Appendices

Contents

A Background and related works

A.1 Definitions and previous results

First, we provide definitions and results used in the main text.

Definition A.1 (d-separation [41])**.** Given a causal DAG $\mathcal{G}$, a node W on a path π is said to be a collider on π if W has converging arrows into W in π, e.g., $\rightarrow W \leftarrow$ or $\leftrightarrow W \leftarrow$. π is said to be blocked by a set $\mathbf{Z}$ if there exists a node W on π satisfying one of the following two conditions: 1) W is a collider, and neither W nor any of its descendants are in $\mathbf{Z}$, or 2) W is not a collider, and W is in $\mathbf{Z}$. Given disjoint sets $\mathbf{X}, \mathbf{Y}$, and $\mathbf{Z}$ in $\mathcal{G}$, $\mathbf{Z}$ is said to *d-separate* $\mathbf{X}$ from $\mathbf{Y}$ in $\mathcal{G}$ if $\mathbf{Z}$ blocks every path from a node in $\mathbf{X}$ to a node in $\mathbf{Y}$ according to the d-separation criterion.

Definition A.2 (DELETE operator, [9, Def. 13])**.** For adjacent nodes X, Y in $\mathcal{E}$ connected as either $X \rightarrow Y$ or $X - Y$, and for any $\mathbf{H} \subseteq \mathbf{Ne}_Y^{\mathcal{E}} \cap \mathbf{Adj}_X^{\mathcal{E}}$, the $\text{DELETE}(X, Y, \mathbf{T})$ operator modifies $\mathcal{E}$ by deleting the edge between X and Y, and for each $T \in \mathbf{T}$, directing any undirected edges $X - T$ as $X \rightarrow T$ and any $Y - T$ as $Y \rightarrow T$.

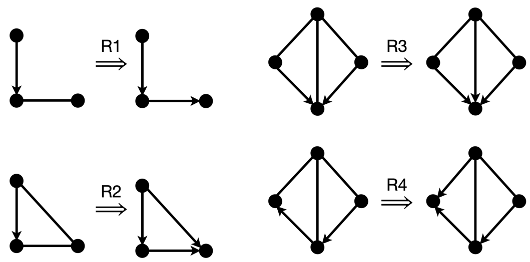


Figure A.1.1: Meek orientation rules for completing partially directed acyclic graphs

The properties and implementation of the TURN operator can be found in [24, Sec. 4.3], though the authors do not provide an exact definition we can reproduce here.

Definition A.3 (Decomposable scoring criterion [9, Sec. 2.3])**.** Let $\mathbf{D}$ be a set of data consisting of iid samples from some distribution $P(\mathbf{v})$. A scoring criterion S is said to be *decomposable* if it can be written as a sum of measures, each of which is a function of only a single node and its parents, as

$$S(\mathcal{G}, \mathbf{D}) = \sum_{V_i \in \mathbf{V}} s(v_i, pa_i^{\mathcal{G}})$$

Each local score $s(v_i, pa_i^{\mathcal{G}})$ depends only on the values of V_i and $\mathbf{Pa}_i$ in $\mathbf{D}$.

Definition A.4 (Consistent scoring criterion [9, Def. 5])**.** Let $\mathbf{D}$ be a set of data consisting of iid samples from some distribution $P(\mathbf{v})$. A scoring criterion S is said to be *consistent* if, as the number of samples goes to infinity, the following two properties hold for any DAGs $\mathcal{G}, \mathcal{H}$:

1. If $P(\mathbf{v})$ is Markov with respect to $\mathcal{G}$ but not $\mathcal{H}$, then $S(\mathcal{G}, \mathbf{D}) > S(\mathcal{H}, \mathbf{D})$.
2. If $P(\mathbf{v})$ is Markov with respect to both $\mathcal{G}$ and $\mathcal{H}$, but $\mathcal{G}$ contains fewer free parameters than $\mathcal{H}$, then $S(\mathcal{G}, \mathbf{D}) > S(\mathcal{H}, \mathbf{D})$.

Definition A.5 (Score-equivalent scoring criterion [9, Sec 2.3])**.** Let $\mathbf{D}$ be a set of data consisting of iid samples from some distribution $P(\mathbf{v})$. A scoring criterion S is said to be *score-equivalent* if, as the number of samples goes to infinity, for any two DAGs $\mathcal{G}, \mathcal{H}$ that are Markov equivalent, $S(\mathcal{G}, \mathbf{D}) = S(H, \mathbf{D})$.

