Active causal structure learning with advice

Davin Choo [1] **Themis Gouleakis** [1] **Arnab Bhattacharyya** [1]

Abstract

We introduce the problem of active causal structure learning with advice. In the typical well-studied setting, the learning algorithm is given the essential graph for the observational distribution and is asked to recover the underlying causal directed acyclic graph (DAG) G^* while minimizing the number of interventions made. In our setting, we are additionally given side information about G^* as advice, e.g. a DAG G purported to be G^*. We ask whether the learning algorithm can benefit from the advice when it is close to being correct, while still having worst-case guarantees even when the advice is arbitrarily bad. Our work is in the same space as the growing body of research on *algorithms with predictions*. When the advice is a DAG G, we design an adaptive search algorithm to recover G^* whose intervention cost is at most $\mathcal{O}(\max\{1, \log \psi\})$ times the cost for verifying G^*; here, ψ is a distance measure between G and G^* that is upper bounded by the number of variables n, and is exactly 0 when $G = G^*$. Our approximation factor matches the state-of-the-art for the advice-less setting.

1. Introduction

A *causal directed acyclic graph* on a set V of n variables is a Bayesian network in which the edges model direct causal effects. A causal DAG can be used to infer not only the observational distribution of V but also the result of any intervention on any subset of variables $V' \subseteq V$. In this work, we restrict ourselves to the *causally sufficient* setting where there are no latent confounders, no selection bias, and no missingness in data.

The goal of *causal structure learning* is to recover the underlying DAG from data. This is an important problem with applications in multiple fields including philosophy, medicine, biology, genetics, and econometrics (Reichenbach, 1956; Hoover, 1990; King et al., 2004; Woodward, 2005; Rubin & Waterman, 2006; Eberhardt & Scheines, 2007; Sverchkov & Craven, 2017; Rotmensch et al., 2017; Pingault et al., 2018). Unfortunately, in general, it is known that observational data can only recover the causal DAG up to an equivalence class (Pearl, 2009; Spirtes et al., 2000). Hence, if one wants to avoid making parametric assumptions about the causal mechanisms, the only recourse is to obtain experimental data from interventions (Eberhardt et al., 2005; 2006; Eberhardt, 2010).

Such considerations motivate the problem of *interventional design* where the task is to find a set of interventions of optimal cost which is sufficient to recover the causal DAG. There has been a series of recent works studying this problem (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; Choo & Shiragur, 2023) under various assumptions. In particular, assuming causal sufficiency, (Choo et al., 2022) gave an adaptive algorithm that actively generates a sequence of interventions of bounded size, so that the total number of interventions is at most $\mathcal{O}(\log n)$ times the optimal.

Typically though, in most applications of causal structure learning, there are domain experts and practitioners who can provide additional "advice" about the causal relations. Indeed, there has been a long line of work studying how to incorporate expert advice into the causal graph discovery process; e.g. see (Meek, 1995a; Scheines et al., 1998; De Campos & Ji, 2011; Flores et al., 2011; Li & Beek, 2018; Andrews et al., 2020; Fang & He, 2020). In this work, we study in a principled way how using purported expert advice can lead to improved algorithms for interventional design.

Before discussing our specific contributions, let us ground the above discussion with a concrete problem of practical importance. In modern virtualized infrastructure, it is increasingly common for applications to be modularized into a large number of interdependent microservices. These microservices communicate with each other in ways that depend on the application code and on the triggering userflow. Crucially, the communication graph between microservices is often unknown to the platform provider as the application code may be private and belong to different entities.

[1] School of Computing, National University of Singapore. Correspondence to: Davin Choo <davin@u.nus.edu>.

However, knowing the graph is useful for various critical platform-level tasks, such as fault localization (Zhou et al., 2019), active probing (Tan et al., 2019), testing (Jha et al., 2019), and taint analysis (Clause et al., 2007). Recently, (Wang et al., 2023) and (Ikram et al., 2022) suggested viewing the microservices communication graph as a sparse causal DAG. In particular, (Wang et al., 2023) show that arbitrary interventions can be implemented as fault injections in a staging environment, so that a causal structure learning algorithm can be deployed to generate a sequence of interventions sufficient to learn the underlying communication graph. In such a setting, it is natural to assume that the platform provider already has an approximate guess about the graph, e.g. the graph discovered in a previous run of the algorithm or the graph suggested by public metadata tagging microservice code. The research program we put forth is to design causal structure learning algorithms that can take advantage of such potentially imperfect advice[1].

1.1. Our contributions

In this work, we study *adaptive intervention design* for recovering *non-parametric* causal graphs *with expert advice*. Specifically, our contributions are as follows.

- **Problem Formulation**. Our work connects the causal structure learning problem with the burgeoning research area of *algorithms with predictions* or *learning-augmented algorithms* (Mitzenmacher & Vassilvitskii, 2022) where the goal is to design algorithms that bypass worst-case behavior by taking advantage of (possibly erroneous) advice or predictions about the problem instance. Most work in this area has been restricted to online algorithms, data structure design, or optimization, as described later in Section 2.5. However, as we motivated above, expert advice is highly relevant for causal discovery, and to the best of our knowledge, ours is the first attempt to formally address the issue of *imperfect* advice in this context.

- **Adaptive Search Algorithm**. We consider the setting where the advice is a DAG G purported to be the orientations of all the edges in the graph. We define a distance measure which is always bounded by n, the number of variables, and equals 0 when $G = G^*$. For any integer $k \geq 1$, we propose an adaptive algorithm to generate a sequence of interventions of size at most k that recovers the true DAG G^*, such that the total number of interventions is $\mathcal{O}(\log \psi(G, G^*) \cdot \log k)$ times the optimal number of interventions of size k. Thus,

our approximation factor is never worse than the factor for the advice-less setting in (Choo et al., 2022). Our search algorithm also runs in polynomial time.

- **Verification Cost Approximation**. For a given upper bound $k \geq 1$, a verifying intervention set for a DAG G^* is a set of interventions of size at most k that, together with knowledge of the Markov equivalence class of G^*, determines the orientations of all edges in G^*. The minimum size of a verifying intervention set for G^*, denoted $\nu_k(G^*)$, is clearly a lower bound for the number of interventions required to learn G^* (regardless of the advice graph G). One of our key technical results is a structural result about ν_1. We prove that for any two DAGs G and G' within the same Markov equivalence class, we always have $\nu_1(G) \leq 2 \cdot \nu_1(G')$ and that this is tight in the worst case. Beyond an improved structural understanding of minimum verifying intervention sets, which we believe is of independent interest, this enables us to "blindly trust" the information provided by imperfect advice to some extent.

Similar to prior works (e.g. (Squires et al., 2020; Choo et al., 2022; Choo & Shiragur, 2023)), we assume causal sufficiency and faithfulness while using ideal interventions. Under these assumptions, running standard causal discovery algorithms (e.g. PC (Spirtes et al., 2000), GES (Chickering, 2002)) will always successfully recover the correct essential graph from data. We also assume that the given expert advice is consistent with observational essential graph. See Appendix A for a discussion about our assumptions.

1.2. Paper organization

In Section 2, we intersperse preliminary notions with related work. Our main results are presented in Section 3 with the high-level technical ideas and intuition given in Section 4. Section 5 provides some empirical validation. See the appendices for full proofs, source code, and experimental details.

2. Preliminaries and Related Work

Basic notions about graphs and causal models are defined in Appendix B. To be *very* brief, if $G = (V, E)$ is a graph on $|V| = n$ nodes/vertices where $V(G)$, $E(G)$, and $A(G) \subseteq E(G)$ denote nodes, edges, and arcs of G respectively, we write $u \sim v$ to denote that two nodes $u, v \in V$ are connected in G, and write $u \rightarrow v$ or $u \leftarrow v$ when specifying a certain direction. The *skeleton* $\mathrm{skel}(G)$ refers to the underlying graph where all edges are made undirected. A *v-structure* in G refers to a collection of three distinct vertices $u, v, w \in V$ such that $u \rightarrow v \leftarrow w$ and $u \not\sim w$. Let $G = (V, E)$ be fully unoriented. For vertices $u, v \in V$, subset of vertices $V' \subseteq V$ and integer $r \geq 0$, we define $\mathrm{dist}_G(u, v)$ as the

---

[1]Note however that the system in (Wang et al., 2023) is not causally sufficient due to confounding user behavior and (Ikram et al., 2022) does not actively perform interventions. So, the algorithm proposed in this work cannot be used directly for the microservices graph learning problem.

shortest path length between u and v, and $N_G^r(V') = \{v \in V : \min_{u \in V'} \text{dist}_G(u,v) \le r\} \subseteq V$ as the set of vertices that are r-hops away from V' in G. A directed acyclic graph (DAG) is a fully oriented graph without directed cycles. 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 have the same skeleton and the 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. 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) and the essential graph $\mathcal{E}(G)$ can be computed from G by orienting v-structures in $\text{skel}(G)$ and applying Meek rules (see Appendix D). In a DAG G, an edge $u \to v$ is a *covered edge* if $\text{Pa}(u) = \text{Pa}(v) \setminus \{u\}$. We use $\mathcal{C}(G) \subseteq E(G)$ to denote the set of covered edges of G.

2.1. Ideal interventions

An *intervention* $S \subseteq V$ is an experiment where all variables $s \in S$ is forcefully set to some value, independent of the underlying causal structure. An intervention is *atomic* if $|S| = 1$ and *bounded size* if $|S| \le k$ for some $k \ge 1$; 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 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., 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).

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 $|A|, |B| \le \alpha \cdot |V|$. 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 $|V| \ge 2$ 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.*

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 (Hauser & Bühlmann, 2012; 2014): Every $\mathcal{I}$-essential graph is a chain graph[2] with chordal[3] chain components. This includes the case of $\mathcal{I} = \emptyset$. Orientations in one chain component do not affect orientations in other components. In other words, to fully orient any essential graph $\mathcal{E}(G^*)$, it is necessary and sufficient to orient every chain component in $\mathcal{E}(G^*)$.

For any intervention set $\mathcal{I} \subseteq 2^V$, we write $R(G, \mathcal{I}) = A(\mathcal{E}_{\mathcal{I}}(G)) \subseteq E$ to mean the set of oriented arcs in the $\mathcal{I}$-essential graph of a DAG G. For cleaner notation, we write $R(G, I)$ for single interventions $\mathcal{I} = \{I\}$ for some $I \subseteq V$, and $R(G, v)$ for single atomic interventions $\mathcal{I} = \{\{v\}\}$ for some $v \in V$. For any interventional set $\mathcal{I} \subseteq 2^V$, define $G^{\mathcal{I}} = G[E \setminus R(G, \mathcal{I})]$ as the *fully directed* subgraph DAG induced by the *unoriented arcs* in $\mathcal{E}_{\mathcal{I}}(G)$, where $G^{\emptyset}$ is the graph obtained after removing all the oriented arcs in the observational essential graph due to v-structures. See Figure 1 for an example. In the notation of $R(\cdot, \cdot)$, the following result justifies studying verification and adaptive search via ideal interventions only on DAGs without v-structures, i.e. moral DAGs (Definition 2.4): 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.[4]

Theorem 2.3 ((Choo & Shiragur, 2023)). *For any DAG $G = (V, E)$ and intervention sets $\mathcal{A}, \mathcal{B} \subseteq 2^V$,*

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

Definition 2.4 (Moral DAG). *A DAG G is called a moral DAG if it has no v-structures. So, $\mathcal{E}(G) = \text{skel}(G)$.*

2.2. Verifying sets

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 D), i.e. $\mathcal{E}_{\mathcal{I}}(G^*) = G^*$. Furthermore, if $\mathcal{I}$ is a verifying set for G^*, then so is $\mathcal{I} \cup S$ for any additional intervention $S \subseteq V$. While there may be multiple verifying sets in general, we are often interested in finding one with a minimum size.

Definition 2.5 (Minimum size verifying set). *An intervention set $\mathcal{I} \subseteq 2^V$ 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 $\mathcal{E}_{\mathcal{I}'}(G^*) \ne G^*$ for any $|\mathcal{I}'| < |\mathcal{I}|$.*

---

[2] A partially directed graph is a *chain graph* if it does *not* contain any partially directed cycles where all directed arcs point in the same direction along the cycle.

[3] A chordal graph is a graph where every cycle of length at least 4 has an edge that is not part of the cycle but connects two vertices of the cycle; see (Blair & Peyton, 1993) for an introduction.

[4] The notation $A \,\dot{\cup}\, B$ denotes disjoint union of sets A and B.

For bounded size interventions, the *minimum verification number* $\nu_k(G)$ denotes the size of the minimum size verifying set for any DAG $G \in [G^*]$; we write $\nu_1(G)$ for atomic interventions. That is, any revealed arc directions when performing interventions on $\mathcal{E}(G^*)$ respects G. (Choo et al., 2022) tells us that it is necessary and sufficient to intervene on a minimum vertex cover of the covered edges $\mathcal{C}(G)$ in order to verify a DAG G, and that $\nu_1(G)$ is efficiently computable given G since $\mathcal{C}(G)$ induces a forest.

Theorem 2.6 ((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.*

For any DAG G, we use $\mathcal{V}(G) \subseteq 2^V$ to denote the set of all *atomic* verifying sets for G. That is, each *atomic* intervention set in $\mathcal{V}(G)$ is a minimum vertex cover of $\mathcal{C}(G)$.

2.3. Adaptive search using ideal interventions

Adaptive search algorithms have been studied in earnest (He & Geng, 2008; Hauser & Bühlmann, 2014; Shanmugam et al., 2015; Squires et al., 2020; Choo et al., 2022; Choo & Shiragur, 2023) as they can use significantly less interventions than non-adaptive counterparts.[5]

Most recently, (Choo et al., 2022) gave an efficient algorithm for computing adaptive interventions with provable approximation guarantees on general graphs.

Theorem 2.7 ((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 $k > 1$.

Meanwhile, in the context of local causal graph discovery where one is interested in only learning a *subset* of causal relationships, the `SubsetSearch` algorithm of (Choo & Shiragur, 2023) incurs a multiplicative overhead that scales logarithmically with the number of relevant nodes when orienting edges within a node-induced subgraph.

Definition 2.8 (Relevant nodes). *Fix a DAG $G^* = (V, E)$ and arbitrary subset $V' \subseteq V$. For any intervention set $\mathcal{I} \subseteq 2^V$ 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 $\mathcal{E}_{\mathcal{I}}(G^*)[V']$.*
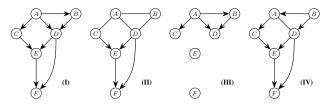

Figure 1. **(I)** Ground truth DAG G^*; **(II)** Observational essential graph $\mathcal{E}(G^*)$ where $C \to E \leftarrow D$ is a v-structure and Meek rules orient arcs $D \to F$ and $E \to F$; **(III)** $G^{\emptyset} = G[E \setminus R(G, \emptyset)]$ where oriented arcs in $\mathcal{E}(G^*)$ are removed from G^*; **(IV)** MPDAG $\tilde{G} \in [G^*]$ incorporating the following partial order advice ($S_1 = \{B\}, S_2 = \{A, D\}, S_3 = \{C, E, F\}$), which can be converted to required arcs $B \to A$ and $B \to D$. Observe that $A \to C$ is oriented by Meek R1 via $B \to A \sim C$, the arc $A \sim D$ is still unoriented, the arc $B \to A$ disagrees with G^*, and there are two possible DAGs consistent with the resulting MPDAG.

For an example of relevant nodes, see Figure 1: For the subset $V' = \{A, C, D, E, F\}$ in (II), only $\{A, C, D\}$ are relevant since incident edges to E and F are all oriented.

Theorem 2.9 ((Choo & Shiragur, 2023)). *Fix an unknown underlying DAG G^*. Given an interventional essential graph $\mathcal{E}_{\mathcal{I}}(G^*)$, node-induced subgraph H with relevant nodes $\rho(\mathcal{I}, V(H))$ and intervention set bound $k \geq 1$, there is a deterministic polynomial time algorithm that computes an intervention set $\mathcal{I}$ adaptively such that $\mathcal{E}_{\mathcal{I} \cup \mathcal{I}'}(G^*)[V(H)] = G^*[V(H)]$, and $|\mathcal{I}'|$ has size*
1. $\mathcal{O}(\log(|\rho(\mathcal{I}, V(H))|) \cdot \nu_1(G^*))$ when $k = 1$
2. $\mathcal{O}(\log(|\rho(\mathcal{I}, V(H))|) \cdot \log(k) \cdot \nu_k(G^*))$ when $k > 1$.

