New metrics and search algorithms for weighted causal DAGs

Davin Choo [*] [1] **Kirankumar Shiragur** [*] [2]

Abstract

Recovering causal relationships from data is an important problem. Using observational data, one can typically only recover causal graphs up to a Markov equivalence class and additional assumptions or interventional data are needed for complete recovery. In this work, under some standard assumptions, we study causal graph discovery via *adaptive interventions with node-dependent interventional costs*. For this setting, we show that no algorithm can achieve an approximation guarantee that is asymptotically better than linear in the number of vertices with respect to the verification number; a well-established benchmark for adaptive search algorithms. Motivated by this negative result, we define a *new benchmark* that captures the worst-case interventional cost for any search algorithm. Furthermore, with respect to this new benchmark, we provide adaptive search algorithms that achieve logarithmic approximations under various settings: atomic, bounded size interventions and generalized cost objectives.

1. Introduction

Causal discovery is a fundamental problem that has found applications in a wide range of fields, including biology/medicine/genetics (King et al., 2004; Cho et al., 2016; Tian, 2016; Sverchkov & Craven, 2017; Rotmensch et al., 2017; Pingault et al., 2018; de Campos et al., 2019), epidemiology, philosophy (Reichenbach, 1956; Woodward, 2005; Eberhardt & Scheines, 2007), and econometrics (Hoover, 1990; Rubin & Waterman, 2006). In most of these applications, directed acyclic graphs (DAGs) are used to represent the causal relationships and the goal is to recover the underlying causal graph from data. It is well known that using observational data, the causal structure can only be learned up to its Markov equivalence class (MEC) and additional assumptions or interventional data is required for the recovery of the ground truth causal graph. Here, we focus our attention on causal discovery using interventions.

There is a rich literature on causal discovery from interventional data, which can be broadly classified into two categories: adaptive (Shanmugam et al., 2015; Greenewald et al., 2019; Squires et al., 2020; Choo et al., 2022; Choo & Shiragur, 2023) versus non-adaptive (Eberhardt et al., 2005; 2006; Eberhardt, 2010; Hu et al., 2014) approaches. Given an essential graph, *non-adaptive* algorithms have to decide beforehand a collection of interventions such that *any* plausible causal graph can be recovered while *adaptive* algorithms can decide on interventions sequentially while using information gleaned from past interventions. Adaptive algorithms are powerful and in some cases, the interventional cost of an optimal adaptive algorithm is exponentially better than any non-adaptive algorithms[1]. While the non-adaptive setting is pretty well understood even in the most general setting of node dependent vertex costs, researchers have only recently made progress on the adaptive front in the special case of unit vertex costs. Unfortunately, unit vertex costs fail to capture many real world scenarios where performing interventions can have varying costs (e.g. it is less costly to force someone to sleep eight hours than to force someone to run ten miles), are unethical (e.g. force someone to smoke), or even practically impossible. See (King et al., 2004; Sverchkov & Craven, 2017; Ness et al., 2018; Lindgren et al., 2018) for more applications of causal learning in settings where interventions have different costs.

Problem setup Motivated by the power of adaptivity and broad applicability of varying costs, in this work, we study causal discovery via *adaptive* interventions with the goal of recovering the true underlying causal graph given the observational MEC while minimizing the total interventional cost when vertices may have *differing* interventional cost.

Under standard assumptions of causal sufficiency, faithfulness and infinite sample regime, in addition to the *search* problem defined above, recent works (Squires et al., 2020; Choo et al., 2022; Choo & Shiragur, 2023) have also stud-

[*] Equal contribution [1] School of Computing, National University of Singapore [2] Broad Institute of MIT and Harvard. Correspondence to: Davin Choo <davin@u.nus.edu>, Kirankumar Shiragur <shiragur@stanford.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

[1] For tree causal graphs, an adaptive algorithm only needs $\mathcal{O}(\log n)$ interventions to recover it while non-adaptive algorithms require $\Omega(n)$ interventions in some cases. See Appendix A.1.

ied a related fundamental problem for the adaptive setting known as the *verification* problem. Given a MEC of an unknown ground truth causal graph G^* and a graph G from the MEC, the goal of the verification problem is determine whether G is G^*. By plugging in G with G^* in the verification problem, we see that the optimal solution to the verification is a natural lower bound for the search problem. We denote the minimum *size* and minimum *cost* solutions to the verification problem as $\nu(G^*)$ and $\overline{\nu}(G^*)$ respectively.

For the special case of unit cost at each vertex, where $\nu(G^*) = \overline{\nu}(G^*)$, (Choo et al., 2022) recently gave an adaptive search algorithm that recovers G^* by performing at most $\mathcal{O}(\log n \cdot \nu(G^*))$ atomic interventions[2], which is only a logarithmic factor worse than necessary. Furthermore, they also argue that no algorithm can achieve an asymptotically better approximation ratio than $\mathcal{O}(\log n)$ with respect to the $\nu(G^*)$ for all the causal graphs G^*. In light of these results, it is natural to ask if such results also hold when vertices have different interventional costs.

While efficient algorithms for the verification and *non-adaptive* search problems with varying vertex costs, and adaptive search problem for *unit vertex costs* are known, to the best of our knowledge, there is no existing efficient *adaptive search algorithm for varying vertex costs*. Existing approaches for the unweighted setting do *not* extend to the weighted setting due to two major difficulties: proving lower bounds for the benchmark, and designing algorithms that are competitive with it. For the lower bound, existing methods only have guarantees with respect to the clique numbers of chain components and are oblivious to individual vertex costs. On the other hand, existing adaptive search algorithms do not account for vertex weights[3]. In fact, we can even show that the previously considered benchmark of the verification number is no longer meaningful in the context of weighted causal graphs. More formally, we prove that no algorithm (even with infinite computational power) can achieve an asymptotically better approximation than $\mathcal{O}(n)$ with respect to the verification cost $\overline{\nu}(G^*)$ for all ground truth causal graphs on n nodes. Therefore, $\overline{\nu}(G^*)$ is too strong and an unreasonable benchmark[4] to compare against in the weighted setting. Motivated by this negative result, we propose the following new benchmark

$$\overline{\nu}^{\max}(G^*) = \max_{G \in [G^*]} \overline{\nu}(G)$$

which captures the intuition that any algorithm has to grapple with the worst-case causal graph in the given MEC[5]. Using this new benchmark, we then provide adaptive search algorithms that are competitive against the $\overline{\nu}^{\max}(G^*)$.

Our main contributions are summarized as follows:

1. We argue that $\overline{\nu}(G^*)$ is not a good benchmark.
2. Define a new benchmark $\overline{\nu}^{\max}(G^*)$ that captures the worst case interventional cost for any search algorithm.
3. Provide an adaptive search algorithm that is $\mathcal{O}(\log^2 n)$ competitive to $\overline{\nu}^{\max}(G^*)$ in the atomic setting.
4. Extend our search results to bounded size interventions and for a generalized cost function that explicitly trade-offs the *number* and *cost* of interventions, with $\nu^{\max}(G^*)$ and $\overline{\nu}^{\max}(G^*)$ being special cases.

Outline of paper We give preliminaries and related work in Section 2. Results are stated in Section 3 and we provide a proof sketch of these results in Section 4. Some empirical results are shown in Section 5 and source code is provided in the supplementary materials. Full proofs and further experimental details are provided in the appendix.

2. Preliminaries

We write $\{1, \ldots, n\}$ as $[n]$ and hide absolute constant multiplicative factors in n using standard asymptotic notations. For any set A, we denote its powerset by 2^A. Throughout, we denote the (unknown) ground truth DAG by G^*.

Graph preliminaries

Let $G = (V, E)$ be a graph on $|V| = n$ vertices. We use $V(G)$, $E(G)$ and $A(G) \subseteq E(G)$ to denote its vertices, edges, and oriented arcs respectively. G is said to be directed or fully oriented if $A(G) = E(G)$, and partially oriented otherwise. For $u, v \in V$, we write $u \sim v$ if these vertices are connected and $u \not\sim v$ otherwise. We use $u \to v$ or $u \leftarrow v$ to specify the arc directions. For any subset $V' \subseteq V$ and $E' \subseteq E$, $G[V']$ and $G[E']$ denote the vertex-induced and edge-induced subgraphs respectively. Consider a vertex $v \in V$ in a directed graph, let $\texttt{Pa}(v), \texttt{Anc}(v), \texttt{Des}(v)$ denote the parents, ancestors and descendants of v respectively.

The *skeleton* $\text{skel}(G)$ of a (partially oriented) graph G refers to the underlying graph where all edges are made undirected. A *v-structure* refers to three distinct vertices $u, v, w \in V$

[2] Interventions that only involve a single vertex each.

[3] For instance, the algorithm of (Choo et al., 2022) searches for clique separators and intervenes on all the vertices in these clique separators. However, we show that (see Theorem 4.1) one cannot always intervene on the costliest vertex in a clique if we hope to have any theoretical guarantees; this is reflected in one of our algorithmic subroutines (see Algorithm 2).

[4] A recent work on subset verification and search (Choo & Shiragur, 2023) also remarked that comparing against an algorithm that *knows* G^* can be overly pessimistic, and suggested that one should "compare against the "best" algorithm that does *not* know G^*". This is consistent with our formulation of taking the maximum over all DAGs within the same Markov equivalence class.

[5] Our benchmark differs from the notion of separating systems studied in the non-adaptive search literature. See Appendix A.2.

such that $u \to v \leftarrow w$ and $u \not\sim w$. 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 $G[E \setminus A]$ obtained by removing all arcs from a chain graph G, each connected component is called a *chain component*. We use $CC(G)$ to denote the set of chain components, where each $H \in CC(G)$ is a subgraph of G and $V = \dot{\bigcup}_{H \in CC(G)} V(H)$. For any partially directed graph, an *acyclic completion* or *consistent extension* refers to an assignment of edge directions to unoriented edges such that the resulting fully directed graph has no directed cycles; we say that a DAG G is *consistent* with a partially directed graph H if G is an acyclic completion of H.

DAGs are fully oriented chain graphs, where vertices represent random variables and the joint probability density f factorizes according to the Markov property: $f(v_1, \ldots, v_n) = \prod_{i=1}^{n} f(v_i \mid \texttt{Pa}(v))$. We can associate a (not necessarily unique) *valid permutation* $\pi : V \to [n]$ to any (partially directed) DAG such that oriented arcs (u, v) satisfy $\pi(u) < \pi(v)$ and unoriented arcs $\{u, v\}$ can be oriented as $u \to v$ without forming directed cycles when $\pi(u) < \pi(v)$. A DAG is called a *moral DAG* if it has no v-structures, in which case its essential graph is just its skeleton. Moral DAGs have a unique source node (a node without incoming arcs), and any subgraph of it is also a moral DAG.

For any DAG G, we denote its *Markov equivalence class* (MEC) by $[G]$ and *essential graph* by $\mathcal{E}(G)$. DAGs in the same MEC $[G]$ have the same skeleton and essential graph $\mathcal{E}(G)$ is a partially directed graph such that an arc $u \to v$ is directed if $u \to v$ in *every* DAG in MEC $[G]$, and an edge $u \sim v$ is undirected if there exists two DAGs $G_1, G_2 \in [G]$ such that $u \to v$ in G_1 and $v \to u$ in G_2. It is known that two graphs are Markov equivalent if and only if they have the same skeleton and v-structures (Verma & Pearl, 1990; Andersson et al., 1997). An arc $u \to v$ is a *covered edge* (Chickering, 1995) if $\texttt{Pa}(u) = \texttt{Pa}(v) \setminus \{u\}$.

We now give a definition and result for graph separators.

Definition 2.1 (α-separator and α-clique separator, Definition 19 from (Choo et al., 2022))**.** 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.

Theorem 2.2 ((Gilbert et al., 1984), 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 involving at most $p - 1$ vertices. The clique C can be computed in $\mathcal{O}(|E|)$ time.*

Interventions and verifying sets

An *intervention* $S \subseteq V$ is an experiment where all variables $s \in S$ are forcefully set to some value, independent of the underlying causal structure. An intervention is *atomic* if $|S| = 1$ and *bounded* if $|S| \leq k$ for some $k > 0$; observational data is a special case where $S = \emptyset$. The effect of interventions is formally captured by Pearl's do-calculus (Pearl, 2009). We call any $\mathcal{I} \subseteq 2^V$ a *intervention set*. An *ideal intervention* on $S \subseteq V$ in G induces an interventional graph G_S where all incoming arcs to vertices $v \in S$ are removed (Eberhardt et al., 2005). It is known that intervening on S allows us to infer the edge orientation of any edge cut by S and $V \setminus S$ (Eberhardt, 2007; Hyttinen et al., 2013; Hu et al., 2014; Shanmugam et al., 2015; Kocaoglu et al., 2017). 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 $S \in \mathcal{I}$. In Appendix C, we give some well-known properties about interventional essential graphs. Here, we highlight one such result that we will later use: intervening on a node v in a moral DAG will orient any arcs $u \to w$ where u is an ancestor of v and w is a descendant of v.

Lemma 2.3 (Lemma 34 of (Choo & Shiragur, 2023))**.** *Let $G = (V, E)$ be a moral DAG. Intervening on vertex w orients all edges $u \to v$ with $w \in \texttt{Des}(u) \cap \texttt{Anc}(v)$.*

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 B). 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 $S \subseteq V$.

Definition 2.4 (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 $w(\mathcal{I}') < w(\mathcal{I})$).

For DAG G and upper bound $k \geq 1$, the *minimum verification number* $\nu_k(G)$ and the *minimum verification cost* $\overline{\nu}_k(G)$ denote the size/cost of the minimum size/cost verifying set respectively. Note that atomic interventions are a special case of bounded size interventions with $k = 1$.

Similar to (Kocaoglu et al., 2017; Ghassami et al., 2018; Lindgren et al., 2018; Addanki et al., 2020), we consider additive vertex costs where each $v \in V$ has an associated intervention cost $w(v)$ in this work. The cost of an intervention $S \subseteq V$ is simply the sum of the vertices involved and the cost of an intervention set $\mathcal{I} \subseteq 2^V$ is the sum of the intervention costs, i.e. $w(\mathcal{I}) = \sum_{S \in \mathcal{I}} w(S) = \sum_{S \in \mathcal{I}} \sum_{v \in S} w(v)$. Since treating a bounded size intervention as k individual atomic interventions can only recover more information, we aim to optimize the following generalized cost function to explicitly trade-off between the cost and size of the intervention set:

$$\max_{G \in [G^*]} \min_{\substack{\mathcal{I} \text{ is a bounded} \\ \text{size verifying} \\ \text{set for } G}} \alpha \cdot w(\mathcal{I}) + \beta \cdot |\mathcal{I}| \quad \text{where } \alpha, \beta \geq 0 \quad (1)$$

Fix any integer $k \geq 1$ and DAG $G \in [G^*]$. Minimizing Equation (1) yields $\overline{\nu}_k(G)$ when $\alpha = 1$ and $\beta = 0$ and $\nu_k(G)$ when $\alpha = 0$ and $\beta = 1$. Prior work (Choo et al., 2022) studied the version of Equation (1) without the maximization over all DAGs in the Markov equivalence class for the verification problem, but not the search problem.

For any bounded size verification set $\mathcal{I} \subseteq 2^V$, we write $\texttt{cost}(\mathcal{I}, \alpha, \beta, k) = \alpha \cdot w(\mathcal{I}) + \beta \cdot |\mathcal{I}|$ to denote its cost relative to Equation (1). For any deterministic adaptive search algorithm A that produces intervention set $\mathcal{I}$ for causal graph G^*, we define $\texttt{cost}(A, G^*, \alpha, \beta, k) = \texttt{cost}(\mathcal{I}, \alpha, \beta, k)$. For any randomized adaptive search algorithms, $\texttt{cost}(A, G^*, \alpha, \beta, k)$ refers to *expected cost*, where the expectation is over all the internal random choices made by A. When restricting to atomic interventions with $k = 1$, we simply write $\texttt{cost}(\mathcal{I}, \alpha, \beta)$ and $\texttt{cost}(A, G^*, \alpha, \beta)$.

2.1. Related work in causal graph discovery

(Hyttinen et al., 2013) was the first to apply the notion of separating systems from the combinatorics literature to causal discovery via *non-adaptive* atomic interventions. This was later extended to interventions of bounded size in the adaptive setting by (Hu et al., 2014; Shanmugam et al., 2015). Meanwhile, (Ghassami et al., 2018) studied the problem of maximizing the number of oriented edges given a fixed budget of non-adaptive atomic interventions.

There has been a flurry of works that explored adaptive search algorithms to fully orient a given essential graph while minimizing the number of interventions used (He & Geng, 2008; Hu et al., 2014; Shanmugam et al., 2015; Kocaoglu et al., 2017; Lindgren et al., 2018; Greenewald et al., 2019; Squires et al., 2020; Choo et al., 2022). More recently, (Choo & Shiragur, 2023) studied the problem of adaptive *subset* search problem where one is only interested in learning the orientations of a subset of target edges.

In the context of *weighted* interventions, one of the earliest works in the setting of additive vertex costs is (Kocaoglu et al., 2017), where they show how to compute minimum cost non-adaptive bounded size interventions in polynomial time. When the maximum number of interventions is fixed and one has to find the minimum cost intervention set, (Lindgren et al., 2018) showed that it is NP-hard and provided a constant factor approximation algorithm.

For lower bounds, prior works such as (Squires et al., 2020; Porwal et al., 2022; Choo et al., 2022) studied bounds for the verification number, where (Choo et al., 2022) eventually gave a complete characterization of $\nu_1(G^*)$ via the minimum vertex cover of the covered edges of G^*.

In the presence of latents (i.e. causal insufficiency), a common causal graph discovery objective is to recover an *ancestral graph* (Richardson & Spirtes, 2002) instead of a DAG. (Addanki et al., 2020) recently studied this problem using non-adaptive interventions under additive vertex costs.

3. Results

Here we state all our main results of the paper. Our first result suggests that comparing against $\overline{\nu}_1(G^*)$ may be too pessimistic for weighted causal graphs as we show that one cannot outperform an approximation of $|V(G^*)| = n$ in the worst case. Theorem 3.1 is information-theoretic and holds even for algorithms that have infinite computation power.

Theorem 3.1. *For any adaptive search algorithm A, deterministic or randomized, there exists a weighted causal graph G^* such that $\texttt{cost}(A, G^*, 1, 0) \in \Omega(n \cdot \overline{\nu}_1(G^*))$.*

Recently, in the context of adaptive *subset* search on *unweighted* causal graphs, (Choo & Shiragur, 2023) showed that comparing against $\nu_1(G^*)$ in the presence of an *adaptive* adversary[6] leads to pessimistic bounds. Instead, they propose to compare against a benchmark that does *not* know G^*. Independently motivated by Theorem 3.1, we propose the following natural benchmark metric that to compare search algorithms against:

$$\overline{\nu}_k^{\max}(G^*) = \max_{G \in [G^*]} \overline{\nu}_k(G) \quad \text{for any integer } k \geq 1 \quad (2)$$

As discussed in the introduction, our new benchmark captures the worst case cost for any optimal algorithm over the DAGs corresponding to a given essential graph. That is, $\max_{G \in [G^*]} \texttt{cost}(\texttt{ALG}, G, 1, 0, k) \geq \nu_k^{\max}(G^*) \geq \nu_k(G^*)$ for any fixed adaptive search algorithm $\texttt{ALG}$. This benchmark also resolves the earlier raised concerns in (Choo & Shiragur, 2023) of "comparing against the "best" algorithm that does *not* know G^*".

We next present an adaptive algorithm that is competitive with respect to Equation (2) when searching over weighted causal graphs using adaptive interventions.

Theorem 3.2. *Fix an essential graph $\mathcal{E}(G^*)$ corresponding to an unknown weighted causal DAG G^*. Algorithm 1 is a deterministic and adaptive algorithm that computes an atomic intervention set $\mathcal{I}$ such that $\mathcal{E}_{\mathcal{I}}(G^*) = G^*$ and $w(\mathcal{I}) \in \mathcal{O}\left(\log^2(n) \cdot \overline{\nu}_1^{\max}(G^*)\right)$. Ignoring the time spent implementing the actual interventions, Algorithm 1 runs in*

[6] An adaptive adversary observes the interventions made by an adaptive algorithm and is allowed to "change its mind" by choosing the ground truth DAG among the set of all DAGs that are consistent with the already revealed information. We remark that Theorem 3.1 holds even in the presence of a *non-adaptive* adversary, and thus is a stronger result in this aspect.

$\mathcal{O}(n \cdot \log^2(n) \cdot d \cdot m)$ time, where d and m are the degeneracy and number of edges of $\mathrm{skel}(\mathcal{E}(G^*))$ respectively.

Theorem 3.2 is the first competitive *adaptive* algorithm for the weighted setting. We later show that one can combine Algorithm 1 with a simple algorithm in a black-box manner to get $w(\mathcal{I}) \in \mathcal{O}(\min\{n \cdot \overline{\nu}(G^*), \log^2(n) \cdot \overline{\nu}_1^{\max}(G^*)\})$. The closest comparable result for *weighted* graph search is the *non-adaptive* search algorithm of (Lindgren et al., 2018) discussed in Section 2.1. However, note that the size of a separating system in the non-adaptive setting can be much larger than $\overline{\nu}_1^{\max}(G^*)$ even when all vertices have unit weight: in the case where the essential graph is a path on n vertices, $\overline{\nu}_1^{\max}(G^*) = \nu_1^{\max}(G^*) = 1$ while any separating system on this path has size $\Omega(n)$; see Appendix A.2.

By tweaking the algorithm of Theorem 3.2 appropriately, our next result provides competitive guarantees with respect to the generalized cost function Equation (1).

Theorem 3.3. *Fix an essential graph $\mathcal{E}(G^*)$ corresponding to an unknown weighted causal DAG G^*. Suppose $\mathcal{I}_1^*$ and $\mathcal{I}_k^*$ are an atomic and bounded size verifying sets minimizing Equation (1) such that $\texttt{cost}(\mathcal{I}_1^*, \alpha, \beta, 1) = \texttt{OPT}_1$ and $\texttt{cost}(\mathcal{I}_k^*, \alpha, \beta, k) = \texttt{OPT}_k$. Then, Algorithm 3 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*
1. $\texttt{cost}(\mathcal{I}, \alpha, \beta, 1) \in \mathcal{O}\left(\log^2 n \cdot \texttt{OPT}_1\right)$
2. $\texttt{cost}(\mathcal{I}, \alpha, \beta, k) \in \mathcal{O}\left(\log n \cdot (\log n + \log k) \cdot \texttt{OPT}_k\right)$.