Definition A.6 (Soft unconditional intervention [24, Sec. 2.1])**.** A soft unconditional intervention on a set of variables $\mathbf{X}$ sets the value of each variable $V_i \in \mathbf{X}$ to an independent random variable U_i from a given set of random variables $\mathbf{U}$. The resulting distribution is given by

$$P_{\mathbf{X}}(\mathbf{v}) = \prod_{V_i \notin \mathbf{X}} P(v_i \mid pa_i) \prod_{V_i \in \mathbf{X}} P^*(v_i)$$

where $P^*(v_i)$ denotes the distribution of $U_i \in \mathbf{U}$ corresponding to $V_i \in \mathbf{X}$.

Definition A.7 ($\mathcal{I}$-Markov property [24, Def. 7])**.** Let $\mathbf{V}$ be a set of variables, $\mathcal{G}$ a causal DAG over $\mathbf{V}$, $\mathcal{I}$ a family of interventional targets, and $(\mathbf{P_I}(\mathbf{v}))_{\mathbf{I} \in \mathcal{I}}$ a corresponding family of interventional distributions. We say $(\mathbf{P_I}(\mathbf{v}))_{\mathbf{I} \in \mathcal{I}}$ satisfies the $\mathcal{I}$-Markov Property of $\mathcal{G}$ if:

1. Each $P_{\mathbf{I}}(\mathbf{v})$ is Markov with respect to the interventional graph $\mathcal{G}_{\overline{\mathbf{I}}}$, and
2. For interventions $\mathbf{I}, \mathbf{J} \in \mathcal{I}$ and variables $V_i \notin \mathbf{I} \cup \mathbf{J}$, $P_{\mathbf{I}}(v_i \mid pa_i) = P_{\mathbf{J}}(v_i \mid pa_i)$.

We let $\mathcal{M}_{\mathcal{I}}(\mathcal{G})$ denote the set of all $(\mathbf{P_I}(\mathbf{v}))_{\mathbf{I} \in \mathcal{I}}$ that are $\mathcal{I}$-Markov with respect to $\mathcal{G}$. Two causal DAGs $\mathcal{G}, \mathcal{H}$ are *$\mathcal{I}$-Markov equivalent* if $\mathcal{M}_{\mathcal{I}}(\mathcal{G}) = \mathcal{M}_{\mathcal{I}}(\mathcal{H})$.

Meek orientation rules. In Fig. A.1.1, we provide Meek's orientation rules used in $\mathcal{I}$-ORIENT to orient an $\mathcal{I}$-MEC. These rules provide an algorithm for completing a PDAG to a *completed* PDAG. They are applied repeatedly to a PDAG until no eligible motifs exist.

Next, introduce some additional definitions and results that will be used in Sec. C.

The *skeleton* of a causal DAG $\mathcal{G}$ (denoted $skel(\mathcal{G})$) is the undirected graph that results from ignoring the edge directions of every edge in $\mathcal{G}$. A triplet of variables (X, Z, Y) in $\mathcal{G}$ is said to be *unshielded* if (X, Z) and (Y, Z) are adjacent but (X, Y) are not. An unshielded triplet is said to be a *v-structure* (or *unshielded collider*) if it is oriented as $X \to Z \leftarrow Y$ in $\mathcal{G}$.

Theorem A.1 (Graphical criterion for Markov equivalence [57, Thm. 1])**.** *Two DAGs are Markov equivalent if and only if they have the same skeletons and same v-structures.*

Based on the above characterization, to obtain the CPDAG for the MEC corresponding to a DAG $\mathcal{G}$, one adds an undirected edge for every adjacency in $\mathcal{G}$; orients any v-structures according to $\mathcal{G}$; then applies Meek's orientation rules to complete the resulting PDAG to a CPDAG.

Definition A.8 (Global Markov property [41])**.** A probability distribution $P(\mathbf{v})$ over a set of variables $\mathbf{V}$ is said to satisfy the global Markov property for a causal DAG $\mathcal{G}$ if, for arbitrary disjoint sets $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subset \mathbf{V}$ with $\mathbf{X}, \mathbf{Y} \neq \emptyset$,

$$\mathbf{X} \perp_d \mathbf{Y} | \mathbf{Z} \implies \mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z} \text{ in } P(\mathbf{v}).$$

Let $\mathbf{Nd}_X^{\mathcal{G}}$ denote the set of non-descendants of a variable X in $\mathcal{G}$, i.e. variables in $\mathcal{G}$ (excluding X itself) to which there is no directed path from X.