Note that $k = 1$ refers to the setting of atomic interventions and we always have $0 \leq |\rho(\mathcal{I}, V(H))| \leq n$.

2.4. Expert advice in causal graph discovery

There are three main types of information that a domain expert may provide (e.g. see the references given in Section 1):

(I) Required parental arcs: $X \to Y$

(II) Forbidden parental arcs: $X \not\to Y$

(III) Partial order or tiered knowledge: A partition of the n variables into $1 \leq t \leq n$ sets $S_1, \ldots, S_t$ such that variables in S_i *cannot come after* S_j, for all $i < j$.

In the context of orienting unoriented $X \sim Y$ edges in an essential graph, it suffices to consider only information of type (I): $X \not\to Y$ implies $Y \to X$, and a partial order can be converted to a collection of required parental arcs.[6]

---

[5]If the essential graph $\mathcal{E}(G^*)$ is a path of n nodes, then non-adaptive algorithms need $\Omega(n)$ atomic interventions to recover G^* while $\mathcal{O}(\log n)$ atomic interventions suffices for adaptive search.

[6]For every edge $X \sim Y$ with $X \in S_i$ and $Y \in S_j$, enforce the required parental arc $X \to Y$ if and only if $i < j$.

Maximally oriented partially directed acyclic graphs (MPDAGs), a refinement of essential graphs under additional causal information, are often used to model such expert advice and there has been a recent growing interest in understanding them better (Perkovic et al., 2017; Perkovic, 2020; Guo & Perkovic, 2021). MPDAGs are obtained by orienting additional arc directions in the essential graph due to background knowledge, and then applying Meek rules. See Figure 1 for an example.

2.5. Other related work

Causal Structure Learning Algorithms for causal structure learning can be grouped into three broad categories, constraint-based, score-based, and Bayesian. Previous works on the first two approaches are described in Appendix C. In Bayesian methods, a prior distribution is assumed on the space of all structures, and the posterior is updated as more data come in. Heckerman (1995) was one of the first works on learning from interventional data in this context, which spurred a series of papers (e.g. Heckerman et al. (1995); Cooper & Yoo (1999); Friedman & Koller (2000); Heckerman et al. (2006)). Research on active experimental design for causal structure learning with Bayesian updates was initiated by Tong & Koller (2000; 2001) and Murphy (2001). Masegosa & Moral (2013) considered a combination of Bayesian and constraint-based approaches. Cho et al. (2016) and Agrawal et al. (2019) have used active learning and Bayesian updates to help recover biological networks. While possibly imperfect expert advice may be used to guide the prior in the Bayesian approach, the works mentioned above do not provide rigorous guarantees about the number of interventions performed or about optimality, and so they are not directly comparable to our results here.

Algorithms with predictions Learning-augmented algorithms have received significant attention since the seminal work of Lykouris & Vassilvitskii (2021), where they investigated the online caching problem with predictions. Based on that model, Purohit et al. (2018) proposed algorithms for the ski-rental problem as well as non-clairvoyant scheduling. Subsequently, Gollapudi & Panigrahi (2019), Wang et al. (2020), and Angelopoulos et al. (2020) improved the initial results for the ski-rental problem. Several works, including (Rohatgi, 2020; Antoniadis et al., 2020a; Wei, 2020), improved the initial results regarding the caching problem. Scheduling problems with machine-learned advice have been extensively studied in the literature (Lattanzi et al., 2020; Bamas et al., 2020a; Antoniadis et al., 2022). There are also results for augmenting classical data structures with predictions (e.g. indexing (Kraska et al., 2018) and Bloom filters (Mitzenmacher, 2018)), online selection and matching problems (Antoniadis et al., 2020b; Dütting et al., 2021), online TSP (Bernardini et al., 2022; Gouleakis et al., 2023),

and a more general framework of online primal-dual algorithms (Bamas et al., 2020b).

In the above line of work, the extent to which the predictions are helpful in the design of the corresponding online algorithms, is quantified by the following two properties. The algorithm is called (i) *α-consistent* if it is *α-competitive* with no prediction error and (ii) *β-robust* if it is *β-competitive* with any prediction error. In the language of learning augmented algorithms or algorithms with predictions, our causal graph discovery algorithm is 1-consistent and $\mathcal{O}(\log n)$-robust when competing against the verification number $\nu_1(G^*)$, the minimum number of interventions necessary needed to recover G^*. Note that even with arbitrarily bad advice, our algorithm uses asymptotically the same number of interventions incurred by the best-known advice-free adaptive search algorithm (Choo et al., 2022).

3. Results

Our exposition here focuses on interpreting and contextualizing our main results while deferring technicalities to Section 4. We first focus on the setting where the advice is a fully oriented DAG $\widetilde{G} \in [G^*]$ within the Markov equivalence class $[G^*]$ of the true underlying causal graph G^*, and explain in Appendix E how to handle the case of partial advice. Full proofs are provided in the appendix.

3.1. Structural property of verification numbers

We begin by stating a structural result about verification numbers of DAGs within the same Markov equivalence class (MEC) that motivates the definition of a metric between DAGs in the same MEC our algorithmic guarantees (Theorem 3.5) are based upon.

Theorem 3.1. *For any DAG G^* with MEC $[G^*]$, we have that $\max_{G \in [G^*]} \nu_1(G) \leq 2 \cdot \min_{G \in [G^*]} \nu_1(G)$.*

Theorem 3.1 is the first known result relating the minimum and maximum verification numbers of DAGs given a fixed MEC. The next result tells us that the ratio of two is tight.

Lemma 3.2 (Tightness of Theorem 3.1). *There exist DAGs G_1 and G_2 from the same MEC with $\nu_1(G_1) = 2 \cdot \nu_1(G_2)$.*

Theorem 3.1 tells us that we can blindly intervene on any minimum verifying set $\widetilde{V} \in \mathcal{V}(\widetilde{G})$ of any given advice DAG $\widetilde{G}$ while incurring only at most a constant factor of 2 more interventions than the minimum verification number $\nu(G^*)$ of the unknown ground truth DAG G^*.

3.2. Adaptive search with imperfect DAG advice

Recall the definition of r-hop from Section 2. To define the quality of the advice DAG $\widetilde{G}$, we first define the notion of *min-hop-coverage* which measures how "far" a given

verifying set of $\widetilde{G}$ is from the set of covered edges of G^*.

Definition 3.3 (Min-hop-coverage). Fix a DAG G^* with MEC $[G^*]$ and consider any DAG $\widetilde{G} \in [G^*]$. For any minimum verifying set $\widetilde{V} \in \mathcal{V}(\widetilde{G})$, we define the *min-hop-coverage* $h(G^*, \widetilde{V}) \in \{0, 1, 2, \ldots, n\}$ as the minimum number of hops such that *both* endpoints of covered edges $\mathcal{C}(G^*)$ of G^* belong in $N^{h(G^*, \widetilde{V})}_{\text{skel}(\mathcal{E}(G^*))}(\widetilde{V})$.

Using min-hop-coverage, we now define a quality measure $\psi(G^*, \widetilde{G})$ for DAG $\widetilde{G} \in [G^*]$ as an advice for DAG G^*.

Definition 3.4 (Quality measure). *Fix a DAG G^* with MEC $[G^*]$ and consider any DAG $\widetilde{G} \in [G^*]$. We define $\psi(G^*, \widetilde{G})$ as follows:*

$$\psi(G^*, \widetilde{G}) = \max_{\widetilde{V} \in \mathcal{V}(\widetilde{G})} \left| \rho\left(\widetilde{V}, N^{h(G^*, \widetilde{V})}_{\text{skel}(\mathcal{E}(G^*))}(\widetilde{V}) \right) \right|$$

By definition, $\psi(G^*, G^*) = 0$ and $\max_{G \in [G^*]} \psi(G^*, G) \leq n$. In words, $\psi(G^*, \widetilde{G})$ only counts the relevant nodes within the min-hop-coverage neighborhood after intervening on the *worst* possible verifying set $\widetilde{V}$ of $\widetilde{G}$. We define ψ via the worst set because any search algorithm *cannot* evaluate $h(G^*, \widetilde{V})$, since G^* is unknown, and can only consider an *arbitrary* $\widetilde{V} \in \mathcal{V}(\widetilde{G})$. See Figure 2 for an example.

Our main result is that it is possible to design an algorithm that leverages an advice DAG $\widetilde{G} \in [G^*]$ and performs interventions to fully recover an unknown underlying DAG G^*, whose performance depends on the advice quality $\psi(G^*, \widetilde{G})$. Our search algorithm only knows $\mathcal{E}(G^*)$ and $\widetilde{G} \in [G^*]$ but knows neither $\psi(G^*, \widetilde{G})$ nor $\nu(G^*)$.

Theorem 3.5. *Fix an essential graph $\mathcal{E}(G^*)$ with an unknown underlying ground truth DAG G^*. Given an advice graph $\widetilde{G} \in [G^*]$ and intervention set bound $k \geq 1$, there exists a deterministic polynomial time algorithm (Algorithm 1) that computes an intervention set $\mathcal{I}$ adaptively such that $\mathcal{E}_{\mathcal{I}}(G^*) = G^*$, and $|\mathcal{I}|$ has size*
1. $\mathcal{O}(\max\{1, \log \psi(G^*, \widetilde{G})\} \cdot \nu_1(G^*))$ when $k = 1$
2. $\mathcal{O}(\max\{1, \log \psi(G^*, \widetilde{G})\} \cdot \log k \cdot \nu_k(G^*))$ when $k > 1$.

Consider first the setting of $k = 1$. Observe that when the advice is perfect (i.e. $\widetilde{G} = G^*$), we use $\mathcal{O}(\nu(G^*))$ interventions, i.e. a constant multiplicative factor of the minimum number of interventions necessary. Meanwhile, even with low quality advice, we still use $\mathcal{O}(\log n \cdot \nu(G^*))$ interventions, asymptotically matching the best known guarantees for adaptive search without advice. To the best of our knowledge, Theorem 3.5 is the first known result that principally employs imperfect expert advice with provable guarantees in the context of causal graph discovery via interventions.

Consider now the setting of bounded size interventions where $k > 1$. The reason why we can obtain such a result is precisely because of our algorithmic design: we deliberately
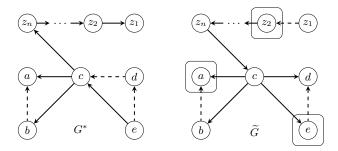

Figure 2. Consider the moral DAGs G^* and $\widetilde{G} \in [G^*]$ on $n + 5$ nodes, where dashed arcs represent the covered edges in each DAG. A minimum sized verifying set $\widetilde{V} = \{a, e, z_2\} \in \mathcal{V}(\widetilde{G})$ of $\widetilde{G}$ is given by the boxed vertices on the right. As $N^1_{\text{skel}(G^*)}(\widetilde{V}) = \{a, b, c, d, e, z_1, z_2, z_3\}$ includes both endpoints of all covered edges of G^*, we see that $h(G^*, \widetilde{V}) = 1$. Intervening on $\widetilde{V} = \{a, e, z_2\}$ in G^* orients the arcs $b \to a \leftarrow c$, $c \leftarrow e \to d$, and $z_3 \to z_2 \to z_1$ respectively which then triggers Meek R1 to orient $c \to b$ via $e \to c \sim b$ and to orient $z_4 \to z_3$ via $e \to c \to \ldots \to z_4 \sim z_3$ (after a few invocations of R1), so $\{a, b, e, z_1, z_2, z_3\}$ will *not* be relevant nodes in $\mathcal{E}_{\widetilde{V}}(G^*)$. Meanwhile, the edge $c \sim d$ remains unoriented in $\mathcal{E}_{\widetilde{V}}(G^*)$, so $\rho(\widetilde{V}, N^1_{\text{skel}(G^*)}(\widetilde{V})) = |\{c, d\}| = 2$. One can check that $\psi(G^*, \widetilde{G}) = 2$ while n could be arbitrarily large. On the other hand, observe that ψ is *not* symmetric: in the hypothetical situation where we use G^* as an advice for $\widetilde{G}$, the min-hop-coverage has to extend along the chain $z_1 \sim \ldots \sim z_n$ to reach $\{z_1, z_2\}$, so $h(G^*, V^*) \approx n$ and $\psi(\widetilde{G}, G^*) \approx n$ since the entire chain remains unoriented with respect to any $V^* \in \mathcal{V}(G^*)$.

designed an algorithm that invokes `SubsetSearch` as a black-box subroutine. Thus, the bounded size guarantees of `SubsetSearch` given by Theorem 2.9 carries over to our setting with a slight modification of the analysis.

4. Techniques

Here, we discuss the high-level technical ideas and intuition behind how we obtain our adaptive search algorithm with imperfect DAG advice. See the appendix for full proofs; in particular, see Appendix F for an overview of Theorem 3.1.

For brevity, we write ψ to mean $\psi(G^*, \widetilde{G})$ and drop the subscript $\text{skel}(\mathcal{E}(G^*))$ of r-hop neighborhoods in this section. We also focus our discussion to the atomic interventions. Our adaptive search algorithm (Algorithm 1) uses `SubsetSearch` as a subroutine.

We begin by observing that `SubsetSearch`$(\mathcal{E}(G^*), A)$ fully orients $\mathcal{E}(G^*)$ into G^* if the covered edges of G^* lie within the node-induced subgraph induced by A.

Lemma 4.1. *Fix a DAG $G^* = (V, E)$ and let $V' \subseteq V$ be any subset of vertices. Suppose $\mathcal{I}_{V'} \subseteq V$ is the set of nodes intervened by* `SubsetSearch`$(\mathcal{E}(G^*), V')$. *If*

$\mathcal{C}(G^*) \subseteq E(G^*[V'])$, then $\mathcal{E}_{\mathcal{I}_{V'}}(G^*) = G^*$.

Motivated by Lemma 4.1, we design Algorithm 1 to repeatedly invoke SubsetSearch on node-induced subgraphs $N^r(\widetilde{V})$, starting from an *arbitrary* verifying set $\widetilde{V} \in \mathcal{V}(\widetilde{G})$ and for *increasing* values of r.

For $i \in \mathbb{N} \cup \{0\}$, let us denote $r(i) \in \mathbb{N} \cup \{0\}$ as the value of r in the i-th invocation of SubsetSearch, where we insist that $r(0) = 0$ and $r(j) > r(j-1)$ for any $j \in \mathbb{N}$. Note that $r = 0$ simply implies that we intervene on the verifying set $\widetilde{V}$, which only incurs $\mathcal{O}(\nu_1(G^*))$ interventions due to Theorem 3.1. Then, we can appeal to Lemma 4.1 to conclude that $\mathcal{E}(G^*)$ is completely oriented into G^* in the t-th invocation if $r(t) \geq h(G^*, \widetilde{V})$.

While the high-level subroutine invocation idea seems simple, one needs to invoke SubsetSearch at *suitably chosen intervals* in order to achieve our theoretical guarantees we promise in Theorem 3.5. We now explain how to do so in three successive attempts while explaining the algorithmic decisions behind each modification introduced.

As a reminder, we *do not* know G^* and thus *do not* know $h(G^*, \widetilde{V})$ for any verifying set $\widetilde{V} \in \mathcal{V}(\widetilde{G})$ of $\widetilde{G} \in [G^*]$.

NAIVE ATTEMPT: INVOKE FOR $r = 0, 1, 2, 3, \ldots$

The most straightforward attempt would be to invoke SubsetSearch repeatedly each time we increase r by 1 until the graph is fully oriented – in the worst case, $t = h(G^*, \widetilde{V})$. However, this may cause us to incur way too many interventions. Suppose there are n_i relevant nodes in the i-th invocation. Using Theorem 2.9, one can only argue that the overall number interventions incurred is $\mathcal{O}(\sum_{i=0}^{t} \log n_i \cdot \nu(G^*))$. However, $\sum_i \log n_i$ could be significantly larger than $\log(\sum_i n_i)$ in general, e.g. $\log 2 + \ldots + \log 2 = (n/2) \cdot \log 2 \gg \log n$. In fact, if G^* was a path on n vertices $v_1 \to v_2 \to \ldots \to v_n$ and $\widetilde{G} \in [G^*]$ misleads us with $v_1 \leftarrow v_2 \leftarrow \ldots \leftarrow v_n$, then this approach incurs $\Omega(n)$ interventions in total.

