Adaptivity Complexity for Causal Graph Discovery

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

[1] School of Computing, National University of Singapore
[2] Broad Institute of MIT and Harvard

Abstract

Causal discovery from interventional data is an important problem, where the task is to design an interventional strategy that learns the hidden ground truth causal graph $G(V, E)$ on $|V| = n$ nodes while minimizing the number of performed interventions. Most prior interventional strategies broadly fall into two categories: non-adaptive and adaptive. Non-adaptive strategies decide on a single fixed set of interventions to be performed while adaptive strategies can decide on which nodes to intervene on sequentially based on past interventions. While adaptive algorithms may use exponentially fewer interventions than their non-adaptive counterparts, there are practical concerns that constrain the amount of adaptivity allowed. Motivated by this trade-off, we study the problem of r-adaptivity, where the algorithm designer recovers the causal graph under a total of r sequential rounds whilst trying to minimize the total number of interventions. For this problem, we provide a r-adaptive algorithm that achieves $O(\min\{r, \log n\} \cdot n^{1/\min\{r,\log n\}})$ approximation with respect to the verification number, a well-known lower bound for adaptive algorithms. Furthermore, for every r, we show that our approximation is tight. Our definition of r-adaptivity interpolates nicely between the non-adaptive ($r = 1$) and fully adaptive ($r = n$) settings where our approximation simplifies to $O(n)$ and $O(\log n)$ respectively, matching the best-known approximation guarantees for both extremes. Our results also extend naturally to the bounded size interventions.

1 INTRODUCTION

Learning causal relationships from data is a fundamental problem that has found applications across various scientific disciplines, including biology [King et al., 2004, Cho et al., 2016, Tian, 2016, Sverchkov and Craven, 2017, Rotmensch et al., 2017, Pingault et al., 2018, de Campos et al., 2019], epidemiology, philosophy [Reichenbach, 1956, Woodward, 2005, Eberhardt and Scheines, 2007], and econometrics [Hoover, 1990, Rubin and Waterman, 2006]. Directed acyclic graphs (DAGs) are a popular choice to model causal relationships and 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 task. Here, we focus our attention on causal discovery using interventions. As interventions are often costly, our objective is to minimize interventions during the recovery process.

There is a rich literature on causal discovery from interventional data, and causal discovery algorithms 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 and Shiragur, 2023] versus non-adaptive [Eberhardt et al., 2005, 2006, Eberhardt, 2010, Hu et al., 2014]. 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 the interventional cost of an optimal adaptive algorithm can even be exponentially better than any non-adaptive algorithms in some cases[1]. However, the sequential nature of adaptive algorithms fundamentally hinders parallelization [Dean et al., 2008, Balkanski and Singer, 2018] and may even be practically infeasible, e.g. hard constraints like timed deadlines

*Equal contribution

[1] On tree causal graphs, $\Omega(n)$ non-adaptive interventions are needed in the worst case while $\mathcal{O}(\log n)$ adaptive ones suffice.

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

may restrict how many rounds of adaptivity one can afford to do whilst minimizing the number of interventions.

The study of adaptivity is natural and has been studied across a wide spectrum of problems in computer science and statistics including parallel algorithms [Valiant, 1975, Cole, 1988, Braverman et al., 2016], communication complexity [Papadimitriou and Sipser, 1982, Duris et al., 1984, Nisan and Widgerson, 1991, Alon et al., 2015], multi-armed bandits [Agarwal et al., 2017], sparse recovery problems [Malioutov et al., 2008, Haupt et al., 2009, Indyk et al., 2011], knapsack [Dean et al., 2008], submodular optimization [Balkanski and Singer, 2018], stochastic set covering [Goemans and Vondrák, 2006], stochastic probing [Gupta et al., 2016, 2017], local search algorithms [Brânzei and Li, 2022], and many more [Scarlett, 2018, Canonne, Clément L and Gur, Tom, 2018, Raskhodnikova and Smith, 2006]. Typically, one can obtain a lower objective cost if more rounds of adaptivity are allowed and researchers have been trying to characterize the trade-off between cost and adaptivity in various problems.

Beyond being a fundamental question in theory, understanding adaptivity is also practically motivated in the problem of causal discovery. Consider a scientific institute where lab technicians carry out experiments suggested by their scientist colleagues to make progress towards a common research goal. A round of interaction involves the experimentalists discussing with the scientists on what experiment(s) should be performed next, where each round of interaction may incur undesirable coordination overheads. Since the institute has enough resources to run multiple (but bounded) number of experiments simultaneously, a batch of experiments is typically proposed in a single round of interaction. In this work, we formally model the interactions between the two parties through the framework of adaptivity in hopes that our theoretical contributions will yield practical insights, e.g. give guidance on how the two parties should interact. Let us now formally define our problem setup.

Problem setup Given $r \geq 1$ adaptivity rounds and a partially oriented observational essential graph of an underlying causal graph $G^* = (V, E)$, adaptively design[2] intervention sets $\mathcal{I}_1, \ldots, \mathcal{I}_r \subseteq 2^V$ such that all arc orientations in G^* will be recovered after intervening on them while minimizing the total number of interventions performed.

Formally, we require $\mathcal{E}_{\mathcal{I}_1 \cup \ldots \cup \mathcal{I}_r}(G^*) = G^*$ while minimizing $\cup_{i=1}^{r} |\mathcal{I}_i|$, where $\mathcal{E}_{\mathcal{I}}(G^*)$ denotes the partially oriented interventional essential graph after performing interventions $\mathcal{I}$. Furthermore, each intervention $I \in \mathcal{I}_i$ in any intervention set $\mathcal{I}_i$ has size $|I| \leq k$. Atomic interventions when $k = 1$ and bounded size interventions when $k > 1$.

[2]The decision of $\mathcal{I}_i$ may depend on outcomes after recovering arc orientations from intervening on $\mathcal{I}_1, \ldots, \mathcal{I}_{i-1}$.

1.1 CONTRIBUTIONS

Under standard assumptions of causal sufficiency, faithfulness and infinite sample regime[3], we provide a r-adaptive search algorithm that recovers the ground truth causal graph by performing at most $\mathcal{O}(\min\{r, \log n\} \cdot n^{1/\min\{r, \log n\}} \cdot \nu_1(G^*))$ atomic interventions, where $\nu_1(G^*)$ is the atomic verification number of G^* [Squires et al., 2020, Choo et al., 2022, Choo and Shiragur, 2023][4], a natural lower bound for the adaptive algorithms. See Definition 6 for definition of $\nu_1(G^*)$ and $\nu_k(G^*)$, the bounded size analog of $\nu_1(G^*)$. To the best of our knowledge, this is the first work to formalize the trade-offs between sequentiality and parallelism by studying adaptivity in the context of causal graph discovery.