We remark that Algorithm 1 is a special case of Algorithm 3 (given in Appendix D) with $\alpha = 1$, $\beta = 0$, and $k = 1$.

Why study bounded size interventions?
One may be able to reduce the *number* of interventions performed if one is allowed to intervene on more than one vertex per intervention. For instance, to fully recover the orientations of a clique on n nodes, it is known that $\Omega(n)$ atomic interventions are required. However, if bounded size interventions are allowed, the lower bound is only $\Omega(n/k)$ and $\tilde{\mathcal{O}}(n/k)$ interventions suffice (Shanmugam et al., 2015). As interventions take up actual wall-clock time and adaptivity demands sequentiality in the decision process, the ability to perform bounded size interventions (ideally in parallel) is particularly important for time-sensitive scenarios.

Significance of our new metric $\overline{\nu}^{\max}(G^*)$
As the previous benchmark of $\overline{\nu}(G^*)$ is overly pessimistic, many algorithms will "look the same" (albeit all with terrible competitive ratios) when compared against $\overline{\nu}(G^*)$ and it is natural to ask if there is a *meaningful* comparison that differentiates them. The new benchmark $\overline{\nu}^{\max}(G^*)$ serves this purpose: an algorithm that is more competitive to $\overline{\nu}^{\max}(G^*)$ would have better worst-case guarantees. Intuitively, $\overline{\nu}^{\max}(G^*)$ shifts the comparisons away from an idealistic "how much will an oracle that knows G^* pay?" to a weaker "how much will the best possible algorithm that only knows $[G^*]$ pay?". The latter question is more realistic/reasonable, and as we have argued, more meaningful in problem instances where vertices have differing costs.

Many adaptive search algorithms guarantee an $\tilde{\mathcal{O}}(n)$ approximation to $\overline{\nu}(G^*)$ which only implies an $\tilde{\mathcal{O}}(n)$ approximation to $\overline{\nu}^{\max}(G^*)$, while Algorithm 1 provably obtains a logarithmic competitive ratio to $\overline{\nu}^{\max}(G^*)$. For instance, the following naive algorithm incurs a cost of $\mathcal{O}(n \cdot \overline{\nu}(G^*))$, but does not yield meaningful guarantees against $\overline{\nu}^{\max}(G^*)$: intervene on vertices one-by-one in an ascending weight ordering until the entire graph is oriented. The proof of $\mathcal{O}(n \cdot \overline{\nu}(G^*))$ is straightforward: the weight $w(v_{final})$ of the final intervened vertex v_{final} is a lower bound for $\overline{\nu}(G^*)$, and we intervened at most n vertices, each of cost lower than $w(v_{final})$, before v_{final}. In Appendix E, we formally show how to combine the this naive algorithm with our algorithms of Theorem 3.2 and Theorem 3.3 in a black-box manner to retain the guarantees against $\overline{\nu}^{\max}(G^*)$ whilst *simultaneously* ensuring that at most $\mathcal{O}(n \cdot \overline{\nu}(G^*))$ cost is incurred. The high-level idea is to simulate both algorithms in parallel until the causal graph is recovered.

4. Techniques

Here, we give some high-level technical ideas behind our algorithmic results (Theorem 3.2 and Theorem 3.3). We first describe how to lower bound the benchmark $\overline{\nu}^{\max}$ before giving our atomic adaptive algorithm (Algorithm 1). Then, we explain how to tweak Algorithm 1 to handle the generalized cost function with bounded size interventions.

4.1. Lower bounding the benchmark

For any interventional essential graph, we know that interventions within a chain component do not help to recover arcs within another chain component (Hauser & Bühlmann, 2014). Using this fact along with the proof strategy of (Choo et al., 2022) for lower bounding $\nu_1(G^*)$, we can show the following lower bound for $\overline{\nu}_1^{\max}(G^*)$.

Theorem 4.1. *For any DAG G^* and its essential graph $\mathcal{E}(G^*)$, we have*

$$\overline{\nu}_1^{\max}(G^*) \geq \max_{\mathcal{I} \subseteq V} \left\{ \sum_{\substack{H \in CC(\mathcal{E}_{\mathcal{I}}(G^*)) \\ |V(H)| \geq 2}} \max \left\{ \zeta_{\mathcal{I},H}^{(1)}, \zeta_{\mathcal{I},H}^{(2)} \right\} \right\}$$

where we maximize over atomic intervention sets $\mathcal{I} \subseteq V$,

$$\zeta_{\mathcal{I},H}^{(1)} = \frac{1}{2} \cdot \max_{\textit{clique } C \in H} \left\{ w(V(C)) - \max_{v \in V(C)} \{w(v)\} \right\}$$

and

$$\zeta_{\mathcal{I},H}^{(2)} = \frac{1}{2} \cdot \max_{v \in V(H)} \left\{ \min \{w(v), \gamma_{H,v}\} \right\}$$

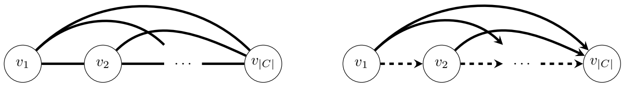


Figure 1. Consider clique C involving vertices $v_1, v_2, \ldots, v_{|C|}$ with $w(v_1) \geq w(v_2) \geq \ldots \geq w(v_{|C|})$. By Lemma 4.2, there exists a DAG consistent with this essential graph by choosing any vertex ordering of our choice within C (see right figure). Covered edges $\{v_1 \to v_2, v_2 \to v_3, \ldots, v_{|C|-1} \to v_{|C|}\}$ are dashed.

where $\gamma_{H,v} = \sum_{i=1}^{t} \max_{\substack{\text{clique } C_i: \\ V(C_i) \subseteq V_i \cap N_H(v)}} \{w(V(C_i))\}$ *with* $V_1, \ldots, V_t \subseteq V(H)$ *being vertex sets of the* $t \geq 1$ *disjoint connected components in* $H[V(H) \setminus \{v\}]$.

The two cases of Theorem 4.1 are pictorially illustrated by Figure 1 and Figure 2 respectively: we lower bound the $\zeta^{(1)}$ and $\zeta^{(2)}$ terms via the minimum cost vertex cover of the covered edges constructed in each figure. In $\zeta^{(1)}$, $w(V(C)) - \max_{v \in V(C)} w(v)$ corresponds to the sum of the weight of all clique vertices *except* the costliest one. In $\zeta^{(2)}$, we check whether it is cheaper to intervene on a particular vertex v or a clique in each connected component "dangling" from v. The proof for both cases relies on being able to pick a "worst case ordering" on the vertices that are consistent with the given essential graph $\mathcal{E}_{\mathcal{I}}(G^*)$.

To argue that we can always fix such an ordering of our choice, we combine a "patching" result of (Choo & Shiragur, 2023) (see the second point of Theorem C.4) with the "maximal clique picking" procedure of (Wienöbst et al., 2021b) from the literature of MEC size counting. Informally, (Wienöbst et al., 2021b) showed that all possible DAGs consistent with any given essential graph can be generated by repeating procedure: Picking a maximal clique C to be the prefix maximal clique; orient all incident edges out of C; apply Meek rules until convergence; repeat.

Lemma 4.2. *Fix an interventional essential graph* $\mathcal{E}_{\mathcal{I}}(G^*)$ *corresponding to an arbitrary moral DAG* G^* *and intervention set* $\mathcal{I} \subseteq 2^V$. *For any clique* C *(not necessarily maximal) in any chain component of* $\mathcal{E}_{\mathcal{I}}(G^*)$ *and any permutation ordering* π *on the vertices* $V(C)$ *of* C*, there exists a DAG* G *consistent with* $\mathcal{E}_{\mathcal{I}}(G^*)$ *such that* $u \to v$ *if and only if* $\pi(u) < \pi(v)$ *for any two clique vertices* $u, v \in V(C)$.

Roughly speaking, given a (not necessarily maximal) clique C' and an ordering π, Lemma 4.2 follows by first picking a maximal clique containing C' to be the prefix via "maximal clique picking", and then picking the vertices within C' one by one according to π via "patching".

Since vertex costs are additive and intervening on vertices in each bounded intervention set atomically can only recover more information about the causal graph, our next result provide lower bounds of the benchmark for bounded size interventions with its atomic counterpart.

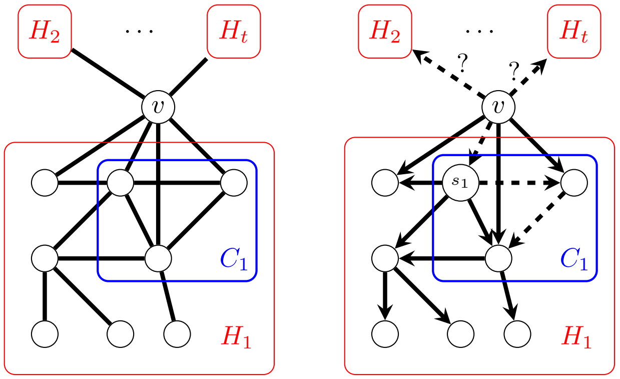


Figure 2. Consider the chain component H with vertex v and "dangling" connected components $H_1, H_2, \ldots, H_t$ in $H[V(H) \setminus \{v\}]$. Suppose C_1 is the costliest clique within $H[V(H_1) \cap N_H(v)]$. By Lemma 4.2, there exists a DAG consistent with this essential graph by letting v be the prefix within H and then letting C_1 be the prefix within H_1 and choosing any vertex ordering of our choice within C_1 (see right figure). Covered edges are dashed. For any connected component H_i with source node s_i, the arc $v \to s_i$ is a covered edge while $v \to u$ is *not* a covered edge, for $u \in V(H_i) \setminus \{s_i\}$.

Theorem 4.3. *For any DAG* G^* *and integer* $k \geq 1$, $\overline{\nu}_k^{\max}(G^*) \geq \overline{\nu}_1^{\max}(G^*)$ *and* $\nu_k^{\max}(G^*) \geq \lceil \nu_1^{\max}(G^*)/k \rceil$.

4.2. A competitive adaptive search algorithm

Here, we present an adaptive search algorithm (Algorithm 1) that is competitive with respect to the lower bounds we presented in the previous section and proves Theorem 3.2. A known result of (Choo & Shiragur, 2023) (see Theorem C.4) allows us to ignore all oriented arcs in an interventional essential graph, without loss of generality, we can always assume that the underlying causal graph is a moral DAG. Given an essential graph of a moral DAG, Algorithm 1 adaptively computes and outputs an atomic intervention set with cost competitive to $\overline{\nu}_1^{\max}(G^*)$.

On a high level, Algorithm 1 is rather similar to the algorithm of (Choo et al., 2022): both algorithms repeatedly apply Theorem 2.2 to compute 1/2-clique separators K_H so that we can break up the chain components and recurse on smaller sized chain components. Such an approach is useful because the lower bound of Theorem 4.1 ensures that the interventions done in each disjoint chain component can be summed up together to compare against $\overline{\nu}_1^{\max}(G^*)$.

Unfortunately, we *cannot* fully intervene on *all* vertices in the clique separators unlike the *unweighted* adaptive search algorithm of (Choo et al., 2022). In the weighted setting, the costliest vertex v_H in a clique separator K_H may be enormous cost[7] and the first case of Theorem 4.1 only guarantees

[7] In the extreme case, consider the example where $w(v_H) \gg$

Algorithm 1 Atomic weighted adaptive search.

```
1:  Input: Essential graph $\mathcal{E}(G^*)$ of a moral DAG $G^*$ and
    weight function $w : V \to \mathbb{R}$.
2:  Output: Atomic intervention set $\mathcal{I}$ s.t. $\mathcal{E}_{\mathcal{I}}(G^*) = G^*$.
3:  Initialize $i = 0$ and $\mathcal{I}_0 = \emptyset$.
4:  while $\mathcal{E}_{\mathcal{I}_i}(G^*)$ still has unoriented edges do
5:      Initialize $\mathcal{J}_i \leftarrow \emptyset$
6:      for $H \in CC(\mathcal{E}_{\mathcal{I}_i}(G^*))$ of size $|H| \geq 2$ do
7:          Find 1/2-clique separator $K_H$ via Theorem 2.2.
8:          Let $S = \{\{v\} : v \in V(K_H)) \setminus \{v_H\}\}$ be
            an atomic intervention set without the costliest
            vertex $v_H = \text{argmax}_{v \in V(K_H)} w(v)$.
9:          Intervene on $S$ and add $S$ to $\mathcal{J}_i$.
10:         Let $Z_{v_H} \in CC(\mathcal{E}_{\mathcal{I}_i \cup S}(G^*))$ be the chain
            component containing $v_H$ after intervening on $S$.
11:         if $v_H$ is not singleton in $Z_{v_H}$ then
12:             Add output of ResolveDangling to $\mathcal{J}_i$.
13:     Update $\mathcal{I}_{i+1} \leftarrow \mathcal{I}_i \cup \mathcal{J}_i$ and $i \leftarrow i + 1$.
14: Return $\mathcal{I}_i$
```

that we can remain competitive by intervening on *all but* v_H. As there may be connected components "dangling off v_H", we invoke ResolveDangling (Algorithm 2) to ensure that the partites induced by the 1/2-clique separators will indeed be separated while we use the second case of Theorem 4.1 to bound the cost of interventions used[8]. Denoting an iteration of the while loop in Algorithm 1 as a *phase*, we show the following two lemmas about Algorithm 1 whose combinations directly yields Theorem 3.2.

Lemma 4.4. *Algorithm 1 terminates after $\mathcal{O}(\log n)$ phases.*

Lemma 4.5. *Each phase in Algorithm 1 incurs a cost of $\mathcal{O}(\log(n) \cdot \overline{\nu}_1^{\max}(G^*))$.*

The first logarithmic factor in Lemma 4.4 is due to the halving of the size of the chain components in each phase while the second logarithmic factor in Lemma 4.5 is due to the subroutine ResolveDangling, which tries to find a prefix clique in each dangling component. The description of the subroutine ResolveDangling is provided in Algorithm 2 and the guarantees of this subroutine are summarized in the following lemma.

Lemma 4.6. *Fix an interventional essential graph $\mathcal{E}_{\mathcal{I}'}(G^*)$ corresponding to an unknown weighted causal moral DAG G^* and some intervention $\mathcal{I}' \subseteq 2^V$. Let H be a chain component of $\mathcal{E}_{\mathcal{I}'}(G^*)$ containing a vertex $v \in V(H)$. Then, Algorithm 2 returns an atomic intervention set $\mathcal{I}$ such that all the outgoing edges of v within H are oriented in $\mathcal{E}_{\mathcal{I}' \cup \mathcal{I}}(H)$ and $w(\mathcal{I}) \in \mathcal{O}(\log n \cdot \overline{\nu}_1^{\max}(G^*))$.*

Algorithm 2 ResolveDangling

```
Input: Interventional essential graph $\mathcal{E}_{\mathcal{I}'}(G^*)$ for some
intervention $\mathcal{I}' \subseteq 2^V$, weight function $w : V \to \mathbb{R}$,
and a chain component $H$ of $\mathcal{E}_{\mathcal{I}'}(G^*)$ that contains
vertex $v \in V(H)$ with $t$ disjoint connected components
$H_1, \ldots, H_t$ in $H[V(H) \setminus \{v\}]$.
Output: Atomic intervention set $\mathcal{I}$ such that all the
outgoing edges of $v$ within $H$ are oriented in $\mathcal{E}_{\mathcal{I} \cup \mathcal{I}'}(H)$.
Initialize $\mathcal{I} \leftarrow \emptyset$.
if $w(v) \leq \sum_{i=1}^{t} \max_{\text{clique } C \text{ in } H_i \cap N_H(v)} w(C)$ do
    Intervene on $v$; Set $\mathcal{I} \leftarrow \mathcal{I} \cup \{\{v\}\}$.
else
    for $i \in \{1, \ldots, t\}$ do
        Initialize $V' \leftarrow V(H_i) \cap N_H(v)$.
        while skel$(\mathcal{E}_{\mathcal{I} \cup \mathcal{I}'}(G^*))[V']$ is not a clique do
            Find a 1/2-clique separator $K$ of $H_i[V']$.
            Intervene on $K$ atomically; Add $V(K)$ to $\mathcal{I}$.
            By Lemma 4.7, find the chain component $Q$
            with only incoming arcs into $K$ (if it exists).
            if $Q$ exists then Set $V' \leftarrow V(Q)$.
            else Set $V' \leftarrow \emptyset$. break
        Intervene on $V'$ atomically; Set $\mathcal{I} \leftarrow \mathcal{I} \cup V'$.
Return $\mathcal{I}$
```

In the remainder, we briefly discuss the technical idea behind Algorithm 2. Let π be an arbitrary consistent ordering of vertices corresponding to the unknown underlying DAG G^*. Suppose there are t disjoint connected components $H_1, \ldots, H_t$ after removing v_H, then the cost incurred by Algorithm 2 is made competitive by using the second case of Theorem 4.1. See Figure 3 for an illustration. If $w(v_H)$ is at most the sum of weights of the heaviest clique across all $\{H_i \cap N_H(v)\}_{i \in [t]}$, then we can simply intervene on v_H to disconnect the partites. Otherwise, within each disjoint connected component H_i, we will search for and intervene on the source vertex $u_i = \text{argmin}_{u \in V(H_i) \cap N_H(v)} \pi(u)$ within each H_i. As we will search for u_i using 1/2-clique separators, Lemma 4.7 guarantees we find it in $\mathcal{O}(\log n)$ iterations.

Lemma 4.7. *Let $\mathcal{E}_{\mathcal{I}}(G)$ be the interventional essential graph of a moral DAG $G = (V, E)$ with respect to intervention set $\mathcal{I} \subseteq V$. Fix any chain component $H \in CC(\mathcal{E}_{\mathcal{I}}(G))$ and vertex $v \in V(H)$. If v is the source node of H, then there are no chain components of $\mathcal{E}_{\mathcal{I} \cup \{v\}}(H)$ with only incoming arcs into v in G. Otherwise, if v is not the source node of H, then there is exactly one chain component of $\mathcal{E}_{\mathcal{I} \cup \{v\}}(H)$ with only incoming arcs into v in G. Furthermore, without further interventions, we can decide if such a chain component exist (and find it) in polynomial time.*

Consider an arbitrary connected component H_i amongst

$\sum_{v \in V \setminus \{v_H\}} w(v)$. In this example, intervening on *everything but* v_H in an atomic fashion trivially recovers *any* DAG in $[G^*]$, i.e. $\overline{\nu}_1^{\max}(G^*) \leq \sum_{v \in V \setminus \{v_H\}} w(v) \ll w(v_H)$.

[8] If the computed clique separator in Step 8 of Algorithm 1 involves only 1 node, $S = \emptyset$ and we break up the partites via ResolveDangling (Algorithm 2).

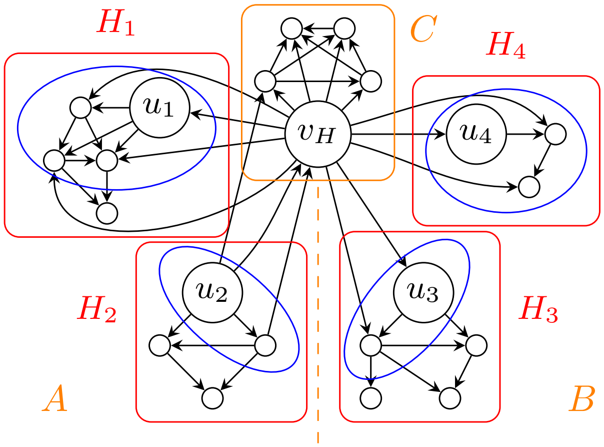


Figure 3. Consider the moral DAG G^* above where C is a 1/2-clique separator with vertices in $A = V(H_1) \cup V(H_2)$ and $B = V(H_3) \cup V(H_4)$, and v_H is the costliest vertex in C. If we were to intervene on every single vertex in C, as per the algorithm of (Choo et al., 2022), then the partites A and B will be disconnected. However, v_H may be very costly and Theorem 4.1 only gives approximation guarantees when intervening on $\mathcal{I} = V(C) \setminus \{v_H\}$. Since the incident edges of v_H may remain unoriented in $\mathcal{E}_{\mathcal{I}}(G^*)$, the partites may still be connected, e.g. the arcs $u_2 \to v_H \to u_3$ remain unoriented in $\mathcal{E}_{\mathcal{I}}(G^*)$. We say that connected components H_1, H_2, H_3, and H_4 are "dangling" from v_H in $\mathcal{E}_{\mathcal{I}}(G^*)$. By Lemma 2.3, it suffices to intervene on all the source vertices u_i in each H_i, and thus `ResolveDangling` searches for u_i amongst the neighbors of v_H in each H_i (the blue ellipses).

$H_1, \ldots, H_t$ with source vertex u_i. If $\pi(v_H) < \pi(u_i)$, Lemma 2.3 tells us that intervening on u_i will orient all $v_H \to z$ arcs for $z \in H_i$ which disconnects H_i from v_H, and thus from other components H_j. Meanwhile, if $\pi(v_H) > \pi(u_i)$, then there is an arc from H_i to v_H. Note that intervening on u_i may not disconnect H_i from v_H, but it will disconnect H_i from the other components[9]. Nonetheless, we can still conclude that the resulting connected component has size at most halved since H_i was part of a partite resulting from a 1/2-clique separator – it may include at most an additional vertex v_H but will not include u_i (since we intervene on u_i). We prove this formally in the appendix.

4.3. Handling the generalized cost objective

To handle the generalized cost objective of Equation (1), we make three algorithmic tweaks to the algorithms presented in the previous section. Firstly, we change the condition of Line 4 in Algorithm 2 to account for the α-β trade-off in Equation (1). Secondly, to compute bounded size interventions to follow orient a clique, we apply the labelling scheme of Lemma 4.8 to use bounded sized interventions when intervening on cliques via the subroutine `CliqueIntervention` (Algorithm 5) with guarantees given in Lemma 4.9. Finally, when searching for a prefix clique in the while-loop Algorithm 2, if one directly applies `CliqueIntervention` on each clique separator K_{H_i}, then one can show an $\mathcal{O}(\log^2 n \cdot \log k)$ approximation. To obtain an $\mathcal{O}(\log n \cdot (\log n + \log k))$ approximation, we show that it suffices to partition $V(K_{H_i})$ into groups of size at most k and intervening on them. Note that this will *not* necessarily orient all internal edges of $V(K_{H_i})$ but is sufficient for the purposes of locating a prefix clique.