Definition A.9 (Local Markov property [41])**.** A probability distribution $P(\mathbf{v})$ over a set of variables $\mathbf{V}$ is said to satisfy the local Markov property for a causal DAG $\mathcal{G}$ if, for every variable $X \in \mathbf{V}$,

$$X \perp\!\!\!\perp \mathbf{Nd}_X^{\mathcal{G}} \mid \mathbf{Pa}_X^{\mathcal{G}} \text{ in } P(\mathbf{v}).$$

Proposition A.1 (Equivalence of Local and Global Markov Properties [30, Prop. 4])**.** *Let $\mathcal{G}$ be a causal DAG over variables $\mathbf{V}$. A probability distribution over $\mathbf{V}$ satisfies the global Markov property for $\mathcal{G}$ if and only if it satisfies the local Markov property for $\mathcal{G}$.*

Definition A.10 (Covered edge [8, Def. 2])**.** An edge $X \to Y$ in a given causal DAG $\mathcal{G}$ is said to be *covered* if $\mathbf{Pa}_Y^{\mathcal{G}} = \mathbf{Pa}_X^{\mathcal{G}} \cup \{X\}$.

Lemma A.1 (Covered edge reversal [8, Lemma. 1])**.** *Let $\mathcal{G}$ be any causal DAG containing the edge $X \to Y$ and let $\mathcal{G}'$ be the DAG that is identical to $\mathcal{G}$ except it instead contains the edge $Y \to X$. Then, $\mathcal{G}'$ is a DAG that is Markov equivalent to $\mathcal{G}$ iff the edge $X \to Y$ is covered in $\mathcal{G}$.*

Theorem A.2 (Transformational characterization of Markov equivalent graphs [8, Thm. 2])**.** *Let $\mathcal{G}, \mathcal{G}'$ be a pair of Markov equivalent causal DAGs. Then, there exists a sequence of covered edge reversals transforming $\mathcal{G}$ to $\mathcal{G}'$.*

Theorem A.3 (Chickering-Meek theorem [9, Thm. 4])**.** *let $\mathcal{G}$ and $\mathcal{H}$ be a pair of causal DAGs such every d-separation that holds in $\mathcal{H}$ also holds in $\mathcal{G}$. Then, there exists a sequence of edge additions and covered edge reversals transforming $\mathcal{G}$ to $\mathcal{H}$ such that after each reversal and addition, $\mathcal{G}$ is DAG and every d-separation that holds $\mathcal{H}$ also holds in $\mathcal{G}$.*

A.2 Related works

A.2.1 Learning from observational data

Algorithms for causal discovery fall into three broad categories: constraint-based, score-based, and hybrid. Constraint-based algorithms like PC [54] and the Sparsest Permutations (SP) algorithm [44] learn the true MEC using statistical tests for whether the chosen type of constraints, typically conditional independencies, hold in the data. A challenge to these approaches is improving the accuracy of conditional independence tests, for e.g. in controlling type I error [52]. Score-based algorithms such as GES [9, 34] use a scoring criterion that reflects fit between data and graph, typically in the form of a likelihood plus a complexity penalty. Hybrid algorithms such as max-min hill climbing [56] use a combination of the two approaches, for e.g., first learning the skeleton using a constraint-based method then orienting edges in the skeleton using a score. There is no general claim about the relative accuracy of these methods; we refer readers to [56] for an extensive empirical analysis, who found, for instance, that GES outperforms PC in accuracy across various sample sizes [56, Tables 4, 5] .

Commonly, causal discovery algorithms struggle with scaling in high-dimensional settings. This has motivated variants such as Parallel-PC [31] and Fast Greedy Equivalence Search (FGES) [43], which offer faster, parallelized implementations of the algorithms. While FGES offers an additional heuristic over GES, i.e., not adding any edge $X \to Y$ in the forward phase if X, Y are uncorrelated in the data, this heuristic is not theoretically guaranteed to recover the true MEC. More recent continuous-optimization based approaches such as NoTears [61] are in principle more scalable, but lack theoretical guarantees and show brittle performance even in simulated settings [38, 45].

A.2.2 Learning with prior knowledge

Causal discovery with background knowledge is a well-studied problem [14]. Such background knowledge may be provided by a domain expert or even a large language model [33].