Theorem 1 (Atomic upper bound)**.** *Let $\mathcal{E}(G^*)$ be the observational essential graph of an underlying causal DAG G^* on n nodes and m edges. There is a r-adaptive algorithm (Algorithm 2) that uses $\mathcal{O}(\min\{r, \log n\} \cdot n^{1/\min\{r, \log n\}} \cdot \nu_1(G^*))$ atomic interventions to recover G^* from $\mathcal{E}(G^*)$. Ignoring the time spent implementing the actual interventions, Algorithm 2 runs in $\mathcal{O}(n^{1+1/r} \cdot (n+m))$ time.*

When $r = 1$, the upper bound becomes $\mathcal{O}(n)$, which is worst case optimal since $\Omega(n)$ non-adaptive interventions are necessary when the given essential graph is a path on n vertices. Meanwhile, when $r = \log n$, the upper bound becomes $\mathcal{O}(\log n \cdot \nu_1(G^*))$, matching the upper bound guarantees of [Choo et al., 2022]. Again, this is worst case optimal when the given essential graph is a path on n vertices.

In fact, for any r, our approximation factor is tight. We formally show this in the following lower bound result.

Theorem 2 (Atomic worst case)**.** *In the worst case, any r-adaptive algorithm needs to use at least $\Omega(\min\{r, \log n\} \cdot n^{1/\min\{r, \log n\}} \cdot \nu_1(G^*))$ atomic interventions against an adaptive adversary.*

We also extend our upper bound results to accommodate bounded size interventions, where each intervention can involve up to k vertices, for some pre-determined bound $k \geq 1$; atomic interventions are a special case of $k = 1$.

Theorem 3 (Bounded upper bound)**.** *Let $\mathcal{E}(G^*)$ be the observational essential graph of an underlying causal DAG G^* on n nodes. There is a polynomial time r-adaptive algorithm that uses $\mathcal{O}(\min\{r, \log n\} \cdot n^{1/\min\{r, \log n\}} \cdot \log k \cdot \nu_k(G^*))$ bounded sized interventions to recover G^* from $\mathcal{E}(G^*)$, where each intervention involves at most $k > 1$ vertices.*

[3]These assumptions are common across all/most theoretical causal graph discovery works and one may also interpret these assumptions as having access to a conditional independence oracle.

[4]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 determining 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.

While the approximation ratio is worse than Theorem 1, note that we are comparing against $\nu_k(G^*)$. Since using bounded size interventions typically translates to using a smaller number of interventions performed as k increases, $\nu_k(G^*)$ could roughly be k times smaller than $\nu_1(G^*)$.

Remark Since non-adaptive results and the upper bound guarantees of [Choo et al., 2022] already matches our upper bounds for regime of $r \geq \log n$, it suffices for us to design and analyze algorithms for regime of $1 < r < \log n$.

1.2 OUTLINE

After giving preliminaries and related work in Section 2, we slowly build up intuition towards our main results. We first solve the problem using atomic interventions when the input graph is a path or a tree (Section 3) before solving the problem in full generality on any graph input, and also generalizing to the setting of bounded size interventions (Section 4). We show how our algorithm performs in practice on synthetic graphs in Section 5 and conclude with some interesting future work directions in Section 6. For a cleaner exposition, some details are deferred to the appendix.

2 PRELIMINARIES

We write $\{1, \ldots, n\}$ as $[n]$ and use standard asymptotic notations such as $\mathcal{O}(\cdot)$ and $\Omega(\cdot)$. $A \dot\cup B$ refers to the union of two disjoint sets A and B. Logarithms are in base 2.

2.1 GRAPH NOTATIONS

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. The graph G is said to be directed or fully oriented if $A(G) = E(G)$, and partially oriented otherwise. For any two vertices $u, v \in V$, we write $u \sim v$ if these vertices are connected in the graph and $u \not\sim v$ otherwise. To specify the arc directions, we use $u \to v$ or $u \leftarrow v$. We use $G[V']$ to denote the vertex-induced subgraph for any subset $V' \subseteq V$ of vertices.

A *clique* is a graph where $u \sim v$ for any pair of vertices $u, v \in V$. A *maximal clique* is an vertex-induced subgraph of a graph that is a clique and ceases to be one if we add any other vertex to the subgraph. For an undirected graph G, $\omega(G)$ refers to the size of its maximum clique.

The *skeleton* $skel(G)$ of a (partially oriented) graph G is the underlying graph where all edges are made undirected. A *v-structure* refers to three distinct vertices $u, v, w \in V$ such that $u \to v \leftarrow w$ and $u \not\sim w$. A simple cycle is a sequence of $k \geq 3$ vertices where $v_1 \sim v_2 \sim \ldots \sim v_k \sim v_1$. The cycle is partially 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 partially directed cycle. In the undirected graph $G[E \setminus A]$ obtained by removing all arcs from a chain graph G, each connected component in $G[E \setminus A]$ 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 $V = \dot\cup_{H \in CC(G)} V(H)$.

2.2 CHORDAL GRAPHS

An undirected graph is chordal if every cycle of length at least 4 has an edge connecting two non-adjacent vertices of the cycle. There are many known characterizations and properties of chordal graphs; see [Blair and Peyton, 1993] for an introduction. One of the most common characterization is the following: A graph G is chordal if and only if perfect elimination ordering (PEO)[5] exists [Fulkerson and Gross, 1965]. Furthermore, a PEO can be computed in linear time via lexicographic BFS [Rose et al., 1976] and can used to prove the following lemma which implies that a chordal graph on n nodes has at most n maximal cliques.

Lemma 4 ([Fulkerson and Gross, 1965]; Lemma 6 in [Blair and Peyton, 1993]). *The set of maximal cliques of a graph G is precisely the sets $\{v_i\} \cup (\{v_{i+1}, \ldots, v_n\} \cap N(v_i))$ for which $\{v_i\} \cup (\{v_{i+1}, \ldots, v_n\} \cap N(v_i))$ is* not *in $\{v_j\} \cup (\{v_{j+1}, \ldots, v_n\} \cap N(v_j))$ for any vertex v_j with $j < i$.*

It is known that chordal graphs have a clique tree representation. One way to construct a clique tree T_G from a chordal graph G in polynomial time is via the "maximum-weight spanning tree property": Let the vertex set of T_G be all maximal cliques of G, assign edge weight as the size of intersection between every pair of maximal cliques, and then compute the *maximum* weight spanning tree.