Lemma 4.8 ((Shanmugam et al., 2015)). *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 $\ell = \lceil \log_a n \rceil$. 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.*

Lemma 4.9. *Given a set of clique vertices $V(C) \subseteq V$ and integer $k \geq 1$, Algorithm 5 returns a set $S \subseteq 2^{V(C)}$ such that each partite in S has at most k vertices. When $k = 1$, $|S| = |V(C)|$ and each vertex appears exactly once in S. When $k > 1$, $|S| \in \mathcal{O}(\log k \cdot |V(C)|/k)$ and each vertex appears at most $\mathcal{O}(\log k)$ times in S.*

In terms of analysis, to lower bound Equation (1), we individually lower bound the cost and size terms. For instance,

$$\begin{aligned} &\max_{G \in [G^*]} \min_{\substack{\mathcal{I} \text{ is a bounded} \\ \text{size verifying} \\ \text{set for } G}} \alpha \cdot w(\mathcal{I}) + \beta \cdot |\mathcal{I}| \\ &\geq \max_{G \in [G^*]} \min_{\substack{\mathcal{I} \text{ is a bounded} \\ \text{size verifying} \\ \text{set for } G}} \alpha \cdot w(\mathcal{I}) = \alpha \cdot \overline{\nu}_k^{\max}(G^*) \ . \end{aligned}$$

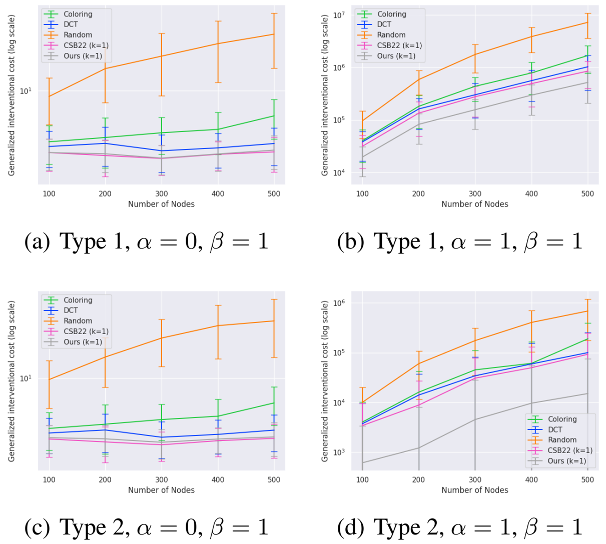
Similarly, $\beta \cdot \nu_k^{\max}(G^*)$ is also a lower bound. Then, we can further use Theorem 4.3 to lower bound $\overline{\nu}_k^{\max}(G^*)$ and $\nu_k^{\max}(G^*)$ via $\overline{\nu}_1^{\max}(G^*)$ and $\nu_1^{\max}(G^*)$ respectively. The additional $\log k$ term for non-atomic interventions occurs because of the multiplicity of vertices in the output of `CliqueIntervention` (see Lemma 4.9).

In summary, our tweaked algorithm for the generalized cost objective has $\mathcal{O}(\log n)$ phases, similar to Algorithm 1, and we incur a cost of $\mathcal{O}((\log n + \log k) \cdot \texttt{OPT}_k)$ in each phase. A description of the tweaked algorithm and a more detailed analysis of it is provided in the appendix.

5. Experiments

Since `ALG` (Algorithm 1) is a special case of `ALG-GENERALIZED` (Algorithm 3) when $\alpha = 0$,

[9] Without loss of generality, suppose $\pi(u_1) = \min_{i \in \{1,\ldots,t\}} \pi(u_i)$ and $\pi(v_H) > \pi(u_1)$. Orienting the arc $u_1 \to v_H$ triggers Meek rule R1 to orient all $v_H \to z$ arcs for $z \notin H_1$, thus disconnecting the H_i's from each other.



(a) Type 1, $\alpha = 0$, $\beta = 1$

(b) Type 1, $\alpha = 1$, $\beta = 1$

(c) Type 2, $\alpha = 0$, $\beta = 1$

(d) Type 2, $\alpha = 1$, $\beta = 1$

Figure 4. Experimental results for atomic interventions (log scale)

$\beta = 1$, and $k = 1$, we implement and benchmark ALG-GENERALIZED. Modifying the synthetic experimental setup of (Squires et al., 2020; Choo et al., 2022; Choo & Shiragur, 2023) for *weighted* causal graphs, we ran experiments for $\alpha \in \{0, 1\}$ and $\beta = 1$ on two different types of weight classes for a graph on n vertices and measured generalized costs incurred:

Type 1. Vertex weights are i.i.d. sampled from an exponential distribution $\exp(n^2)$ with parameter n^2 to simulate the setting where there is a spread in the costs of the vertices.
Type 2. A random $p = 0.1$ fraction of vertices are assigned weight n^2 while the others are assigned weight 1 to simulate the setting with a few randomly chosen high cost vertices.

Figure 4 shows a subset of our experiments; see Appendix G for more experimental details and results, where we also investigate the impact of size for bounded size interventions.

Qualitative discussion of experimental results

The Y-axis measures the generalized cost $\alpha \cdot w(\mathcal{I}) + \beta \cdot |\mathcal{I}|$, for intervention set $\mathcal{I} \subseteq 2^V$. Ignoring the magnitude of Y-axis values, fixing either α or β while scaling the other recovers any possible observable trend. For atomic interventions, $(\alpha, \beta) = (0, 1)$ recovers the unweighted setting and Figure 4 illustrates the effects of $\alpha = 0$ versus $\alpha = 1$.

When $\alpha = 0$, the generalized cost function is simply the number of interventions used that the other state-of-the-art methods were designed for. Here, ALG-GENERALIZED incurs a similar cost despite having additional overheads to ensure theoretical guarantees for general $\alpha \geq 0$.

For $\alpha > 0$, the generalized cost function is affected by the vertex weights, and ALG-GENERALIZED incurs noticeably less cost than the others already when $\alpha = 1$ (note: log scale). This gap will only increase as we increase the value of α to make the generalized cost put more weightage on the total additive vertex cost of the intervention $\mathcal{I}$.

As our experimental instances were randomly generated, it does look like existing algorithms, such as (Choo et al., 2022), is competitive with our weight-sensitive algorithm ALG-GENERALIZED on such random instances, even though they are oblivious to vertex weights. However, we can easily create many instances where these algorithms performs arbitrarily worse. For instance, consider the star graph G^* on n nodes where the leaves have weight 1 and the centroid has weight $w \gg n$; imagine $w = n^{10000}$. On G^*, (Choo et al., 2022) will intervene on the centroid, incurring w while ALG-GENERALIZED will never intervene on the centroid and in the worst case intervene on all the leaves (paying at most $n - 1$) to fully orient G^* from $\mathcal{E}(G^*)$.

ALG-GENERALIZED has a similar running time[10] as the other state-of-the-art algorithms across all experiments.

6. Conclusion and future directions

In our work, we make standard assumptions of causal sufficiency, faithfulness, and infinite sample regime. As these assumptions may be too strong in some practical settings, one should view our work as providing theoretical foundations to the *feasibility* of the weighted search problem (i.e. what *can* be done in an optimistic setting) and it is of paramount practical importance to weaken/remove such assumptions in future work. In addition, we also state some possible future directions that we think are interesting:

1. Understand the optimal approximation ratio of our new benchmark $\overline{\nu}^{\max}(G^*) = \max_{G \in [G^*]} \overline{\nu}(G)$.
2. Design subset search algorithms à la (Choo & Shiragur, 2023) that are competitive with respect to $\overline{\nu}^{\max}(G^*)$, for both unweighted and weighted settings.
3. Provide an efficient algorithm to compute $\overline{\nu}^{\max}(G^*)$. We remark that, for a given G^*, efficient computation of $\nu_1(G^*)$ and $\overline{\nu}_1(G^*)$ are known (Choo et al., 2022).

Acknowledgements

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-PhD/2021-08-013). Part of this work was done while the authors were visiting the Simons Institute for the Theory of Computing. We would like to thank Saravanan Kandasamy, Jiaqi Zhang, and the ICML reviewers for valuable feedback and discussions.

[10] ALG-GENERALIZED is faster than all benchmarked algorithms except (Choo et al., 2022). This is expected as both are based on 1/2-clique separators but ALG-GENERALIZED has additional computational overheads to handle dangling components.

References

Addanki, R., Kasiviswanathan, S., McGregor, A., and Musco, C. Efficient Intervention Design for Causal Discovery with Latents. In *International Conference on Machine Learning*, pp. 63–73. PMLR, 2020.

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

Chickering, D. M. A Transformational Characterization of Equivalent Bayesian Network Structures. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, pp. 87–98, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603859.

Cho, H., Berger, B., and Peng, J. Reconstructing Causal Biological Networks through Active Learning. *PLoS ONE*, 11(3):e0150611, 2016.

Choo, D. and Shiragur, K. Subset verification and search algorithms for causal DAGs. In *International Conference on Artificial Intelligence and Statistics*, 2023.

Choo, D., Shiragur, K., and Bhattacharyya, A. Verification and search algorithms for causal DAGs. *Advances in Neural Information Processing Systems*, 35, 2022.

de Campos, L. M., Cano, A., Castellano, J. G., and Moral, S. Combining gene expression data and prior knowledge for inferring gene regulatory networks via Bayesian networks using structural restrictions. *Statistical Applications in Genetics and Molecular Biology*, 18(3), 2019.

Eberhardt, F. Causation and Intervention. *Unpublished doctoral dissertation, Carnegie Mellon University*, pp. 93, 2007.

Eberhardt, F. Causal Discovery as a Game. In *Causality: Objectives and Assessment*, pp. 87–96. PMLR, 2010.

Eberhardt, F. and Scheines, R. Interventions and Causal Inference. *Philosophy of science*, 74(5):981–995, 2007.

Eberhardt, F., Glymour, C., and Scheines, R. On the number of experiments sufficient and in the worst case necessary to identify all causal relations among N variables. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pp. 178–184, 2005.

Eberhardt, F., Glymour, C., and Scheines, R. N-1 Experiments Suffice to Determine the Causal Relations Among N Variables. In *Innovations in machine learning*, pp. 97–112. Springer, 2006.

Ghassami, A., Salehkaleybar, S., Kiyavash, N., and Bareinboim, E. Budgeted Experiment Design for Causal Structure Learning. In *International Conference on Machine Learning*, pp. 1724–1733. PMLR, 2018.

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

Greenewald, K., Katz, D., Shanmugam, K., Magliacane, S., Kocaoglu, M., Boix-Adserà, E., and Bresler, G. Sample Efficient Active Learning of Causal Trees. *Advances in Neural Information Processing Systems*, 32, 2019.

Hauser, A. and Bühlmann, P. Two Optimal Strategies for Active Learning of Causal Models from Interventions. *International Journal of Approximate Reasoning*, 55(4): 926–939, 2014.

He, Y.-B. and Geng, Z. Active Learning of Causal Networks with Intervention Experiments and Optimal Designs. *Journal of Machine Learning Research*, 9:2523–2547, 2008.

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

Hu, H., Li, Z., and Vetta, A. Randomized Experimental Design for Causal Graph Discovery. *Advances in Neural Information Processing Systems*, 27, 2014.

Hyttinen, A., Eberhardt, F., and Hoyer, P. O. Experiment Selection for Causal Discovery. *Journal of Machine Learning Research*, 14:3041–3071, 2013.

King, R. D., Whelan, K. E., Jones, F. M., Reiser, P. G. K., Bryant, C. H., Muggleton, S. H., Kell, D. B., and Oliver, S. G. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427(6971): 247–252, 2004.

Kocaoglu, M., Dimakis, A., and Vishwanath, S. Cost-Optimal Learning of Causal Graphs. In *International Conference on Machine Learning*, pp. 1875–1884. PMLR, 2017.

Koller, D. and Friedman, N. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Lindgren, E. M., Kocaoglu, M., Dimakis, A. G., and Vishwanath, S. Experimental Design for Cost-Aware Learning of Causal Graphs. *Advances in Neural Information Processing Systems*, 31, 2018.

Meek, C. Causal Inference and Causal Explanation with Background Knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, pp. 403–410, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603859.

Ness, R. O., Sachs, K., Mallick, P., and Vitek, O. A Bayesian Active Learning Experimental Design for Inferring Signaling Networks. *Journal of Computational Biology: a Journal of Computational Molecular Cell Biology*, 25(7): 709–725, 2018.

Pearl, J. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.

Pingault, J.-B., O'reilly, P. F., Schoeler, T., Ploubidis, G. B., Rijsdijk, F., and Dudbridge, F. Using genetic data to strengthen causal inference in observational research. *Nature Reviews Genetics*, 19(9):566–580, 2018.

Porwal, V., Srivastava, P., and Sinha, G. Almost Optimal Universal Lower Bound for Learning Causal DAGs with Atomic Interventions. In *International Conference on Artificial Intelligence and Statistics*, 2022.

Reichenbach, H. *The Direction of Time*, volume 65. University of California Press, 1956.

Richardson, T. and Spirtes, P. Ancestral graph markov models. *The Annals of Statistics*, 30(4):962–1030, 2002.

Rotmensch, M., Halpern, Y., Tlimat, A., Horng, S., and Sontag, D. Learning a Health Knowledge Graph from Electronic Medical Records. *Scientific reports*, 7(1):1–11, 2017.

Rubin, D. B. and Waterman, R. P. Estimating the Causal Effects of Marketing Interventions Using Propensity Score Methodology. *Statistical Science*, pp. 206–222, 2006.

Shanmugam, K., Kocaoglu, M., Dimakis, A. G., and Vishwanath, S. Learning Causal Graphs with Small Interventions. *Advances in Neural Information Processing Systems*, 28, 2015.

Squires, C., Magliacane, S., Greenewald, K., Katz, D., Kocaoglu, M., and Shanmugam, K. Active Structure Learning of Causal DAGs via Directed Clique Trees. *Advances in Neural Information Processing Systems*, 33:21500–21511, 2020.

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

Tian, T. Bayesian Computation Methods for Inferring Regulatory Network Models Using Biomedical Data. *Translational Biomedical Informatics: A Precision Medicine Perspective*, pp. 289–307, 2016.

Verma, T. and Pearl, J. Equivalence and Synthesis of Causal Models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, UAI '90, pp. 255–270, USA, 1990. Elsevier Science Inc. ISBN 0444892648.

Wienöbst, M., Bannach, M., and Liśkiewicz, M. Extendability of Causal Graphical Models: Algorithms and Computational Complexity. In *Uncertainty in Artificial Intelligence*, pp. 1248–1257. PMLR, 2021a.

Wienöbst, M., Bannach, M., and Liśkiewicz, M. Polynomial-Time Algorithms for Counting and Sampling Markov Equivalent DAGs. In *Proccedings of the 35th Conference on Artificial Intelligence, AAAI*, 2021b.

Woodward, J. *Making Things Happen: A Theory of Causal Explanation*. Oxford University Press, 2005.

Yao, A. C.-C. Probabilistic computations: Toward a unified measure of complexity. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 222–227. IEEE Computer Society, 1977.

A. Adaptive versus non-adaptive interventions

Separating systems are the central mathematical objects for non-adaptive intervention design. Roughly speaking, a separating system on a set of elements is a collection of subsets such that for every pair of elements from the set, there exists at least one subset which contains exactly one element from the pair.

Instead of all pairs of elements, let us consider the (typically smaller) G-separating system for a given graph G. It is known (Kocaoglu et al., 2017) that the optimal non-adaptive intervention set to learn a moral DAG G^* is a $\mathrm{skel}(G^*)$-separating system.

Definition A.1 (G-separating system; Definition 3 of (Kocaoglu et al., 2017))**.** Given an undirected graph $G = (V, E)$, a set of subsets $\mathcal{I} \subseteq 2^V$ is a G-separating system if for every edge $\{u, v\} in E$, there exists $I \in \mathcal{I}$ such that either ($u \in I_i$ and $v \notin I_i$) or ($u \notin I_i$ and $v \in I_i$).

Theorem A.2 (Theorem 1 of (Kocaoglu et al., 2017))**.** *For any undirected graph G, an intervention set $\mathcal{I}$ learns* every *causal graph D with* $\mathrm{skel}(D) = G$ *if and only if $\mathcal{I}$ is a G-separating system.*

Path example Consider an essential graph which is an undirected path on n vertices. There are n possible DAGs corresponding to this Markov equivalence class, each of which can be uniquely identified by picking one of the vertices as a source and orienting all edges away from it. By Theorem A.2, we see that $\Omega(n)$ atomic interventions are necessary.

A.1. Adaptive can be exponentially stronger

Consider an essential graph which is an undirected path on n vertices described above where we know that one has intervene on at least $\Omega(n)$ vertices using non-adaptive atomic interventions. If we allow adaptive interventions, $\mathcal{O}(\log n)$ atomic interventions suffice by simulating binary search: intervene on the "center" vertex to uncover its incident edge orientations; orient one half using Meek rule R1; repeat.

A.2. New benchmark is different from separating system

Recall our newly proposed metric: $\overline{\nu}_k^{\max}(G^*) = \max_{G \in [G^*]} \overline{\nu}_k(G)$ for any integer $k \geq 1$.

Consider an essential graph which is an undirected path on n vertices described above where we know that one has intervene on at least $\Omega(n)$ vertices using non-adaptive atomic interventions. Under our newly proposed metric, $\overline{\nu}_1^{\max}(G^*) = \nu_1^{\max}(G^*) = 1$ since intervening on the source vertex always suffices to fully orient the entire DAG.

B. Meek rules

Remark This section of well-known facts is adapted from the appendices of (Choo et al., 2022; Choo & Shiragur, 2023).

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 (Meek, 1995). Given any edge orientation information, one can always repeatedly apply Meek rules till a fixed point to maximize the number of oriented arcs.

Definition B.1 (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 B.2 (The four Meek rules (Meek, 1995), see 5 for an illustration)**.**

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

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

R3 Edge $\{a, b\} \in E \setminus A$ 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 $\{a, b\} \in E \setminus A$ 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 $b \not\sim d$.

There exists an algorithm (Algorithm 2 of (Wienöbst et al., 2021a)) that runs in $\mathcal{O}(d \cdot |E|)$ time and computes the closure under Meek rules, where d is the degeneracy of the graph skeleton[11].

[11] 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

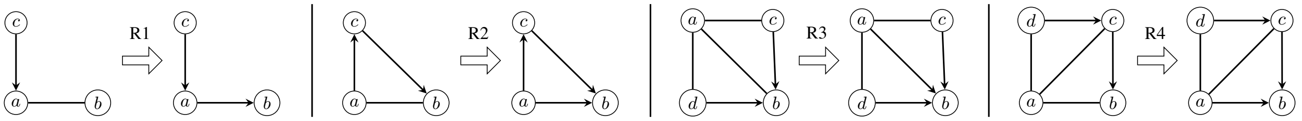


Figure 5. An illustration of the four Meek rules

C. Additional known results

Lemma C.1 (Yao's lemma (Yao, 1977))**.** *Let $\mathcal{A}$ be the space of all possible deterministic algorithms over probability distribution p, and $\mathcal{X}$ be the space of problem inputs over probability distribution q. Denote probability distributions over $\mathcal{A}$ and $\mathcal{X}$ by p_a and q_x respectively. Then,*

$$\max_{x \in \mathcal{X}} \mathbb{E}_p[c(A, x)] \geq \min_{a \in \mathcal{A}} \mathbb{E}_q[c(a, X)]$$

In other words, Lemma C.1 tells us that in order to lower bound the cost of any randomized algorithm, it suffices to find a "bad" input distribution such that any deterministic incurs a high cost.

Lemma C.2 (Modified lemma 1 of (Hauser & Bühlmann, 2014); Appendix B of (Choo et al., 2022))**.** *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*

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

Lemma C.3 (Lemma 21 of (Choo et al., 2022))**.** *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. Then,*

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

Theorem C.4 ((Choo & Shiragur, 2023))**.** *For any intervention set $\mathcal{I} \subseteq 2^V$, define $R(G, \mathcal{I}) = A(\mathcal{E}_{\mathcal{I}}(G)) \subseteq E$ as the set of oriented arcs in the $\mathcal{I}$-essential graph of a DAG G and define $G^{\mathcal{I}} = G[E \setminus R(G, \mathcal{I})]$ as the* fully directed *subgraph DAG induced by the* unoriented arcs *in G, where $G^{\emptyset}$ is the graph obtained after removing all the oriented arcs in the observational essential graph due to v-structures. Then, for any DAG $G = (V, E)$ and intervention sets $\mathcal{A}, \mathcal{B} \subseteq 2^V$,*

1. ***"Suffices to study moral DAGs"**: $R(G, \mathcal{A} \cup \mathcal{B}) = R(G^{\mathcal{A}}, \mathcal{B}) \,\dot\cup\, R(G^{\mathcal{B}}, \mathcal{A}) \,\dot\cup\, (R(G, \mathcal{A}) \cap R(G, \mathcal{B}))$*
2. ***"Patching"**: Any acyclic completion of $\mathcal{E}(G^{\mathcal{A}})$ can be combined with $R(G, \mathcal{A})$ to obtain a valid DAG that belongs to both $\mathcal{E}(G)$ and $\mathcal{E}_{\mathcal{A}}(G)$.*

The first point of Theorem C.4 justifies why it suffices to only study verification and adaptive search via ideal interventions on moral DAGs: since $R(G, \mathcal{I}) = R(G^{\emptyset}, \mathcal{I}) \,\dot\cup\, R(G, \emptyset)$, any oriented arcs in the observational graph can be removed *before performing any interventions* as the optimality of the solution is unaffected.

The second point of Theorem C.4 tells us one can freely orient any chain component within any interventional essential graph and still be able to find a consistent DAG within the equivalence class. This is useful in the lower bound analysis of our proposed benchmark later.

Definition C.5 (Separation of covered edges; Definition 8 of (Choo et al., 2022))**.** 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 $u \sim v$.

Theorem C.6 (Theorem 9 of (Choo et al., 2022))**.** *An intervention set $\mathcal{I}$ is an atomic verifying set for DAG G if and only if $\mathcal{I}$ separates every covered edges of G.*

Theorem C.7 (Theorem 12 of (Choo et al., 2022))**.** *For any DAG G and integer $k \geq 1$, $\nu_k(G) \geq \lceil \nu_1(G)/k \rceil$.*

graph is typically smaller than the maximum degree of the graph.