TWEAK 1: ONLY INVOKE PERIODICALLY

Since Theorem 2.9 provides us a logarithmic factor in the analysis, we could instead consider only invoking SubsetSearch after the number of nodes in the subgraph *increases by a polynomial factor*. For example, if we invoked SubsetSearch with n_i previously, then we will wait until the number of relevant nodes surpasses n_i^2 before invoking SubsetSearch again, where we define $n_0 \geq 2$ for simplicity. Since $\log n_i \geq 2 \log n_{i-1}$, we can see via an inductive argument that the number of interventions used in the final invocation will dominate the total number of interventions used so far: $n_t \geq 2 \log n_{t-1} \geq \log n_{t-1} + 2 \log n_{t-2} \geq \ldots \geq \sum_{i=0}^{t-1} \log n_i$. Since $n_i \leq n$ for any i, we can already prove that $\mathcal{O}(\log n \cdot \nu_1(G^*))$ inter-
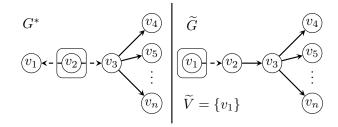

Figure 3. Consider the ground truth DAG G^* with unique minimum verifying set $\{v_2\}$ and an advice DAG $\widetilde{G} \in [G^*]$ with chosen minimum verifying set $\widetilde{V} = \{v_1\}$. So, $h(G^*, \widetilde{V}) = 1$ and ideally we want to argue that our algorithm uses a constant number of interventions. Without tweak 2 and $n_0 = 2$, an algorithm that increases hop radius until the number of relevant nodes is squared will *not* invoke SubsetSearch until $r = 3$ because $\rho(\widetilde{V}, N^1) = 1 < n_0^2$ and $\rho(\widetilde{V}, N^2) = 2 < n_0^2$. However, $\rho(\widetilde{V}, N^3) = n - 1$ and we can only conclude that the algorithm uses $\mathcal{O}(\log n)$ interventions by invoking SubsetSearch on a subgraph on $n - 1$ nodes.

ventions suffice, matching the advice-free bound of Theorem 2.7. However, this approach and analysis does *not* take into account the quality of $\widetilde{G}$ and is *insufficient* to relate n_t with the advice measure ψ.

TWEAK 2: ALSO INVOKE ONE ROUND BEFORE

Suppose the final invocation of SubsetSearch is on $r(t)$-hop neighborhood while incurring $\mathcal{O}(\log n_t \cdot \nu_1(G^*))$ interventions. This means that $\mathcal{C}(G^*)$ lies within $N^{r(t)}(\widetilde{V})$ but *not* within $N^{r(t-1)}(\widetilde{V})$. That is, $N^{r(t-1)}(\widetilde{V}) \subsetneq N^{h(G^*, \widetilde{V})}(\widetilde{V}) \subseteq N^{r(t)}(\widetilde{V})$. While this tells us that $n_{t-1} \leq |\rho(\widetilde{V}, N^{r(t-1)}(\widetilde{V}))| < |\rho(\widetilde{V}, N^{h(G^*, \widetilde{V})}(\widetilde{V}))| = \psi$, what we want is to conclude that $n_t \in \mathcal{O}(\psi)$. Unfortunately, even when $\psi = r(t-1) + 1$, it could be the case that $|\rho(\widetilde{V}, N^{h(G^*, \widetilde{V})}(\widetilde{V}))| \ll |N^{r(t)}(\widetilde{V})|$ as the number of relevant nodes could blow up within a single hop (see Figure 3). To control this potential blow up in the analysis, we can introduce the following technical fix: whenever we want to invoke SubsetSearch on $r(i)$, first invoke SubsetSearch on $r(i) - 1$ and terminate earlier if the graph is already fully oriented into G^*.

PUTTING TOGETHER

Algorithm 1 presents our full algorithm where the inequality $\rho(\mathcal{I}_i, N^r_{\text{skel}(\mathcal{E}(G^*))}(\widetilde{V})) \geq n_i^2$ corresponds to the first tweak while the terms C_i and C_i' correspond to the second tweak.

In Appendix H, we explain why our algorithm (Algorithm 1) is simply the classic "binary search with prediction"[7] when the given essential graph $\mathcal{E}(G^*)$ is an undirected path. So,

---

[7]e.g. see https://en.wikipedia.org/wiki/Learning_augmented_algorithm#Binary_search

another way to view our result is a *generalization* that works on essential graphs of arbitrary moral DAGs.

Algorithm 1 Adaptive search algorithm with advice.

1: **Input**: Essential graph $\mathcal{E}(G^*)$, advice DAG $\widetilde{G} \in [G^*]$, intervention size $k \in \mathbb{N}$
2: **Output**: An intervention set $\mathcal{I}$ such that each intervention involves at most k nodes and $\mathcal{E}_{\mathcal{I}}(G^*) = G^*$.
3: Let us write SS to mean SubsetSearch.
4: Let $\widetilde{V} \in \mathcal{V}(\widetilde{G})$ be any atomic verifying set of $\widetilde{G}$.
5: **if** $k = 1$ **then**
6:     Define $\mathcal{I}_0 = \widetilde{V}$ as an atomic intervention set.
7: **else**
8:     Define $k' = \min\{k, |\widetilde{V}|/2\}$, $a = \lceil |\widetilde{V}|/k' \rceil \geq 2$, and $\ell = \lceil \log_a |C| \rceil$. Compute labelling scheme on $\widetilde{V}$ with $(|\widetilde{V}|, k, a)$ via Lemma 4.3 and define $\mathcal{I}_0 = \{S_{x,y}\}_{x \in [\ell], y \in [a]}$, where $S_{x,y} \subseteq \widetilde{V}$ is the subset of vertices whose x^{th} letter in the label is y.
9: **end if**
10: Intervene on $\mathcal{I}_0$ and initialize $r \leftarrow 0$, $i \leftarrow 0$, $n_0 \leftarrow 2$.
11: **while** $\mathcal{E}_{\mathcal{I}_i}(G^*)$ still has undirected edges **do**
12:     **if** $\rho(\mathcal{I}_i, N^r_{\text{skel}(\mathcal{E}(G^*))}(\widetilde{V})) \geq n_i^2$ **then**
13:         Increment $i \leftarrow i + 1$ and record $r(i) \leftarrow r$.
14:         Update $n_i \leftarrow \rho(\mathcal{I}_i, N^r_{\text{skel}(\mathcal{E}(G^*))}(\widetilde{V}))$
15:         $C_i \leftarrow \text{SS}(\mathcal{E}_{\mathcal{I}_i}(G^*), N^{r-1}_{\text{skel}(\mathcal{E}(G^*))}(\widetilde{V}), k)$
16:         **if** $\mathcal{E}_{\mathcal{I}_{i-1} \cup C_i}(G^*)$ still has undirected edges **then**
17:             $C'_i \leftarrow \text{SS}(\mathcal{E}_{\mathcal{I}_{i-1} \cup C_i}(G^*), N^r_{\text{skel}(\mathcal{E}(G^*))}(\widetilde{V}), k)$
18:             Update $\mathcal{I}_i \leftarrow \mathcal{I}_{i-1} \cup C_i \cup C'_i$.
19:         **else**
20:             Update $\mathcal{I}_i \leftarrow \mathcal{I}_{i-1} \cup C_i$.
21:         **end if**
22:     **end if**
23:     Increment $r \leftarrow r + 1$.
24: **end while**
25: **return** $\mathcal{I}_i$

For bounded size interventions, we rely on the following known results.

Theorem 4.2 (Theorem 12 of (Choo et al., 2022)). *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. If $\nu_1(G) = \ell$, then $\nu_k(G) \geq \lceil \frac{\ell}{k} \rceil$ and there exists a polynomial time algo. to compute a bounded size intervention set $\mathcal{I}$ of size $|\mathcal{I}| \leq \lceil \frac{\ell}{k} \rceil + 1$.*

Lemma 4.3 (Lemma 1 of (Shanmugam et al., 2015)). *Let (n, k, a) be parameters where $k \leq n/2$. There exists a polynomial time labeling scheme that produces distinct ℓ length labels for all elements in $[n]$ using letters from the integer alphabet $\{0\} \cup [a]$ where $\ell = \lceil \log_a n \rceil$. Further, in every digit (or position), any integer letter is used at most $\lceil n/a \rceil$ times. This labelling scheme is a separating system: for any $i, j \in [n]$, there exists some digit $d \in [\ell]$ where the labels of i and j differ.*

Theorem 4.2 enables us to easily relate $\nu_1(G)$ with $\nu_k(G)$ while Lemma 4.3 provides an efficient labelling scheme to partition a set of n nodes into a set $S = \{S_1, S_2, \ldots\}$ of bounded size sets, each S_i involving at most k nodes. By invoking Lemma 4.3 with $a \approx n'/k$ where n' is related to $\nu_1(G)$, we see that $|S| \approx \frac{n'}{k} \cdot \log k$. As $\nu_k(G) \approx \nu_1(G)/k$, this is precisely why the bounded intervention guarantees in Theorem 2.7, Theorem 2.9 and Theorem 3.5 have an additional multiplicative $\log k$ factor.

5. Empirical validation

While our main contributions are theoretical, we also performed some experiments to empirically validate that our algorithm is practical, outperforms the advice-free baseline when the advice quality is good, and still being at most a constant factor worse when the advice is poor.

Motivated by Theorem 2.3, we experimented on synthetic moral DAGs from Wienöbst et al. (2021b): For each undirected chordal graph, we use the uniform sampling algorithm of Wienöbst et al. (2021b) to uniformly sample 1000 moral DAGs $\widetilde{G}_1, \ldots, \widetilde{G}_{1000}$ and randomly choose one of them as G^*. Then, we give $\{(\mathcal{E}(G^*), \widetilde{G}_i)\}_{i \in [1000]}$ as input to Algorithm 1.

Figure 4 shows one of the experimental plots; more detailed experimental setup and results are given in Appendix I. On the X-axis, we plot $\psi(G^*, \widetilde{V}) = \left| \rho\left(\widetilde{V}, N^{h(G^*, \widetilde{V})}_{\text{skel}(\mathcal{E}(G^*))}(\widetilde{V})\right) \right|$, which is a *lower bound* and proxy[8] for $\psi(G^*, \widetilde{G})$. On the Y-axis, we aggregate advice DAGs based on their quality measure and also show (in dashed lines) the empirical distribution of quality measures of all DAGs within the Markov equivalence class.

As expected from our theoretical analyses, we see that the number of interventions by our advice search starts from $\nu_1(G^*)$, is lower than advice-free search of (Choo et al., 2022) when $\psi(G^*, \widetilde{V})$ is low, and gradually increases as the advice quality degrades. Nonetheless, the number of interventions used is always theoretically bounded below $\mathcal{O}(\psi(G^*, \widetilde{V}) \cdot \nu_1(G^*))$; we do not plot $\psi(G^*, \widetilde{V}) \cdot \nu_1(G^*)$ since plotting it yields a "squashed" graph as the empirical counts are significantly smaller. In this specific graph instance, Figure 4 suggests that our advice search outperforms its advice-free counterpart when given an advice DAG $\widetilde{G}$ that is better than $\sim 40\%$ of all possible DAGs consistent with the observational essential graph $\mathcal{E}(G^*)$.

---

[8]We do not know if there is an efficient way to compute $\psi(G^*, \widetilde{G})$ besides the naive (possibly exponential time) enumeration over all possible minimum verifying sets.
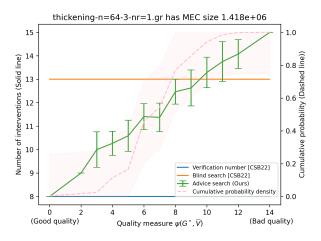
Figure 4. Experimental plot for one of the synthetic graphs G^*, with respect to $1000 \ll |[G^*]| \approx 1.4 \times 10^6$ uniformly sampled advice DAGs $\widetilde{G}$ from the MEC $[G^*]$. The solid lines indicate the number of atomic interventions used while the dotted lines indicate the empirical cumulative probability density of $\widetilde{G}$. The true cumulative probability density lies within the shaded area with probability at least 0.99 (see Appendix I for details).

6. Conclusion and discussion

In this work, we gave the first result that utilizes imperfect advice in the context of causal discovery. We do so in a way that the performance (i.e. the number of interventions in our case) does not degrade significantly even when the advice is inaccurate, which is consistent with the objectives of learning-augmented algorithms. Specifically, we show a smooth bound that matches the number of interventions needed for verification of the causal relationships in a graph when the advice is completely accurate and also depends logarithmically on the distance of the advice to the ground truth. This ensures robustness to "bad" advice, the number of interventions needed is asymptotically the same as in the case where no advice is available.

Our results do rely on the widely-used assumptions of sufficiency and faithfulness as well as access to ideal itervensions; see Appendix A for a more detailed discussion. Since wrong causal conclusions may be drawn when these assumptions are violated by the data, thus it is of great interest to remove/weaken these assumptions while maintaining strong theoretical guarantees in future work.

6.1. Interesting future directions to explore

Partial advice In Appendix E, we explain why having a DAG $\widetilde{G}$ as advice may not always be possible and explain how to extend our results to the setting of *partial advice* by considering the worst case DAG consistent with the given partial advice $\mathcal{A}$. The question is

whether one can design and analyze a better algorithm than a trivial $\max_{\widetilde{G} \in \mathcal{A}}$. For example, maybe one could pick $\widetilde{G} = \arg\min_{G \in \mathcal{A}} \max_{H \in [G^*]} \psi(H, G)$? The motivation is as follows: If $[G^*]$ is a disc in $\mathbb{R}^2$ and ψ is the Euclidean distance, then $\widetilde{G}$ should be the point within $\mathcal{A}$ that is closest to the center of the disc. Note that we can only optimize with respect to $\max_{H \in [G^*]}$ because we do not actually know G^*. It remains to be seen if such an object can be efficiently computed and whether it gives a better bound than $\max_{\widetilde{G} \in \mathcal{A}}$.

Incorporating expert confidence The notion of "confidence level" and "correctness" of an advice are orthogonal issues – an expert can be confidently wrong. In this work, we focused on the case where the expert is fully confident but may be providing imperfect advice. It is an interesting problem to investigate how to principally handle both issues simultaneously; for example, what if the advice is not a DAG $\widetilde{G} \in [G^*]$ in the essential graph but a distribution over all DAGs in $[G^*]$? Bayesian ideas may apply here.

Better analysis? Empirically, we see that the log factor is a rather loose upper bound both for blind search and advice search. *Can there be a tighter analysis?* (Choo et al., 2022) tells us that $\Omega(\log n \cdot \nu_1(G^*))$ is unavoidable when $\mathcal{E}(G^*)$ is a path on n vertices with $\nu_1(G^*) = 1$ but this is a special class of graphs. What if $\nu_1(G^*) > 1$? Can we give tighter bounds in other graph parameters? Furthermore, in some preliminary testing, we observed that implementing tweak 2 or ignoring it yield similar empirical performance and we wonder if there is a tighter analysis without tweak 2 that has similar guarantees.

Acknowledgements

References

Agrawal, R., Squires, C., Yang, K., Shanmugam, K., and Uhler, C. ABCD-Strategy: Budgeted Experimental Design for Targeted Causal Structure Discovery. In *International Conference on Artificial Intelligence and Statistics*, pp. 3400–3409. PMLR, 2019.

Andersen, H. When to expect violations of causal faithful-

ness and why it matters. *Philosophy of Science*, 80(5): 672–683, 2013.

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.

Andrews, B., Spirtes, P., and Cooper, G. F. On the Completeness of Causal Discovery in the Presence of Latent Confounding with Tiered Background Knowledge. In *International Conference on Artificial Intelligence and Statistics*, pp. 4002–4011. PMLR, 2020.