The following result is one of the many useful properties of clique trees which we exploit in our algorithm later.

Lemma 5 (Lemma 5 of [Blair and Peyton, 1993]). *Let $T_G = (K, S)$ be the clique tree of a chordal graph $G = (V, E)$. For any two adjacent maximal cliques K_i and K_j in T_G, let T_i and T_j be the subtrees obtained by removing the edge $\{K_i, K_j\}$ from T_G. Then, vertices v_i and v_j are disconnected in $G[V \setminus (V(K_i) \cap V(K_j))]$.*

2.3 CAUSAL GRAPH BASICS

Directed acyclic graphs (DAGs), a special case of chain graphs where *all* edges are directed, are commonly used as graphical causal models [Pearl, 2009] where vertices represents random variables and the joint probability density f factorizes according to the Markov property:

[5] An ordering σ is a PEO if for $1 \leq i \leq n$, the node-induced subgraph $G[\{v_1, \ldots, v_{i-1}\} \cap N(v_i)]$ on v_i's neighbors is a clique.

$f(v_1, \ldots, v_n) = \prod_{i=1}^{n} f(v_i \mid pa(v_i))$, where $pa(v_i)$ is the values taken by v_i's parents in the DAG.

For any DAG G, we denote its *Markov equivalence class* (MEC) by $[G]$ and *essential graph* by $\mathcal{E}(G)$. Two graphs are Markov equivalent if and only if they have the same skeleton and v-structures [Verma and Pearl, 1990, Andersson et al., 1997]. DAGs in the same MEC have the same skeleton; Essential graph 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.

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 $\mathcal{I} \subseteq 2^V$ an *intervention set*: an intervention set is a set of interventions where each intervention corresponds to a subset of variables. 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., 2012]. 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 $S \in \mathcal{I}$. There are several known properties about $\mathcal{I}$-essential graph properties (e.g. see [Hauser and Bühlmann, 2012, 2014]). For instance, every $\mathcal{I}$-essential graph is a chain graph with chordal chain components; this includes the case of $S = \emptyset$. Also, orientations in one chain component do not affect orientations in other components. Thus, to fully orient any essential graph $\mathcal{E}(G^*)$, it is necessary and sufficient to orient every chain component in $\mathcal{E}(G^*)$.

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

Definition 6 (Verifying set and verifying number)**.** 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 verifying set* if $\mathcal{E}_{\mathcal{I}'}(G^*) \neq G^*$ for any $|\mathcal{I}'| < |\mathcal{I}|$. The *verification number* $\nu_k(G^*)$ denotes the size of the minimum size verifying set of G^* when each intervention has size at most $k \geq 1$.

Recently, [Choo et al., 2022] proved two fundamental results with respect to verification number. Theorem 7 characterizes verification number of any given causal DAG while Theorem 8 gives an adaptive search algorithm that is competitive to the verification number of the underlying causal DAG.

Theorem 7 ([Choo et al., 2022])**.** *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. An atomic intervention set $\mathcal{I}$ is a minimal sized verifying set for G if and only if $\mathcal{I}$ is a minimum vertex cover of covered edges $\mathcal{C}(G)$ of G. A minimal sized atomic verifying set can be computed in polynomial time since the edge-induced subgraph on $\mathcal{C}(G)$ is a forest.*

Theorem 8 ([Choo et al., 2022])**.** *Fix an unknown underlying DAG G^*. Given an essential graph $\mathcal{E}(G^*)$ and intervention set bound $k \geq 1$, there is a deterministic polynomial time algorithm that computes an intervention set $\mathcal{I}$ adaptively such that $\mathcal{E}_{\mathcal{I}}(G^*) = G^*$, and $|\mathcal{I}|$ has size*
1. $\mathcal{O}(\log(n) \cdot \nu_1(G^*))$ when $k = 1$
2. $\mathcal{O}(\log(n) \cdot \log(k) \cdot \nu_k(G^*))$ when $k > 1$.

To obtain a competitive bound with respect to $\nu_1(G^*)$, [Choo et al., 2022] proved a stronger (but non-computable) lower bound (Theorem 8) on $\nu_1(G^*)$. We will rely on this lemma to show a competitive bound for our algorithm later.

Lemma 9. *For any causal DAG G^*,*

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

For any intervention set $\mathcal{I} \subseteq 2^V$, we write $R(G, \mathcal{I}) \subseteq E$ to mean 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. The next result explains why it suffices to study causal graph discovery via interventions on causal graphs without v-structures: 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.

Theorem 10 ([Choo and Shiragur, 2023])**.** *For any DAG $G = (V, E)$ and intervention sets $\mathcal{A}, \mathcal{B} \subseteq 2^V$,*

$$R(G, \mathcal{A} \cup \mathcal{B}) = R(G^{\mathcal{A}}, \mathcal{B}) \dot{\cup} R(G^{\mathcal{B}}, \mathcal{A}) \dot{\cup} (R(G, \mathcal{A}) \cap R(G, \mathcal{B}))$$

We will also borrow the following definition of *relevant nodes* from [Choo and Shiragur, 2023].

Definition 11 (Relevant nodes)**.** Fix a DAG $G^* = (V, E)$ and arbitrary subset $V' \subseteq V$. For any intervention set $\mathcal{I} \subseteq V$ and resulting interventional essential graph $\mathcal{E}_{\mathcal{I}}(G^*)$, we define the *relevant nodes* $\rho(\mathcal{I}, V') \subseteq V'$ as the set of nodes within V' that is adjacent to some unoriented arc within the node-induced subgraph $\mathcal{E}_{\mathcal{I}}(G^*)[V']$.

2.4 RELATED WORK

There are two broad classes of causal graph discovery algorithms. Given the observational essential graph, non-adaptive algorithms need to recover any underlying causal graph using a single fixed set of interventions while adaptive algorithms are allowed to adapt their interventional decisions based on the outcomes of earlier interventions.

Non-adaptive search 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 undirected graph G.

Definition 12 (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 $v \in I_i$).

To simplify notation, we write G-separating system to refer to $\mathrm{skel}(G)$-separation system or $\mathrm{skel}(\mathcal{E}(G))$-separation system, for any causal DAG G. It is known that the optimal non-adaptive intervention set to learn a moral DAG G^* is a G^*-separating system [Kocaoglu et al., 2017].