Theorem C.8 (Proposition 3 of (Eberhardt et al., 2006); Theorem 4 of (Shanmugam et al., 2015); Lemma 17 of (Choo et al., 2022))**.** *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 covered edges of G. Using atomic interventions to orient G, $n-1$ adaptive interventions are necessary in the worst case. Using interventions involving at most $k \geq 1$ vertices each to orient G, $\frac{n}{2k}$ randomized adaptive interventions are necessary. In any case, to orient G, the total number of variables being intervened upon is at least $n/2$.*

Note that Theorem C.8 holds even if the clique is just a subgraph of a larger causal DAG as long as there is no non-clique vertex u such that $\pi(v_i) < \pi(u) < \pi(v_j)$ and $v_i \to u \to v_j$ for any two clique vertices v_i and v_j with $i < j$. Within the proofs of Theorem 4.1 and Theorem D.3, we rely on this observation in combination with Lemma 4.2, which make the clique a prefix within an ordering of interest. This allows us to lower bound our benchmark since our benchmark cares about the worst case ordering.

The next result of (Wienöbst et al., 2021b) is used together with Theorem C.4 to argue that we can always pick an unoriented clique (not necessarily maximal) to be the prefix of a given interventional essential graph.

Definition C.9 (Acyclic moral orientation)**.** An *acyclic moral orientation* is a complete orientation of a partially directed DAG such that it does not create a new v-structure.

Lemma C.10 (Maximal clique picking; (Wienöbst et al., 2021b))**.** *Every acyclic moral orientation of an undirected graph can be represented by a topological ordering which starts with a maximal clique.*

The next result is a lemma in the appendix of (Choo & Shiragur, 2023) that is used to prove Lemma 2.3. We will later use it to prove a generalization of Lemma 2.3 that holds for bounded size interventions.

Lemma C.11. *Let $G = (V, E)$ be a moral DAG. If $u \to v$ in G, then $u \to w$ in G for any two vertices $u, v \in V$ and for all $w \in \texttt{Des}(u) \cap \texttt{Anc}(v)$.*

D. Handling the generalized cost objective

As discussed in Section 4.3, we make algorithmic tweaks to account for Equation (1) and bounded size interventions: see the blue lines in `ALG-GENERALIZED` (Algorithm 3) and `ResolveDanglingGeneralized` (Algorithm 4).

Algorithm 3 `ALG-GENERALIZED`. A weighted adaptive search competitive with respect to Equation (1).

```
Input: Essential graph $\mathcal{E}(G^*)$ of a moral DAG $G^*$, weight function $w : V \to \mathbb{R}$, and integer $k \geq 1$.
Output: Bounded size intervention set $\mathcal{I}$ such that $\mathcal{E}_{\mathcal{I}}(G^*) = G^*$.
Initialize $i = 0$ and $\mathcal{I}_0 = \emptyset$.
while $\mathcal{E}_{\mathcal{I}_i}(G^*)$ still has unoriented edges do
  Initialize $\mathcal{J}_i \leftarrow \emptyset$
  for $H \in CC(\mathcal{E}_{\mathcal{I}_i}(G^*))$ of size $|H| \geq 2$ do
    Find 1/2-clique separator $K_H$ via Theorem 2.2.
    Denote $v_H$ as the costliest vertex $v_H = \mathrm{argmax}_{v \in V(K_H)} w(v)$.
    Let $S$ be the intervention set output by CliqueIntervention on the subclique $V(K_H) \setminus \{v_H\}$ without $v_H$.
    Intervene on $S$ and add $S$ to $\mathcal{J}_i$.
    Let $Z_{v_H} \in CC(\mathcal{E}_{\mathcal{I}_i \cup \mathcal{J}_i}(G^*))$ be the chain component containing $v_H$ after intervening on $S$.
    if $v_H$ is not singleton in $Z_{v_H}$ then
      Add output of ResolveDanglingGeneralized to $\mathcal{J}_i$.
    end if
    Update $\mathcal{I}_{i+1} \leftarrow \mathcal{I}_i \cup \mathcal{J}_i$ and $i \leftarrow i + 1$.
  end for
end while
Return $\mathcal{I}_i$
```

The correctness of Algorithm 4 relies on Lemma 4.7 Lemma D.1, and Lemma D.2. Note that Algorithm 4 does *not* attempt to fully orient the edges within the 1/2-clique separator while searching for a prefix clique of size at most k. Since we are *not* guaranteed to know the source node of K so we cannot hope to directly apply Lemma 4.7, and thus we need to prove generalized version of Lemma D.2 to justify why `ResolveDanglingGeneralized` terminates after $\mathcal{O}(\log n)$ iterations. In fact, Lemma 4.7 is the special case where the clique K is a single vertex.

Algorithm 4 `ResolveDanglingGeneralized`. A subroutine for `ALG-GENERALIZED`.

```
Input: Interventional essential graph $\mathcal{E}_{\mathcal{I}'}(G^*)$ for some intervention $\mathcal{I}' \subseteq 2^V$, weight function $w : V \to \mathbb{R}$, a chain component $H$ of $\mathcal{E}_{\mathcal{I}'}(G^*)$ that contains vertex $v \in V(H)$ with $t$ disjoint connected components $H_1, \ldots, H_t$ in $H[V(H) \setminus \{v\}]$, and an integer $k \geq 1$.
Output: Bounded size intervention set $\mathcal{I}$ such that the $t$ components are mutually disjoint in $\mathcal{E}_{\mathcal{I} \cup \mathcal{I}'}(H)$.
Initialize $\mathcal{I} \leftarrow \emptyset$.
if $\alpha \cdot w(v) + \beta \leq \sum_{i=1}^{t} \max_{\text{clique } C \text{ in } H_i \cap N_H(v)} \alpha \cdot w(C) + \beta \cdot |V(C)|$ then
    Intervene on $v$ and set $\mathcal{I} \leftarrow \{\{v\}\}$.
else
    for $i \in \{1, \ldots, t\}$ do
        Initialize $V' \leftarrow V(H_i) \cap N_H(v)$.
        while skel$(\mathcal{E}(G^*))[V']$ is *not* a clique, or $|V'| > k$ do
            Find a 1/2-clique separator $K$ of $H_i[V']$ using Theorem 2.2.
            Arbitrarily partition the vertices of $K$ into sets $S_1, \ldots, S_{\lceil |V(K)|/k \rceil} \subseteq V$, each involving at most $k$ vertices.
            Intervene on the vertices in the sets $\mathcal{S} = \{S_1, \ldots, S_{\lceil |V(K)|/k \rceil}\}$ and add $S_1, \ldots, S_{\lceil |V(K)|/k \rceil}$ to $\mathcal{I}$.
            By Lemma D.1, identify $S_{source} \in \mathcal{S}$ which is the set containing the source node of $K$.
            By Lemma D.2, determine if there exists a chain component $Q$ with only incoming arcs to $S_{source}$. If so, find it.
            if $Q$ exists then
                Restrict $V'$ to $V(Q)$.
            else
                Set $V' \leftarrow V(S_{source})$.
            end if
        end while
        Let $S$ be the intervention set output by CliqueIntervention on the clique $H_i[V']$ involving at most $k$ vertices.
        Intervene on $S$ and add $S$ to $\mathcal{I}$.
    end for
end if
Return $\mathcal{I}$
```

Lemma D.1. *Consider any arbitrary directed clique $G = (V, E)$ and any integer $k \geq 1$. Without loss of generality, $V = \{v_1, \ldots, v_n\}$ and $\pi(v_1) < \ldots < \pi(v_n)$, i.e. v_1 is the source of G. Suppose we arbitrarily partition the vertex set into sets $\mathcal{S} = \{S_1, \ldots, S_{\lceil n/k \rceil}\}$, each of size at most k. Then, the set $S_{source} \in \mathcal{S}$ containing v_1 is the* unique *set in $\mathcal{S}$ that has a vertex without any incoming arcs from the other sets.*

Lemma D.2. *Let $\mathcal{E}_{\mathcal{I}}(G)$ be the interventional essential graph of a moral DAG $G = (V, E)$ with respect to intervention set $\mathcal{I} \subseteq 2^V$. Fix any chain component $H \in CC(\mathcal{E}_{\mathcal{I}}(G))$ and let K be an arbitrary clique in H. If K contains the source node of H, then there are no chain components of $\mathcal{E}_{\mathcal{I} \cup \{V(K)\}}(H)$ with only incoming arcs into K in G. Otherwise, if K does not contain the source node of H, then there is exactly one chain component of $\mathcal{E}_{\mathcal{I} \cup \{V(K)\}}(H)$ with only incoming arcs into K in G. Furthermore, without further interventions, we can decide if such a chain component exist (and find it) in polynomial time.*

To obtain bounded size interventions for intervening on cliques, we invoke Lemma 4.8 through the subroutine `CliqueIntervention` (Algorithm 5). Lemma 4.9 states the guarantees of `CliqueIntervention`.

Lemma 4.9. *Given a set of clique vertices $V(C) \subseteq V$ and integer $k \geq 1$, Algorithm 5 returns a set $S \subseteq 2^{V(C)}$ such that each partite in S has at most k vertices. When $k = 1$, $|S| = |V(C)|$ and each vertex appears exactly once in S. When $k > 1$, $|S| \in \mathcal{O}(\log k \cdot |V(C)|/k)$ and each vertex appears at most $\mathcal{O}(\log k)$ times in S.*

Analogous to Theorem 4.1 and Lemma 4.6, we prove Theorem D.3 and Lemma D.4 for our tweaked algorithm with respect to the generalized cost Equation (1).

Theorem D.3. *Fix an essential graph $\mathcal{E}(G^*)$ corresponding to an unknown weighted causal DAG G^*. Suppose $\mathcal{I}_1^*$ and $\mathcal{I}_k^*$ are an atomic and bounded size intervention sets minimizing Equation (1) such that $\mathcal{E}_{\mathcal{I}_1^*}(G^*) = \mathcal{E}_{\mathcal{I}_k^*}(G^*) = G^*$,*

Algorithm 5 `CliqueIntervention`. A labelling subroutine based on Lemma 4.8.

```
**Input**: A set of clique vertices $C \subseteq V$, integer $k \geq 1$.
**Output**: Partition $S$ of $C$.
**if** $k = 1$ **then**
    Define $\mathcal{I} = \{\{v\} : v \in C\}$
**else**
    Define $k' = \min\{k, |C|/2\}$, $a = \lceil |C|/k' \rceil \geq 2$, and $\ell = \lceil \log_a |C| \rceil$. Compute labelling scheme on $C$ with $(|C|, k, a)$ via Lemma 4.8 and define $\mathcal{I} = \{S_{x,y}\}_{x \in [\ell], y \in [a]}$, where $S_{x,y} \subseteq Q$ is the subset of vertices whose $x^{th}$ letter in the label is $y$.
**end if**
**Return** $\mathcal{I}$
```

$\texttt{cost}(\mathcal{I}_1^*, \alpha, \beta, 1) = \texttt{OPT}_1$, and $\texttt{cost}(\mathcal{I}_k^*, \alpha, \beta, k) = \texttt{OPT}_k$. Then, maximizing over intervention sets $\mathcal{I} \subseteq V$, we have

$$\texttt{OPT}_1 \geq \max_{\substack{\mathcal{I} \subseteq 2^V \\ \mathcal{I} \text{ atomic}}} \left\{ \sum_{\substack{H \in CC(\mathcal{E}_{\mathcal{I}}(G^*)) \\ |V(H)| \geq 2}} \max \left\{ \zeta_{\mathcal{I},H}^{(3)}, \zeta_{\mathcal{I},H}^{(4)} \right\} \right\} \quad \text{and} \quad \texttt{OPT}_k \geq \max_{\substack{\mathcal{I} \subseteq 2^V \\ \mathcal{I} \text{ bounded size}}} \left\{ \sum_{\substack{H \in CC(\mathcal{E}_{\mathcal{I}}(G^*)) \\ |V(H)| \geq 2}} \max \left\{ \zeta_{\mathcal{I},H}^{(5)}, \zeta_{\mathcal{I},H}^{(6)} \right\} \right\}$$

where

$$\zeta_{\mathcal{I},H}^{(3)} = \frac{1}{2} \cdot \max_{\text{clique } C \in H} \left\{ \alpha \cdot \left(w(V(C)) - \max_{v \in V(C)} \{w(v)\} \right) + \beta \cdot |V(C)| \right\} ,$$

$$\zeta_{\mathcal{I},H}^{(4)} = \frac{1}{2} \cdot \max_{v \in V(H)} \left\{ \min \left\{ \alpha \cdot w(v) + \beta, \sum_{i=1}^{t} \max_{\substack{\text{clique } C_i: \\ V(C_i) \subseteq V_i \cap N_H(v)}} \{ \alpha \cdot w(V(C_i)) + \beta \cdot |V(C_i)| \} \right\} \right\} ,$$

$$\zeta_{\mathcal{I},H}^{(5)} = \frac{1}{2} \cdot \max_{\text{clique } C \in H} \left\{ \alpha \cdot \left(w(V(C)) - \max_{v \in V(C)} \{w(v)\} \right) + \frac{\beta}{k} \cdot |V(C)| \right\} ,$$

$$\zeta_{\mathcal{I},H}^{(6)} = \frac{1}{2} \cdot \max_{v \in V(H)} \left\{ \min \left\{ \alpha \cdot w(v) + \frac{\beta}{k}, \sum_{i=1}^{t} \max_{\substack{\text{clique } C_i: \\ V(C_i) \subseteq V_i \cap N_H(v)}} \left\{ \alpha \cdot w(V(C_i)) + \frac{\beta}{k} \cdot |V(C_i)| \right\} \right\} \right\} ,$$

and $V_1, \ldots, V_t \subseteq V(H)$ are vertex sets of the $t \geq 1$ disjoint connected components in $H[V(H) \setminus \{v\}]$ in $\zeta_{\mathcal{I},H}^{(4)}$ and $\zeta_{\mathcal{I},H}^{(6)}$.

Lemma D.4. *Fix an interventional essential graph $\mathcal{E}_{\mathcal{I}'}(G^*)$ corresponding to an unknown weighted causal moral DAG G^* and some intervention $\mathcal{I}' \subseteq 2^V$. Suppose $\mathcal{I}_1^*$ and $\mathcal{I}_k^*$ are atomic and bounded size intervention sets minimizing Equation (1) such that $\mathcal{E}_{\mathcal{I}_1^*}(G^*) = \mathcal{E}_{\mathcal{I}_k^*}(G^*) = G^*$, $\texttt{cost}(\mathcal{I}_1^*, \alpha, \beta, 1) = \texttt{OPT}_1$, and $\texttt{cost}(\mathcal{I}_k^*, \alpha, \beta, k) = \texttt{OPT}_k$. Let H be a chain component of $\mathcal{E}_{\mathcal{I}'}(G^*)$ containing a vertex $v \in V(H)$. Then,*

- *When $k = 1$, Algorithm 4 returns an atomic intervention set $\mathcal{I}$ such that connected components in $H[V(H) \setminus \{v\}]$ are mutually disjoint in $\mathcal{E}_{\mathcal{I}}(H)$ and $\texttt{cost}(\mathcal{I}, \alpha, \beta, 1) \in \mathcal{O}(\log n \cdot \texttt{OPT}_1)$.*
- *When $k > 1$, Algorithm 4 returns a bounded size intervention set $\mathcal{I}$ such that connected components in $H[V(H) \setminus \{v\}]$ are mutually disjoint in $\mathcal{E}_{\mathcal{I}}(H)$ and $\texttt{cost}(\mathcal{I}, \alpha, \beta, k) \in \mathcal{O}((\log n + \log k) \cdot \texttt{OPT}_k)$.*

Denote an iteration of the while loop in Algorithm 1 as a *phase*. Since Algorithm 1 and Algorithm 3 are essentially the same in terms of how they recurse on smaller chain components of at most half the size in each phase, we can also obtain Lemma D.5.

Lemma D.5. *$\texttt{ALG-GENERALIZED}$ (Algorithm 3) terminates after $\mathcal{O}(\log n)$ phases.*

Using Theorem D.3, we can also obtain Lemma D.6.

Lemma D.6. *Suppose $\mathcal{I}_1^*$ and $\mathcal{I}_k^*$ are an atomic and bounded size verifying sets respectively for G^* that minimizes Equation (1) with $\texttt{cost}(\mathcal{I}_1^*) = \texttt{OPT}_1$ and $\texttt{cost}(\mathcal{I}_k^*) = \texttt{OPT}_k$. Each phase in $\texttt{ALG-GENERALIZED}$ (Algorithm 3) incurs a cost of $\mathcal{O}(\log n \cdot \texttt{OPT}_1)$ when $k = 1$ and $\mathcal{O}\left((\log n + \log k) \cdot \texttt{OPT}_k\right)$ when $k > 1$.*

Algorithm 6 Naive weighted adaptive search.

```
Input: Essential graph $\mathcal{E}(G^*)$ of a moral DAG $G^*$ and weight function $w : V \to \mathbb{R}$.
Output: Atomic intervention set $\mathcal{I}$ s.t. $\mathcal{E}_{\mathcal{I}}(G^*) = G^*$.
Sort the vertices in non-decreasing weight ordering.
while $\mathcal{E}_{\mathcal{I}_i}(G^*)$ still has unoriented edges do
    Intervene on the next cheapest unintervened vertex and add it to $\mathcal{I}$.
end while
Return $\mathcal{I}_i$
```

Then, Theorem 3.3 follows directly by combining Lemma D.5 and Lemma D.6.

See Appendix F for the full proofs of Lemma 4.9, Theorem D.3, Lemma D.5, and Lemma D.6.

E. Blackbox combination of algorithms

In this section, we describe a deterministic naive algorithm (Algorithm 6) which provably incurs a cost of $\mathcal{O}(n \cdot \overline{\nu}(G^*))$ and show how to combine this algorithm in a blackbox manner with any other deterministic algorithm to augment it with the provable guarantee of incurring a cost of at most $\mathcal{O}(n \cdot \overline{\nu}(G^*))$.

Lemma E.1. *Fix an essential graph $\mathcal{E}(G^*)$ corresponding to an unknown weighted causal DAG G^*. Algorithm 6 is a deterministic and adaptive algorithm that computes an atomic intervention set $\mathcal{I}$ in polynomial time such that $\mathcal{E}_{\mathcal{I}}(G^*) = G^*$ and $w(\mathcal{I}) \in \mathcal{O}\left(n \cdot \overline{\nu}_1(G^*)\right)$.*

Proof. The weight $w(v_{final})$ of the final intervened vertex v_{final} is a lower bound for $\overline{\nu}(G^*)$. Meanwhile, we intervened at most n vertices before v_{final}, each of which has cost lower than $w(v_{final})$. □

Theorem E.2. *Fix an essential graph $\mathcal{E}(G^*)$ corresponding to an unknown weighted causal DAG G^*. Let A be a deterministic algorithm that that computes an atomic intervention set $\mathcal{I}$ in polynomial time such that $\mathcal{E}_{\mathcal{I}}(G^*) = G^*$ and $w(\mathcal{I}) \in \mathcal{O}(C)$. Then, there is a deterministic and adaptive algorithm that computes an atomic intervention set $\mathcal{I}$ in polynomial time such that $\mathcal{E}_{\mathcal{I}}(G^*) = G^*$ and $w(\mathcal{I}) \in \mathcal{O}\left(\min\{C, n \cdot \overline{\nu}_1(G^*)\}\right)$.*

Proof. Let A_{naive} denote Algorithm 6. We will run both A and A_{naive} in parallel with a budget constraint (that doubles whenever it is exhausted) until we fully orient the causal graph.

More precisely, our new algorithm A_{new} is defined as follows:

1. We initialize a budget of $B = \min_{v \in V} w(v)$ to the minimum vertex cost. Without loss of generality, we may assume that $B > 0$ by first intervening on all vertices with 0 cost.

2. "Simulate" A until the total accumulated cost is at most B. If the graph is fully oriented at any point in time, terminate.

3. "Simulate" A_{naive} until the total accumulated cost is at most B. If the graph is fully oriented at any point in time, terminate.

4. Double the value of B (or increase it to the next weight so that there is no "empty iteration") and return to step 2.

By "simulate", we mean that we accumulate the cost of vertices but only intervene on vertices that has not been intervened on previously. We can do this because A and A_{naive} are deterministic.

Since A_{new} only terminates whenever either A or A_{naive} succeeds in fully orienting the causal graph, A_{new} will correctly fully orient any input graph. Note that we always have $B \in \mathcal{O}(C)$ and $B \in \mathcal{O}(n \cdot \overline{\nu}_1(G^*))$ at any point of the modified algorithm whenever neither algorithm terminated. Since we always double the budget (any constant factor multiplication works), the above asymptotic upper bound also holds for B_{final}, where B_{final} is the final value of B when the algorithm terminates. Furthermore, the cost of A_{new} is at most $2 \cdot B_{final}$ since we pay at most B_{final} for running A and at most B_{final} for running A_{naive}.

A_{new} runs in polynomial time because value of B changes at most $\mathcal{O}\left(\min\left\{n, \log \frac{\max_{v \in V} w(v)}{\min_{v \in V} w(v)}\right\}\right)$ times and each simulation of A and A_{naive} runs in polynomial time. □

Remark about implementation Both A and A_{naive} are actually intervening on the same causal graph, and A_{new} does not actually discard information gained from A when simulating A_{naive} (and vice versa). This may actually help the algorithm to terminate faster (at a lower cost) than running A or A_{naive} independently.

F. Deferred proofs

F.1. Why $\overline{\nu}_1(G^*)$ is not an ideal benchmark

Theorem 3.1. *For any adaptive search algorithm A, deterministic or randomized, there exists a weighted causal graph G^* such that $\texttt{cost}(A, G^*, 1, 0) \in \Omega(n \cdot \overline{\nu}_1(G^*))$.*

Proof. Let $G^* = (V, E, w)$ be a weighted causal directed tree where $|V| = n$ and $\text{skel}(G^*)$ is a star graph in which $n - 1$ vertices have degree 1 (non-center nodes) and a single vertex has degree n – 1 (center node). The weights of the nodes are given as follows,

$$w(v) = \begin{cases} n-1 & v \text{ is a center} \\ 1 & \text{otherwise .} \end{cases}$$

We let one of the $n - 1$ non-center nodes be the root of G^*. As intervening on the root suffices to fully orient G^* from its essential graph $\mathcal{E}(G^*)$, we see that $\overline{\nu}_1(G^*) = 1$.

Observe that any adaptive search algorithm that intervenes on the center of the star immediately incurs $n-1 \in \Omega(n \cdot \overline{\nu}_1(G^*))$. Meanwhile, intervening on any leaf vertex that is *not* the root will only orient the single edge incident to it so $n-1$ non-center node interventions are needed in the worst case, incurring a cost of $\Omega(n \cdot \overline{\nu}_1(G^*))$.