Perfect background knowledge. Many algorithms under this umbrella, such as tiered-FCI [50], the K2 algorithm [15], and Knowledge-Guided GES [22], assume that the background knowledge is correct, i.e., consistent with the ground truth. They use this knowledge to constrain the search, for e.g., by never inserting user-forbidden edges. If the expert's background knowledge is imperfect, such methods necessarily fail to recover the true MEC. Even if the knowledge is perfect, it is unknown whether these methods will output the true graph; for e.g., in Knowledge-Guided GES [22], restricting insertions no longer guarantees reachability of an MEC with respect to which the given data is Markov in the forward phase, as in GES.

Misspecified background knowledge. The constraint-based approach in [39] allows some misspecification in the expert knowledge in the form of missing or excess adjacencies but not incorrect orientations. However, their approach does not guarantee recovery of the true MEC. Certain score-based approaches treat knowledge about edges a 'soft' prior [1, 4, 7, 26, 40] to guide the search, but lack theoretical guarantees on the output graph. One exception with theoretical guarantees is the Sparsest Permutations (SP) algorithm [44], which can initialize the search to an ordering over variables provided by an expert.

A.2.3 Learning from interventional data

Observational learning algorithms can only learn a causal graph up to its observational Markov equivalence class (MEC). The MEC is the limit of what can be identified from observational data without further assumptions. However, MECs can often be large and uninformative for downstream causal tasks [25]. Interventional data can help significantly refine observational MECs [24], and is becoming increasingly available, for e.g., in biological settings due to advances in single-cell technologies [17, 49]. This has motivated the design of algorithms for causal discovery from observational and interventional data such as the score-based Greedy Interventional Equivalence Search (GIES) [24] and the CI-based Interventional Greedy Sparsest Permutations (I-GSP) [58]. However, in [58], it was shown that GIES is inconsistent, i.e.. not guaranteed to recover the true interventional MEC in the sample limit. I-GSP is asymptotically consistent, but being permutation-based, struggles to scale in high-dimensional settings.

B Discussion and examples

B.1 Design of Less Greedy Equivalence Search

In this section, we elaborate on certain design choices made in the LGES algorithm.

Choice of DAG in SAFEINSERT. As specified in Def. 2 and Alg. 5, at a given MEC $\mathcal{E}$, SAFEINSERT chooses some DAG $\mathcal{G}$ from $\mathcal{E}$ to perform its check for conditional independence. In theory, any any procedure for choosing a DAG from an MEC suffices. In our implementation, we use Alg. 7, a simple polynomial-time algorithm for converting a CPDAG to DAG presented in [18] and used in the original GES [9] Another possibility is the polynomial-time algorithm for uniform random sampling of a DAG from an MEC [59]. We leave investigation of DAG choice to future work.

Backward phase of LGES. LGES only modifies the forward phase of GES, while keeping the backward phase intact. Since a score-decreasing INSERT implies independence under some conditioning set (Prop. 1), discarding all $X - Y$ insertions when a single score-decreasing INSERT$(X, Y, \mathbf{T})$ operator is found promotes more robust edge insertions, i.e., between variables that are not found to be conditionally independent. This significantly reduces false adjacencies in the learned graph (Fig. D.1.1). One may ask, why not use an analogous strategy in the backward phase, discarding all $X - Y$ deletions when a single score-decreasing DELETE$(X, Y, \mathbf{T})$ operator is found? This is because one score-decreasing deletion implies dependence given one conditioning set, not given any conditioning set. However, a true $X - Y$ adjacency implies dependence given any conditioning set. So, skipping all deletions due to one score decrease would lead to many false adjacencies and compromise soundness. A true adjacency would imply that all $X - Y$ deletes are score-decreasing, so LGES/GES will not apply any of them. Experiments support this: false non-adjacencies are rare (Fig. D.1.1).

Contrast with PC algorithm. There is a surface-level similarity between LGES and the PC algorithm [54]. The PC algorithm begins with a complete graph, then removes edges between variables for which it finds some separating set. In particular, for adjacent variables X, Y in the current graph $\mathcal{G}$, it iterates over all subsets $\mathbf{Z} \subseteq \mathbf{Adj}_X^{\mathcal{G}} \cup \mathbf{Adj}_Y^{\mathcal{G}}$ to find some $\mathbf{Z}$ such that $X \perp\!\!\!\perp Y \mid \mathbf{Z}$. If X, Y are non-adjacent in the ground-truth graph, then the PC algorithm is guaranteed to find a $\mathbf{Z}$ which separates them.