Theorem 13 (Theorem 1 of [Kocaoglu et al., 2017])**.** *For any undirected graph G, an intervention set $\mathcal{I}$ learns* every *possible causal graph D with $\mathrm{skel}(D) = G$ if and only if $\mathcal{I}$ is a G-separating system.*

Adaptive search There exists essential graphs[6] where non-adaptive interventions require exponentially more interventions than if one could use adaptive interventions.

Adaptive search has been studied for special graph classes by [Shanmugam et al., 2015, Greenewald et al., 2019, Squires et al., 2020]. More recently, [Choo et al., 2022] showed that $\mathcal{O}(\log n \cdot \nu_1(G^*))$ atomic inteventions suffices to fully recover for any general causal graph G^* on n nodes while [Choo and Shiragur, 2023] showed that $\mathcal{O}(\log |\rho(\emptyset, V(H))| \cdot \nu_1(G^*))$ atomic inteventions suffices to fully recover edge directions within a subgraph H of G^*. Both papers also gave results in terms of bounded size interventions where their guarantees incur an additional $\mathcal{O}(\log k)$ factor when comparing to $\nu_k(G^*)$ by invoking Lemma 14 suitably.

Lemma 14 (Lemma 1 of [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 label index, 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.*

[6] If the essential graph is an undirected path on n vertices, then a G-sparating system has size $\Omega(n)$ while adaptive search only requires $\mathcal{O}(\log n)$ atomic interventions by "binary search".

3 WARMUP: PATHS AND TREES

When the essential graph $\mathcal{E}(G^*)$ is a path or tree, there are n possible DAGs, each associated with setting a node as a root and orienting all edges away from it. It is known that $\nu(G^*) = 1$ because the covered edges for each DAG are precisely the edges incident to the hidden root (e.g. see [Greenewald et al., 2019, Choo et al., 2022]).

In this section, we investigate how to optimally solve with r-adaptivity on these special classes of graphs.

3.1 PATHS

When the adaptivity parameter $r = 1$, any 1-adaptive algorithm behaves *exactly* like a non-adaptive algorithm. Thus, it is necessary and sufficient to intervene on a G^*-seperating system, and such a system on a path has size $\Theta(n)$.

Meanwhile, when the adaptive parameter $r = 2$, we already see an interesting trade-off occurring: *how much (and how) should we intervene now versus later?* Our next lemma tells us how to balance the number of interventions done in the first and second adaptive rounds in a worst case fashion.

Lemma 15. *Suppose $\mathcal{E}(G^*)$ is a path on n vertices. There is a 2-adaptive algorithm that uses at most $\mathcal{O}(\sqrt{n})$ interventions in total. In the worst case, $\Omega(\sqrt{n})$ is necessary for any 2-adaptive algorithm.*

Proof. Without loss of generality, up to the inclusion of floors and ceilings, let us assume that n is a square number.

Suppose the vertices on the path are labelled $v_1, v_2, \ldots, v_n$ where $v_i \sim v_{i+1}$ for $1 \leq i \leq n-1$.

In the first round, we intervene on $\sqrt{n}$ evenly spaced vertices: $v_{\sqrt{n}}, v_{2\sqrt{n}}, v_{3\sqrt{n}}, \ldots, v_n$. After applying Meek's rule R1, at most one segment of $v_{i\cdot\sqrt{n}}, v_{i\cdot\sqrt{n}+1}, v_{i\cdot\sqrt{n}+2}, \ldots, v_{(i+1)\cdot\sqrt{n}}$ will remain unoriented. That is, the total number of relevant vertices is at most $\sqrt{n}$. In the second round, we intervene on all these relevant vertices, incurring a total of $\mathcal{O}(\sqrt{n})$ interventions.

For the worst case lower bound, consider any arbitrary 2-adaptive algorithm A. If A makes strictly less than $\sqrt{n}$ interventions in the first round, then there exists a consecutive sequence of $\sqrt{n}$ vertices $v_i, v_{i+1}, \ldots, v_{i+\sqrt{n}}$ which is not intervened by A. If the root node of the path was v_i, then this entire segment $v_i \sim v_{i+1} \sim \ldots \sim v_{i+\sqrt{n}}$ remains unoriented after the first round of interventions. Thus, in the

second round, A needs to at least intervene on a separating system of this segment, which has a size at least $\Omega(\sqrt{n})$. Therefore, in the worst case, any 2-adaptive algorithm needs to perform at least $\Omega(\sqrt{n})$ number of interventions to orient $\mathcal{E}(G^*)$ when it is a path on n vertices. □

The key technical algorithmic idea in Lemma 15 is the strategy of "balanced partitioning", which balances the worst case size of the largest possible unoriented component after a round of intervention. See Fig. 1 for an example.

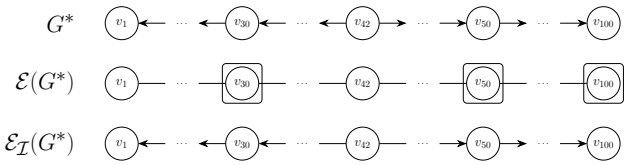

Figure 1: Path example with $n = 100$ with v_{42} as the hidden source. After intervening on $\mathcal{I} = \{v_{10}, v_{20}, \ldots, v_{100}\}$ atomically and applying Meek rules, only the segment $v_{41} \sim v_{42} \sim \ldots \sim v_{49}$ remains unoriented.

Interestingly, we can generalize this "balanced partitioning" strategy to larger values of adaptivity parameter r as follows: Suppose we perform L interventions per round in the first $r-1$ rounds and then intervene on the G-separating system on the remaining relevant vertices in the final round, where L is an integer that depends on r which we define later. In each of the $r-1$ rounds, we will choose the L vertices judiciously by a "balanced partitioning". If we further insist that the final G-separating system has size at most L, then we get a recurrence relation $\frac{n}{(L+1)^{r-1}} \leq L$ while incurring a total of $\approx r \cdot L$ interventions. We formally prove this next.

Lemma 16. *Suppose $\mathcal{E}(G^*)$ is a path on n vertices. There is a r-adaptive algorithm that uses at most $\mathcal{O}(r \cdot n^{1/r})$ interventions in total.*

Proof. Observe that no matter how we intervene on a path, the remaining relevant vertices will form a subpath.

Let $L = \lceil n^{1/r} \rceil$. In the first $r-1$ rounds, we will partition the remaining unoriented subpath into equal length segments. For instance, if the subpath length is currently ℓ, then we partition it into segments of length $\lceil \ell/(L+1) \rceil$ or $\lfloor \ell/(L+1) \rfloor$. Thus, the length of the subpath with relevant vertices in the next round will be at most $\lceil \ell/(L+1) \rceil$.