For randomized algorithms, we will use Yao's lemma (Lemma C.1): the expected worst case performance of a *randomized* algorithm is at least as much the expected performance of the best *deterministic* algorithm over some distribution of inputs.

Consider the distribution of G^* by uniformly picking the root amongst the leaves. Any deterministic algorithm A can be uniquely mapped to a sequence σ_A of vertices that it will intervene on, until $\mathcal{E}(G^*)$ is fully oriented. Since the center is never the root in our distribution, any algorithm A that intervene on the center within its first $n - 1$ choices strictly perform worse than the alternative algorithm A' that shifts the choice of intervening to the last vertex, i.e. if $\sigma_A(j)$ is the center, then

$$\sigma_{A'}(i) = \begin{cases} \sigma_A(i) & \text{if } i < j \\ \sigma_A(i+1) & \text{if } j \leq i < n-1 \\ \text{center} & \text{if } i = n \end{cases}$$

Then, for any algorithm A that does *not* intervene on the center within its first $n - 1$ choices, we see that the intervention set $\mathcal{I}$ produced by A has expected cost

$$\mathbb{E}[w(\mathcal{I})] = \frac{1}{n-1} \cdot (1 + 2 + 3 + \ldots + (n-1)) \in \Omega(n \cdot \overline{\nu}_1(G^*))$$

□

F.2. Lower bounding the benchmark

As Theorem 4.1 relies on Lemma 4.2, we will prove Lemma 4.2 first.

Lemma 4.2. *Fix an interventional essential graph $\mathcal{E}_{\mathcal{I}}(G^*)$ corresponding to an arbitrary moral DAG G^* and intervention set $\mathcal{I} \subseteq 2^V$. For any clique C (not necessarily maximal) in any chain component of $\mathcal{E}_{\mathcal{I}}(G^*)$ and any permutation ordering π on the vertices $V(C)$ of C, there exists a DAG G consistent with $\mathcal{E}_{\mathcal{I}}(G^*)$ such that $u \to v$ if and only if $\pi(u) < \pi(v)$ for any two clique vertices $u, v \in V(C)$.*

Proof. Let H be the chain component containing the clique C of interest. By Theorem C.4, we can orient H independently of all other chain components and still obtain a DAG that is consistent with $\mathcal{E}_{\mathcal{I}}(G^*)$.

Let C' be a *maximal clique* that includes C. By Lemma C.10, there is an acyclic moral orientation of H such that the vertices of C' appear before all other vertices in H. Let this acyclic moral orientation be the DAG H'. Then, we see that the interventional essential graph $\mathcal{E}_{V(H) \setminus V(C')}(H')$ by intervening on every single vertex outside of C' has only one chain component, which is precisely the clique C'. By Theorem C.4, we can orient C' independently and still obtain a DAG that is consistent with $\mathcal{E}_{\mathcal{I}}(G^*)$. So, we can order all vertices in $V(C') \setminus V(C)$ *after* the vertices in $V(C)$ and order the vertices within C according to the given desired ordering π. □

Theorem 4.1. *For any DAG G^* and its essential graph $\mathcal{E}(G^*)$, we have*

$$\overline{\nu}_1^{\max}(G^*) \geq \max_{\mathcal{I} \subseteq V} \left\{ \sum_{\substack{H \in CC(\mathcal{E}_{\mathcal{I}}(G^*)) \\ |V(H)| \geq 2}} \max \left\{ \zeta^{(1)}_{\mathcal{I},H}, \zeta^{(2)}_{\mathcal{I},H} \right\} \right\}$$

where we maximize over atomic intervention sets $\mathcal{I} \subseteq V$,

$$\zeta^{(1)}_{\mathcal{I},H} = \frac{1}{2} \cdot \max_{\text{clique } C \in H} \left\{ w(V(C)) - \max_{v \in V(C)} \{w(v)\} \right\}$$

and

$$\zeta^{(2)}_{\mathcal{I},H} = \frac{1}{2} \cdot \max_{v \in V(H)} \left\{ \min \{w(v), \gamma_{H,v}\} \right\}$$

where $\gamma_{H,v} = \sum_{i=1}^{t} \max_{\substack{\text{clique } C_i: \\ V(C_i) \subseteq V_i \cap N_H(v)}} \{w(V(C_i))\}$ with $V_1, \ldots, V_t \subseteq V(H)$ being vertex sets of the $t \geq 1$ disjoint connected components in $H[V(H) \setminus \{v\}]$.

Proof. Fix an underlying causal graph G^* and consider an arbitrary atomic intervention set $\mathcal{I} \subseteq V$. We will prove for $\mathcal{I}$ and then the claim follows by taking a maximization over all possible atomic intervention sets. We will prove the two cases separately by mirroring parts of the proof of Lemma C.3 in how we invoke Lemma C.2.

Fix an *arbitrary* atomic intervention set $\mathcal{I} \subseteq V$ and consider an *arbitrary* DAG $\widetilde{G}$ that is consistent with $\mathcal{E}_{\mathcal{I}}(G^*)$. That is, $\text{skel}(\widetilde{G}) = \text{skel}(\mathcal{E}_{\mathcal{I}}(G^*))$ and all the oriented edges in $\mathcal{E}_{\mathcal{I}}(G^*)$ appear in the same direction in $\widetilde{G}$. Fix a chain component $H \in CC(\mathcal{E}_{\mathcal{I}}(\widetilde{G}))$ and let $\mathcal{I}' \subseteq V$ be any atomic verifying set of $\widetilde{G}$, that is, $\mathcal{E}_{\mathcal{I}'}(\widetilde{G}) = \widetilde{G}$ and $\mathcal{E}_{\mathcal{I}'}(\widetilde{G})[V(H)] = \widetilde{G}[V(H)]$. Note that,

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

where the first equality is due to Lemma C.2 and the last equality is because $\mathcal{I}'$ is a verifying set of $\widetilde{G}$. So, $(\mathcal{I}' \setminus \mathcal{I}) \cap V(H)$ is a verifying set for $\widetilde{G}[V(H)]$, and so is $\mathcal{I}' \cap V(H)$. Thus, by minimality of $\overline{\nu}_1$, we have

$$\overline{\nu}_1(\widetilde{G}[V(H)]) \leq w(\mathcal{I}' \cap V(H)) \tag{3}$$

for *any* atomic verifying set $\mathcal{I}' \subseteq V$ of $\widetilde{G}$.

We now independently lower bound $\overline{\nu}_1(\widetilde{G}[V(H)])$ by $\zeta^{(1)}_{\mathcal{I},H}$ and $\zeta^{(2)}_{\mathcal{I},H}$. To do so, we will construct a DAG $\widetilde{G}$ that is consistent with the interventional essential graph $\mathcal{E}_{\mathcal{I}}(G^*)$ by making vertices of some unoriented clique the prefix of its chain component by using Lemma 4.2, and then invoking Equation (3) to lower bound the interventional cost in each chain component H. Note that when we fix the ordering of vertices within a chain component, it does not affect the ordering of the vertices outside of that chain component.

Lower bounding via $\zeta^{(1)}_{\mathcal{I},H}$: For each connected component $H \in CC(\mathcal{E}_{\mathcal{I}}(G^*))$, fix an arbitrary clique C in H. Suppose the vertices in C are $v_1, \ldots, v_{|C|}$ with $w(v_1) \geq \ldots \geq w(v_{|C|})$. By Lemma 4.2, there exists a valid orientation π of H such that all the vertices in C appear at the start of the ordering. For any such ordering π, the covered edges are $v_{\pi(1)} \to v_{\pi(2)} \to \ldots \to v_{\pi(|C|)}$ and we know that any atomic verifying set must include a minimum vertex cover of these covered edges due to Theorem C.6. Let $\widetilde{G}$ be one such DAG which imposes the descending weight ordering π on the vertices within H, i.e. $w(v_{\pi(i)}) = w(v_i)$. Consider the set of *disjoint* alternating covered edges $\pi^{-1}(1) \to \pi^{-1}(2)$, $\pi^{-1}(3) \to \pi^{-1}(4)$, and so on. Amongst these disjoint alternating covered edges, at least one endpoint must be intervened upon, incurring a cost of at least $\sum_{\text{even } i} w(v_i)$. That is, $\overline{\nu}_1(\widetilde{G}[V(H)]) \geq \sum_{\text{even } i} w(v_i)$.

Since $w(v_1) \geq \ldots \geq w(v_{|C|})$, we see that

$$w(C) = w(v_1) + \sum_{\text{even } i} w(v_i) + \sum_{\substack{\text{odd } i \\ i \geq 3}} w(v_i) \leq w(v_1) + \sum_{\text{even } i} w(v_i) + \sum_{\substack{\text{odd } i \\ i \geq 3}} w(v_{i-1}) \leq w(v_1) + 2 \cdot \sum_{\text{even } i} w(v_i) \ .$$

Therefore,

$$\overline{\nu}_1(\widetilde{G}[V(H)]) \geq \sum_{\text{even } i} w(v_i) \geq \frac{1}{2} \cdot (w(V(C)) - w(v_1)) \ .$$

By maximizing amongst the cliques within H, we see that $\overline{\nu}_1(\widetilde{G}[V(H)]) \geq \zeta^{(1)}_{\mathcal{I},H}$.

Lower bounding via $\zeta^{(2)}_{\mathcal{I},H}$:

For each connected component $H \in CC(\mathcal{E}_{\mathcal{I}}(G^*))$, fix an arbitrary vertex v in H. To bound $\gamma_{H,v}$, it suffices to consider *arbitrary* cliques C_i in each disjoint connected components in $H[V \setminus \{v\}]$, and then taking the maximum.

Consider a minimum cost atomic verifying set $\mathcal{I}$ of $\widetilde{G}[V(H)]$ with $w(\mathcal{I}) = \overline{\nu}_1(\widetilde{G}[V(H)])$.

Case 1: $v \in \mathcal{I}$. Then, $\overline{\nu}_1(\widetilde{G}[V(H)]) \geq w(v) \geq \frac{w(v)}{2} \geq \frac{1}{2} \cdot \min\left\{ w(v), \sum_{i=1}^{t} w(V(C_i)) \right\}$.

By maximizing amongst the cliques within each connected component, we see that $\overline{\nu}_1(\widetilde{G}[V(H)]) \geq \zeta^{(2)}_{\mathcal{I},H}$.

Case 2: $v \not\in \mathcal{I}$.

By Lemma 4.2, there exists DAGs consistent with $\mathcal{E}(G^*)$ that can be generated by letting v be the first prefix vertex in $\mathcal{E}(G^*)$, followed by vertices in *descending weight ordering* within each clique C_i, across all t components. Let $\widetilde{G}$ be one such DAG and suppose the vertices in clique $C_i = \{u_{i,1}, \ldots, u_{i,|C_i|}\}$ have weights $w(u_{i,1}) \geq \ldots w(u_{i,|C_i|})$ and $\pi(v) < \pi(u_{i,1}) < \ldots < \pi(u_{i,|C_i|})$. We see that the set $\{v \to u_{i,1}, u_{i,1} \to u_{i,2}, \ldots, u_{i,|C_i|-1} \to u_{i,|C_i|}\}_{i=1}^{t}$ are all covered edges of $\widetilde{G}$. By Theorem C.6, *any* verification set must include a minimum vertex cover of these edges. In particular, since $v \not\in \mathcal{I}$, we must have $\{u_{i,1}\}_{i=1}^{t} \subseteq \mathcal{I}$.

Let $A \subseteq E(G^*)$ be the covered edges of $\widetilde{G}$. From above, we know that $\{v \to u_{i,1}, u_{i,1} \to u_{i,2}, \ldots, u_{i,|C_i|-1} \to u_{i,|C_i|}\}_{i=1}^{t} \subseteq A$. Define $B = A \setminus \{v \to u_{i,1}, u_{i,1} \to u_{i,2}\}_{i=1}^{t}$ as the remaining covered edges in the above discussion after removing edges covered by $\{u_{i,1}\}_{i=1}^{t}$. That is, *conditioned on not using* v, A's minimum cost vertex cover has cost $\sum_{i=1}^{t} w(u_{i,1})$ plus the cost B's minimum cost vertex cover.

For each clique $C_i = \{u_{i,1}, \ldots, u_{i,|C_i|}\}$ amongst the disjoint cliques, consider the set of *disjoint* alternating covered edges $u_{i,2} \to u_{i,3}$, $u_{i,4} \to u_{i,5}$, and so on. Amongst these disjoint alternating covered edges, at least one endpoint must be chosen for any vertex cover of B, incurring a cost of at least $\sum_{\substack{\text{odd } i \\ i \geq 3}} w(u_{i,j})$.

Since $w(u_{i,1}) \geq \ldots \geq w(u_{i,|C_i|})$, we see that

$$w(V(C_i)) = w(u_{i,1}) + w(u_{i,2}) + \sum_{\substack{\text{even } i \\ i \geq 4}} w(u_{i,j}) + \sum_{\substack{\text{odd } i \\ i \geq 3}} w(u_{i,j}) \leq 2 \cdot \left(w(u_{i,1}) + \sum_{\substack{\text{odd } i \\ i \geq 3}} w(u_{i,j}) \right) \ .$$

So, the minimum cost vertex cover of B is at least $\frac{1}{2} \sum_{i=1}^{t} (w(V(C_i)) - 2 \cdot w(u_{i,1}))$ and

$$\begin{aligned} \overline{\nu}_1(\widetilde{G}[V(H)]) &\geq \sum_{i=1}^{t} w(u_{i,1}) + \frac{1}{2} \sum_{i=1}^{t} (w(V(C_i)) - 2 \cdot w(u_{i,1})) \\ &\geq \frac{1}{2} \sum_{i=1}^{t} w(V(C_i)) \\ &\geq \frac{1}{2} \cdot \min\left\{ w(v), \sum_{i=1}^{t} w(V(C_i)) \right\} \ . \end{aligned}$$

By maximizing amongst the cliques within each connected component, we see that $\overline{\nu}_1(\widetilde{G}[V(H)]) \geq \zeta^{(2)}_{\mathcal{I},H}$.

Putting together:

Since $\mathcal{I}^*$ is the minimum cost verifying set,

$$\begin{aligned}\overline{\nu}^{\max}_1(G^*) &= \max_{G \in [G^*]} \overline{\nu}_1(G) \geq \overline{\nu}_1(\widetilde{G}) = w(\mathcal{I}^*)\\ &\overset{(*)}{\geq} \sum_{\substack{H \in CC(\mathcal{E}_{\mathcal{I}}(G^*))\\ |V(H)| \geq 2}} w(\mathcal{I}^* \cap V(H)) \geq \sum_{\substack{H \in CC(\mathcal{E}_{\mathcal{I}}(G^*))\\ |V(H)| \geq 2}} \overline{\nu}_1(\widetilde{G}[V(H)]) \geq \sum_{\substack{H \in CC(\mathcal{E}_{\mathcal{I}}(G^*))\\ |V(H)| \geq 2}} \max\{\zeta^{(1)}_{\mathcal{I},H}, \zeta^{(2)}_{\mathcal{I},H}\}\end{aligned}$$

where the inequality $(*)$ is because some edges may have already been oriented by $\mathcal{I}$.

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

Theorem 4.3. *For any DAG G^* and integer $k \geq 1$, $\overline{\nu}^{\max}_k(G^*) \geq \overline{\nu}^{\max}_1(G^*)$ and $\nu^{\max}_k(G^*) \geq \lceil \nu^{\max}_1(G^*)/k \rceil$.*

Proof. **Proof for $\overline{\nu}^{\max}_k(G^*) \geq \overline{\nu}^{\max}_1(G^*)$**:

Observe that intervening on all vertices in a bounded size intervention one-by-one in an atomic fashion will not increase the cost and can only recover more information about the causal graph. Let us formalize this: Suppose $\mathcal{I}^*_k \subseteq 2^V$ is a minimum cost bounded size verifying set. Define $\mathcal{I} = \cup_{S \in \mathcal{I}^*_k} S$ as an atomic intervention set that involves all vertices in $\mathcal{I}^*_k$ exactly once. So, by construction, $w(\mathcal{I}) \leq w(\mathcal{I}^*_k)$. By Theorem C.6, we know that $\mathcal{I}^*_k$ must separate all covered edges of G^*. Meanwhile, by construction, $\mathcal{I}$ also separates all covered edges of G^* while having $w(\mathcal{I}) \leq \sum_{S \in \mathcal{I}^*_k} \sum_{v \in S} w(v) = w(\mathcal{I}^*_k)$. Thus, $\overline{\nu}^{\max}_1(G^*) \leq \overline{\nu}^{\max}_k(G^*)$.

Proof for $\nu^{\max}_k(G^*) \geq \nu^{\max}_1(G^*)$:

Observe that

$$\nu^{\max}_k(G^*) = \max_{G \in [G^*]} \nu_k(G) \geq \max_{G \in [G^*]} \lceil \nu_1(G)/k \rceil = \left\lceil \max_{G \in [G^*]} \nu_1(G)/k \right\rceil = \lceil \nu^{\max}_1(G^*)/k \rceil$$

where the inequality is due to Theorem C.7. □

F.3. A competitive adaptive search algorithm

Lemma 4.6. *Fix an interventional essential graph $\mathcal{E}_{\mathcal{I}'}(G^*)$ corresponding to an unknown weighted causal moral DAG G^* and some intervention $\mathcal{I}' \subseteq 2^V$. Let H be a chain component of $\mathcal{E}_{\mathcal{I}'}(G^*)$ containing a vertex $v \in V(H)$. Then, Algorithm 2 returns an atomic intervention set $\mathcal{I}$ such that all the outgoing edges of v within H are oriented in $\mathcal{E}_{\mathcal{I}' \cup \mathcal{I}}(H)$ and $w(\mathcal{I}) \in \mathcal{O}(\log n \cdot \overline{\nu}^{\max}_1(G^*))$.*

Proof. Since the underlying graph is a moral DAG, intervening on v or $\{\operatorname{argmin}_{u \in V(H_i) \cap N_H(v)} \pi(u)\}_{i \in [t]}$ ensures that all the outgoing edges of v in H are oriented (Lemma 2.3). Suppose $u_i = \operatorname{argmin}_{u \in V(H_i) \cap N_H(v)} \pi(u)$. If $\pi(v) > \min_{i \in \{1,\ldots,t\}} \pi(u_i)$, then intervening on $u_1, \ldots, u_t$ will disconnect[12] H_i's from each other[13]. Otherwise, if $\pi(v) < \min_{i \in \{1,\ldots,t\}} \pi(u_i)$, Lemma 2.3 tells us that intervening on u_i will orient all $v \to z$ arcs for $z \in H_i$. In both cases, we orient all the outgoing edges of v within H.

The if-case of `ResolveDangling` directly intervenes on v while the else-case of `ResolveDangling` repeatedly recurses on a connected subgraph of $H_i[V']$, towards the source u_i. Since K_{H_i} is a 1/2-clique separator of H_i at each iteration, the size of V' is at least halved in each iteration of the while-loop, and there will be at most $\mathcal{O}(\log n)$ iterations. By the $\zeta^{(2)}$ term of Theorem 4.1, we see that for each iteration, the cost of finding all the $\{u_i\}_{i \in [t]}$ has a cost at most $2 \cdot \overline{\nu}^{\max}_1(G^*)$. Put together, we see that $w(\mathcal{I}) \in \mathcal{O}(\log n \cdot \overline{\nu}^{\max}_1(G^*))$. □

[12] Every path between H_i and H_j, for $i \neq j$ will involve an oriented arc. Such arcs will be removed when considering chain components, disconnecting the path.

[13] Without loss of generality, suppose $\pi(u_1) = \min_{i \in \{1,\ldots,t\}} \pi(u_i)$. Orienting the arc $u_1 \to v_H$ triggers Meek rule R1 to orient all $v \to z$ arcs for $z \not\in H_1$, thus disconnecting H_i's from each other.

Lemma 4.7. *Let $\mathcal{E}_{\mathcal{I}}(G)$ be the interventional essential graph of a moral DAG $G = (V, E)$ with respect to intervention set $\mathcal{I} \subseteq V$. Fix any chain component $H \in CC(\mathcal{E}_{\mathcal{I}}(G))$ and vertex $v \in V(H)$. If v is the source node of H, then there are no chain components of $\mathcal{E}_{\mathcal{I} \cup \{v\}}(H)$ with only incoming arcs into v in G. Otherwise, if v is not the source node of H, then there is exactly one chain component of $\mathcal{E}_{\mathcal{I} \cup \{v\}}(H)$ with only incoming arcs into v in G. Furthermore, without further interventions, we can decide if such a chain component exist (and find it) in polynomial time.*

Proof. Apply Lemma D.2 with $K = \{v\}$. □

Lemma 4.4. *Algorithm 1 terminates after $\mathcal{O}(\log n)$ phases.*

Proof. In each phase, we are essentially breaking up the graph into small subgraphs using Theorem 2.2 where the size of the chain components decreases by a factor of two.

Note that we do not intervene on *all* the vertices in the clique separator K_H, but only intervene on $V(K_H) \setminus \{v_H\}$. So, we need to argue that partites A and B (with respect to the 1/2-clique separator K_H) are disconnected before we recurse in the next phase. To do so, we use Lemma 4.6: invoking `ResolveDangling` on (Z_{v_H}, w, v_H) ensures that all outgoing edges from v_H will be oriented, so we obtain two disconnected chain component partites A and B.

Since the maximum chain component size initially at most n and is always halved after a phase, Algorithm 1 terminates after $\mathcal{O}(\log n)$ phases. □

Lemma 4.5. *Each phase in Algorithm 1 incurs a cost of $\mathcal{O}(\log(n) \cdot \overline{\nu}_1^{\max}(G^*))$.*

Proof. By the $\zeta^{(1)}$ term of Theorem 4.1, intervening on $V(K_H) \setminus \{v_H\}$ across all chain components $H \in CC(\mathcal{E}_{\mathcal{I}_i}(G^*))$ incurs a cost of at most $2 \cdot \overline{\nu}_1^{\max}(G^*)$. By Lemma 4.6, `ResolveDangling` incurs returns intervention set $\mathcal{I}$ of weight $w(\mathcal{I}) \in \mathcal{O}(\log n \cdot \overline{\nu}_1^{\max}(G^*))$. □

Theorem 3.2. *Fix an essential graph $\mathcal{E}(G^*)$ corresponding to an unknown weighted causal DAG G^*. Algorithm 1 is a deterministic and adaptive algorithm that computes an atomic intervention set $\mathcal{I}$ such that $\mathcal{E}_{\mathcal{I}}(G^*) = G^*$ and $w(\mathcal{I}) \in \mathcal{O}\left(\log^2(n) \cdot \overline{\nu}_1^{\max}(G^*)\right)$. Ignoring the time spent implementing the actual interventions, Algorithm 1 runs in $\mathcal{O}(n \cdot \log^2(n) \cdot d \cdot m)$ time, where d and m are the degeneracy and number of edges of* skel$(\mathcal{E}(G^*))$ *respectively.*

Proof. Direct consequence of combining Lemma 4.4 and Lemma 4.5.

