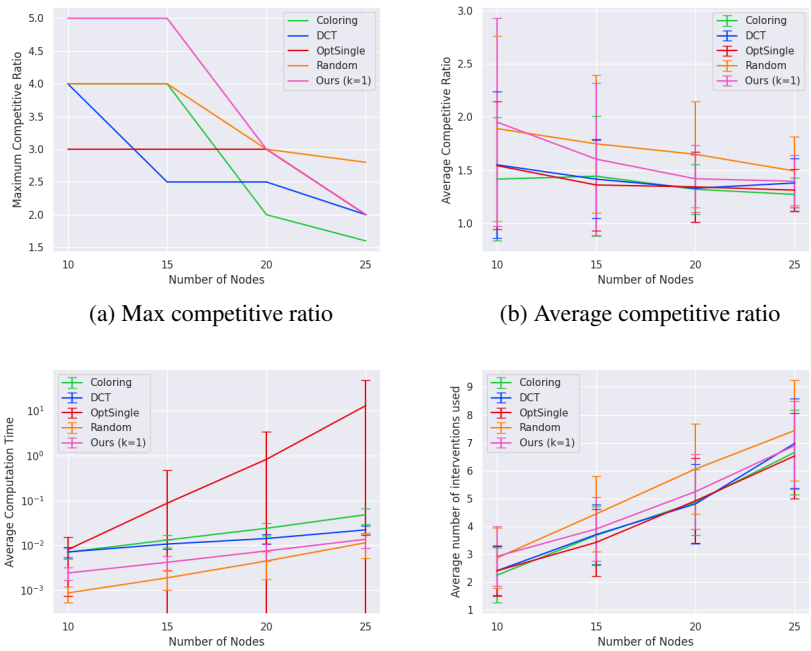
The competitive ratio for an input DAG G^* is measured in terms of *total atomic interventions used to orient the essential graph of G^* to become G^**, divided by *minimum number of interventions needed to orient G^** (i.e. the verification number $\nu_1(G^*)$ of G^*). For non-atomic interventions, we know (Lemma 30) that $\nu_k(G^*) \geq \lceil \nu_1(G^*)/k \rceil$, so we use $\lceil \nu_1(G^*)/k \rceil$ as the denominator of the competitive ratio computation. While the competitive ratio increases as k increases (Theorem 16), the number of interventions used decreases as k increases.

Time is measured as the total amount of time taken to finish computing the nodes to intervene and performing the interventions. Note that our algorithm can beat `random` in terms of runtime in some cases because `random` uses significantly more interventions and hence more overall computation.

H.2.4 Experimental results

Qualitatively, our Algorithm 1 with $k = 1$ has a similar competitive ratio to the current best-known atomic intervention policies in the literature (`DCT` and `Coloring`) while running significantly faster for some graphs (roughly $\sim$10x faster on tree-like graphs).

Experiment 1 Graph class 1 with $n \in \{10, 15, 20, 25\}$ and density $\rho = 0.1$. This is the same setup as [SMG+20]. Additionally, we run Algorithm 1 with $k = 1$. See Fig. 8.



(a) Max competitive ratio

(b) Average competitive ratio

(c) Time

(d) Average interventions count

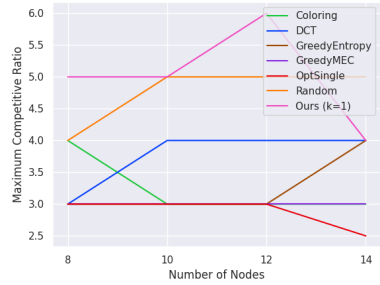
Figure 8: Plots for experiment 1

Experiment 2 Graph class 1 with $n \in \{8, 10, 12, 14\}$ and density $\rho = 0.1$. This is the same setup as [SMG+20]. Additionally, we run Algorithm 1 with $k = 1$. Note that this is the same graph class as experiment 1, but on smaller graphs because some slower algorithms are being run. See Fig. 9.

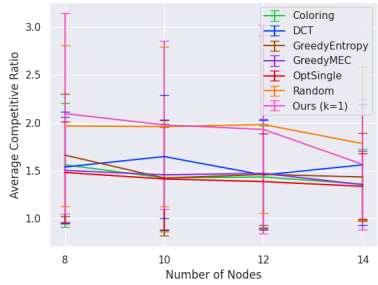
Experiment 3 Graph class 2 with $n \in \{100, 200, 300, 400, 500\}$ and $(\text{degree}, e_{\min}, e_{\max}) = (4, 2, 5)$. This is the same setup as [SMG+20]. Additionally, we run Algorithm 1 with $k = 1$. See Fig. 10.

Experiment 4 Graph class 1 with $n \in \{10, 15, 20, 25\}$ and density $\rho = 0.1$. We run Algorithm 1 with $k \in$ 1,2,3,5 on the same graph class as experiment 1, but on larger graphs. See Fig. 11.

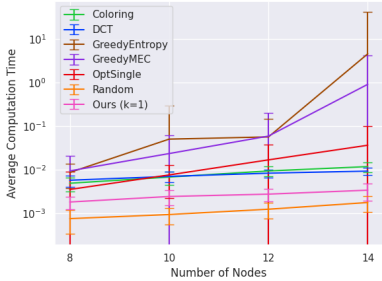
Experiment 5 Graph class 2 with $n \in \{100, 200, 300, 400, 500\}$ and $(\text{degree}, e_{\min}, e_{\max}) = (40, 20, 50)$. We run Algorithm 1 with $k \in \{1, 2, 3, 5\}$ on the same graph class as experiment 3, but on denser graphs. See Fig. 12.