Angelopoulos, S., Dürr, C., Jin, S., Kamali, S., and Renault, M. Online Computation with Untrusted Advice. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

Antoniadis, A., Coester, C., Elias, M., Polak, A., and Simon, B. Online Metric Algorithms with Untrusted Predictions. In *International Conference on Machine Learning*, pp. 345–355. PMLR, 2020a.

Antoniadis, A., Gouleakis, T., Kleer, P., and Kolev, P. Secretary and Online Matching Problems with Machine Learned Advice. *Advances in Neural Information Processing Systems*, 33:7933–7944, 2020b.

Antoniadis, A., Jabbarzade, P., and Shahkarami, G. A Novel Prediction Setup for Online Speed-Scaling. In *18th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

Bamas, É., Maggiori, A., Rohwedder, L., and Svensson, O. Learning Augmented Energy Minimization via Speed Scaling. *Advances in Neural Information Processing Systems*, 33:15350–15359, 2020a.

Bamas, E., Maggiori, A., and Svensson, O. The Primal-Dual method for Learning Augmented Algorithms. *Advances in Neural Information Processing Systems*, 33:20083–20094, 2020b.

Bernardini, G., Lindermayr, A., Marchetti-Spaccamela, A., Megow, N., Stougie, L., and Sweering, M. A Universal Error Measure for Input Predictions Applied to Online Graph Problems. In *Advances in Neural Information Processing Systems*, 2022.

Blair, J. R. S. and Peyton, B. W. An introduction to chordal graphs and clique trees. In *Graph theory and sparse matrix computation*, pp. 1–29. Springer, 1993.

Canonne, C. L. A short note on learning discrete distributions. *arXiv preprint arXiv:2002.11457*, 2020.

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.

Chickering, D. M. Optimal Structure Identification with Greedy Search. *Journal of Machine Learning Research*, 3:507–554, 2002.

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.

Clause, J., Li, W., and Orso, A. Dytan: A Generic Dynamic Taint Analysis Framework. In *Proceedings of the 2007 international symposium on Software testing and analysis*, pp. 196–206, 2007.

Colombo, D., Maathuis, M. H., Kalisch, M., and Richardson, T. S. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, pp. 294–321, 2012.

Cooper, G. F. and Yoo, C. Causal Discovery from a Mixture of Experimental and Observational Data. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pp. 116–125, 1999.

De Campos, C. P. and Ji, Q. Efficient Structure Learning of Bayesian Networks using Constraints. *The Journal of Machine Learning Research*, 12:663–689, 2011.

Dütting, P., Lattanzi, S., Paes Leme, R., and Vassilvitskii, S. Secretaries with Advice. In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pp. 409–429, 2021.

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.

Fang, Z. and He, Y. IDA with Background Knowledge. In *Conference on Uncertainty in Artificial Intelligence*, pp. 270–279. PMLR, 2020.

Flores, M. J., Nicholson, A. E., Brunskill, A., Korb, K. B., and Mascaro, S. Incorporating expert knowledge when learning Bayesian network structure: A medical case study. *Artificial intelligence in medicine*, 53(3):181–204, 2011.

Friedman, N. and Koller, D. Being Bayesian about Network Structure. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pp. 201–210, 2000.

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.

Gollapudi, S. and Panigrahi, D. Online Algorithms for Rent-or-Buy with Expert Advice. In *International Conference on Machine Learning*, pp. 2319–2327. PMLR, 2019.

Gouleakis, T., Lakis, K., and Shahkarami, G. Learning-Augmented Algorithms for Online TSP on the Line. In *37th AAAI Conference on Artificial Intelligence*. AAAI, 2023.

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.

Guo, R. and Perkovic, E. Minimal Enumeration of All Possible Total Effects in a Markov Equivalence Class. In *International Conference on Artificial Intelligence and Statistics*, pp. 2395–2403. PMLR, 2021.

Hauser, A. and Bühlmann, P. Characterization and Greedy Learning of Interventional Markov Equivalence Classes of Directed Acyclic Graphs. *The Journal of Machine Learning Research*, 13(1):2409–2464, 2012.

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.

Heckerman, D. A Bayesian Approach to Learning Causal Networks. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pp. 285–295, 1995.

Heckerman, D., Geiger, D., and Chickering, D. M. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data . *Machine learning*, 20:197–243, 1995.

Heckerman, D., Meek, C., and Cooper, G. A Bayesian Approach to Causal Discovery. *Innovations in Machine Learning*, pp. 1–28, 2006.

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.

Ikram, M. A., Chakraborty, S., Mitra, S., Saini, S., Bagchi, S., and Kocaoglu, M. Root Cause Analysis of Failures in Microservices through Causal Discovery. In *Advances in Neural Information Processing Systems*, 2022.

Jha, S., Banerjee, S., Tsai, T., Hari, S. K., Sullivan, M. B., Kalbarczyk, Z. T., Keckler, S. W., and Iyer, R. K. ML-based Fault Injection for Autonomous Vehicles: A Case for Bayesian Fault Injection. In *2019 49th annual IEEE/IFIP international conference on dependable systems and networks (DSN)*, pp. 112–124. IEEE, 2019.

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.

Kraska, T., Beutel, A., Chi, E. H., Dean, J., and Polyzotis, N. The Case for Learned Index Structures. In *Proceedings of the 2018 international conference on management of data*, pp. 489–504, 2018.

Lattanzi, S., Lavastida, T., Moseley, B., and Vassilvitskii, S. Online Scheduling via Learned Weights. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1859–1877. SIAM, 2020.

Li, A. and Beek, P. Bayesian Network Structure Learning with Side Constraints. In *International conference on probabilistic graphical models*, pp. 225–236. PMLR, 2018.

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.

Lykouris, T. and Vassilvitskii, S. Competitive Caching with Machine Learned Advice. *Journal of the ACM (JACM)*, 68(4):1–25, 2021.

Masegosa, A. R. and Moral, S. An interactive approach for Bayesian network learning using domain/expert knowledge. *International Journal of Approximate Reasoning*, 54(8):1168–1181, 2013.

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, 1995a. Morgan Kaufmann Publishers Inc. ISBN 1558603859.

Meek, C. Strong completeness and faithfulness in Bayesian networks. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pp. 411–418, 1995b.

Mitzenmacher, M. A Model for Learned Bloom Filters, and Optimizing by Sandwiching. *Advances in Neural Information Processing Systems*, 31, 2018.

Mitzenmacher, M. and Vassilvitskii, S. Algorithms with Predictions. *Communications of the ACM*, 65(7):33–35, 2022.

Murphy, K. P. Active Learning of Causal Bayes Net Structure. Technical report, UC Berkeley, 2001.

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

Perkovic, E. Identifying causal effects in maximally oriented partially directed acyclic graphs. In *Conference on Uncertainty in Artificial Intelligence*, pp. 530–539. PMLR, 2020.

Perkovic, E., Kalisch, M., and Maathuis, M. H. Interpreting and using CPDAGs with background knowledge. In *Proceedings of the 2017 Conference on Uncertainty in Artificial Intelligence (UAI2017)*, pp. ID–120. AUAI Press, 2017.

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.

Purohit, M., Svitkina, Z., and Kumar, R. Improving Online Algorithms via ML Predictions. *Advances in Neural Information Processing Systems*, 31, 2018.

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

Rohatgi, D. Near-Optimal Bounds for Online Caching with Machine Learned Advice. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1834–1845. SIAM, 2020.

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.

Scheines, R., Spirtes, P., Glymour, C., Meek, C., and Richardson, T. he TETAD Project: Constraint Based Aids to Causal Model Specification. *Multivariate Behavioral Research*, 33(1):65–117, 1998.

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

Solus, L., Wang, Y., and Uhler, C. Consistency guarantees for greedy permutation-based causal inference algorithms. *Biometrika*, 108(4):795–814, 2021.

Spirtes, P., Glymour, C. N., Scheines, R., and Heckerman, D. *Causation, Prediction, and Search*. MIT press, 2000.

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.

Tan, C., Jin, Z., Guo, C., Zhang, T., Wu, H., Deng, K., Bi, D., and Xiang, D. NetBouncer: Active Device and Link Failure Localization in Data Center Networks. In *Proceedings of the 16th USENIX Conference on Networked Systems Design and Implementation*, pp. 599–613, 2019.

Tong, S. and Koller, D. Active Learning for Parameter Estimation in Bayesian Networks. *Advances in Neural Information Processing Systems*, 13, 2000.

Tong, S. and Koller, D. Active Learning for Structure in Bayesian Networks. In *International joint conference on artificial intelligence*, volume 17, pp. 863–869. Citeseer, 2001.

Uhler, C., Raskutti, G., Bühlmann, P., and Yu, B. Geometry of the faithfulness assumption in causal inference. *The Annals of Statistics*, pp. 436–463, 2013.

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.

Wang, Q., Aliaga, J. R., Jha, S., Shanmugam, K., Bagehorn, F., Yang, X., Filepp, R., Abe, N., and Shwartz, L. Fault Injection based Interventional Causal Learning for Distributed Applications. In *Innovative Applications of Artificial Intelligence Conference*, 2023.

Wang, S., Li, J., and Wang, S. Online Algorithms for Multi-shop Ski Rental with Machine Learned Advice. *Advances in Neural Information Processing Systems*, 33: 8150–8160, 2020.

Wang, Y., Solus, L., Yang, K., and Uhler, C. Permutation-based Causal Inference Algorithms with Interventions. *Advances in Neural Information Processing Systems*, 30, 2017.

Wei, A. Better and Simpler Learning-Augmented Online Caching. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

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.

Zhang, J. and Spirtes, P. Strong Faithfulness and Uniform Consistency in Causal Inference . In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pp. 632–639, 2002.

Zhou, X., Peng, X., Xie, T., Sun, J., Ji, C., Liu, D., Xiang, Q., and He, C. Latent error prediction and fault localization for microservice applications by learning from system trace logs. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 683–694, 2019.

A. Remark about assumptions

Under *causal sufficiency*, there are no hidden confounders (i.e. unobserved common causes to the observed variables). While causal sufficiency may not always hold, it is still a reasonable assumption to make in certain applications such as studying gene regulatory networks (e.g. see (Wang et al., 2017)).

Faithfulness assumes that independencies that occur in the data do not occur due to "cancellations" in the functional relationships, but rather due to the causal graph structure. It is known (Meek, 1995b; Spirtes et al., 2000) that, under many natural parameterizations and settings, the set of unfaithful parameters for any given causal DAG has zero Lebesgue measure (i.e. faithfulness holds; see also Section 3.2 of (Zhang & Spirtes, 2002) for a discussion about faithfulness). However, one should be aware that the faithfulness assumption may be violated in reality (Andersen, 2013; Uhler et al., 2013), especially in the presence of sampling errors in the finite sample regime.

Ideal interventions assume hard interventions (forcefully setting a variable value) and the ability to obtain as many interventional samples as desired, ensuring that we always recover the directions of all edges cut by interventions. Without this assumption, we may fail to correctly infer some arc directions and our algorithms will only succeed with some success probability.

Our assumption that the given expert advice is consistent with observational essential graph is purely for simplicity and can be removed by deciding which part of the given advice to discard so that the remaining advice is consistent. However, we feel that deciding which part of the inconsistent advice to discard will unnecessarily complicate our algorithmic contributions without providing any useful insights, and thus we made such an assumption.

B. Additional Preliminaries

For any set A, we denote its powerset by 2^A. We write $\{1, \ldots, n\}$ as $[n]$ and hide absolute constant multiplicative factors in n using standard asymptotic notations $\mathcal{O}(\cdot)$, $\Omega(\cdot)$, and $\Theta(\cdot)$. The indicator function $\mathbb{1}_{\text{predicate}}$ is 1 if the predicate is true and 0 otherwise. Throughout, we use G^* to denote the (unknown) ground truth DAG, its Markov equivalence class by $[G^*]$ and the corresponding essential graph by $\mathcal{E}(G^*)$. We write $A \dot\cup B$ and $A \setminus B$ to represent the disjoint union and set difference of two sets A and B respectively.

B.1. Graph basics

We consider partially oriented graphs without parallel edges.

Let $G = (V, E)$ be a graph on $|V| = n$ nodes/vertices where $V(G)$, $E(G)$, and $A(G) \subseteq E(G)$ denote nodes, edges, and arcs of G respectively. The graph G is said to be fully oriented if $A(G) = E(G)$, fully unoriented if $A(G) = \emptyset$, and partially oriented otherwise. For any subset $V' \subseteq V$ and $E' \subseteq E$, we use $G[V']$ and $G[E']$ to denote the node-induced and edge-induced subgraphs respectively. We write $u \sim v$ to denote that two nodes $u, v \in V$ are connected in G, and write $u \to v$ or $u \leftarrow v$ when specifying a certain direction. The *skeleton* $\text{skel}(G)$ refers to the underlying graph where all edges are made undirected. A *v-structure* in G refers to a collection of three distinct vertices $u, v, w \in V$ such that $u \to v \leftarrow w$ and $u \not\sim w$. A directed cycle refers to a sequence of $k \geq 3$ vertices where $v_1 \to v_2 \to \ldots \to v_k \to v_1$. An *acyclic completion / consistent extension* of a partially oriented graph refers to an assignment of edge directions to the unoriented edges $E(G) \setminus A(G)$ such that the resulting fully oriented graph has no directed cycles.

Suppose $G = (V, E)$ is fully unoriented. For vertices $u, v \in V$, subset of vertices $V' \subseteq V$ and integer $r \geq 0$, define $\text{dist}_G(u, v)$ as the shortest path length between u and v, $\text{dist}_G(V', v) = \min_{u \in V'} \text{dist}_G(u, v)$, and $N_G^r(V') = \{v \in V : \text{dist}_G(v, V') \leq r\} \subseteq V$ as the set of vertices that are r-hops away from V', i.e. r-hop neighbors of V'. We omit the subscript G when it is clear from context.

Suppose $G = (V, E)$ is fully oriented. For any vertex $v \in V$, we write $\text{Pa}(v), \text{Anc}(v), \text{Des}(v)$ to denote the parents, ancestors and descendants of v respectively and we write $\text{Des}[v] = \text{Des}(v) \cup \{v\}$ and $\text{Anc}[v] = \text{Anc}(v) \cup \{v\}$ to include v itself. We define $\text{Ch}(v) \subseteq \text{Des}(v)$ as the set of *direct children* of v, that is, for any $w \in \text{Ch}(v)$ there does *not* exists $z \in V \setminus \{v, w\}$ such that $z \in \text{Des}(v) \cap \text{Anc}(w)$. Note that, $\text{Ch}(v) \subseteq \{w \in V : v \to w\} \subseteq \text{Des}(v)$.

B.2. Causal graph basics

A directed acyclic graph (DAG) is a fully oriented graph without directed cycles. By representing random variables by nodes, DAGs are commonly used as graphical causal models (Pearl, 2009), where the joint probability density f factorizes according to the Markov property: $f(v_1, \ldots, v_n) = \prod_{i=1}^{n} f(v_i \mid pa(v))$, where $pa(v)$ denotes the values taken by v's parents. One can associate a (not necessarily unique) *valid permutation / topological ordering* $\pi : V \to [n]$ 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 $\pi(u) < \pi(v)$.

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 have the same skeleton and the 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. 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). In fact, the essential graph $\mathcal{E}(G)$ can be computed from G by orienting v-structures in the skeleton $\mathrm{skel}(G)$ and applying Meek rules (see Appendix D). An edge $u \to v$ is a *covered edge* (Chickering, 1995) if $\mathrm{Pa}(u) = \mathrm{Pa}(v) \setminus \{u\}$. We use $\mathcal{C}(G) \subseteq E(G)$ to denote the set of covered edges of G. The following is a well-known result relating covered edges and MECs.

Lemma B.1 ((Chickering, 1995)). *If G and G' belong in the same MEC if and only if there exists a sequence of covered edge reversals to transform between them.*

C. Additional Related Works on Causal Structure Learning

Constraint-based algorithms, such as ours, use information about conditional independence relations to identify the underlying structure. From purely observational data, the PC (Spirtes et al., 2000), FCI (Spirtes et al., 2000) and RFCI algorithms (Colombo et al., 2012) have been shown to consistently recover the essential graph, assuming causal sufficiency, faithfulness, and i.i.d. samples. The problem of recovering the DAG using constraints from interventional data was first studied by Eberhardt et al. (2006; 2005); Eberhardt (2007). Many recent works (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; Choo & Shiragur, 2023) have followed up on these themes.