To analyze the running time, let us consider the running time of the subroutines:

- Algorithm 1 has $\mathcal{O}(\log n)$ phases where each phase may execute the `ResolveDangling` subroutine.
- There are at most $t \leq n$ components within the `ResolveDangling` subroutine and the while loops terminates after $\mathcal{O}(\log n)$ iterations.
- Throughout, computing clique separators can be done in $\mathcal{O}(m)$ time (Theorem 2.2, (Gilbert et al., 1984)).
- Throughout, executing Meek rules after performing an intervention can be done in $\mathcal{O}(d \cdot m)$ time (Appendix B, (Wienöbst et al., 2021a)).
- Within the `ResolveDangling` subroutine, finding the chain component Q can be done in $\mathcal{O}(m)$ time.

Thus, Algorithm 1 runs in $\mathcal{O}(n \cdot \log^2(n) \cdot d \cdot m)$ time. Since $d \leq n$ and $m \leq n^2$, the overall running time is polynomial in n. □

F.4. Handling the generalized cost objective

Lemma D.1. *Consider any arbitrary directed clique $G = (V, E)$ and any integer $k \geq 1$. Without loss of generality, $V = \{v_1, \ldots, v_n\}$ and $\pi(v_1) < \ldots < \pi(v_n)$, i.e. v_1 is the source of G. Suppose we arbitrarily partition the vertex set into sets $\mathcal{S} = \{S_1, \ldots, S_{\lceil n/k \rceil}\}$, each of size at most k. Then, the set $S_{source} \in \mathcal{S}$ containing v_1 is the* unique *set in $\mathcal{S}$ that has a vertex without any incoming arcs from the other sets.*

Proof. By definition of a source node, all edges in G will point *away* from v_1. Meanwhile, since G is a clique, every other vertex v_i will have an arc $v_1 \to v_i$. So, S_{source} is the unique set in $\mathcal{S}$ that has a vertex without any incoming arcs from the other sets. □

To prove Lemma D.2, we rely on the next lemma (Lemma F.1) which generalizes Lemma 2.3: the latter is the special case of the former where S is a single vertex. Given a moral DAG, Lemma 2.3 of (Choo & Shiragur, 2023) tells us that intervening on a single vertex w will split up the graph into separate chain components such that all ancestors of w will belong in a single chain component. Lemma F.1 generalizes this fact to the setting of bounded size interventions.

Lemma F.1. *Let $G = (V, E)$ be a moral DAG and π be an arbitrary consistent ordering of G. Intervening on vertex set $S = \{s_1, s_2, \ldots, s_k\} \subseteq V$ orients all edges $u \to v$ with $s_1 \in \texttt{Des}(u) \cap \texttt{Anc}(v)$, where $\pi(s_1) < \pi(s_2) < \ldots < \pi(s_k)$.*

Proof. Note that $u \not\in S$ as $s_1 \in \texttt{Des}(u)$, but v could possibly be a vertex in S.

By Lemma C.11, we know that there are arcs $u \to w$ for all $w \in \texttt{Des}(u) \cap \texttt{Anc}(v)$.

Let

$$s_i = \operatorname*{argmax}_{\substack{z \in S;\\ z \in \texttt{Des}(u) \cap (\texttt{Anc}(v) \cup \{v\})}} \{\pi(z)\}$$

be a vertex in S that lies between u and v, with the largest ordering. The vertex s_i is well-defined because $s_1 \in \texttt{Des}(u) \cap \texttt{Anc}(v) \subseteq \texttt{Des}(u) \cap (\texttt{Anc}(v) \cup \{v\})$.

If $s_i = v$, then $u \to v$ is trivially oriented when we intervene on S because $u \not\in S$. In the rest of the proof, we may assume that $s_i \neq v$, i.e. $s_i \in \texttt{Anc}(v)$. Let

$$w = \operatorname*{argmax}_{\substack{z \in \texttt{Des}(s_i) \cap (\texttt{Anc}(v) \cup \{v\});\\ (s_i \to z) \in E}} \{\pi(z)\}$$

denote the vertex with incoming arc from s_i and ancestral to v, with the largest ordering. The vertex w is well-defined because $s_i \in \texttt{Anc}(v)$ and thus there is a sequence of directed arcs from s_i to v. Note that w could be v and $w \not\in S$ by maximality of s_i.

When we intervene on S, we recover all arc directions incident to s_i, except maybe the arcs internal within S. In particular, we will recover the arcs $u \to s_i$ and $s_i \to w$.

If $w = v$, then Meek rule R2 recovers $u \to w = v$ via $u \to s_i \to w \sim u$.

Otherwise, if $w \neq v$, then let $w = w_0 \to w_1 \to \ldots \to w_\ell = v$ be the sequence of directed arcs from w to v in G. By maximality of w, there is no arc from s_i to any of the vertices $\{w_1, \ldots, w_\ell\}$. So, by repeatedly applying Meek R1, we recover

- $w_0 \to w_1$ via $s_i \to w_0 \sim w_1$
- $w_1 \to w_2$ via $w_0 \to w_1 \sim w_2$
- $\ldots$
- $w_{\ell-1} \to w_\ell$ via $w_{\ell-2} \to w_{\ell-1} \sim w_\ell$

Furthermore, we know that the arcs $u \to w_0, u \to w_1, \ldots u \to w_\ell$ exist due to Lemma C.11. So, by repeatedly applying Meek R2, we recover

- $u \to w_0$ via $u \to s_i \to w_0 \sim u$
- $u \to w_1$ via $u \to w_0 \to w_1 \sim u$
- $\ldots$
- $u \to w_\ell$ via $u \to w_{\ell-1} \to w_\ell \sim u$

That is, the arc $u \to w_\ell = v$ will be oriented. □

Lemma D.2. *Let $\mathcal{E}_{\mathcal{I}}(G)$ be the interventional essential graph of a moral DAG $G = (V, E)$ with respect to intervention set $\mathcal{I} \subseteq 2^V$. Fix any chain component $H \in CC(\mathcal{E}_{\mathcal{I}}(G))$ and let K be an arbitrary clique in H. If K contains the source node of H, then there are no chain components of $\mathcal{E}_{\mathcal{I} \cup \{V(K)\}}(H)$ with only incoming arcs into K in G. Otherwise, if K does not contain the source node of H, then there is exactly one chain component of $\mathcal{E}_{\mathcal{I} \cup \{V(K)\}}(H)$ with only incoming arcs into K in G. Furthermore, without further interventions, we can decide if such a chain component exist (and find it) in polynomial time.*

Proof. Let us denote H_{source} as the source node of H and K_{source} as the source node of K.

Case 1: $K_{source} = H_{source}$

Suppose, for a contradiction, that there was a chain component in $\mathcal{E}_{\mathcal{I} \cup \{V(K)\}}(H)$ with incoming arcs into K in G. Since G is moral, this chain component must have an edge with K_{source}. However, since $K_{source} = H_{source}$, this arc must be *outgoing* from K_{source}. Contradiction.

Case 2: $K_{source} \neq H_{source}$, i.e. $K_{source} \in \texttt{Des}(H_{source})$

Recall that chain components do not have oriented arcs, so H must be moral. Since K is a clique in the chain component H, there was an unoriented directed path from $H_{source} \to u_1 \to \ldots \to u_{last} \to K_{source}$ before intervening on K. Since Meek rules can only orient arcs with an endpoint that is a descendant of vertices in K, we see that the arcs $H_{source} \to u_1 \to \ldots \to u_{last}$ remain unoriented after intervening on K.

Claim 2.1: There exists one such chain component. Let A be the chain component containing H_{source} after intervening on K. From the above discussion, A has an arc into K in G, namely $u_{last} \to K_{source}$. For A to have any incoming arcs from K, A must contain some descendant of K_{source}. However, by Lemma F.1, any arc joining an ancestor and descendant of K_{source} would be oriented, thus ancestors and descendants of K_{source} will belong in different chain components in $\mathcal{E}_{\mathcal{I} \cup \{V(K)\}}(H)$. Thus, A only has incoming arcs into K in G.

Claim 2.2: There does not exist two such chain components. Suppose, for a contradiction, that there is another chain components B in $\mathcal{E}_{\mathcal{I} \cup \{V(K)\}}(H)$ with incoming arcs into K in G. Since G is moral, B must have an edge into K_{source}, say $b \to K_{source}$. Again, since G is moral, there must be an edge between b and u_{last}. Since Meek rules can only orient arcs with an endpoint the arc $b \sim u_{last}$ remains unoriented after intervening on K, so A and B are actually the same chain component. Contradiction.

Running time We can enumerate over all chain components of H and checking each edge at most twice in order to determine whether there is a chain component in $\mathcal{E}_{\mathcal{I} \cup \{V(K)\}}(H)$ with incoming arcs into K in G, and if so find it. □

Lemma 4.9. *Given a set of clique vertices $V(C) \subseteq V$ and integer $k \geq 1$, Algorithm 5 returns a set $S \subseteq 2^{V(C)}$ such that each partite in S has at most k vertices. When $k = 1$, $|S| = |V(C)|$ and each vertex appears exactly once in S. When $k > 1$, $|S| \in \mathcal{O}(\log k \cdot |V(C)|/k)$ and each vertex appears at most $\mathcal{O}(\log k)$ times in S.*

Proof. By construction and Lemma 4.8, each partite in S has at most k vertices.

When $k = 1$, the output $|S| = |V(C)|$ and each vertex appears exactly once in S.

When $k > 1$, the output $|S| \leq \left\lceil \frac{|V(C)|}{k'} \right\rceil \cdot \left\lceil \log_{\left\lceil \frac{|V(C)|}{k'} \right\rceil} |V(C)| \right\rceil$ and each vertex appears $\left\lceil \log_{\left\lceil \frac{|V(C)|}{k'} \right\rceil} |V(C)| \right\rceil$ times in S, where $k' = \min\{k, |V(C)|/2\} > 1$. Since $k' \leq k$, we have $\left\lceil \frac{|V(C)|}{k'} \right\rceil \in \mathcal{O}\left(\frac{|V(C)|}{k} \right)$. So, it remains to bound $\left\lceil \log_{\left\lceil \frac{|V(C)|}{k'} \right\rceil} |V(C)| \right\rceil$.

When $1 < k \leq \frac{|V(C)|}{2}$, we see that $k' = k$. So,

$$\left\lceil \log_{\left\lceil \frac{|V(C)|}{k'} \right\rceil} |V(C)| \right\rceil = \left\lceil \log_{\left\lceil \frac{|V(C)|}{k} \right\rceil} |V(C)| \right\rceil = \left\lceil \frac{\log |V(C)|}{\log \left\lceil \frac{|V(C)|}{k} \right\rceil} \right\rceil \in \mathcal{O}(\log k)$$

For the final asymptotic inclusion, consider the following argument with $\log$ being base 2 and $1 < k \leq x/2$:

$$\begin{aligned}
& \frac{\log x}{\log(x/k)} \leq \log k + 1 \\
\iff \quad & \log x \leq \log k \cdot \log(x/k) + \log(x/k) \\
\iff \quad & \log k \leq \log k \cdot \log(x/k) \\
\iff \quad & 1 \leq \log(x/k) \\
\iff \quad & 2 \leq x/k
\end{aligned}$$

When $k > \frac{|V(C)|}{2}$, we see that $k' = \frac{|V(C)|}{2}$. So,

$$\left\lceil \log_{\left\lceil \frac{|V(C)|}{k'} \right\rceil} |V(C)| \right\rceil = \lceil \log_2 |V(C)| \rceil \leq \lceil \log_2 2k \rceil \in \mathcal{O}(\log k)$$

The claim follows since we always have

$$\left\lceil \log_{\left\lceil \frac{|V(C)|}{k'} \right\rceil} |V(C)| \right\rceil \in \mathcal{O}(\log k)$$

□

Theorem D.3. *Fix an essential graph $\mathcal{E}(G^*)$ corresponding to an unknown weighted causal DAG G^*. Suppose $\mathcal{I}_1^*$ and $\mathcal{I}_k^*$ are an atomic and bounded size intervention sets minimizing Equation (1) such that $\mathcal{E}_{\mathcal{I}_1^*}(G^*) = \mathcal{E}_{\mathcal{I}_k^*}(G^*) = G^*$, $\texttt{cost}(\mathcal{I}_1^*, \alpha, \beta, 1) = \texttt{OPT}_1$, and $\texttt{cost}(\mathcal{I}_k^*, \alpha, \beta, k) = \texttt{OPT}_k$. Then, maximizing over intervention sets $\mathcal{I} \subseteq V$, we have*

$$\texttt{OPT}_1 \geq \max_{\substack{\mathcal{I} \subseteq 2^V \\ \mathcal{I} \text{ atomic}}} \left\{ \sum_{\substack{H \in CC(\mathcal{E}_{\mathcal{I}}(G^*)) \\ |V(H)| \geq 2}} \max\left\{ \zeta^{(3)}_{\mathcal{I},H}, \zeta^{(4)}_{\mathcal{I},H} \right\} \right\} \quad \textit{and} \quad \texttt{OPT}_k \geq \max_{\substack{\mathcal{I} \subseteq 2^V \\ \mathcal{I} \text{ bounded size}}} \left\{ \sum_{\substack{H \in CC(\mathcal{E}_{\mathcal{I}}(G^*)) \\ |V(H)| \geq 2}} \max\left\{ \zeta^{(5)}_{\mathcal{I},H}, \zeta^{(6)}_{\mathcal{I},H} \right\} \right\}$$

where

$$\begin{aligned}
\zeta^{(3)}_{\mathcal{I},H} &= \frac{1}{2} \cdot \max_{\text{clique } C \in H} \left\{ \alpha \cdot \left(w(V(C)) - \max_{v \in V(C)} \{w(v)\} \right) + \beta \cdot |V(C)| \right\} , \\
\zeta^{(4)}_{\mathcal{I},H} &= \frac{1}{2} \cdot \max_{v \in V(H)} \left\{ \min \left\{ \alpha \cdot w(v) + \beta, \sum_{i=1}^{t} \max_{\substack{\text{clique } C_i: \\ V(C_i) \subseteq V_i \cap N_H(v)}} \{ \alpha \cdot w(V(C_i)) + \beta \cdot |V(C_i)| \} \right\} \right\} , \\
\zeta^{(5)}_{\mathcal{I},H} &= \frac{1}{2} \cdot \max_{\text{clique } C \in H} \left\{ \alpha \cdot \left(w(V(C)) - \max_{v \in V(C)} \{w(v)\} \right) + \frac{\beta}{k} \cdot |V(C)| \right\} , \\
\zeta^{(6)}_{\mathcal{I},H} &= \frac{1}{2} \cdot \max_{v \in V(H)} \left\{ \min \left\{ \alpha \cdot w(v) + \frac{\beta}{k}, \sum_{i=1}^{t} \max_{\substack{\text{clique } C_i: \\ V(C_i) \subseteq V_i \cap N_H(v)}} \left\{ \alpha \cdot w(V(C_i)) + \frac{\beta}{k} \cdot |V(C_i)| \right\} \right\} \right\} ,
\end{aligned}$$

and $V_1, \ldots, V_t \subseteq V(H)$ are vertex sets of the $t \geq 1$ disjoint connected components in $H[V(H) \setminus \{v\}]$ in $\zeta^{(4)}_{\mathcal{I},H}$ and $\zeta^{(6)}_{\mathcal{I},H}$.

Proof. The proof is similar to Theorem 4.1 but we specialize the bounds to take into account of Equation (1).

Common argument

Fix an *arbitrary* intervention set $\mathcal{I} \subseteq 2^V$. We will prove the two cases separately by mirroring parts of the proof of Lemma C.3 in how we invoke Lemma C.2.

Consider an arbitrary DAG $\widetilde{G} \in [G^*]$. Let $\mathcal{I}' \subseteq V$ be any atomic verifying set of $\widetilde{G}$ and fix a chain component $H \in CC(\mathcal{E}_{\mathcal{I}}(G^*))$. That is, suppose $\mathcal{E}_{\mathcal{I}'}(G^*) = \widetilde{G}$ and $\mathcal{E}_{\mathcal{I}'}(G^*)[V(H)] = \widetilde{G}[V(H)]$. Then,

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

where the first equality is due to Lemma C.2 and the last equality is because $\mathcal{I}'$ is a verifying set of $\widetilde{G}$. So, $(\mathcal{I}' \setminus \mathcal{I}) \cap V(H)$ is a verifying set for $\widetilde{G}[V(H)]$, and so is $\mathcal{I}' \cap V(H)$. Thus, by minimality of ν_1 and ν_k, we have

$$\nu_1(\widetilde{G}[V(H)]) \leq |\mathcal{I}' \cap V(H)| \qquad \text{and} \qquad \overline{\nu}_1(\widetilde{G}[V(H)]) \leq w(\mathcal{I}' \cap V(H)) \tag{4}$$

for *any* atomic verifying set $\mathcal{I}' \subseteq V$ of $\widetilde{G}$.

Repeating the exact same argument for bounded size verifying sets, we have

$$\nu_k(\widetilde{G}[V(H)]) \leq |\mathcal{I}' \cap V(H)| \qquad \text{and} \qquad \overline{\nu}_k(\widetilde{G}[V(H)]) \leq w(\mathcal{I}' \cap V(H)) \tag{5}$$

for *any* bounded size verifying set $\mathcal{I}' \subseteq 2^V$ of $\widetilde{G}$.

We now independently lower bound via $\zeta^{(3)}_{\mathcal{I},H}$, $\zeta^{(4)}_{\mathcal{I},H}$, $\zeta^{(5)}_{\mathcal{I},H}$, and $\zeta^{(6)}_{\mathcal{I},H}$ by using Lemma 4.2: in any interventional essential graph, we can always pick a consistent ordering by making any unoriented clique the prefix of its chain component.

Case A: Lower bounding via $\zeta^{(3)}_{\mathcal{I},H}$ when $k = 1$:

Fix an arbitrary clique C in H. Suppose the vertices in C are $v_1, \ldots, v_{|C|}$ with $w(v_1) \geq \ldots \geq w(v_{|C|})$. By Lemma 4.2, there exists a valid orientation π of H such that all the vertices in C appear at the start of the ordering. For any such ordering π, the covered edges are $v_{\pi(1)} \to v_{\pi(2)} \to \ldots \to v_{\pi(|C|)}$ and we know that any atomic verifying set must include a minimum vertex cover of these covered edges due to Theorem C.6.

Fix the ordering π where $w(v_{\pi(i)}) = w(v_i)$ and let the DAG $\widetilde{G} \in [G^*]$ correspond to this ordering, i.e. π is in descending weight ordering. Consider the set of *disjoint* alternating covered edges $\pi^{-1}(1) \to \pi^{-1}(2)$, $\pi^{-1}(3) \to \pi^{-1}(4)$, and so on. Amongst these disjoint alternating covered edges, at least one endpoint must be intervened upon, incurring a cost of at least $\sum_{\text{even } i} w(v_i)$. That is, $\nu_1(\widetilde{G}) \geq \sum_{\text{even } i} w(v_i)$. From the proof of Theorem 4.1, we know that

$$\overline{\nu}_1(\widetilde{G}[V(H)]) \geq \frac{1}{2} \cdot \left(w(V(C)) - \max_{v \in V(C)} w(v) \right) .$$

Meanwhile, Theorem C.8 tells us that orienting C requires at least $|V(C)|/2$ atomic interventions even if we allow randomization and adaptivity. So,

$$\nu_1(\widetilde{G}[V(H)]) \geq |V(C)|/2 .$$

Therefore, for *any* atomic verifying set $\mathcal{I}$ of $\widetilde{G}[V(H)]$,

$$\begin{aligned}
\alpha \cdot w(\mathcal{I}) + \beta \cdot |\mathcal{I}| &\geq \alpha \cdot \overline{\nu}_1(\widetilde{G}[V(H)]) + \beta \cdot \nu_1(\widetilde{G}[V(H)]) \\
&\geq \alpha \cdot \left(\frac{1}{2} \cdot \left(w(V(C)) - \max_{v \in V(C)} w(v) \right) \right) + \beta \cdot (|V(C)|/2) \\
&= \frac{1}{2} \cdot \left\{ \alpha \cdot \left(w(V(C)) - \max_{v \in V(C)} w(v) \right) + \beta \cdot |V(C)| \right\} .
\end{aligned}$$

By maximizing amongst the cliques within H, we see that $\alpha \cdot w(\mathcal{I}) + \beta \cdot |\mathcal{I}| \geq \zeta^{(3)}_{\mathcal{I},H}$.

Case B: Lower bounding via $\zeta^{(4)}_{\mathcal{I},H}$ when $k = 1$:

It suffices to prove this for *arbitrary* cliques C_i in each disjoint connected components in $H[V \setminus \{v\}]$, and then taking the maximum. Consider a minimum cost atomic verifying set $\mathcal{I}$ of $\widetilde{G}[V(H)]$.

Case 1: $v \in \mathcal{I}$. Then,

$$\begin{aligned}
\alpha \cdot w(\mathcal{I}) + \beta \cdot |\mathcal{I}| &\geq \alpha \cdot \overline{\nu}_1(\widetilde{G}[V(H)]) + \beta \cdot \nu_1(\widetilde{G}[V(H)]) \\
&\geq \alpha \cdot w(v) + \beta \\
&\geq \frac{1}{2} \cdot \left\{ \alpha \cdot w(v) + \beta, \sum_{i=1}^{t} \alpha \cdot w(V(C_i)) + \beta \cdot |V(C_i)| \right\}
\end{aligned}$$

By maximizing amongst the cliques within each connected component, we see that $\alpha \cdot w(\mathcal{I}) + \beta \cdot |\mathcal{I}| \geq \zeta^{(4)}_{\mathcal{I},H}$.

Case 2: $v \not\in \mathcal{I}$.