After $r-1$ rounds, the length of the remaining unoriented subpath is at most (via repeated applications of Lemma 17):

$$\left\lceil \left\lceil \left\lceil \frac{n}{L+1} \right\rceil /(L+1) \right\rceil \ldots \right\rceil \leq \ldots \leq \left\lceil \frac{n}{(L+1)^{r-1}} \right\rceil$$

Since $L = \lceil n^{1/r} \rceil$, we see that $\left\lceil \frac{n}{(L+1)^{r-1}} \right\rceil \leq L$. Therefore, by intervening on all the remaining relevant vertices in the final r-th round, the total number of interventions performed will be $\mathcal{O}(r \cdot L) \subseteq \mathcal{O}(r \cdot n^{1/r})$. □

Lemma 17 (See appendix for proof)**.** *For positive integer n, and arbitrary real numbers m, x, we have* $\left\lceil \frac{\lceil \frac{x}{m} \rceil}{n} \right\rceil = \lceil \frac{x}{mn} \rceil$.

3.2 TREES

While the high-level strategy employed in Lemma 16 also works for trees, how we perform the "balanced partitioning" is slightly different since the vertices are no longer arranged in a line. Instead, we rely on Algorithm 1 to partition a tree on n nodes into subtrees using L node removals such that each subtree has size at most $\lceil n/(L+1) \rceil$.

The correctness of Algorithm 1 is proven in Lemma 18 and the resulting r-adaptive search result is given in Lemma 19.

Algorithm 1 Balanced partitioning on trees.

Input: A tree $G = (V, E)$ with $|V| = n$, an integer L.
Output: A subset of vertices $A \subseteq V$ such that $|A| \leq L$ and subtrees in $G[V \setminus A]$ have size at most $\lceil n/(L+1) \rceil$.

Initialize $A \leftarrow \emptyset$
while tree $G = (V, E)$ has size $|V| > \lceil \frac{n}{L+1} \rceil$ **do**
  Root G arbitrarily.
  Compute size of subtrees T_u at each node $u \in V$
  if there is a subtree T_u of size $1 + \lceil \frac{n}{L+1} \rceil$ **then**
    Add u to A; Update $G \leftarrow G[V \setminus V(T_u)]$.
  else
    Find a subtree T_u of size $|V(T_u)| > 1 + \lceil \frac{n}{L+1} \rceil$, s.t. $|V(T_w)| \leq \lceil \frac{n}{L+1} \rceil$ for *all* children w of u.
    Add u to A; Update $G \leftarrow G[V \setminus V(T_u)]$.
  end if
end while
return A

Lemma 18. *Given a tree $G = (V, E)$ with $|V| = n$ and an integer $L \leq n$, Algorithm 1 runs in polynomial time and returns a subset of vertices $A \subseteq V$ such that $|A| \leq L$ and subtrees in $G[V \setminus A]$ have size at most $\lceil n/(L+1) \rceil$.*

Proof. We first prove the correctness then the running time.

Correctness Consider an arbitrary iteration of the while loop. We will argue two things:

1. The size of G decreases by at least $\lceil \frac{n}{L+1} \rceil$.
2. The newly pruned subtree(s) have size at most $\lceil \frac{n}{L+1} \rceil$.

Since $n - L \cdot \lceil \frac{n}{L+1} \rceil \leq \lceil \frac{n}{L+1} \rceil$, the while loop terminates after at most L iterations. Thus, $|A| \leq L$ as we only add one vertex per iteration to A.

if-case: Since $|V(T_u)| = 1 + \lceil \frac{n}{L+1} \rceil$, removing $V(T_u)$ from V decreases the size of G by at least $\lceil \frac{n}{L+1} \rceil$. Furthermore, the subtree pruned by u's removal has size exactly $\lceil \frac{n}{L+1} \rceil$.

else-case: This case happens when *no* subtrees has size exactly $1 + \lceil \frac{n}{L+1} \rceil$. So, if a subtree has size strictly larger than $\lceil \frac{n}{L+1} \rceil$, then it must have size at least $2 + \lceil \frac{n}{L+1} \rceil$.

We first show how to find such a subtree T_u in Line 8:

1. Initialize u as the root node.
2. If all children w of u have subtree sizes $|V(T_w)| \leq \lceil \frac{n}{L+1} \rceil$, we have found T_u and can stop.
3. Otherwise, update u to any child w with subtree size $|V(T_w)| > \lceil \frac{n}{L+1} \rceil$, and go to step 2. Note that $|V(T_w)| \geq 2 + \lceil \frac{n}{L+1} \rceil$ from the above discussion.

Since $|V(T_u)| > 1 + \lceil \frac{n}{L+1} \rceil$, removing $V(T_u)$ from V decreases the size of G by at least $\lceil \frac{n}{L+1} \rceil$. Furthermore, each subtree pruned by u's removal has size at most $\lceil \frac{n}{L+1} \rceil$ since all subtrees of children w of u has size at most $\lceil \frac{n}{L+1} \rceil$.

Running time Rooting the tree and computing sizes of each rooted subtree can be done in polynomial time via depth-first search and dynamic programming. Finding the node u in within the while loop can also be done in polynomial time. Since the while loop executes at most $L \leq n$ times, the algorithm runs in polynomial time. □

Lemma 19. *Suppose $\mathcal{E}(G^*)$ is a tree on n vertices. There is a r-adaptive algorithm that uses at most $\mathcal{O}(r \cdot n^{1/r})$ interventions in total.*

Proof. Since the underlying causal DAG is a tree, whenever $\mathcal{E}(G^*)$ is not fully oriented, the chain components are singletons and at most one tree (of size at least two).

Let $L = \lceil n^{1/r} \rceil$. Consider the following algorithm:

1. Run Algorithm 1 on the tree chain component of $\mathcal{E}(G^*)$ to obtain a subset of vertices A of size $|A| \leq L$.
2. Intervene on all vertices in A.
3. If the essential graph is still not fully oriented and we have used less than $r-1$ rounds, go back to step 1.
4. In the final round, we intervene on all vertices[7] in the tree chain component (if it exists).

In the first $r-1$ adaptive rounds, the new tree chain component must have size at most one of the pruned subtrees, i.e. size at most $1/(L+1)$ factor smaller than before. After $r-1$ rounds, the largest possible size of the tree chain component is at most (via repeated applications of Lemma 17):

$$\left\lceil \left\lceil \left\lceil \frac{n}{L+1} \right\rceil /(L+1) \right\rceil \dots \right\rceil \leq \dots \leq \left\lceil \frac{n}{(L+1)^{r-1}} \right\rceil$$

[7] We could have further optimized by intervening on a G-separating system of the remaining tree component but this does not affect the asymptotics in the worst case. For instance, if the underlying causal DAG is a single path, then the final round will still be a path and thus the G-separating system size is still roughly at least half the remaining nodes.