Score-based methods maximize a particular score function over the space of graphs. For observational data, the GES algorithm (Chickering, 2002) uses the BIC to iteratively add edges. Extending the GES, Hauser & Bühlmann (2012) proposed the GIES algorithm that uses passive interventional data to orient more edges. Hybrid methods, like Solus et al. (2021) for observational and Wang et al. (2017) for interventional data, use elements of both approaches.

D. Meek rules

Meek rules are a set of 4 edge orientation rules that are sound and complete with respect to any given set of arcs that has a consistent DAG extension (Meek, 1995a). Given any edge orientation information, one can always repeatedly apply Meek rules till a unique fixed point (where no further rules trigger) to maximize the number of oriented arcs.

Definition D.1 (The four Meek rules (Meek, 1995a), see Figure 5 for an illustration).

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

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

R3 Edge $\{a, b\} \in E(G) \setminus A(G)$ 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 $\{a, b\} \in E(G) \setminus A(G)$ 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 $\mathcal{O}(d \cdot |E(G)|)$ time and computes the closure under Meek rules, where d is the degeneracy of the graph skeleton[9].

---

[9] A d-degenerate graph is an undirected graph in which every subgraph has a vertex of degree at most d. Note that the degeneracy of a graph is typically smaller than the maximum degree of the graph.
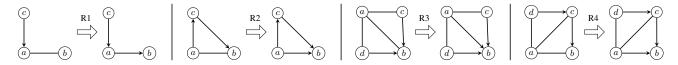
Figure 5. An illustration of the four Meek rules

E. Imperfect *partial* advice via MPDAGs

In the previous sections, we discuss advice that occurs in the form of a DAG $\widetilde{G} \in [G^*]$. However, this may be too much to ask for in certain situations. For example:

- The Markov equivalence class may be too large for an expert to traverse through and propose an advice DAG.

- The expert only has opinions about a subset of a very large causal graph involving millions of nodes / edges.

As discussed in Section 2.4, we can formulate such partial advice as MPDAGs. Given a MPDAG as expert advice, a natural attempt would be to sample a DAG $\widetilde{G}$ from it to use the full advice. Unfortunately, it is #P-complete even to count the number of DAGs consistent with a given MPDAG in general (Wienöbst et al., 2021b) and we are unaware of any efficient way to sample uniformly at random from it. Instead, we propose to pick an arbitrary DAG $\widetilde{G}$ as advice within the given MPDAG: pick any unoriented edge, orient arbitrarily, apply Meek rules, repeat until fully oriented. The following result follows naturally by maximizing over all possible DAGs consistent with the given partial advice.

Theorem E.1. *Fix an essential graph $\mathcal{E}(G^*)$ with an unknown underlying ground truth DAG G^*. Given a set $\mathcal{A}$ of DAGs consistent with the given partial advice and intervention set bound $k \geq 1$, there exists 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. $\mathcal{O}(\max\{1, \log\max_{\widetilde{G} \in \mathcal{A}} \psi(G^*, \widetilde{G})\} \cdot \nu_1(G^*))$
2. $\mathcal{O}(\max\{1, \log\max_{\widetilde{G} \in \mathcal{A}} \psi(G^*, \widetilde{G})\} \cdot \log k \cdot \nu_k(G^*))$
when $k = 1$ and $k > 1$ respectively.

F. Technical Overview for Theorem 3.1

As discussed in Section 2, it suffices to prove Theorem 3.1 with respect to moral DAGs.

Our strategy for proving Theorem 3.1 is to consider two arbitrary DAGs G_s (source) and G_t (target) in the same equivalence class and transform a verifying set for G_s into a verifying set for G_t using Lemma B.1 (see Algorithm 2 for the explicit algorithm[10]). Instead of proving Theorem 3.1 by analyzing the exact sequence of covered edges produced by Algorithm 2[11] when transforming between the DAGs $G_{\min} = \text{argmin}_{G \in [G^*]} \nu_1(G)$ and $G_{\max} = \text{argmax}_{G \in [G^*]} \nu_1(G)$, we will prove something more general.

Observe that taking both endpoints of any maximal matching of covered edges is a valid verifying set that is at most *twice* the size of the minimum verifying set. This is because maximal matching is a 2-approximation to the minimum vertex cover. Motivated by this observation, our proof for Theorem 3.1 uses the following transformation argument (Lemma F.3): for two DAGs G and G' that differ only on the arc direction of a single covered edge $x \sim y$, we show that given a conditional-root-greedy (CRG) maximal matching[12] on the covered edges of G, we can obtain another CRG maximal matching *of the same size* on the covered edges of G', after reversing $x \sim y$ and transforming G to G'.

So, starting from G_s, we compute a CRG maximal matching, then we apply the transformation argument above on the sequence of covered edges given by Algorithm 2 until we get a CRG maximal matching of G_t *of the same size*. Thus, we can conclude that the minimum vertex cover sizes of G_s and G_t differ by a factor of at most two. This argument holds for *any* pair of DAGs (G_s, G_t) from the same MEC.

---

[10]Lemma 2 of (Chickering, 1995) guarantees that $x \to y$ is a covered edge of the current G_s whenever step 9 is executed.

[11]The correctness of Algorithm 2 is given in (Chickering, 1995) where the key idea is to show that $x \to y$ found in this manner is a covered edge. This is proven in Lemma 2 of (Chickering, 1995).

[12]A special type of maximal matching (see Definition F.1).

Algorithm 2 (Chickering, 1995): Transforms between two DAGs within the same MEC via covered edge reversals

---

1: **Input**: Two DAGs $G_s = (V, E_s)$ and $G_t = (V, E_t)$
2: **Output**: A sequence seq of covered edge reversals that transforms G_s to G_t
3: seq $\leftarrow \emptyset$
4: **while** $G_s \neq G_t$ **do**
5:     Fix an arbitrary valid ordering π for G_s.
6:     Let $A \leftarrow A(G_s) \setminus A(G_t)$ be the set of differing arcs.
7:     Let $y \leftarrow \text{argmin}_{y \, \in \, V \, : \, \text{Pa}_A(y) \neq \emptyset}\{\pi(y)\}$.
8:     Let $x \leftarrow \text{argmax}_{z \, \in \, \text{Pa}_A(y)}\{\pi(z)\}$.
9:     Add $x \rightarrow y$ to seq.
10:    Update G_s by replacing $x \rightarrow y$ with $y \rightarrow x$.
11: **end while**
12: **return** seq

We now define what is a conditional-root-greedy (CRG) maximal matching. As the set of covered edges $\mathcal{C}(G)$ of any DAG G induces a forest (see Theorem 2.6), we define the CRG maximal matching using a particular greedy process on the tree structure of $\mathcal{C}(G)$. The CRG maximal matching is unique with respect to a fixed valid ordering π of G and subset S. We will later consider CRG maximal matchings with $S = A(G_s) \cap A(G_t)$, where the arc set S *remains unchanged throughout the entire transformation process*.

Definition F.1 (Conditional-root-greedy (CRG) maximal matching). Given a DAG $G = (V, E)$ with a valid ordering π_G and a subset of edges $S \subseteq E$, we define the conditional-root-greedy (CRG) maximal matching $M_{G,\pi_G,S}$ as the *unique* maximal matching on $\mathcal{C}(G)$ computed via Algorithm 3: greedily choose arcs $x \rightarrow y$ where the x has no incoming arcs by minimizing $\pi_G(y)$, conditioned on *favoring arcs outside of S*.

Algorithm 3 Conditional-root-greedy maximal matching

1: **Input**: A DAG $G = (V, E)$, a valid ordering π_G, a subset of edges $S \subseteq E$
2: **Output**: A CRG maximal matching $M_{G,\pi_G,S}$
3: Initialize $M_{G,\pi_G,S} \leftarrow \emptyset$ and $C \leftarrow \mathcal{C}(G)$
4: **while** $C \neq \emptyset$ **do**
5:     $x \leftarrow \text{argmin}_{z \, \in \, \{u \in V \, | \, u \rightarrow v \in C\}}\{\pi_G(z)\}$
6:     $y \leftarrow \text{argmin}_{z \in V \, : \, x \rightarrow z \, \in \, C}\{\pi_G(z) + n^2 \cdot \mathbb{1}_{x \rightarrow z \in S}\}$
7:     Add the arc $x \rightarrow y$ to $M_{G,\pi_G,S}$
8:     Remove all arcs with x or y as endpoints from C
9: **end while**
10: **return** $M_{G,\pi_G,S}$

To prove the transformation argument (Lemma F.3), we need to first understand how the status of covered edges evolve when we perform a single edge reversal. The following lemma may be of independent interest beyond this work.

Lemma F.2 (Covered edge status changes due to covered edge reversal). *Let G^* be a moral DAG with MEC $[G^*]$ and consider any DAG $G \in [G^*]$. Suppose $G = (V, E)$ has a covered edge $x \rightarrow y \in \mathcal{C}(G) \subseteq E$ and we reverse $x \rightarrow y$ to $y \rightarrow x$ to obtain a new DAG $G' \in [G^*]$. Then, all of the following statements hold:*

1. *$y \rightarrow x \in \mathcal{C}(G')$. Note that this is the covered edge that was reversed.*

2. *If an edge e does not involve x or y, then $e \in \mathcal{C}(G)$ if and only if $e \in \mathcal{C}(G')$.*

3. *If $x \in \text{Ch}_G(a)$ for some $a \in V \setminus \{x, y\}$, then $a \rightarrow x \in \mathcal{C}(G)$ if and only if $a \rightarrow y \in \mathcal{C}(G')$.*

4. *If $b \in \text{Ch}_G(y)$ and $x \rightarrow b \in E(G)$ for some $b \in V \setminus \{x, y\}$, then $y \rightarrow b \in \mathcal{C}(G)$ if and only if $x \rightarrow b \in \mathcal{C}(G')$.*

Using Lemma F.2, we derive our transformation argument.

Lemma F.3. *Consider two moral DAGs G_1 and G_2 from the same MEC such that they differ only in one covered edge direction: $x \to y \in E(G_1)$ and $y \to x \in E(G_2)$.*

Let vertex a be the direct parent of x in G_1, if it exists. Let $S \subseteq E$ be a subset such that $a \to x \in S$ and $x \to y, y \to x \notin S$ (if a does not exist, ignore condition $a \to x \in S$).

Suppose π_{G_1} is an ordering for G_1 such that $y = \operatorname{argmin}_{z \,:\, x \to z \in \mathcal{C}(G_1)}\{\pi_{G_1}(z) + n^2 \cdot \mathbb{1}_{x \to z \in S}\}$ and denote $M_{G_1, \pi_{G_1}, S}$ as the corresponding CRG maximal matching for $\mathcal{C}(G_1)$. Then, there exists an explicit modification of π_{G_1} to π_{G_2}, and $M_{G_1, \pi_{G_1}, S}$ to a CRG maximal matching $M_{G_2, \pi_{G_2}, S}$ for $\mathcal{C}(G_2)$ such that $|M_{G_1, \pi_{G_1}, S}| = |M_{G_2, \pi_{G_2}, S}|$.

To be precise, given π_{G_1}, we will define π_{G_2} in our proofs as follows:

$$
\pi_{G_2}(v) = \begin{cases} \pi_{G_1}(x) & \text{if } v = y \\ \pi_{G_1}(u) & \text{if } v = x \\ \pi_{G_1}(y) & \text{if } v = u \\ \pi_{G_1}(v) & \text{else} \end{cases} \tag{1}
$$

As discussed earlier, Theorem 3.1 follows by picking $G_s = \operatorname{argmax}_{G \in [G^*]} \nu_1(G)$ and $G_t = \operatorname{argmin}_{G \in [G^*]} \nu_1(G)$, applying Algorithm 2 to find a transformation sequence of covered edge reversals between them, and repeatedly applying Lemma F.3 with the conditioning set $S = A(G_s) \cap A(G_t)$ to conclude that G_s and G_t have the same sized CRG maximal matchings, and thus implying that $\min_{G \in [G^*]} \nu_1(G) = \nu_1(G_s) \leq 2 \cdot \nu_1(G_t) = 2 \cdot \operatorname{argmax}_{G \in [G^*]} \nu_1(G)$. Note that we keep the conditioning set S *unchanged throughout the entire transformation process* from G_s to G_t.

For an illustrated example of conditional-root-greedy (CRG) maximal matchings and how we update the permutation ordering, see Figure 6 and Figure 7.

Figure 6. Consider the following simple setup of two DAGs G_1 and G_2 which agree on all arc directions except for $x \to y$ in G_1 and $y \to x$ in G_2. Dashed arcs represent the covered edges in each DAG. The numbers below each vertex indicate the π_{G_1} and π_{G_2} orderings respectively. In G_1, $u = \operatorname{argmin}_{z \in \mathrm{Ch}_{G_1}(x)}\{\pi_{G_1}(z)\}$. Observe that Equation (1) modifies the ordering only for $\{x, y, u\}$ (in blue) while keeping the ordering of all other vertices fixed. Suppose $S = A(G_1) \cap A(G_2) = \{a \to b, a \to x, a \to y, a \to u, x \to b, x \to u, y \to b\}$. With respect to π_{G_1} and S, The conditional-root-greedy maximal matchings (see Algorithm 3) are $M_{G_1, \pi_{G_1}, S} = \{a \to x, y \to b\}$ and $M_{G_2, \pi_{G_2}, S} = \{a \to y, x \to b\}$.

G. Deferred proofs

G.1. Preliminaries

Our proofs rely on some existing results which we first state and explain below.

Lemma G.1 (Lemma 27 of (Choo et al., 2022)). *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. If $\mathcal{I} \subseteq 2^V$ is a verifying set, then $\mathcal{I}$ separates all unoriented covered edge $u \sim v$ of G.*

Lemma G.2 (Lemma 28 of (Choo et al., 2022)). *Fix an essential graph $\mathcal{E}(G^*)$ and $G \in [G^*]$. If $\mathcal{I} \subseteq 2^V$ is an intervention set that separates every unoriented covered edge $u \sim v$ of G, then $\mathcal{I}$ is a verifying set.*

Lemma G.1 tells us that we have to intervene on one of the endpoints of *any* covered edge in order to orient it while Lemma G.2 tells us that doing so for all covered edges suffices to orient the entire causal DAG.
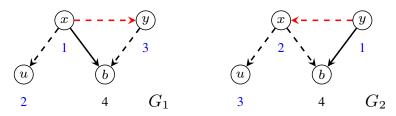
Figure 7. Consider the following simple setup of two DAGs G_3 and G_4 which agree on all arc directions except for $x \to y$ in G_3 and $y \to x$ in G_4. Dashed arcs represent the covered edges in each DAG. The numbers below each vertex indicate the π_{G_3} and π_{G_4} orderings respectively. Observe that $\mathcal{C}(G_3) = \{x \to u, x \to y, y \to b\}$. If we define $S = A(G_3) \cap A(G_4) = \{x \to b, x \to u, y \to b\}$, we see that the conditional-root-greedy maximal matchings (see Algorithm 3) are $M_{G_3, \pi_{G_3}, S} = \{x \to y\}$ and $M_{G_4, \pi_{G_4}, S} = \{y \to x\}$. Note that Algorithm 3 does *not* choose $x \to u \in \mathcal{C}(G_1)$ despite $\pi(u) < \pi(y)$ because $x \to u \in S$, so $\pi(y) < \pi(u) + n^2$.

G.2. Verification numbers of DAGs within same MEC are bounded by a factor of two

We use the following simple lemma in our proof of Lemma F.2.

Lemma G.3. *For any covered edge $x \to y$ in a DAG $G = (V, E)$, we have $y \in Ch_G(x)$. Furthermore, each vertex only appears as an endpoint in the collection of covered edges $\mathcal{C}(G)$ at most once.*

Proof. For the first statement, suppose, for a contradiction, that $y \notin Ch(x)$. Then, there exists some $z \in V \setminus \{x, y\}$ such that $z \in Des(x) \cap Anc(y)$. Fix an arbitrary ordering π for G and let $z^* = \mathrm{argmax}_{z \in Des(x) \cap Anc(y)} \{\pi(z)\}$. Then, we see that $z^* \to y$ while $z^* \not\to x$ since $z^* \in Des(x)$. So, $x \to y$ *cannot* be a covered edge. Contradiction.