By Lemma 4.2, there exists DAGs consistent with $\mathcal{E}(G^*)$ that can be generated by letting v be the first prefix vertex in $\mathcal{E}(G^*)$, followed by vertices in *descending weight ordering* within each clique C_i, across all t components. Let $\widetilde{G}$ be one such DAG and suppose the vertices in clique $C_i = \{u_{i,1}, \ldots, u_{i,|C_i|}\}$ have weights $w(u_{i,1}) \geq \ldots w(u_{i,|C_i|})$ and $\pi(v) < \pi(u_{i,1}) < \ldots < \pi(u_{i,|C_i|})$. We see that the set $\{v \to u_{i,1}, u_{i,1} \to u_{i,2}, \ldots, u_{i,|C_i|-1} \to u_{i,|C_i|}\}_{i=1}^{t}$ are all covered edges of $\widetilde{G}$. By Theorem C.6, *any* verification set must include a minimum vertex cover of these edges. In particular, since $v \not\in \mathcal{I}$, we must have $\{u_{i,1}\}_{i=1}^{t} \subseteq \mathcal{I}$.

Conditioned on not using v, we know, from the proof of Theorem 4.1, that

$$\overline{\nu}_1(\widetilde{G}[V(H)]) \geq \frac{1}{2} \cdot \sum_{i=1}^{t} w(V(C_i)) \;.$$

Meanwhile, Theorem C.8 tells us that orienting all the C_i's require at least $\sum_{i=1}^{t} |V(C_i)|/2$ atomic interventions, even if we allow randomization and adaptivity. So,

$$\nu_1(\widetilde{G}[V(H)]) \geq \frac{1}{2} \cdot \sum_{i=1}^{t} |V(C_i)| \;.$$

Therefore,

$$\begin{aligned}
\alpha \cdot w(\mathcal{I}) + \beta \cdot |\mathcal{I}| &\geq \alpha \cdot \overline{\nu}_1(\widetilde{G}[V(H)]) + \beta \cdot \nu_1(\widetilde{G}[V(H)]) \\
&\geq \alpha \cdot \left(\frac{1}{2} \cdot \sum_{i=1}^{t} w(V(C_i)) \right) + \beta \cdot \left(\frac{1}{2} \cdot \sum_{i=1}^{t} |V(C_i)| \right) \\
&= \frac{1}{2} \cdot \left(\alpha \cdot \sum_{i=1}^{t} w(V(C_i)) + \beta \cdot |V(C_i)| \right)
\end{aligned}$$

By maximizing amongst the cliques within each connected component, we see that $\alpha \cdot w(\mathcal{I}) + \beta \cdot |\mathcal{I}| \geq \zeta^{(4)}_{\mathcal{I},H}$.

Case C: Lower bounding via $\zeta^{(5)}_{\mathcal{I},H}$ when $k > 1$:

We use the exact same proof outline as $\zeta^{(3)}_{\mathcal{I},H}$ while invoking Theorem 4.3. This gives the following inequalities:

$$\overline{\nu}_k(\widetilde{G}[V(H)]) \geq \overline{\nu}_1(\widetilde{G}[V(H)]) \geq \frac{1}{2} \cdot \left(w(V(C)) - \max_{v \in V(C)} w(v) \right) \;.$$

and

$$\nu_k(\widetilde{G}[V(H)]) \geq \left\lceil \frac{\nu_1(\widetilde{G}[V(H)])}{k} \right\rceil \geq \left\lceil \frac{|V(C)|}{2k} \right\rceil \;.$$

Therefore, for *any* bounded size verifying set $\mathcal{I}$ of $\widetilde{G}[V(H)]$,

$$\alpha \cdot w(\mathcal{I}) + \beta \cdot |\mathcal{I}| = \frac{1}{2} \cdot \left\{ \alpha \cdot \left(w(V(C)) - \max_{v \in V(C)} w(v) \right) + \beta \cdot \frac{|V(C)|}{k} \right\} \;.$$

By maximizing amongst the cliques within H, we see that $\alpha \cdot w(\mathcal{I}) + \beta \cdot |\mathcal{I}| \geq \zeta^{(5)}_{\mathcal{I},H}$.

Case D: Lower bounding via $\zeta^{(6)}_{\mathcal{I},H}$ when $k > 1$:

We use the exact same proof outline as $\zeta^{(4)}_{\mathcal{I},H}$ while invoking Theorem 4.3. Let $\mathcal{I}$ be an *arbitrary* bounded size verifying set of $\widetilde{G}[V(H)]$.

Conditioned on using v, we trivially get $\alpha \cdot w(\mathcal{I}) + \beta \cdot |\mathcal{I}| \geq \alpha \cdot w(v) + \beta$ like before. By maximizing amongst the cliques within H, we see that $\alpha \cdot w(\mathcal{I}) + \beta \cdot |\mathcal{I}| \geq \zeta^{(6)}_{\mathcal{I},H}$.

Meanwhile, *conditioned on not using* v, we get the following inequalities:

$$\overline{\nu}_k(\widetilde{G}[V(H)]) \geq \overline{\nu}_1(\widetilde{G}[V(H)]) \geq \frac{1}{2} \cdot \sum_{i=1}^{t} w(V(C_i)) \ .$$

and

$$\nu_k(\widetilde{G}[V(H)]) \geq \left\lceil \frac{\nu_1(\widetilde{G}[V(H)])}{k} \right\rceil \geq \left\lceil \frac{1}{2} \cdot \sum_{i=1}^{t} \frac{|V(C_i)|}{k} \right\rceil \geq \frac{1}{2} \cdot \sum_{i=1}^{t} \frac{|V(C_i)|}{k} \ .$$

Therefore,

$$\alpha \cdot w(\mathcal{I}) + \beta \cdot |\mathcal{I}| = \frac{1}{2} \cdot \left\{ \alpha \cdot \sum_{i=1}^{t} w(V(C_i)) + \beta \cdot \frac{|V(C_i)|}{k} \right\} \ .$$

By maximizing amongst the cliques within H, we see that $\alpha \cdot w(\mathcal{I}) + \beta \cdot |\mathcal{I}| \geq \zeta^{(6)}_{\mathcal{I},H}$.

Putting together

For $k = 1$, recall that $\mathcal{I}_1^* \subseteq V$ is the atomic intervention set optimizing Equation (1) such that $\mathcal{E}_{\mathcal{I}_1^*}(G^*) = G^*$. So,

$$\begin{aligned}
\texttt{OPT}_1 &= \alpha \cdot w(\mathcal{I}_1^*) + \beta \cdot |\mathcal{I}_1^*| \\
&\overset{(*)}{\geq} \sum_{\substack{H \in CC(\mathcal{E}_{\mathcal{I}}(G^*)) \\ |V(H)| \geq 2}} \alpha \cdot w(\mathcal{I}_1^* \cap V(H)) + \beta \cdot |\mathcal{I}_1^* \cap V(H)| \\
&\geq \sum_{\substack{H \in CC(\mathcal{E}_{\mathcal{I}}(G^*)) \\ |V(H)| \geq 2}} \alpha \cdot \overline{\nu}_1(\widetilde{G}[V(H)]) + \beta \cdot \nu_1(\widetilde{G}[V(H)]) \\
&\geq \sum_{\substack{H \in CC(\mathcal{E}_{\mathcal{I}}(G^*)) \\ |V(H)| \geq 2}} \max\{\zeta^{(3)}_{\mathcal{I},H}, \zeta^{(4)}_{\mathcal{I},H}\}
\end{aligned}$$

where the inequality $(*)$ is because some edges may have already been oriented by $\mathcal{I}$ and the last two inequalities follow from arguments in cases A and B. Finally, the claim follows by taking the maximum over all possible atomic interventions $\mathcal{I} \subseteq V$.

For $k > 1$, recall that $\mathcal{I}_k^* \subseteq 2^V$ is the bounded size intervention set optimizing Equation (1) such that $\mathcal{E}_{\mathcal{I}_k^*}(G^*) = G^*$. So,

$$\begin{aligned}
\texttt{OPT}_k &= \sum_{S \in \mathcal{I}_k^*} \alpha \cdot w(S) + \beta \cdot |S| \\
&\overset{(*)}{\geq} \sum_{\substack{H \in CC(\mathcal{E}_{\mathcal{I}}(G^*)) \\ |V(H)| \geq 2}} \sum_{S \in \mathcal{I}_k^*} \alpha \cdot w(S \cap V(H)) + \beta \cdot |S \cap V(H)| \\
&\geq \sum_{\substack{H \in CC(\mathcal{E}_{\mathcal{I}}(G^*)) \\ |V(H)| \geq 2}} \alpha \cdot \overline{\nu}_k(\widetilde{G}[V(H)]) + \beta \cdot \nu_k(\widetilde{G}[V(H)]) \\
&\geq \sum_{\substack{H \in CC(\mathcal{E}_{\mathcal{I}}(G^*)) \\ |V(H)| \geq 2}} \max\{\zeta^{(5)}_{\mathcal{I},H}, \zeta^{(6)}_{\mathcal{I},H}\}
\end{aligned}$$

where the inequality $(*)$ is because some edges may have already been oriented by $\mathcal{I}$ and the last two inequalities follow from arguments in cases C and D. Finally, the claim follows by taking the maximum over all possible bounded size interventions $\mathcal{I} \subseteq 2^V$. □

Lemma D.4. *Fix an interventional essential graph $\mathcal{E}_{\mathcal{I}'}(G^*)$ corresponding to an unknown weighted causal moral DAG G^* and some intervention $\mathcal{I}' \subseteq 2^V$. Suppose $\mathcal{I}_1^*$ and $\mathcal{I}_k^*$ are atomic and bounded size intervention sets minimizing Equation (1) such that $\mathcal{E}_{\mathcal{I}_1^*}(G^*) = \mathcal{E}_{\mathcal{I}_k^*}(G^*) = G^*$, $\texttt{cost}(\mathcal{I}_1^*, \alpha, \beta, 1) = \texttt{OPT}_1$, and $\texttt{cost}(\mathcal{I}_k^*, \alpha, \beta, k) = \texttt{OPT}_k$. Let H be a chain component of $\mathcal{E}_{\mathcal{I}'}(G^*)$ containing a vertex $v \in V(H)$. Then,*

- *When $k = 1$, Algorithm 4 returns an atomic intervention set $\mathcal{I}$ such that connected components in $H[V(H) \setminus \{v\}]$ are mutually disjoint in $\mathcal{E}_{\mathcal{I}}(H)$ and $\texttt{cost}(\mathcal{I}, \alpha, \beta, 1) \in \mathcal{O}(\log n \cdot \texttt{OPT}_1)$.*
- *When $k > 1$, Algorithm 4 returns a bounded size intervention set $\mathcal{I}$ such that connected components in $H[V(H) \setminus \{v\}]$ are mutually disjoint in $\mathcal{E}_{\mathcal{I}}(H)$ and $\texttt{cost}(\mathcal{I}, \alpha, \beta, k) \in \mathcal{O}((\log n + \log k) \cdot \texttt{OPT}_k)$.*

Proof. The proof strategy exactly follows Lemma 4.6 except we have to account for the subroutine call to `CliqueIntervention` via Lemma 4.9.

Since the underlying graph is a moral DAG, intervening on v or $\text{argmin}_{u \in (\cup_{i=1}^{t} V(H_1)) \cap N_H(v)} \pi(u)$ ensures that the partites will indeed become separated. Suppose $u_i = \text{argmin}_{u \in V(H_i) \cap N_H(v)} \pi(u)$. If $\pi(v_H) > \min_{i \in \{1,\ldots,t\}} \pi(u_i)$, then intervening on $u_1, \ldots, u_t$ will disconnect[14] H_i's from each other[15]. Otherwise, if $\pi(v_H) < \min_{i \in \{1,\ldots,t\}} \pi(u_i)$, Lemma 2.3 tells us that intervening on u_i will orient all $v_H \to z$ arcs for $z \in H_i$.

The if-case of `ResolveDanglingGeneralized` directly intervenes on v while the else-case of `ResolveDanglingGeneralized` repeatedly recurses on a connected subgraph of $H_i[V']$, towards the source $\text{argmin}_{u \in (\cup_{i=1}^{t} V(H_1)) \cap N_H(v)} \pi(u)$. Since the size of V' is at least halved in each iteration of the while-loop, it can have at most $\mathcal{O}(\log n)$ iterations.

Note that, in each iteration (out of $O(\log n)$ iterations) of `ResolveDanglingGeneralized` except the last one, we partition the clique seperators into sets of size at most k and intervene on them. Suppose $S \subseteq \mathcal{I}$ is the intervention set output of that iteration, by Theorem D.3 the cost of this step, that is $\texttt{cost}(S, \alpha, \beta, k) \in \mathcal{O}(\texttt{OPT}_k)$ for all $k \geq 1$. In the last step of our prcoedure `ResolveDanglingGeneralized`, we invoke `CliqueIntervention` and in the remainder of the proof we bound the cost incurred by this subroutine.

Accounting for `CliqueIntervention` subroutine calls

Suppose $S \subseteq \mathcal{I}$ is the intervention set output of `CliqueIntervention` on some clique C in the last step.

When $k = 1$, we know from Lemma 4.9 that $|S| = |V(C)|$ and each vertex appears exactly once in S. By $\zeta^{(4)}$ term of Theorem D.3, $\texttt{cost}(S, \alpha, \beta, 1) \in \mathcal{O}(\texttt{OPT}_1)$. So, across all $\mathcal{O}(\log n)$ iterations, $\texttt{cost}(\mathcal{I}, \alpha, \beta, 1) \in \mathcal{O}(\log n \cdot \texttt{OPT}_1)$.

When $k > 1$, we know from Lemma 4.9 that $|S| \in \mathcal{O}(\log k \cdot |V(C)|/k)$ and each vertex appears at most $\mathcal{O}(\log k)$ in S. By $\zeta^{(6)}$ term of Theorem D.3, $\texttt{cost}(S, \alpha, \beta, k) \in \mathcal{O}(\log k \cdot \texttt{OPT}_k)$, where the $\mathcal{O}(\log k)$ multiplicity of each vertex occurrence increases the α term while the $\mathcal{O}(\log k)$ multiplicative factor size overhead increases the β term. As we invoke, `CliqueIntervention` only in the last step and as we incur only a cost of $\mathcal{O}(\texttt{OPT}_k)$ in all the remaining steps, our total cost across all $\mathcal{O}(\log n)$ iterations is, $\texttt{cost}(\mathcal{I}, \alpha, \beta, k) \in \mathcal{O}(\log n \cdot \texttt{OPT}_k + \log k \cdot \texttt{OPT}_k)$. We conclude the proof. □

Lemma D.5. `ALG-GENERALIZED` *(Algorithm 3) terminates after $\mathcal{O}(\log n)$ phases.*

Proof. The proof exactly follows Lemma 4.4 except we use `ResolveDanglingGeneralized` instead of `ResolveDangling` to ensure that the partites A and B (with respect to 1/2-clique separator K_H) are separated before we recurse in the next phase. For completeness, we repeat the entire argument below.

In each phase, we are essentially breaking up the graph into small subgraphs using Theorem 2.2 where the size of the chain components decrease by a factor of two.

Note that we do not intervene on *all* the vertices in the clique separator K_H, but only intervene on $V(K_H) \setminus \{v_H\}$ on line 8, we need to argue that partites A and B (with respect to the 1/2-clique separator K_H) are separated before we recurse in

[14] Every path between H_i and H_j, for $i \neq j$ will involve an oriented arc. Such arcs will be removed when considering chain components, disconnecting the path.

[15] Without loss of generality, suppose $\pi(u_1) = \min_{i \in \{1,\ldots,t\}} \pi(u_i)$. Orienting the arc $u_1 \to v_H$ triggers Meek rule R1 to orient all $v \to z$ arcs for $z \notin H_1$, thus disconnecting H_i's from each other.

the next phase. Lemma D.4 ensures that `ResolveDanglingGeneralized` on (Z_{v_H}, w, v_H) separates any connected components that may be "dangling" from v_H after intervening on $V(K_H) \setminus \{v_H\}$.

Since the maximum chain component size initially at most n and is always halved after a phase, Algorithm 3 terminates after $\mathcal{O}(\log n)$ phases. □

Lemma D.6. *Suppose $\mathcal{I}_1^*$ and $\mathcal{I}_k^*$ are an atomic and bounded size verifying sets respectively for G^* that minimizes Equation (1) with $\texttt{cost}(\mathcal{I}_1^*) = \texttt{OPT}_1$ and $\texttt{cost}(\mathcal{I}_k^*) = \texttt{OPT}_k$. Each phase in `ALG-GENERALIZED` (Algorithm 3) incurs a cost of $\mathcal{O}(\log n \cdot \texttt{OPT}_1)$ when $k = 1$ and $\mathcal{O}\left((\log n + \log k) \cdot \texttt{OPT}_k\right)$ when $k > 1$.*

Proof. Fix an arbitrary phase i with an intermediate interventional essential graph $\mathcal{E}_{\mathcal{I}}(G^*)$. Suppose that $\mathcal{J}_i \subseteq 2^V$ is the intervention set computed by `ALG-GENERALIZED` in phase i. By construction, $\mathcal{J}_i$ is made up by at most two calls from `CliqueIntervention` – one from line 9 and one within `ResolveDanglingGeneralized`.

Case $k = 1$: By the $\zeta^{(3)}$ term of Theorem D.3, a function call to `CliqueIntervention` from line 9 incurs a cost of $\mathcal{O}(\texttt{OPT}_1)$. By Lemma D.4, `ResolveDanglingGeneralized` incurs a cost of $\mathcal{O}(\log n \cdot \texttt{OPT}_1)$.

Case $k > 1$: By the $\zeta^{(5)}$ term of Theorem D.3, a function call to `CliqueIntervention` from line 9 incurs a cost of $\mathcal{O}(\texttt{OPT}_k)$. By Lemma D.4, `ResolveDanglingGeneralized` incurs a cost of $\mathcal{O}((\log n + \log k) \cdot \texttt{OPT}_k)$. □

Theorem 3.3. *Fix an essential graph $\mathcal{E}(G^*)$ corresponding to an unknown weighted causal DAG G^*. Suppose $\mathcal{I}_1^*$ and $\mathcal{I}_k^*$ are an atomic and bounded size verifying sets minimizing Equation (1) such that $\texttt{cost}(\mathcal{I}_1^*, \alpha, \beta, 1) = \texttt{OPT}_1$ and $\texttt{cost}(\mathcal{I}_k^*, \alpha, \beta, k) = \texttt{OPT}_k$. Then, Algorithm 3 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*
1. $\texttt{cost}(\mathcal{I}, \alpha, \beta, 1) \in \mathcal{O}\left(\log^2 n \cdot \texttt{OPT}_1\right)$
2. $\texttt{cost}(\mathcal{I}, \alpha, \beta, k) \in \mathcal{O}\left(\log n \cdot (\log n + \log k) \cdot \texttt{OPT}_k\right)$.

Proof. Direct consequence of combining Lemma D.5 and Lemma D.6.

The algorithm runs in polynomial time because the following it uses a polynomial number of phases and each phase can be computed in polynomial time:

- Computation of 1/2-clique separators run in polynomial time (Theorem 2.2)
- Enumerating all maximal cliques in chordal graph can be done in polynomial time[16]
- Labelling scheme computation of Lemma 4.8 can be computed in polynomial time
- Applying Meek rules till convergence can be made to run in polynomial time (Wienöbst et al., 2021a)

□

G. Experiments

In this section, we provide more details about our experiments.

All our experiments are conducted on an Ubuntu server with two AMD EPYC 7532 CPU and 256GB DDR4 RAM. Source code implementation and experimental scripts are available at `https://github.com/cxjdavin/new-metrics-and-search-algorithms-for-weighted-causal-DAGs`.

We base our evaluation on the experimental framework of (Choo et al., 2022)[17], which in turn is based on (Squires et al., 2020)[18]. In the following, we replicate some of the experimental setup details from Appendix H of (Choo et al., 2022).

[16] e.g. see `https://en.wikipedia.org/wiki/Chordal_graph`
[17] Available at `https://github.com/cxjdavin/subset-verification-and-search-algorithms-for-causal-DAGs`
[18] Available at `https://github.com/csquires/dct-policy`

G.1. Synthetic graph classes

The synthetic graphs are random connected moral DAGs.

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 n^{th} 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 (Koller & Friedman, 2009) 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.

G.2. Weights and generalized cost parameters

We ran experiments for $\alpha \in \{0, 1\}$ and $\beta = 1$ on two different types of weight classes for a graph on n vertices:

Type 1 The weight of each vertex is independently sampled from an exponential distribution $\exp(n^2)$ with parameter n^2. This is to simulate the setting where there is a spread in the costs of the vertices.

Type 2 A randomly chosen $p = 0.1$ fraction of vertices are assigned weight n^2 while the others are assigned weight 1. This is to simulate the setting where there are a few randomly chosen high cost vertices.

So, we have 4 sets of experiments in total, where each set follows the 5 experiments performed in (Choo et al., 2022).

Experiment 1 Graph class 1 with $n \in \{10, 15, 20, 25\}$ and density $\rho = 0.1$.

Experiment 2 Graph class 1 with $n \in \{8, 10, 12, 14\}$ and density $\rho = 0.1$.

Experiment 3 Graph class 2 with $n \in \{100, 200, 300, 400, 500\}$ and $(\text{degree}, e_{\min}, e_{\max}) = (4, 2, 5)$.

Experiment 4 Graph class 1 with $n \in \{10, 15, 20, 25\}$ and density $\rho = 0.1$.

Experiment 5 Graph class 2 with $n \in \{100, 200, 300, 400, 500\}$ and $(\text{degree}, e_{\min}, e_{\max}) = (40, 20, 50)$.

G.3. Algorithms benchmarked

The following algorithms perform *atomic interventions*. Our algorithm `weighted_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 (Squires et al., 2020)

`coloring`: `Coloring` of (Shanmugam et al., 2015)

`opt_single`: `OptSingle` of (Hauser & Bühlmann, 2014)

`greedy_minmax`: `MinmaxMEC` of (He & Geng, 2008)

`greedy_entropy`: `MinmaxEntropy` of (He & Geng, 2008)

`separator`: Algorithm of (Choo et al., 2022); Takes in a parameter k for bounded-size interventions.

`weighted_separator`: Our Algorithm 3; Takes in a parameter k for bounded-size interventions.