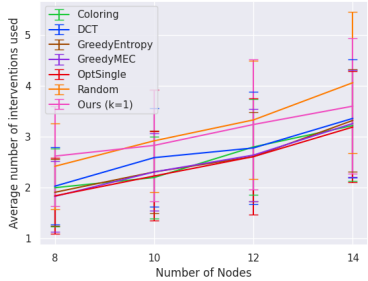
(a) Max competitive ratio



(b) Average competitive ratio

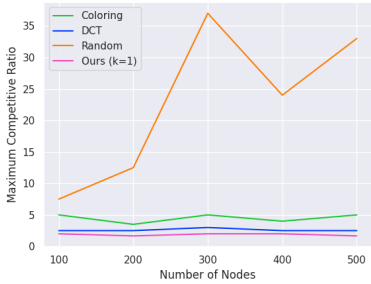


(c) Time

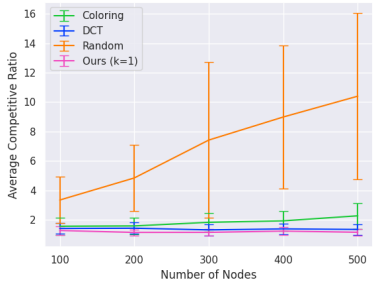


(d) Average interventions count

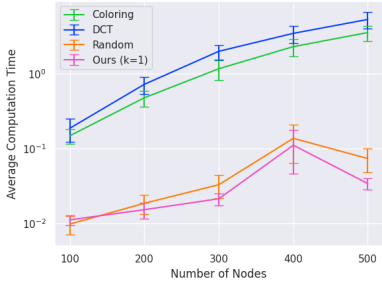
Figure 9: Plots for experiment 2



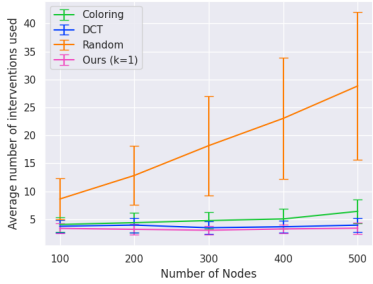
(a) Max competitive ratio



(b) Average competitive ratio

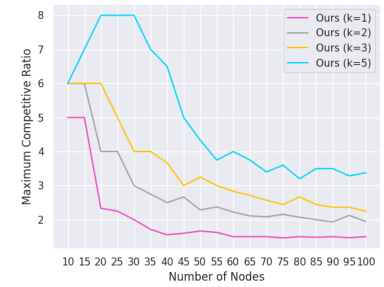


(c) Time

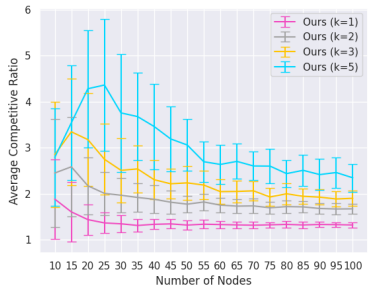


(d) Average interventions count

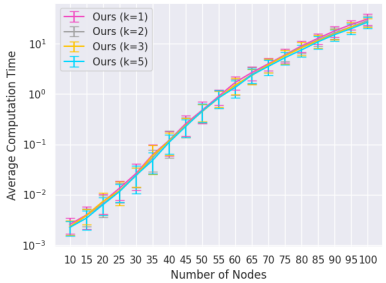
Figure 10: Plots for experiment 3



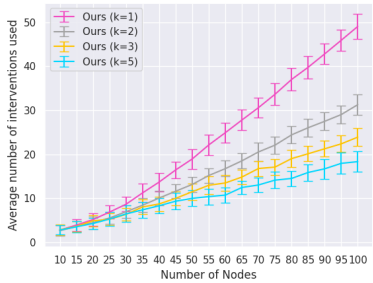
(a) Max competitive ratio



(b) Average competitive ratio

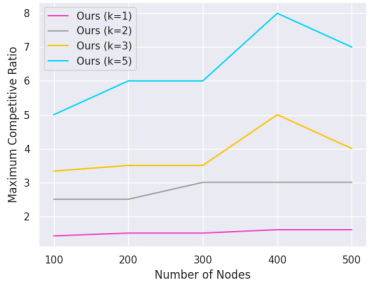


(c) Time

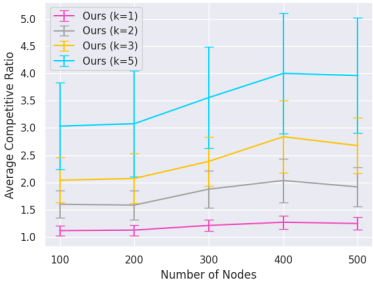


(d) Average interventions count

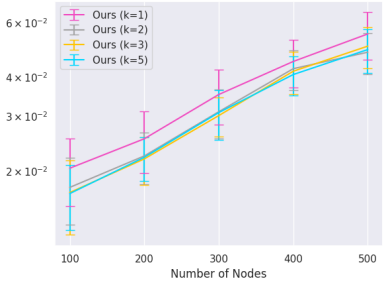
Figure 11: Plots for experiment 4



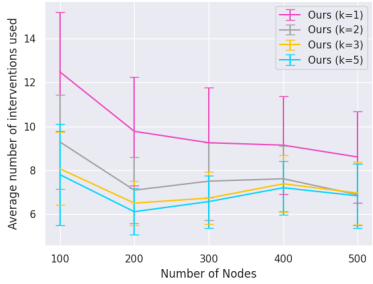
(a) Max competitive ratio



(b) Average competitive ratio



(c) Time



(d) Average interventions count

Figure 12: Plots for experiment 5