For the second statement, suppose, for a contradiction, that there are two covered edges $u \to x, v \to x \in \mathcal{C}(G)$ that ends with x. Since $u \to x \in \mathcal{C}(G)$, we must have $v \to u$. Since $v \to x \in \mathcal{C}(G)$, we must have $u \to v$. We cannot have both $u \to v$ and $v \to u$ simultaneously. Contradiction. $\square$

Lemma F.2 (Covered edge status changes due to covered edge reversal). *Let G^* be a moral DAG with MEC $[G^*]$ and consider any DAG $G \in [G^*]$. Suppose $G = (V, E)$ has a covered edge $x \to y \in \mathcal{C}(G) \subseteq E$ and we reverse $x \to y$ to $y \to x$ to obtain a new DAG $G' \in [G^*]$. Then, all of the following statements hold:*

1. *$y \to x \in \mathcal{C}(G')$. Note that this is the covered edge that was reversed.*

2. *If an edge e does not involve x or y, then $e \in \mathcal{C}(G)$ if and only if $e \in \mathcal{C}(G')$.*

3. *If $x \in Ch_G(a)$ for some $a \in V \setminus \{x, y\}$, then $a \to x \in \mathcal{C}(G)$ if and only if $a \to y \in \mathcal{C}(G')$.*

4. *If $b \in Ch_G(y)$ and $x \to b \in E(G)$ for some $b \in V \setminus \{x, y\}$, then $y \to b \in \mathcal{C}(G)$ if and only if $x \to b \in \mathcal{C}(G')$.*

Proof. The only parental relationships that changed when we reversing $x \to y$ to $y \to x$ are $Pa_{G'}(y) = Pa_G(y) \setminus \{x\}$ and $Pa_{G'}(x) = Pa_G(x) \cup \{y\}$. For any other vertex $u \in V \setminus \{x, y\}$, we have $Pa_{G'}(u) = Pa_G(u)$. The first two points have the same proof: as parental relationships of both endpoints are unchanged, the covered edge status is unchanged.

3. Since $x \to y \in \mathcal{C}(G)$, we have $a \to y \in E(G)$. We prove both directions separately.

   Suppose $a \to x \in \mathcal{C}(G)$. Then, $Pa_G(a) = Pa_G(x) \setminus \{a\}$. Since $x \to y \in \mathcal{C}(G)$, then $Pa_G(x) = Pa_G(y) \setminus \{x\}$. So, we have $Pa_{G'}(a) = Pa_G(a) = Pa_G(x) \setminus \{a\} = Pa_G(y) \setminus \{x, a\} = Pa_{G'}(y) \setminus \{a\}$. Thus, $a \to y \in \mathcal{C}(G')$.

   Suppose $a \to x \notin \mathcal{C}(G)$. Then, one of the two cases must occur:

   (a) There exists some vertex u such that $u \to a$ and $u \not\to x$ in G.
       Since $x \to y$ is a covered edge, $u \not\to x$ implies $u \not\to y$ in G. Therefore, $a \to y \notin \mathcal{C}(G')$ due to $u \to a$.
   (b) There exists some vertex v such that $v \to x$ and $v \not\to a$ in G.
       There are two possibilities for $v \not\to a$: $v \not\sim a$ or $v \leftarrow a$. If $v \not\sim a$, then $v \to x \leftarrow a$ is a v-structure. If $v \leftarrow a$, then $x \notin Ch(a)$ since we have $a \to v \to x$. Both possibilities lead to contradictions.

   The first case implies $a \to y \notin \mathcal{C}(G')$ while the second case cannot happen.

19

4. We prove both directions separately.

Suppose $y \to b \in \mathcal{C}(G)$. Then, $\texttt{Pa}_G(y) = \texttt{Pa}_G(b) \setminus \{y\}$. Since $x \to y \in \mathcal{C}(G)$, then $\texttt{Pa}_G(x) = \texttt{Pa}_G(y) \setminus \{x\}$. So, we have $\texttt{Pa}_{G'}(b) \setminus \{x\} = \texttt{Pa}_G(b) \setminus \{x\} = \texttt{Pa}_G(y) \cup \{y\} \setminus \{x\} = \texttt{Pa}_G(x) \cup \{y\} = \texttt{Pa}_{G'}(x)$. Thus, $x \to b \in \mathcal{C}(G')$.

Suppose $y \to b \notin \mathcal{C}(G)$. Then, one of the two cases must occur:

- There exists some vertex $u \to y$ and $u \not\to b$.
  Since $x \to y$ is a covered edge, $u \to y$ implies $u \to x$. Therefore, $x \to b \notin \mathcal{C}(G')$ due to $u \not\to b$.
- There exists some vertex $v \to b$ and $v \not\to y$.
  There are two possibilities for $v \not\to y$: $v \not\sim y$ or $v \leftarrow y$. If $v \not\sim y$, then $v \to b \leftarrow y$ is a v-structure. If $v \leftarrow y$, then $b \notin \texttt{Ch}(y)$ since we have $y \to v \to b$. Both possibilities lead to contradictions.

The first case implies $x \to b \notin \mathcal{C}(G')$ while the second case cannot happen. $\qquad\square$

Lemma F.3. *Consider two moral DAGs G_1 and G_2 from the same MEC such that they differ only in one covered edge direction: $x \to y \in E(G_1)$ and $y \to x \in E(G_2)$.*

Let vertex a be the direct parent of x in G_1, if it exists. Let $S \subseteq E$ be a subset such that $a \to x \in S$ and $x \to y, y \to x \notin S$ (if a does not exist, ignore condition $a \to x \in S$).

Suppose π_{G_1} is an ordering for G_1 such that $y = \mathrm{argmin}_{z\,:\,x \to z \in \mathcal{C}(G_1)}\{\pi_{G_1}(z) + n^2 \cdot \mathbb{1}_{x \to z \in S}\}$ and denote $M_{G_1, \pi_{G_1}, S}$ as the corresponding CRG maximal matching for $\mathcal{C}(G_1)$. Then, there exists an explicit modification of π_{G_1} to π_{G_2}, and $M_{G_1, \pi_{G_1}, S}$ to a CRG maximal matching $M_{G_2, \pi_{G_2}, S}$ for $\mathcal{C}(G_2)$ such that $|M_{G_1, \pi_{G_1}, S}| = |M_{G_2, \pi_{G_2}, S}|$.

Proof. Define $u = \mathrm{argmin}_{z \in \texttt{Ch}_{G_1}(x)}\{\pi_{G_1}(z)\}$ as the lowest ordered child of x. Note that Algorithm 3 chooses $x \to y$ instead of $x \to u$ by definition of y. This implies that $x \to u \in S$ whenever $u \neq y$.

Let us define π_{G_2} as follows:

$$
\pi_{G_2}(v) = \begin{cases} \pi_{G_1}(x) & \text{if } v = y \\ \pi_{G_1}(u) & \text{if } v = x \\ \pi_{G_1}(y) & \text{if } v = u \\ \pi_{G_1}(v) & \text{else} \end{cases}
$$

Clearly, $\pi_{G_1}(x) < \pi_{G_1}(y)$ and $\pi_{G_2}(x) > \pi_{G_2}(y)$. Meanwhile, for any other two adjacent vertices v and v', observe that $\pi_{G_1}(v) < \pi_{G_1}(v') \iff \pi_{G_2}(v) < \pi_{G_2}(v')$ so π_{G_2} agrees with the arc orientations of π_{G_1} except for $x \sim y$. See Figure 6 for an illustrated example.

Define vertex b as follows:

$$
b = \mathrm{argmin}_{z \in V\,:\,z \in \texttt{Des}(x) \text{ and } y \to z \in \mathcal{C}(G_1)}\{\pi_{G_1}(z) + n^2 \cdot \mathbb{1}_{x \to z \in S}\}
$$

If vertex b exists, then we know that $b \in \texttt{Ch}_{G_1}(y)$ and $x \to b \in \mathcal{C}(G_2)$ by Lemma G.3 and Lemma F.2. By minimality of b, Definition F.1 will choose $y \to b$ if picking a covered edge starting with y for $M_{G_1, \pi_{G_1}, S}$. So, we can equivalently define vertex b as follows:

$$
b = \mathrm{argmin}_{z \in V\,:\,z \in \texttt{Des}(y) \text{ and } x \to z \in \mathcal{C}(G_2)}\{\pi_{G_2}(z) + n^2 \cdot \mathbb{1}_{x \to z \in S}\}
$$

By choice of π_{G_2}, Definition F.1 will choose $x \to b$ if picking a covered edge starting with x for $M_{G_2, \pi_{G_2}, S}$.

We will now construct a same-sized maximal matching $M_{G_2, \pi_{G_2}, S}$ from $M_{G_1, \pi_{G_1}, S}$ (Step 1), argue that it is maximal matching of $\mathcal{C}(G_2)$ (Step 2), and that it is indeed a conditional-root-greedy matching for $\mathcal{C}(G_2)$ with respect to π_{G_2} and S (Step 3). There are three cases that cover all possibilities:

Case 1 Vertex a exists, $a \to x \in M_{G_1, \pi_{G_1}, S}$, and vertex b exists.

Case 2 Vertex a exists, $a \to x \in M_{G_1, \pi_{G_1}, S}$, and vertex b does not exist.

Case 3 $a \to x \notin M_{G_1, \pi_{G_1}, S}$.
This could be due to vertex a not existing, or $a \to x \notin \mathcal{C}(G_1)$, or $M_{G_1, \pi_{G_1}, S}$ containing a covered edge ending at a so $a \to x$ was removed from consideration.

Step 1: Construction of $M_{G_2,\pi_{G_2},S}$ such that $|M_{G_2,\pi_{G_2},S}| = |M_{G_1,\pi_{G_1},S}|$.

By Lemma F.2, covered edge statuses of edges whose endpoints do not involve x or y will remain unchanged. By definition of y, we know that Definition F.1 will choose $x \to y$ if picking a covered edge starting with x for $M_{G_1,\pi_{G_1},S}$.

Since $a \to x \in M_{G_1,\pi_{G_1}}$ in cases 1 and 2, we know that there is no arcs of the form $x \to \cdot$ in $M_{G_1,\pi_{G_1},S}$. Since there is no arc of the form $\cdot \to x$ in $M_{G_1,\pi_{G_1},S}$ in case 3, we know that $x \to y \in M_{G_1,\pi_{G_1},S}$.

Case 1 Define $M_{G_2,\pi_{G_2},S} = M_{G_1,\pi_{G_1},S} \cup \{a \to y, x \to b\} \setminus \{a \to x, y \to b\}$.

Case 2 Define $M_{G_2,\pi_{G_2},S} = M_{G_1,\pi_{G_1},S} \cup \{a \to y\} \setminus \{a \to x\}$.

Case 3 Define $M_{G_2,\pi_{G_2},S} = M_{G_1,\pi_{G_1},S} \cup \{y \to x\} \setminus \{x \to y\}$.

By construction, we see that $|M_{G_2,\pi_{G_2},S}| = |M_{G_1,\pi_{G_1},S}|$.

Step 2: $M_{G_2,\pi_{G_2},S}$ is a maximal matching of the covered edge $\mathcal{C}(G_2)$ of G_2.

To prove that $M_{G_2,\pi_{G_2},S}$ is a maximal matching of $\mathcal{C}(G_2)$, we argue in three steps:

 2(i) Edges of $M_{G_2,\pi_{G_2},S}$ belong to $\mathcal{C}(G_2)$.

 2(ii) $M_{G_2,\pi_{G_2},S}$ is a matching of $\mathcal{C}(G_2)$.

 2(iii) $M_{G_2,\pi_{G_2},S}$ is maximal matching of $\mathcal{C}(G_2)$.

Step 2(i): Edges of $M_{G_2,\pi_{G_2},S}$ belong to $\mathcal{C}(G_2)$.

By Lemma F.2, covered edge statuses of edges whose endpoints do not involve x or y will remain unchanged. Since $M_{G_1,\pi_{G_1},S}$ is a matching, it has at most one edge e involving endpoint x and at most one edge e' involving endpoint y (e' could be e).

Case 1 Since b exists, the edges in $M_{G_1,\pi_{G_1},S}$ with endpoints involving $\{x, y\}$ are $a \to x$ and $y \to b$. By Lemma F.2, we know that $a \to y, x \to b \in \mathcal{C}(G_2)$.

Case 2 Since b does not exist, the only edge in $M_{G_1,\pi_{G_1},S}$ with endpoints involving $\{x, y\}$ is $a \to x$. By Lemma F.2, we know that $a \to y \in \mathcal{C}(G_2)$.

Case 3 Since $a \to x \notin M_{G_1,\pi_{G_1},S}$, we have $x \to y \in M_{G_1,\pi_{G_1},S}$ by minimality of y.

In all cases, we see that $M_{G_2,\pi_{G_2},S} \subseteq \mathcal{C}(G_2)$.

Step 2(ii): $M_{G_2,\pi_{G_2},S}$ is a matching of $\mathcal{C}(G_2)$.

It suffices to argue that there are *no* two edges in $M_{G_2,\pi_{G_2},S}$ sharing an endpoint. Since $M_{G_1,\pi_{G_1},S}$ is a matching, this can only happen via newly added endpoints in $M_{G_2,\pi_{G_2},S}$.

Case 1 The endpoints of newly added edges are exactly the endpoints of removed edges.

Case 2 Since we removed $a \to x$ and added $a \to y$, it suffices to check that there are no edges in $M_{G_1,\pi_{G_1},S}$ involving y. This is true since b does not exist in Case 2.

Case 3 The endpoints of newly added edges are exactly the endpoints of removed edges.

Therefore, we conclude that $M_{G_2,\pi_{G_2},S}$ is a matching of $\mathcal{C}(G_2)$.

Step 2(iii): $M_{G_2, \pi_{G_2}, S}$ **is a maximal matching of** $\mathcal{C}(G_2)$**.**

For any $u \to v \in \mathcal{C}(G_2)$, we show that there is some edge in $M_{G_2, \pi_{G_2}, S}$ with at least one of u or v is an endpoint. By Lemma F.2, covered edge statuses of edges whose endpoints do not involve x or y will remain unchanged, so it suffices to consider $|\{u, v\} \cap \{x, y\}| \geq 1$.

We check the following 3 scenarios corresponding to $|\{u, v\} \cap \{x, y\}| \geq 1$ below:

(i) $y \in \{u, v\}$.

   The endpoints of $M_{G_2, \pi_{G_2}}$ always contains y.

(ii) $y \notin \{u, v\}$ and $x \to v \in \mathcal{C}(G_2)$, for some $v \in V \setminus \{x, y\}$.

   Since $x \to v \in \mathcal{C}(G_2)$ and $y \to x$ in G_2, it must be the case that $y \to v$ in G_2. Since G_1 and G_2 agrees on all arcs except $x \sim y$, we have that $y \to v$ in G_1 as well. Since $x \to v \in \mathcal{C}(G_2)$, we know that $v \in \text{Ch}_{G_2}(x)$ via Lemma G.3. So, we have $y \to v \in \mathcal{C}(G_1)$ via Lemma F.2. Since the set $\{v : y \to v \in \mathcal{C}(G_1)\}$ is non-empty, vertex b exists. In both cases 1 and 3, the endpoints of $M_{G_2, \pi_{G_2}}$ includes x.

(iii) $y \notin \{u, v\}$ and $u \to x \in \mathcal{C}(G_2)$, for some $u \in V \setminus \{x, y\}$.

   By Lemma G.3, we know that $x \in \text{Ch}_{G_2}(u)$. Meanwhile, since $y \to x \in \mathcal{C}(G_2)$, we must have $u \to y$ in G_2. However, this implies that $x \notin \text{Ch}_{G_2}(u)$ since $u \to y \to x$ exists. This is a contradiction, so this situation cannot happen.

As the above argument holds for any $u \to v \in \mathcal{C}(G_2)$, we see that $M_{G_2, \pi_{G_2}}$ is maximal matching for $\mathcal{C}(G_2)$.