LGES, on the other hand, starts with an empty (or user-provided) graph. It then avoids inserting the edge $X \to Y$ if it finds a set separating X and Y. However, LGES only searches over restricted space of separating sets. ConservativeInsert checks if $X \perp\!\!\!\perp Y \mid \mathbf{Pa}_Y^{\mathcal{G}}$ for any $\mathcal{G}$ in the current MEC. SafeInsert checks if $X \perp\!\!\!\perp Y \mid \mathbf{Pa}_Y^{\mathcal{G}}$ for a fixed $\mathcal{G}$ in the current MEC. There may exist a set $\mathbf{Z}$ separating X and Y that does not equal $\mathbf{Pa}_Y^{\mathcal{G}}$ for some $\mathcal{G}$ in the current MEC. LGES may not find this $\mathbf{Z}$, and thus insert an edge between X and Y. In this way, it differs from the PC algorithm.

The advantage of LGES is that it does not require conducting any scoring operations beyond those which GES already conducts. On the other hand, an approach which iterates over all possible separating sets of X and Y would require significantly more scoring operations. Still, future work might consider the theoretical guarantees and empirical performance of such a score-based approach.

B.2 Limitations of Less Greedy Equivalence Search

Causal sufficiency. In this work, we assume that the underlying system is *Markovian*, i.e. no two observed variables have an unobserved common cause. This is also known as the *causal sufficiency* assumption. While this assumption is standard in causal discovery, it can be violated in practice. In settings with unobserved confounders, i.e., *non-Markovian* settings, the equivalence class of graphs that can be identified is typically even larger (and hence more uninformative) than in the Markovian case. One reason for this is that two variables may be non-adjacent in the true graph while still being inseparable by any set due to the existence of an *inducing path* between them [57, Def. 2]. As a result, performing causal inference from equivalence classes of non-Markovian graphs is challenging.

Still, there has been work on causal discovery in non-Markovian settings. Constraint-based approaches include the FCI algorithm [53, 60] and its interventional variants [27, 29, 32], guaranteed to recover the true equivalence class in the sample limit. While there has been progress towards score-based approaches [6, 13, 46, 55], finding algorithms that are asymptotically correct remains an open problem.

There is a fundamental theoretical challenge to generalizing our approach to the non-Markovian setting. GES relies on two useful properties of Markovian DAGs. First is the local Markov property [30] of such DAGs, allowing for a locally consistent scoring criterion (Def. 1). Second are the transformational characterizations of two Markovian causal DAGs $\mathcal{G}_1$ and $\mathcal{G}_2$ when (a) $\mathcal{G}_1$ and G_2 encode exactly the same d-separations (Thm. A.2) and (b) (a) $\mathcal{G}_1$ encodes a subset of the d-separations encoded in $\mathcal{G}_2$ (Thm. A.3). A 'transformational characterization' is a procedure whereby $\mathcal{G}_1$ can be transformed into $\mathcal{G}_2$ by a sequence of single-edge changes that satisfy certain criteria. The transformational characterization of (a) has been generalized to non-Markovian causal DAGS [60]. Moreover, there exists a local Markov property for non-Markovian causal DAGs, as well as an efficient algorithm to test condition (b) [28]. However, it remains an open problem to fully recover (b) in the non-Markovian setting.

Other assumptions. In this work, we assume we are given a distribution that is Markov and *faithful* with respect to some causal DAG. This is a standard assumption in causal discovery, often justified by the fact that the set of distributions that are Markov but not faithful with respect to a given DAG has Lebesgue measure zero. Moreover, if we assume only that the given distribution is Markov with respect to some causal DAG, we can never rule out the true DAG being the fully connected DAG. Still, there has been work on relaxing the faithfulness assumption, giving rise to the Sparsest Permutations algorithm [44].

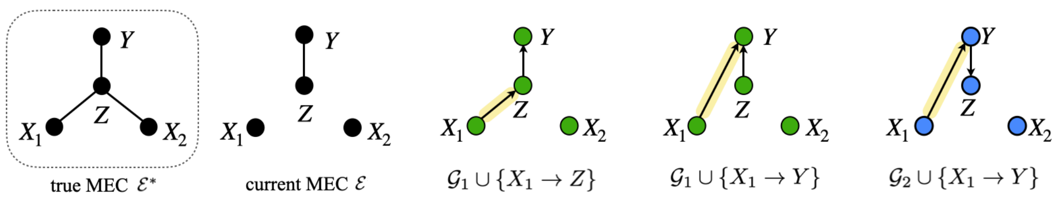


Figure B.3.1: Figs. 2, 3 partially reproduced for convenience. GES is given a distribution $P(\mathbf{v})$ whose true MEC is represented by $\mathcal{E}^*$ (left). GES is currently at MEC $\mathcal{E}$, evaluating which INSERT operator to apply next. Each INSERT corresponds to picking some $\mathcal{G} \in \mathcal{E}$ and adding some edge to it.