Since $L = \lceil n^{1/r} \rceil$, we see that $\left\lceil \frac{n}{(L+1)^{r-1}} \right\rceil \leq L$. Therefore, by intervening on all the remaining relevant vertices in the final r-th round, the total number of interventions performed will be $\mathcal{O}(r \cdot L) \subseteq \mathcal{O}(r \cdot n^{1/r})$. □

Fig. 2 illustrates a potential partitioning example when the essential graph is a tree on $n = 16$ nodes and $r = 2$.

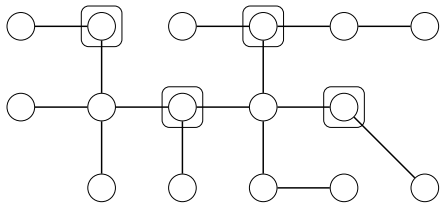

Figure 2: Tree example on $n = 16$ nodes where the boxed vertices represent a subset $A \subseteq V$ of size $|A| = \sqrt{n} = 4$. Observe that the subtrees in $G[V \setminus A]$ each are of size at most $\lceil n/(\sqrt{n}+1) \rceil = \lceil 16/5 \rceil = 4$.

4 FULL GENERALITY

Building upon the intuition developed in Section 3, we now show how to obtain Theorem 1. We also present a matching worst case lower bound (Theorem 2), and extend our results to the bounded size intervention settings (Theorem 3).

Let us now describe the algorithmic idea behind our atomic intervention result. By Theorem 10, we can remove all oriented edges from earlier rounds of interventions and focus on each undirected chain component. Let L be a suitably chosen value that depends on r, which we define later. Since these chain components are chordal graphs, we can efficiently compute their clique tree representation for each chain component G. Invoking Algorithm 1 to compute a "balanced partitioning" on the clique tree T_G, we obtain L maximal cliques from each chain component G. Intervening on all these cliques incurs a cost of $\mathcal{O}(L \cdot \nu_1(G^*))$ interventions, as the sum of maximal cliques in each chain component is a lower bound for the verification number (see Lemma 9). Performing this operation for $r-1$ rounds leaves us with chain components, each with at most L maximal cliques. In the final r-th round, we intervene on all the vertices in these unoriented connected components which additionally incurs a cost of at most $\mathcal{O}(L \cdot \nu_1(G^*))$. Therefore, the total cost in all iterations is at most $\mathcal{O}(r \cdot L \cdot \nu_1(G^*))$. Setting $L \approx n^{1/r}$ above gives the desired guarantees.

To formally prove our result, we first begin with following lemma which tells us that intervening on all vertices within a maximal clique of a chordal graph breaks up the chain component according to its clique tree representation.

Lemma 20. *Let $T_G = (K, S)$ be the clique tree of a chordal graph $G = (V, E)$. For any maximal clique K_i in T_G, if two*

maximal cliques K_1 and K_2 are disjoint in $T_G[K \setminus \{K_i\}]$, then $v_1 \in V(K_1)$ and $v_2 \in V(K_2)$ are not in the same connected component of $G[V \setminus V(K_i)]$.

Proof. Since K_1 and K_2 are disjoint in $T_G[K \setminus \{K_i\}]$, the removed K_i must lie on the unique path between K_1 and K_2 in the tree T_G. As $V(K_i) = \cup_{\{K_i,K_j\} \in S}(V(K_i) \cap V(K_j))$, Lemma 5 tells us that the removal of K_i will makes v_1 and v_2 disconnected in $G[V \setminus V(K_i)]$. □

We are now ready to prove Theorem 1 using Algorithm 2.

Algorithm 2 Adaptivity-sensitive search.

Input: Essential graph $\mathcal{E}(G^*)$, adaptivity param. $r \geq 1$.
Output: A sequence of intervention sets $\mathcal{I}_1, \ldots, \mathcal{I}_r$ such that $\mathcal{E}_{\mathcal{I}_1,\ldots,\mathcal{I}_r}(G^*) = G^*$.

1: Initialize $L = \lceil n^{1/r} \rceil$.
2: **for** $i = 1, \ldots, r-1$ **do**
3: Initialize $\mathcal{I}_i \leftarrow \emptyset$
4: **for** chain comp. $H \in CC(\mathcal{E}_{\mathcal{I}_1,\ldots,\mathcal{I}_{i-1}}(G^*))$ **do**
5: **if** H is a clique **then** Set $V' \leftarrow V(H)$.
6: **else** Compute clique tree T_H of H. Compute L-balanced partitioning S of T_H via Algorithm 1. Let $V' \leftarrow \cup_{K_j \in S} V(K_j)$.
7: Add V' to $\mathcal{I}_i$.
8: **end for**
9: Intervene on all vertices in $\mathcal{I}_i$.
10: **end for**
11: Define $\mathcal{I}_r$ as all remaining relevant vertices and intervene on vertices in $\mathcal{I}_r$.
12: **return** $\mathcal{I}_1, \ldots, \mathcal{I}_r$

To visualize the full generalization of Algorithm 2, think of Fig. 2 as the clique tree representation of a chain component of the essential graph where each node is a maximal clique. When we intervene on a "node" in Fig. 2, we actually intervene on the entire maximally clique in an atomic fashion so that all incident edges to the clique vertices are oriented.

Theorem 1 (Atomic upper bound)**.** *Let $\mathcal{E}(G^*)$ be the observational essential graph of an underlying causal DAG G^* on n nodes and m edges. There is a r-adaptive algorithm (Algorithm 2) that uses $\mathcal{O}(\min\{r, \log n\} \cdot n^{1/\min\{r,\log n\}} \cdot \nu_1(G^*))$ atomic interventions to recover G^* from $\mathcal{E}(G^*)$. Ignoring the time spent implementing the actual interventions, Algorithm 2 runs in $\mathcal{O}(n^{1+1/r} \cdot (n+m))$ time.*

Proof. Consider Algorithm 2. Since we always intervene on all relevant vertices outside of the for loop, we are guaranteed to fully recover G^*.

Number of interventions

By Lemma 18, we know that invoking Algorithm 1 for any clique tree T_H on maximal cliques K_H of H returns a set $A_H \subseteq K_H$ of at most $|A_H| \leq L$ clique nodes such that subtrees in $T_H[K_H \setminus A_H]$ have size at most $\lceil |T_H|/(L+1) \rceil$.