Step 3: $M_{G_2, \pi_{G_2}, S}$ **is a conditional-root-greedy maximal matching.**

We now compare the execution of Algorithm 3 on (π_{G_1}, S) and (π_{G_2}, S). Note that S remains unchanged.

We know the following:

- Since $\pi_{G_2}(y) = \pi_{G_1}(x)$ and $a \to x \in S$, if a exists and $a \to x$ is chosen by Algorithm 3 on (π_{G_1}, S), then it means that there are *no* $a \to v$ arc in $\mathcal{C}(G_1)$ such that $a \to v \notin S$. So, $a \to y$ will be chosen by Algorithm 3 on (π_{G_2}, S) if a exists.

- Since $\pi_{G_2}(y) = \pi_{G_1}(x)$, x is chosen as a root by Algorithm 3 on (π_{G_1}, S) if and only if y is chosen as a root by Algorithm 3 on (π_{G_2}, S).

- By definition of b, if it exists, then $y \to b \in M_{G_1, \pi_{G_1}, S} \iff x \to b \in M_{G_2, \pi_{G_2}, S}$.

- By the definition of π_{G_2}, we see that Algorithm 3 makes the "same decisions" when choosing arcs rooted on $V \setminus \{a, x, y, b\}$.

Therefore, $M_{G_2, \pi_{G_2}, S}$ is indeed a conditional-root-greedy maximal matching for $\mathcal{C}(G_2)$ with respect to π_{G_2} and S. □

Theorem 3.1. *For any DAG G^* with MEC $[G^*]$, we have that $\max_{G \in [G^*]} \nu_1(G) \leq 2 \cdot \min_{G \in [G^*]} \nu_1(G)$.*

Proof. Consider any two DAGs $G_s, G_t \in [G^*]$. To transform $G_s = (V, E_s)$ to $G_t = (V, E_t)$, Algorithm 2 flips covered edges one by one such that $|E_s \setminus E_t|$ decreases in a monotonic manner. We will repeatedly apply Lemma F.3 with $S = A(G_s) \cap A(G_t)$ on the sequence of covered edge reversals produced by Algorithm 2.

Let π_{G_s} be an arbitrary ordering for G_s and we compute an initial conditional-root-greedy maximal matching for $\mathcal{C}(G_s)$ with respect to some ordering π_{G_s} and conditioning set S. To see why Lemma F.3 applies at each step for reversing a covered edge from $x \to y$ to $y \to x$, we need to ensure the following:

1. If x has a parent vertex a (i.e. $x \in \text{Ch}_{G_1}(a)$), then $a \to x \in S$.

   If $a \to x \notin S$, then then $a \to x$ is a covered edge that should be flipped to transform from G_s to G_t. However, this means that Algorithm 2 would pick $a \to x$ to reverse instead of picking $x \to y$ to reverse. Contradiction.

2. $x \to y, y \to x \notin S$.

   This is satisfied by the definition of $S = E_s \cap E_t$ since reversing $x \to y$ to $y \to x$ implies that neither of them are in S.

3. $y = \mathrm{argmin}_{z \,:\, x \to z \in \mathcal{C}(G_1)}\{\pi_{G_1}(z) + n^2 \cdot \mathbb{1}_{x \to z \in S}\}$.

   Since $x \to y \notin S$, this is equivalent to checking if $y = \mathrm{argmin}_{z \,:\, x \to z \in \mathcal{C}(G_1)}\{\pi_{G_1}(z)\}$. This is satisfied by line 7 of Algorithm 2.

4. $M_{G_1, \pi_{G_1}, S}$ is a conditional-root-greedy maximal matching for $\mathcal{C}(G_1)$ with respect to some ordering π_{G_1} and conditioning set S.

   This is satisfied since we always maintain a conditional-root-greedy maximal matching and S is unchanged throughout.

By applying Lemma F.3 with $S = A(G_s) \cap A(G_t)$ repeatedly on the sequence of covered edge reversals produced by Algorithm 2, we see that there exists a conditional-root-greedy maximal matching $M_{G_s, \pi_{G_s}}$ for $\mathcal{C}(G_s)$ and a conditional-root-greedy maximal matching $M_{G_t, \pi_{G_t}}$ for $\mathcal{C}(G_t)$ such that $|M_{G_s, \pi_{G_s}}| = |M_{G_t, \pi_{G_t}}|$.

The claim follows since maximal matching is a 2-approximation to minimum vertex cover, and the verification number $\nu(G)$ of any DAG G is the size of the minimum vertex cover of its covered edges $\mathcal{C}(G)$. $\square$

Lemma 3.2 (Tightness of Theorem 3.1). *There exist DAGs G_1 and G_2 from the same MEC with $\nu_1(G_1) = 2 \cdot \nu_1(G_2)$.*

Proof. See Figure 8. $\square$

Figure 8. The ratio of 2 in Theorem 3.1 is tight: G_1 and G_2 belong in the same MEC with $\nu(G_1) = 2$ and $\nu(G_2) = 1$. The dashed arcs represent the covered edges and the boxed vertices represent a minimum vertex cover of the covered edges.

G.3. Adaptive search with imperfect advice

Lemma 4.1. *Fix a DAG $G^* = (V, E)$ and let $V' \subseteq V$ be any subset of vertices. Suppose $\mathcal{I}_{V'} \subseteq V$ is the set of nodes intervened by* SubsetSearch$(\mathcal{E}(G^*), V')$. *If $\mathcal{C}(G^*) \subseteq E(G^*[V'])$, then $\mathcal{E}_{\mathcal{I}_{V'}}(G^*) = G^*$.*

Proof. By Theorem 2.9, SubsetSearch fully orients edges within the node-induced subgraph induced by V', i.e. SubsetSearch will perform atomic interventions on $\mathcal{I}_{V'} \subseteq V$ resulting in $\mathcal{E}_{\mathcal{I}_{V'}}(G^*)[V'] = G^*[V']$. Since $\mathcal{C}(G^*) \subseteq E(G^*[V'])$ and all covered edges $\mathcal{C}(G^*)$ were oriented, then according to Lemma G.1, it must be the case that $V^* \subseteq \mathcal{I}_{V'}$ for some minimum vertex cover V^* of $\mathcal{C}(G^*)$, so we see that $R(G^*, V^*) \subseteq R(G^*, \mathcal{I}_{V'})$. By Lemma G.2, we have $R(G^*, V^*) = A(G^*)$ and so SubsetSearch$(\mathcal{E}(G^*), V')$ fully orients $\mathcal{E}(G^*)$. $\square$

We will now prove our main result (Theorem 3.5) which shows that the number of interventions needed is a function of the quality of the given advice DAG. Let us first recall how we defined the quality of a given advice and restate our algorithm.

Definition 3.4 (Quality measure). *Fix a DAG G^* with MEC $[G^*]$ and consider any DAG $\widetilde{G} \in [G^*]$. We define $\psi(G^*, \widetilde{G})$ as follows:*

$$\psi(G^*, \widetilde{G}) = \max_{\widetilde{V} \in \mathcal{V}(\widetilde{G})} \left| \rho \left(\widetilde{V}, N^{h(G^*, \widetilde{V})}_{\text{skel}(\mathcal{E}(G^*))}(\widetilde{V}) \right) \right|$$

Algorithm 1 Adaptive search algorithm with advice.

1: **Input**: Essential graph $\mathcal{E}(G^*)$, advice DAG $\widetilde{G} \in [G^*]$, intervention size $k \in \mathbb{N}$
2: **Output**: An intervention set $\mathcal{I}$ such that each intervention involves at most k nodes and $\mathcal{E}_{\mathcal{I}}(G^*) = G^*$.
3: Let $\widetilde{V} \in \mathcal{V}(\widetilde{G})$ be any atomic verifying set of $\widetilde{G}$.
4: **if** $k = 1$ **then**
5:     Define $\mathcal{I}_0 = \widetilde{V}$ as an atomic intervention set.
6: **else**
7:     Define $k' = \min\{k, |\widetilde{V}|/2\}$, $a = \lceil |\widetilde{V}|/k' \rceil \geq 2$, and $\ell = \lceil \log_a |C| \rceil$. Compute labelling scheme on $\widetilde{V}$ with $(|\widetilde{V}|, k, a)$ via Lemma 4.3 and define $\mathcal{I}_0 = \{S_{x,y}\}_{x \in [\ell], y \in [a]}$, where $S_{x,y} \subseteq \widetilde{V}$ is the subset of vertices whose x^{th} letter in the label is y.
8: **end if**
9: Intervene on $\mathcal{I}_0$ and initialize $r \leftarrow 0$, $i \leftarrow 0$, $n_0 \leftarrow 2$.
10: **while** $\mathcal{E}_{\mathcal{I}_i}(G^*)$ still has undirected edges **do**
11:     **if** $\rho(\mathcal{I}_i, N^r_{\text{skel}(\mathcal{E}(G^*))}(\widetilde{V})) \geq n_i^2$ **then**
12:         Increment $i \leftarrow i + 1$ and record $r(i) \leftarrow r$.
13:         Update $n_i \leftarrow \rho(\mathcal{I}_i, N^r_{\text{skel}(\mathcal{E}(G^*))}(\widetilde{V}))$
14:         $C_i \leftarrow \texttt{SubsetSearch}(\mathcal{E}_{\mathcal{I}_i}(G^*), N^{r-1}_{\text{skel}(\mathcal{E}(G^*))}(\widetilde{V}), k)$
15:         **if** $\mathcal{E}_{\mathcal{I}_{i-1} \cup C_i}(G^*)$ still has undirected edges **then**
16:             $C'_i \leftarrow \texttt{SubsetSearch}(\mathcal{E}_{\mathcal{I}_{i-1} \cup C_i}(G^*), N^r_{\text{skel}(\mathcal{E}(G^*))}(\widetilde{V}), k)$
17:             Update $\mathcal{I}_i \leftarrow \mathcal{I}_{i-1} \cup C_i \cup C'_i$.
18:         **else**
19:             Update $\mathcal{I}_i \leftarrow \mathcal{I}_{i-1} \cup C_i$.
20:         **end if**
21:     **end if**
22:     Increment $r \leftarrow r + 1$.
23: **end while**
24: **return** $\mathcal{I}_i$

Theorem 3.5. *Fix an essential graph $\mathcal{E}(G^*)$ with an unknown underlying ground truth DAG G^*. Given an advice graph $\widetilde{G} \in [G^*]$ and intervention set bound $k \geq 1$, there exists a deterministic polynomial time algorithm (Algorithm 1) that computes an intervention set $\mathcal{I}$ adaptively such that $\mathcal{E}_{\mathcal{I}}(G^*) = G^*$, and $|\mathcal{I}|$ has size*
1. $\mathcal{O}(\max\{1, \log \psi(G^*, \widetilde{G})\} \cdot \nu_1(G^*))$ when $k = 1$
2. $\mathcal{O}(\max\{1, \log \psi(G^*, \widetilde{G})\} \cdot \log k \cdot \nu_k(G^*))$ when $k > 1$.

Proof. Consider Algorithm 1. Observe that $n_0 = 2$ ensures that $n_0^2 > n_0$.

In this proof, we will drop the subscript $\text{skel}(\mathcal{E}(G^*))$ when we discuss the r-hop neighbors $N^r_{\text{skel}(\mathcal{E}(G^*))}(\cdot)$. We first prove the case where $k = 1$ then explain how to tweak the proof for the case of $k > 1$.

If Algorithm 1 terminates when $i = 0$, then $\mathcal{I} = \mathcal{I}_0 = \widetilde{V}$ and Theorem 3.1 tells us that $|\mathcal{I}| \in \mathcal{O}(\nu_1(G^*))$.

Now, suppose Algorithm 1 terminates with $i = t$, for some final round $t > 0$. As Algorithm 1 uses an arbitrary verifying set of $\widetilde{G}$ in step 3, we will argue that $\mathcal{O}(\max\{1, \log |N^{h(G^*, \widetilde{V})}(\widetilde{V})|\} \cdot \nu(G^*))$ interventions are used in the while-loop, for any arbitrary chosen $\widetilde{V} \in \mathcal{V}(\widetilde{G})$. The theorem then follows by taking a maximization over all possibilities in $\mathcal{V}(\widetilde{G})$.

In Line 12, $r(i)$ records the hop value such that $\rho(\mathcal{I}_i, N^{r(i)}(\widetilde{V})) \geq n_i^2$, for any $0 \leq i < t$. By construction of the algorithm, we know the following:

1. For any $0 < i \leq t$,

$$n_i = \rho(\mathcal{I}_i, N^{r(i)}(\widetilde{V})) \geq n_{i-1}^2 > \rho(\mathcal{I}_i, N^{r(i)-1}(\widetilde{V})) \tag{2}$$

   because $r(i) - 1$ did *not* trigger Algorithm 1 to record $r(i)$.

2. By Theorem 2.9 and Equation (2), for any $1 \leq i \leq t$,

$$
\begin{aligned}
|C_i| &\in \mathcal{O}(\log \rho(\mathcal{I}_i, N^{r(i)-1}(\widetilde{V})) \cdot \nu_1(G^*)) && \subseteq \mathcal{O}(\log n_{i-1} \cdot \nu_1(G^*)) \\
|C_i'| &\in \mathcal{O}(\log \rho(\mathcal{I}_i, N^{r(i)}(\widetilde{V})) \cdot \nu_1(G^*)) && \subseteq \mathcal{O}(\log n_i \cdot \nu_1(G^*))
\end{aligned}
\tag{3}
$$

   Note that the bound for $|C_i'|$ is an over-estimation (but this is okay for our analytical purposes) since some nodes previously counted for $\rho(\mathcal{I}_i, N^{r(i)}(\widetilde{V}))$ may no longer be relevant in $\mathcal{E}_{\mathcal{I}_i \cup C_i}(G^*)$ after intervening on C_i.

3. Since $n_{i-1} \leq \sqrt{n_i}$ for any $0 < i \leq t$, we know that $n_j \leq n_t^{1/2^{t-j}}$ for any $0 \leq j \leq t$. So, for any $0 \leq t' \leq t$, we have

$$\sum_{i=0}^{t'} \log(n_i) \leq \sum_{i=0}^{t'} \log\left(n_{t'}^{1/2^{t'-i}}\right) = \sum_{i=0}^{t'} \frac{\log(n_{t'})}{2^{t'-i}} \leq 2 \cdot \log(n_{t'}) \tag{4}$$

4. By definition of t, $h(G^*, \widetilde{V})$, and Lemma 4.1,

$$r(t-1) < h(G^*, \widetilde{V}) \leq r(t) \tag{5}$$

   and

$$N^{r(t-1)}(\widetilde{V}) \subsetneq N^{h(G^*, \widetilde{V})}(\widetilde{V}) \subseteq N^{r(t)}(\widetilde{V}) \tag{6}$$

Combining Equation (2), Equation (3), and Equation (4), we get

$$\sum_{i=1}^{t-1}(|C_i| + |C_i'|) \in \mathcal{O}\left(\left(\sum_{i=1}^{t-1} \log n_{i-1} + \log n_i\right) \cdot \nu_1(G^*)\right) \subseteq \mathcal{O}\left(\sum_{i=1}^{t-1} \log n_i \cdot \nu_1(G^*)\right) \subseteq \mathcal{O}\left(\log n_{t-1} \cdot \nu_1(G^*)\right) \tag{7}$$

To relate $|\mathcal{I}_t|$ with $|N^{h(G^*, \widetilde{V})}(\widetilde{V})|$, we consider two scenarios depending on whether the essential graph was fully oriented after intervening on C_t or C_t'.