G.4. Experimental results

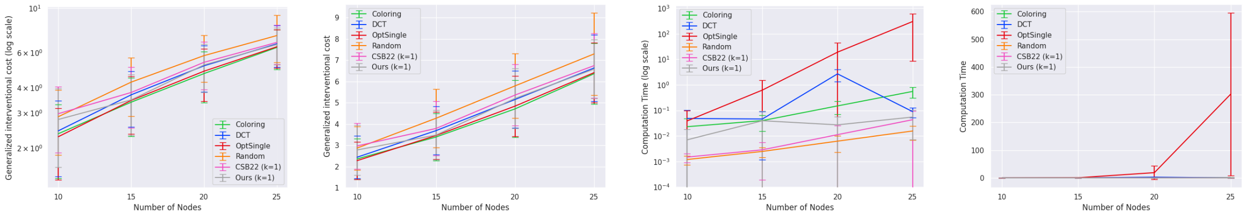
In all experiments, `ALG-GENERALIZED` has a similar run time[19]. When $\alpha = 0$ and $\beta = 1$, the generalized cost function is simply the number of interventions used, and `ALG-GENERALIZED` incurs a similar cost to the existing state-of-the-art

[19] `ALG-GENERALIZED` is faster than all benchmarked algorithms except (Choo et al., 2022). This is expected as both use an approach based on 1/2-clique separators but `ALG-GENERALIZED` has additional computational overhead to handle dangling components.

algorithms. Meanwhile, when $\alpha = 1$ and $\beta = 1$, the generalized cost function is affected by the vertex weights, and `ALG-GENERALIZED` incurs noticeably less generalized cost.

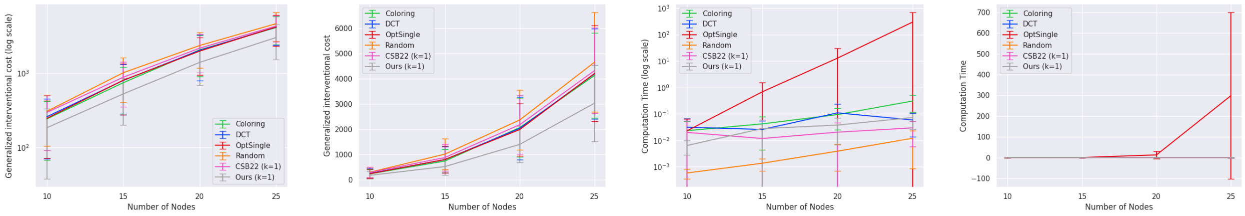
We also tested the bounded size implementation for $k \in \{1, 3, 5\}$ and observe that the lines "flip", for both weight types. When $(\alpha, \beta) = (0, 1)$, $k = 1$ is worst and $k = 3$ is best. When $(\alpha, \beta) = (1, 1)$, $k = 3$ is worst and $k = 1$ is best. This matches what we expect from our theoretical analyses.

G.4.1. EXPERIMENT 1



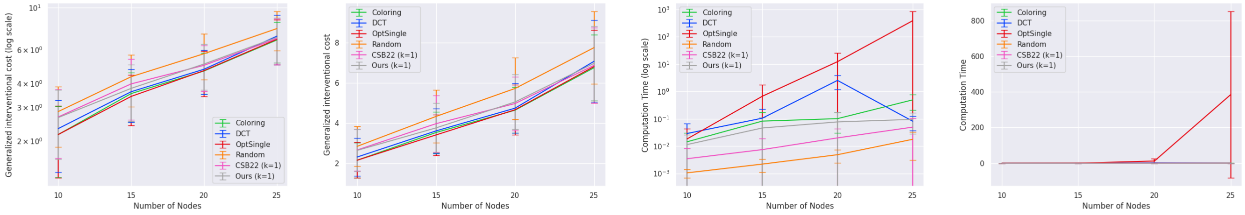
(a) Generalized cost (log scale) (b) Generalized cost (c) Time taken, in secs (log scale) (d) Time taken, in secs

Figure 6. Experiment 1, Type 1, $\alpha = 0$, $\beta = 1$



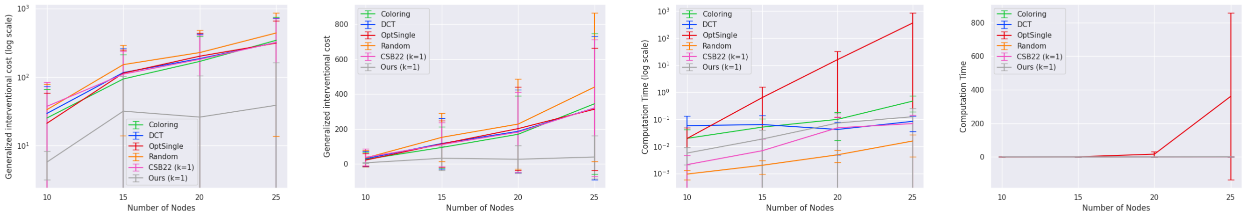
(a) Generalized cost (log scale) (b) Generalized cost (c) Time taken, in secs (log scale) (d) Time taken, in secs

Figure 7. Experiment 1, Type 1, $\alpha = 1$, $\beta = 1$



(a) Generalized cost (log scale) (b) Generalized cost (c) Time taken, in secs (log scale) (d) Time taken, in secs

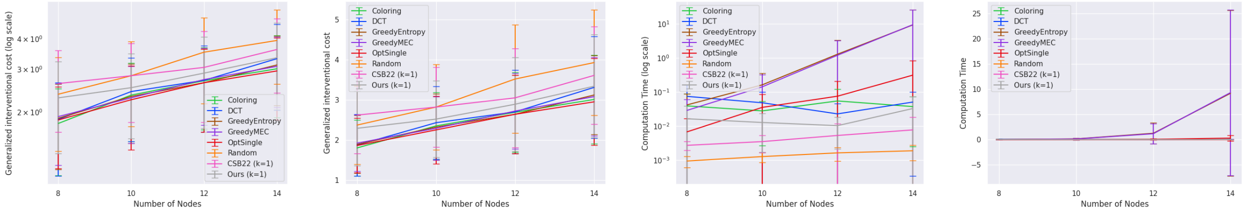
Figure 8. Experiment 1, Type 2, $\alpha = 0$, $\beta = 1$



(a) Generalized cost (log scale) (b) Generalized cost (c) Time taken, in secs (log scale) (d) Time taken, in secs

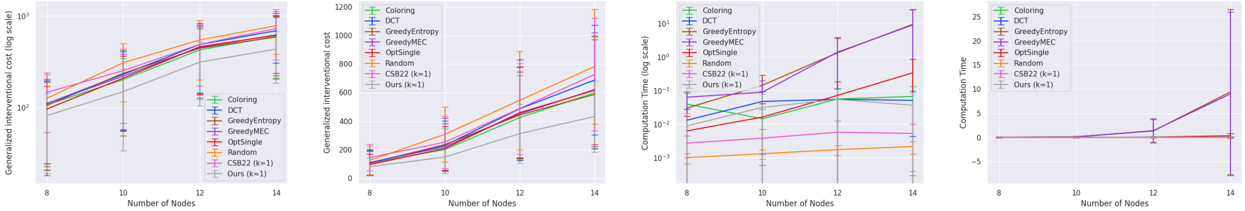
Figure 9. Experiment 1, Type 2, $\alpha = 1$, $\beta = 1$

G.4.2. Experiment 2



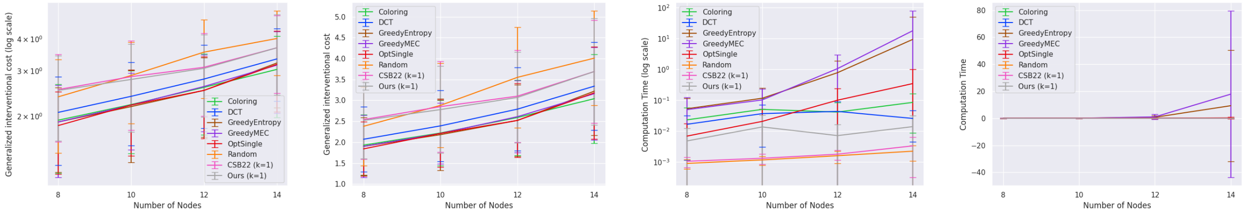
(a) Generalized cost (log scale) (b) Generalized cost (c) Time taken, in secs (log scale) (d) Time taken, in secs

Figure 10. Experiment 1, Type 1, $\alpha = 0$, $\beta = 1$



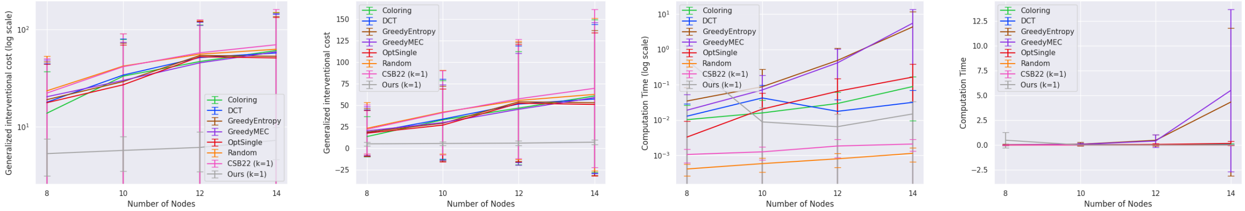
(a) Generalized cost (log scale) (b) Generalized cost (c) Time taken, in secs (log scale) (d) Time taken, in secs

Figure 11. Experiment 1, Type 1, $\alpha = 1$, $\beta = 1$



(a) Generalized cost (log scale) (b) Generalized cost (c) Time taken, in secs (log scale) (d) Time taken, in secs

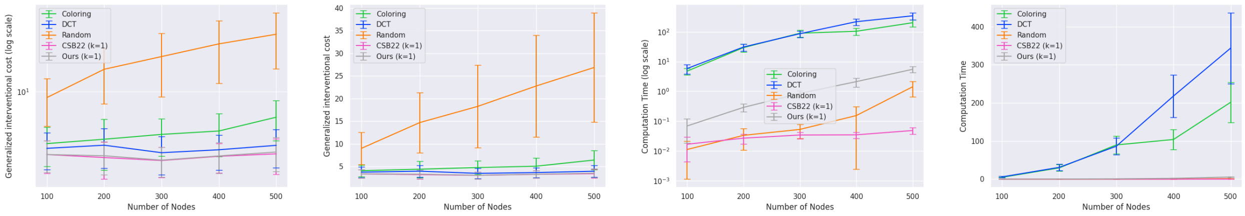
Figure 12. Experiment 1, Type 2, $\alpha = 0$, $\beta = 1$



(a) Generalized cost (log scale) (b) Generalized cost (c) Time taken, in secs (log scale) (d) Time taken, in secs

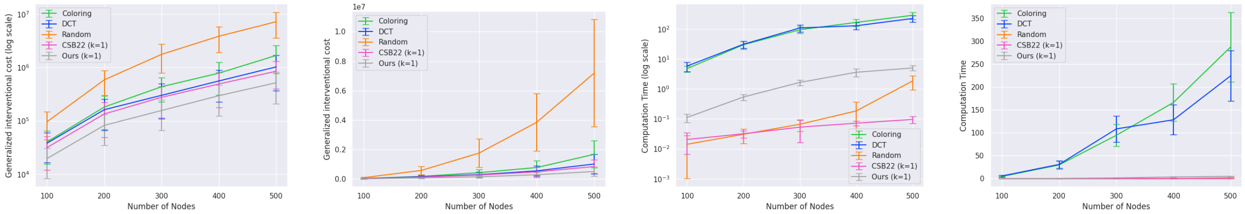
Figure 13. Experiment 1, Type 2, $\alpha = 1$, $\beta = 1$

G.4.3. Experiment 3



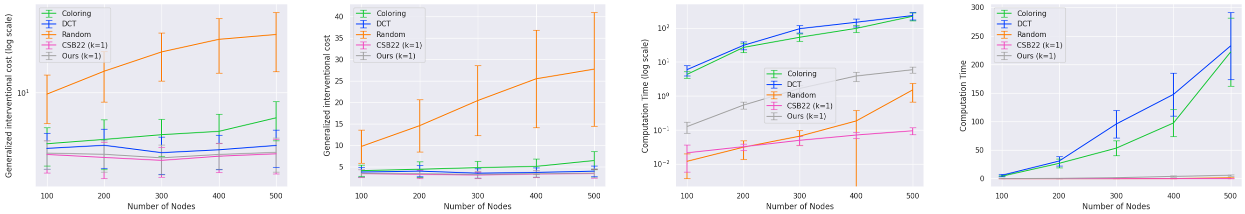
(a) Generalized cost (log scale) (b) Generalized cost (c) Time taken, in secs (log scale) (d) Time taken, in secs

Figure 14. Experiment 1, Type 1, $\alpha = 0$, $\beta = 1$



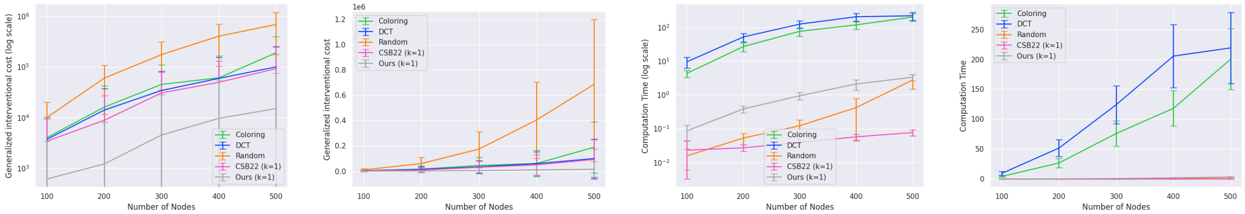
(a) Generalized cost (log scale) (b) Generalized cost (c) Time taken, in secs (log scale) (d) Time taken, in secs

Figure 15. Experiment 1, Type 1, $\alpha = 1$, $\beta = 1$



(a) Generalized cost (log scale) (b) Generalized cost (c) Time taken, in secs (log scale) (d) Time taken, in secs

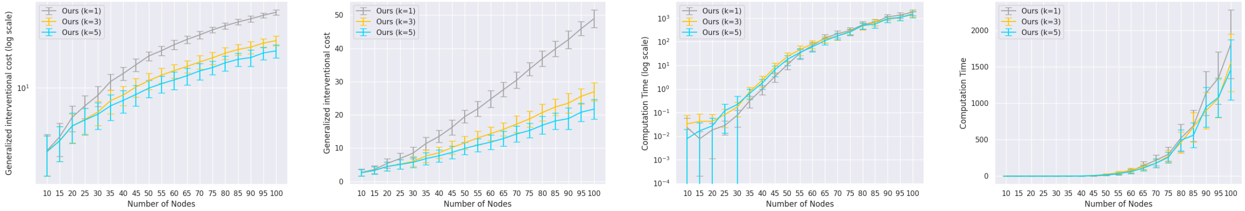
Figure 16. Experiment 1, Type 2, $\alpha = 0$, $\beta = 1$



(a) Generalized cost (log scale) (b) Generalized cost (c) Time taken, in secs (log scale) (d) Time taken, in secs

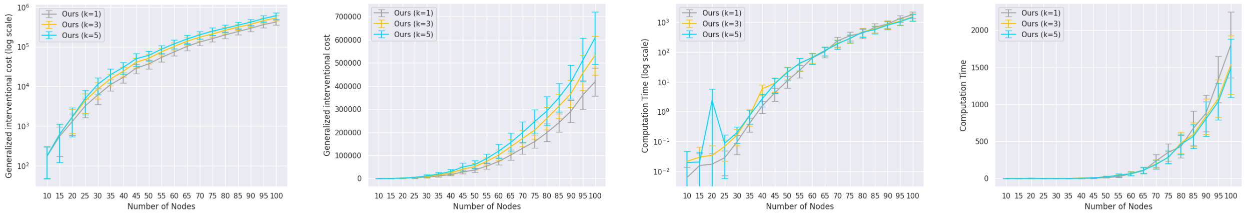
Figure 17. Experiment 1, Type 2, $\alpha = 1$, $\beta = 1$

G.4.4. Experiment 4



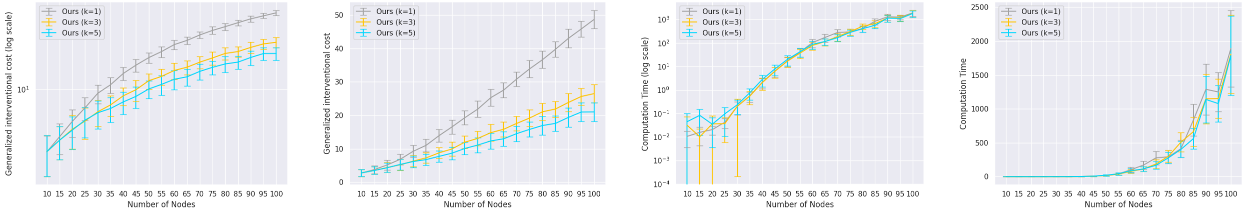
(a) Generalized cost (log scale) (b) Generalized cost (c) Time taken, in secs (log scale) (d) Time taken, in secs

Figure 18. Experiment 1, Type 1, $\alpha = 0$, $\beta = 1$



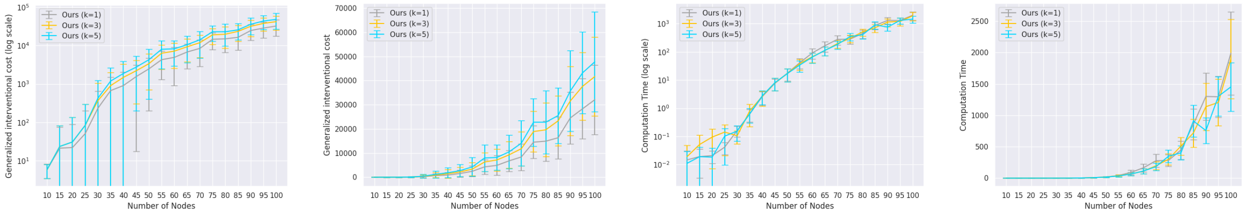
(a) Generalized cost (log scale) (b) Generalized cost (c) Time taken, in secs (log scale) (d) Time taken, in secs

Figure 19. Experiment 1, Type 1, $\alpha = 1$, $\beta = 1$



(a) Generalized cost (log scale) (b) Generalized cost (c) Time taken, in secs (log scale) (d) Time taken, in secs

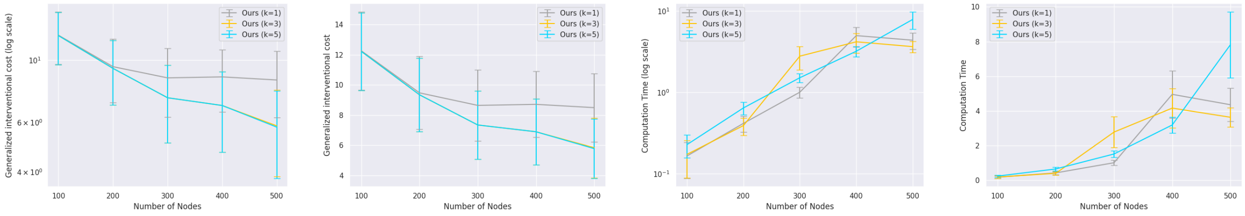
Figure 20. Experiment 1, Type 2, $\alpha = 0$, $\beta = 1$



(a) Generalized cost (log scale) (b) Generalized cost (c) Time taken, in secs (log scale) (d) Time taken, in secs

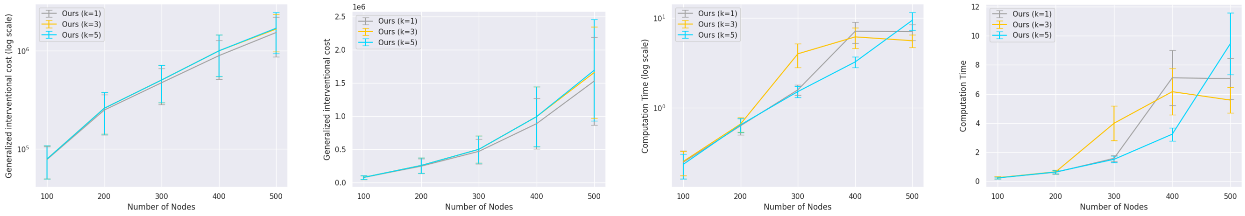
Figure 21. Experiment 1, Type 2, $\alpha = 1$, $\beta = 1$

G.4.5. Experiment 5



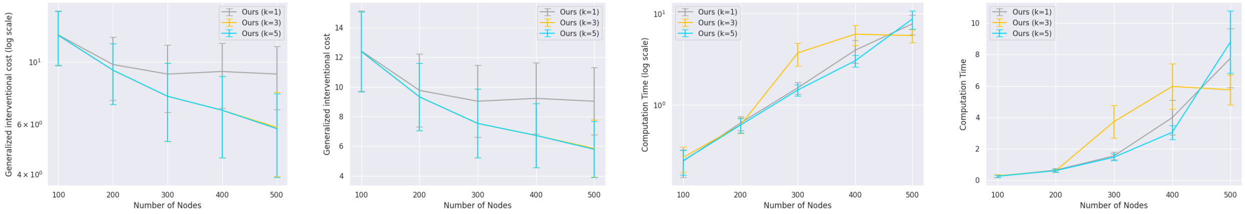
(a) Generalized cost (log scale) (b) Generalized cost (c) Time taken, in secs (log scale) (d) Time taken, in secs

Figure 22. Experiment 1, Type 1, $\alpha = 0$, $\beta = 1$



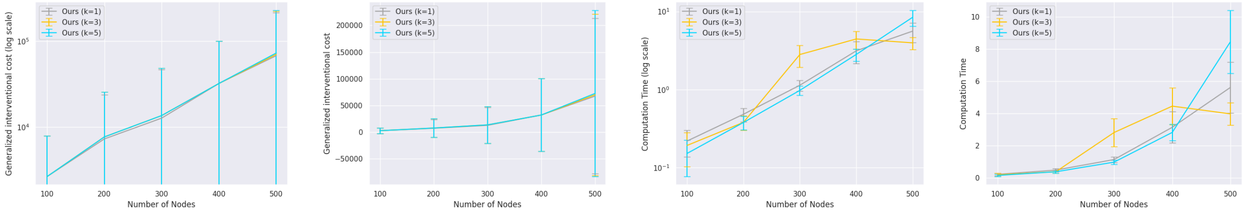
(a) Generalized cost (log scale) (b) Generalized cost (c) Time taken, in secs (log scale) (d) Time taken, in secs

Figure 23. Experiment 1, Type 1, $\alpha = 1$, $\beta = 1$



(a) Generalized cost (log scale) (b) Generalized cost (c) Time taken, in secs (log scale) (d) Time taken, in secs

Figure 24. Experiment 1, Type 2, $\alpha = 0$, $\beta = 1$



(a) Generalized cost (log scale) (b) Generalized cost (c) Time taken, in secs (log scale) (d) Time taken, in secs

Figure 25. Experiment 1, Type 2, $\alpha = 1$, $\beta = 1$