By Lemma 9,

$$\left| \bigcup_{H \in CC(\mathcal{E}_{\mathcal{I}_1 \cup \ldots \cup \mathcal{I}_{i-1}}(G^*))} A_H \right| = \sum_{H \in CC(\mathcal{E}_{\mathcal{I}_1 \cup \ldots \cup \mathcal{I}_{i-1}}(G^*))} |A_H| \in \mathcal{O}(L \cdot \nu_1(G^*))$$

So, within the for loop, we incur at most $\mathcal{O}(r \cdot L \cdot \nu_1(G^*))$ interventions.

By Lemma 20, we know that each iteration reduces the maximum number of maximal cliques in any chain component size by a factor of $L+1$. After $r-1$ rounds, the largest clique tree in any chain component has at most

$$\frac{n}{(L+1)^{r-1}} = \frac{n}{(\lceil n^{1/r} \rceil + 1)^{r-1}} \leq \lceil n^{1/r} \rceil = L$$

maximal cliques. So, if we intervene on all remaining relevant vertices in the final r-th round, this incurs at most $\mathcal{O}(L \cdot \nu_1(G^*))$ interventions via Lemma 9.

Therefore, in total, we use $\mathcal{O}(r \cdot L \cdot \nu_1(G^*)) \subseteq \mathcal{O}(r \cdot n^{1/r} \cdot \nu_1(G^*))$ atomic interventions.

Running time

Throughout, executing Meek rules after performing an intervention can be done in $\mathcal{O}(d \cdot m) \subseteq \mathcal{O}(n \cdot m)$ time [Wienöbst et al., 2021], where d is the degeneracy of the input graph. Now, consider an arbitrary iteration of the while loop. There are at most n chain components. Given a chordal graph with n nodes and m edges, a clique tree can computed in $\mathcal{O}(n+m)$ time [Blair and Peyton, 1993, Galinier et al., 1995]. Given a tree with n nodes, computing a L-balanced partitioning (Algorithm 1) takes $\mathcal{O}(n)$ time using depth-first search. So, each iteration takes $\mathcal{O}(n \cdot (n+m))$ time. Since there are $L = \lceil n^{1/r} \rceil$ iterations, the overall running time is $\mathcal{O}(L \cdot n \cdot (n+m)) \subseteq \mathcal{O}(n^{1+1/r} \cdot (n+m))$. □

Atomic worst case. Our lower bound (Theorem 2) generalizes the idea behind the lower bound proof in Lemma 15 (for the special case of $r = 2$): on a path essential graph with $\nu_1(G^*) = 1$, the adaptive adversary repeatedly hides the source node in the largest possible unoriented segment based on the current round of interventions. See Appendix C for the full proof of Theorem 2.

Bounded size interventions. With bounded size interventions, each intervention is now allowed to involve $k \geq 1$ vertices, for some pre-determined upper bound k. Algorithmically, we tweak Lines 7 and 11 in Algorithm 2 to use the labelling lemma of Lemma 14, which incurs additional $\mathcal{O}(\log k)$ multiplicative factor when comparing with $\nu_k(G^*)$. See Appendix B for the tweaked algorithm and Appendix C for the full proof of Theorem 3.

5 EXPERIMENTS

While our main contributions are theoretical, we also performed some experiments to empirically validate that Algorithm 2 is practical and that fewer interventions are generally needed when higher levels of adaptivity are allowed.

Using the synthetic experimental setup of [Squires et al., 2020, Choo et al., 2022, Choo and Shiragur, 2023], we benchmarked our adaptivity-sensitive search algorithm with varying values of r together with the algorithm of [Choo et al., 2022]. For full experimental details, implementation details, and source code, please see Appendix D.

Checks to avoid redundant interventions The current implementation of [Choo et al., 2022]'s `separator` algorithm is actually n-adaptive because it performs "checks" before performing each intervention — if the vertices in the proposed intervention set S do *not* have any unoriented incident arcs, then the intervention set S will be skipped. One may think of such interventions as "redundant" since they do not yield any new information about the underlying causal graph. As such, we ran two versions of their algorithm: one without checks (i.e. $\mathcal{O}(\log n)$-adaptive) and one with checks (i.e. n-adaptive). Note that each check corresponds to an adaptivity round because an intervention within a batch of interventions may turn out to be redundant, but we will only know this after performing a check after some of the interventions within that batch have been executed.

Scaling our algorithm with checks Since $n^{\frac{1}{\log n}} = 2$, running Algorithm 2 (as it is) with adaptivity parameters $r \in \Omega(\log n)$ does not make much sense. As such, we define a checking budget $b = r - \lceil \log n \rceil$ and greedily perform up to b checks whilst executing Algorithm 2. This allows Algorithm 2 to scale naturally for $r \in \Omega(\log n)$.

Empirical trends Fig. 3 shows a subset of our results (see Appendix D for all experimental results). As expected, we observe that higher rounds of adaptivity leads to lower number of interventions required. When $r = n$, Algorithm 2 can match [Choo et al., 2022] with its full adaptivity.

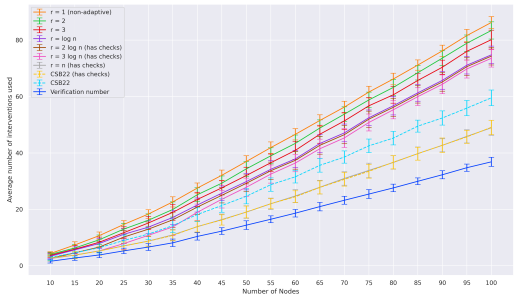

Figure 3: Subset of experimental results

6 CONCLUSION

In our work, we define r-adaptivity that interpolates between non-adaptivity (for $r = 1$) and full adaptivity (for $r = n$). We provide a r-adaptive algorithm that achieves $\mathcal{O}(\min\{r, \log n\} \cdot n^{1/\min\{r, \log n\}})$ approximation with respect to the verification number.

Let us denote ν^r as the necessary number of interventions required when allowed r rounds of adaptive interventions. Amongst $r \in \{1, 2, \ldots, n\}$, the only two values of ν^r that we currently understand are the extremes of ν^1 (non-adaptive) and ν^n (full adaptivity), where ν^1 is the size of a $\text{skel}(G^*)$-separating system and ν^n corresponds to the verification number of G^*. Furthermore, we also know that ν^r is non-decreasing as r decreases from n to 1. Ideally, given an input r, we want to compete against ν^r. Unfortunately, as we only currently understand ν^1 and ν^n, and ν^1 could potentially even be exponentially larger than ν^n (e.g. when the essential graph is a path), we gave approximations in terms of ν^1 in this work while it may be the case that ν^1 is a weak lower bound for r-adaptive algorithms, especially for smaller values of r. As such, understanding the correct bound for a given r is an interesting open direction to pursue and with the hopes that one can design an corresponding r-adaptive search algorithm achieving a good approximation with respect to ν^r, possibly better than Algorithm 2.