Scenario 1: Fully oriented after intervening on C_t, i.e. $\mathcal{E}_{\mathcal{I}_{t-1} \cup C_t}(G^*) = G^*$. Then,

$$\mathcal{I}_t = C_t \,\dot{\cup}\, \mathcal{I}_{t-1} = C_t \,\dot{\cup}\, (C_{t-1} \,\dot{\cup}\, C_{t-1}') \,\dot{\cup}\, \mathcal{I}_{t-2} = \ldots = C_t \,\dot{\cup}\, \bigcup_{i=1}^{t-1}(C_i \,\dot{\cup}\, C_i') \,\dot{\cup}\, \widetilde{V}$$

In this case, $h(G^*, \widetilde{V}) = r(t) - 1$. By definition, $n_{t-1} \leq |N^{r(t-1)}(\widetilde{V})|$ and we have

$$n_{t-1} \leq |N^{r(t-1)}(\widetilde{V})| < |N^{h(G^*, \widetilde{V})}(\widetilde{V})| \tag{8}$$

since $N^{r(t-1)}(\widetilde{V}) \subsetneq N^{h(G^*, \widetilde{V})}(\widetilde{V})$. So,

$$
\begin{aligned}
|\mathcal{I}_t| - |\widetilde{V}| &= |C_t| + \sum_{i=1}^{t-1}(|C_i| + |C_i'|) \\
&\in \mathcal{O}\left(\log n_{t-1} \cdot \nu_1(G^*)\right) + \mathcal{O}\left(\log n_{t-1} \cdot \nu_1(G^*)\right) && \text{By Equation (3) and Equation (7)} \\
&\subseteq \mathcal{O}\left(\log |N^{h(G^*, \widetilde{V})}(\widetilde{V})| \cdot \nu_1(G^*)\right) && \text{Equation (8)}
\end{aligned}
$$

Scenario 2: Fully oriented after intervening on C_t', i.e. $\mathcal{E}_{\mathcal{I}_{t-1} \cup C_t \cup C_t'}(G^*) = G^*$. Then,

$$\mathcal{I}_t = C_t \,\dot{\cup}\, C_t' \,\dot{\cup}\, \mathcal{I}_{t-1} = \ldots = C_t \,\dot{\cup}\, C_t' \,\dot{\cup}\, \bigcup_{i=1}^{t-1}(C_i \,\dot{\cup}\, C_i') \,\dot{\cup}\, \widetilde{V}$$

In this case, $h(G^*, \widetilde{V}) = r(t)$ and $N^{h(G^*, \widetilde{V})}(\widetilde{V}) = N^{r(t)}(\widetilde{V})$. So,

$$n_t \leq |N^{r(t)}(\widetilde{V})| = |N^{h(G^*, \widetilde{V})}(\widetilde{V})| \tag{9}$$

So,

$$
\begin{aligned}
|\mathcal{I}_t| - |\widetilde{V}| &= |C_t| + |C_t'| + \sum_{i=1}^{t-1} (|C_i| + |C_i'|) \\
&\in \mathcal{O}\left((\log n_{t-1} + n_t) \cdot \nu_1(G^*)\right) + \mathcal{O}\left(\log n_{t-1} \cdot \nu_1(G^*)\right) \qquad \text{By Equation (3) and Equation (7)} \\
&\subseteq \mathcal{O}\left(\log |N^{h(G^*, \widetilde{V})}(\widetilde{V})| \cdot \nu_1(G^*)\right) \qquad \text{Equation (9)}
\end{aligned}
$$

Since $|\widetilde{V}| \in \mathcal{O}(\nu_1(G^*))$, we can conclude

$$|\mathcal{I}_t| \in \mathcal{O}\left(\nu(G^*) + \log |N^{h(G^*, \widetilde{V})}(\widetilde{V})| \cdot \nu_1(G^*)\right) \subseteq \mathcal{O}\left(\max\left\{1, \log |N^{h(G^*, \widetilde{V})}(\widetilde{V})|\right\} \cdot \nu_1(G^*)\right)$$

in either scenario, as desired. The theorem then follows by taking a maximization over all $\widetilde{V} \in \mathcal{V}(\widetilde{G})$.

Adapting the proof for $k > 1$

By Theorem 4.2, $\nu_k(G^*) \geq \lceil \nu_1(G^*)/k \rceil$. So, $|\mathcal{I}_0| \in \mathcal{O}(\log k \cdot \nu_k(G^*))$ via Lemma 4.3. The rest of the proof follows the same structure except that we use the bounded size guarantee of Theorem 2.9, which incurs an additional multiplicative $\log k$ factor.

Polynomial running time

By construction, the Algorithm 1 is deterministic. Furthermore, Algorithm 1 runs in polynomial time because:

- Hop information and relevant nodes can be computed in polynomial time via breadth first search and maintaining suitable neighborhood information.

- It is known that performing Meek rules to obtain essential graphs takes polynomial time ((Wienöbst et al., 2021a)).

- Algorithm 1 makes at most two calls to `SubsetSearch` whenever the number of relevant nodes is squared. Each `SubsetSearch` call is known to run in polynomial time (Theorem 2.9). Since this happens each time the number of relevant nodes is squared, this can happen at most $\mathcal{O}(\log n)$ times.

$\square$

Theorem E.1. *Fix an essential graph $\mathcal{E}(G^*)$ with an unknown underlying ground truth DAG G^*. Given a set $\mathcal{A}$ of DAGs consistent with the given partial advice and intervention set bound $k \geq 1$, there exists 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. $\mathcal{O}(\max\{1, \log \max_{\widetilde{G} \in \mathcal{A}} \psi(G^*, \widetilde{G})\} \cdot \nu_1(G^*))$
2. $\mathcal{O}(\max\{1, \log \max_{\widetilde{G} \in \mathcal{A}} \psi(G^*, \widetilde{G})\} \cdot \log k \cdot \nu_k(G^*))$
when $k = 1$ and $k > 1$ respectively.

Proof. Apply Theorem 3.5 while taking a maximization over all possible advice DAGs $\widetilde{G}$ consistent with the given partial advice. $\square$

H. Path essential graph

In this section, we explain why our algorithm (Algorithm 1) is simply the classic "binary search with prediction"[13] when the given essential graph $\mathcal{E}(G^*)$ is an undirected path on n vertices. So, another way to view our result is a *generalization* that works on essential graphs of arbitrary moral DAGs.

---

[13]e.g. see https://en.wikipedia.org/wiki/Learning_augmented_algorithm#Binary_search

When the given essential graph is a path $\mathcal{E}(G^*)$ on n vertices, we know that there are n possible DAGs in the Markov equivalence class where each DAG corresponds to choosing a single root node and having all edges pointing away from it. Observe that a verifying set of any DAG is then simply the root node as the set of of covered edges in any rooted tree are precisely the edges incident to the root.

Therefore, given any $\widetilde{G} \in [G^*]$, we se that $h(G^*, \widetilde{V})$ measures the number of hops between the root of the advice DAG $\widetilde{G}$ and the root of the true DAG G^*. Furthermore, by Meek rule R1, whenever we intervene on a vertex u on the path, we will fully orient the "half" of the path that points away from the root while the subpath between u and the root remains unoriented (except the edge directly incident to u). So, one can see that Algorithm 1 is actually mimicking exponential search from the root of $\widetilde{G}$ towards the root of G^*. Then, once the root of G^* lies within the r-hop neighborhood H, `SubsetSearch` uses $\mathcal{O}(\log |V(H)|)$ interventions, which matches the number of queries required by binary search within a fixed interval over $|V(H)|$ nodes.

I. Experiments

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

All experiments were run on a laptop with Apple M1 Pro chip and 16GB of memory. Source code implementation and experimental scripts are available at https://github.com/cxjdavin/active-causal-structure-learning-with-advice.

I.1. Experimental setup

For experiments, we evaluated our advice algorithm on the synthetic graph instances of (Wienöbst et al., 2021b)[14] on graph instances of sizes $n = \{16, 32, 64\}$. For each undirected chordal graph instance, we do the following:

1. Set $m = 1000$ as the number of advice DAGs that we will sample.

2. Use the uniform sampling algorithm of (Wienöbst et al., 2021b) to uniformly sample m advice DAGs $\widetilde{G}_1, \ldots, \widetilde{G}_m$.

3. Randomly select G^* from one of $\widetilde{G}_1, \ldots, \widetilde{G}_m$.

4. For each $\widetilde{G} \in \{\widetilde{G}_1, \ldots, \widetilde{G}_m\}$,

   - Compute a minimum verifying set $\widetilde{V}$ of $\widetilde{G}$.
   - Define and compute $\psi(G^*, \widetilde{V}) = \left| \rho \left(\widetilde{V}, N^{h(G^*, \widetilde{V})}_{\text{skel}(\mathcal{E}(G^*))}(\widetilde{V}) \right) \right|$.
   - Compute a verifying set using $(\mathcal{E}(G^*), \widetilde{G})$ as input to Algorithm 1.

5. Aggregate the sizes of the verifying sets used based on $\psi(G^*, \widetilde{V})$ and compute the mean and standard deviations.

6. Compare against verification number $\nu_1(G^*)$ and the number of interventions used by the fully adaptive search (without advice, which we denote as "blind search" in the plots) of (Choo et al., 2022).

7. Compute the empirical distribution of the quality measure amongst the m advice DAGs, then use standard sample complexity arguments for estimating distributions up to ε error in TV distance to compute a confidence interval for which the true cumulative probability density of all DAGs within the MEC lies within[15]. To be precise, it is known that for a discrete distribution P on k elements, when there are $m \geq \max\{k/\varepsilon^2, (2/\varepsilon^2) \cdot \ln(2/\delta)\}$ uniform samples, the probability that the TV distance between the true distribution P and the empirical distribution P is less than ε is at least $1 - \delta$. Since the *upper bound* on the domain size of quality measure is the number of nodes n, by setting $m = 1000$ and $\delta = 0.01$, we can compute $\varepsilon = \max\{\sqrt{n/m}, \sqrt{(2/m) \cdot \ln(2/\delta)}\}$ and conclude that the probability that the true cumulative probability density of all DAGs within the MEC lies within ε distance (clipped to be between 0 and 1) of the empirical distribution is at least 99%.

---

[14]See Appendix E of (Wienöbst et al., 2021b) for details about each class of synthetic graphs. Instances are available at https://github.com/mwien/CliquePicking/tree/master/aaai_experiments
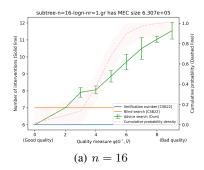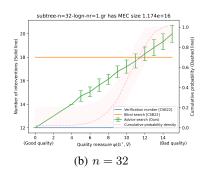[15]For example, see Theorem 1 of (Canonne, 2020).

I.2. Experimental remarks

- The uniform sampling code of (Wienöbst et al., 2021b) is written in Julia and it uses a non-trivial amount of memory, which may make it unsuitable for running on a shared server with memory constraints.

- Note that $\psi(G^*, \widetilde{V}) \leq \psi(G^*, \widetilde{G}) = \max_{\widetilde{V} \in \mathcal{V}(\widetilde{G})} \left| \rho \left(\widetilde{V}, N^{h(G^*, \widetilde{V})}_{\text{skel}(\mathcal{E}(G^*))}(\widetilde{V}) \right) \right|$. We use $\psi(G^*, \widetilde{V})$ as a proxy for $\psi(G^*, \widetilde{G})$ because we do not know if there is an efficient way to compute the latter besides the naive (possibly exponential time) enumeration over all possible minimum verifying sets.

- We also experimented with an "unsafe" variant of Algorithm 1 where we ignore the second tweak of intervening one round before. In our synthetic experiments, both variants use a similar number of interventions.

- We do not plot the theoretical upper bounds $\mathcal{O}(\log \psi(G^*, \widetilde{V}) \cdot \nu_1(G^*))$ or $\mathcal{O}(\log n \cdot \nu_1(G^*))$ because these values are a significantly higher than the other curves and result in "squashed" (and less interesting/interpretable) plots.

- Even when $\psi(G^*, \widetilde{V}) = 0$, there could be cases where (Choo et al., 2022) uses more interventions than $\nu_1(G^*)$. For example, consider Figure 8 with $G^* = G_2$ and $\widetilde{G} = G_1$. After intervening on $\widetilde{V} = \{b, c\}$, the entire graph will be oriented so the $\psi(G^*, \widetilde{V}) = 0$ while $\nu_1(G^*) = 1 < 2 = |\widetilde{V}|$. Fortunately, Theorem 3.1 guarantees that $|\widetilde{V}| \leq 2 \cdot \nu_1(G^*)$.

- Note that the error bar may appear "lower" than the verification number even though all intervention sizes are at least as large as the verification number. For instance, if $\nu_1(G^*) = 6$ and we used $(6, 6, 7)$ interventions on three different $\widetilde{G}$'s, each with $\psi(G^*, \widetilde{V}) = 0$. In this case, the mean is $6.3333\ldots$ while the standard deviation is $0.4714\ldots$, so the error bar will display an interval of $[5.86\ldots, 6.80\ldots]$ whose lower interval is below $\nu_1(G^*) = 6$.

I.3. All experimental plots

For details about the synthetic graph classes, see Appendix E of (Wienöbst et al., 2021b). Each experimental plot is for one of the synthetic graphs G^*, with respect to $1000 \ll |[G^*]|$ uniformly sampled advice DAGs $\widetilde{G}$ from the MEC $[G^*]$. The solid lines indicate the number of atomic interventions used while the dotted lines indicate the empirical cumulative probability density of $\widetilde{G}$. The true cumulative probability density lies within the shaded area with probability at least 0.99.

(a) $n = 16$ (b) $n = 32$ (c) $n = 64$
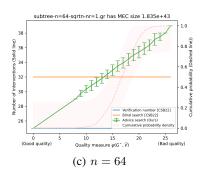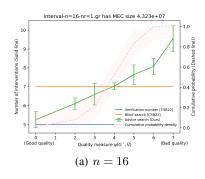
Figure 9. Subtree-logn synthetic graphs

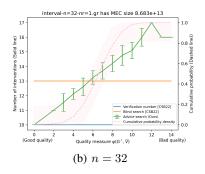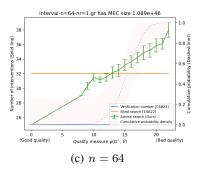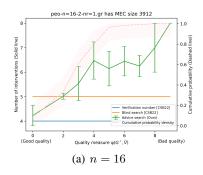*Figure 10.* Subtree-2logn synthetic graphs
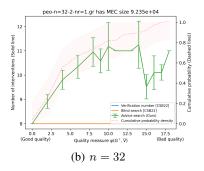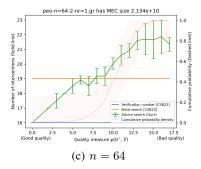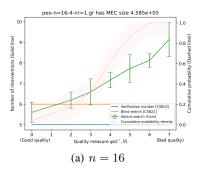

Figure 11. Subtree-sqrtn synthetic graphs
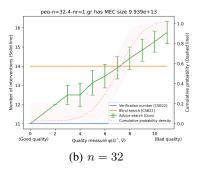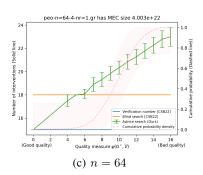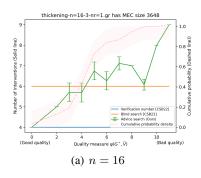

Figure 12. Interval synthetic graphs
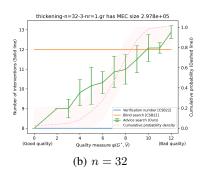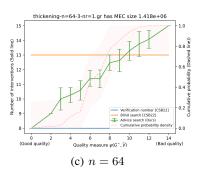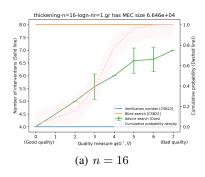

Figure 13. peo-2 synthetic graphs
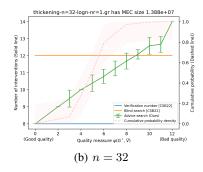
Figure 14. peo-4 synthetic graphs

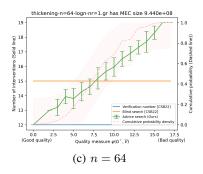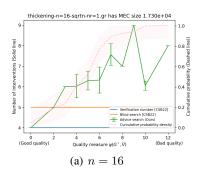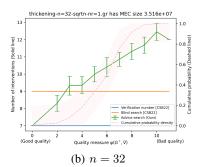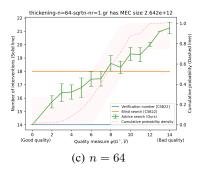*Figure 15.* Thickening-3 synthetic graphs

Figure 16. Thickening-logn synthetic graphs

Figure 17. Thickening-sqrtn synthetic graphs