Here, we studied the complexity of adaptivity for causal discovery under the standard assumptions of causal sufficiency, faithfulness, and infinite sample regime. When these assumptions are violated, wrong causal conclusions may be drawn and possibly lead to unintended downstream consequences. Hence, it is of great interest to remove/weaken these assumptions while maintaining theoretical guarantees.

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 the UAI reviewers for valuable feedback and discussions.

References

Arpit Agarwal, Shivani Agarwal, Sepehr Assadi, and Sanjeev Khanna. Learning with limited rounds of adaptivity: Coin tossing, multi-armed bandits, and ranking from pairwise comparisons. In *Conference on Learning Theory*, pages 39–75. PMLR, 2017.

Noga Alon, Noam Nisan, Ran Raz, and Omri Weinstein. Welfare maximization with limited interaction. In *2015*

IEEE 56th Annual Symposium on Foundations of Computer Science, pages 1499–1512. IEEE, 2015.

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

Eric Balkanski and Yaron Singer. The adaptive complexity of maximizing a submodular function. In *Proceedings of the 50th annual ACM SIGACT symposium on theory of computing*, pages 1138–1151, 2018.

Jean RS Blair and Barry Peyton. An introduction to chordal graphs and clique trees. In *Graph theory and sparse matrix computation*, pages 1–29. Springer, 1993.

Simina Brânzei and Jiawei Li. The query complexity of local search and brouwer in rounds. In *Conference on Learning Theory*, pages 5128–5145. PMLR, 2022.

Mark Braverman, Jieming Mao, and S Matthew Weinberg. Parallel algorithms for select and partition with noisy comparisons. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 851–862, 2016.

Canonne, Clément L and Gur, Tom. An Adaptivity Hierarchy Theorem for Property Testing. *computational complexity*, 27:671–716, 2018.

Hyunghoon Cho, Bonnie Berger, and Jian Peng. Reconstructing Causal Biological Networks through Active Learning. *PLoS ONE*, 11(3):e0150611, 2016.

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

Davin Choo, Kirankumar Shiragur, and Arnab Bhattacharyya. Verification and search algorithms for causal DAGs. *Advances in Neural Information Processing Systems*, 35, 2022.

Richard Cole. Parallel merge sort. *SIAM Journal on Computing*, 17(4):770–785, 1988.

Luis M. de Campos, Andrés Cano, Javier G. Castellano, and Serafín Moral. 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.

Brian C Dean, Michel X Goemans, and Jan Vondrák. Approximating the stochastic knapsack problem: The benefit of adaptivity. *Mathematics of Operations Research*, 33 (4):945–964, 2008.

Pavol Duris, Zvi Galil, and Georg Schnitger. Lower bounds on communication complexity. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 81–91, 1984.

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

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

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

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

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

Frederick Eberhardt, Clark Glymour, and Richard Scheines. On the Number of Experiments Sufficient and in the Worst Case Necessary to Identify All Causal Relations Among N Variables. *arXiv preprint arXiv:1207.1389*, 2012.

Delbert Fulkerson and Oliver Gross. Incidence matrices and interval graphs. *Pacific journal of mathematics*, 15(3): 835–855, 1965.

Philippe Galinier, Michel Habib, and Christophe Paul. Chordal graphs and their clique graphs. *WG*, 95:358–371, 1995.

Michel Goemans and Jan Vondrák. Stochastic covering and adaptivity. In *LATIN 2006: Theoretical Informatics: 7th Latin American Symposium, Valdivia, Chile, March 20-24, 2006. Proceedings 7*, pages 532–543. Springer, 2006.

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

Anupam Gupta, Viswanath Nagarajan, and Sahil Singla. Algorithms and adaptivity gaps for stochastic probing. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 1731–1747. SIAM, 2016.

Anupam Gupta, Viswanath Nagarajan, and Sahil Singla. Adaptivity gaps for stochastic probing: Submodular and xos functions. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1688–1702. SIAM, 2017.

Jarvis Haupt, Robert Nowak, and Rui Castro. Adaptive Sensing for Sparse Recovery. In *2009 IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop*, pages 702–707. IEEE, 2009.

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

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

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

Huining Hu, Zhentao Li, and Adrian Vetta. Randomized Experimental Design for Causal Graph Discovery. *Advances in Neural Information Processing Systems*, 27, 2014.

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

Piotr Indyk, Eric Price, and David P Woodruff. On the Power of Adaptivity in Sparse Recovery. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 285–294. IEEE, 2011.

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

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

Dmitry M Malioutov, Sujay Sanghavi, and Alan S Willsky. Compressed sensing with sequential observations. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3357–3360. IEEE, 2008.

Noam Nisan and Avi Widgerson. Rounds in communication complexity revisited. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 419–429, 1991.

Christos H Papadimitriou and Michael Sipser. Communication complexity. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 196–200, 1982.

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

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

Sofya Raskhodnikova and Adam Smith. A Note on Adaptivity in Testing Properties of Bounded Degree Graphs. In *Electronic Colloquium on Computational Complexity (ECCC)*. Citeseer, 2006.

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

Donald J. Rose, R. Endre Tarjan, and George S. Lueker. Algorithmic Aspects of Vertex Elimination on Graphs. *SIAM Journal on Computing*, 5(2):266–283, 1976. doi: 10.1137/0205021.

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

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

Jonathan Scarlett. Noisy adaptive group testing: Bounds and algorithms. *IEEE Transactions on Information Theory*, 65(6):3646–3661, 2018.

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

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

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

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

Leslie G Valiant. Parallelism in comparison problems. *SIAM Journal on Computing*, 4(3):348–355, 1975.

Thomas Verma and Judea Pearl. Equivalence and Synthesis of Causal Models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, UAI '90, page 255–270, USA, 1990. Elsevier Science Inc. ISBN 0444892648.

Marcel Wienöbst, Max Bannach, and Maciej Liśkiewicz. Extendability of Causal Graphical Models: Algorithms and Computational Complexity. In *Uncertainty in Artificial Intelligence*, pages 1248–1257. PMLR, 2021.

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