We further assume in this work that we are given a scoring criterion that is decomposable and consistent. This does not strictly mean we make parametric assumptions. Existing scores such as BIC satisfy these criteria as long as the model is a *curved exponential family* [9, 23]. This includes multinomial (discrete) and linear-Gaussian models. For continuous data, the linear-Gaussian assumption can be violated in practice. In this case, one can discretize the data before using it for causal discovery, as in [48]. However, since the parameter space of multinomial models is quite large, and information is lost during discretization, it would be valuable future work to investigate scores for other models of continuous data.

B.3 Extended example of GES trajectories

Finally, we return to our explanation of the two trajectories GES might take in Ex. 2, Fig. 2.

Example B.1. (Ex. 2 continued). Recall that GES is given a distribution $P(\mathbf{v})$ whose true MEC is $\mathcal{E}^*$ (Fig. B.3.1, left). GES is currently at the MEC $\mathcal{E}$, evaluating which INSERT operator to apply. Each INSERT operator corresponds to picking some DAG $\mathcal{G} \in \mathcal{E}$ and adding some edge to it. One such operator corresponds to picking $\mathcal{G}_1 \in \mathcal{E}$, and adding the edge $X_1 \to Z$ to it. Another such operator corresponds to picking $\mathcal{G}_2 \in \mathcal{E}$, and adding the edge $X_1 \to Y$ to it. Both operators, shown in Fig. B.3.1, result in a score increase by local consistency (Def. 1) since $Z \not\perp\!\!\!\perp X_1 \mid \mathbf{Pa}_Z^{\mathcal{G}_1}$ and $Y \not\perp\!\!\!\perp X_1 \mid \mathbf{Pa}_Y^{\mathcal{G}_2}$ in $P(\mathbf{v})$. Although $\mathcal{G}_1 \cup X_1 \to Z$ looks 'closer' to the true MEC $\mathcal{E}^*$ than $\mathcal{G}_2 \cup X_! \to Y$, when tested empirically, the latter often scores more than the former (Ex. 2).

Moreover, even in the sample limit, the consistency (Def. A.4) of the scoring criterion does not guarantee that $\mathcal{G}_2 \cup \{X_1 \to Y\}$ will score lower than $\mathcal{G}_1 \cup \{X_1 \to Z\}$. Neither the global nor the local consistency of the score provide an immediate guarantee for which will score higher. Global consistency only allows us to compare graphs when $P(\mathbf{v})$ is Markov with respect to at least one of them; however, $P(\mathbf{v})$ is not Markov with respect to $\mathcal{G}_2 \cup \{X_1 \to Y\}$ or $\mathcal{G}_1 \cup \{X_1 \to Z\}$. Local consistency (Def. 1) does not let us compare these graphs either, since they differ by more than an edge addition. Therefore, GES may move either to the MEC of $\mathcal{G}_1 \cup \{X_1 \to Z\}$ (as in τ_1, Fig. 2) or to the MEC of $\mathcal{G}_2 \cup \{X_1 \to Y\}$ (as in τ_2, Fig. 2). It is unknown a priori which operator is the highest-scoring. □

C Proofs and pseudocode

In this section, we provide proofs, additional results, and pseudocode for the algorithms of the main paper.

Theorem 1 (Correctness of GGES). Let $\mathcal{E}$ denote the Markov equivalence class that results from GGES (Alg. 3) initialized from an arbitrary MEC $\mathcal{E}_0$ and let $P(\mathbf{v})$ denote the distribution from which the data $\mathbf{D}$ was generated. Then, as the number of samples goes to infinity, $\mathcal{E}$ is the Markov equivalence class underlying $P(\mathbf{v})$.

Proof. The proof is similar to that of [9, Lemma 9, 10]. First, we will prove the correctness of GGES run with the forward-turning-backward phase order.

Forward phase. First, we show that $P(\mathbf{v})$ is Markov with respect to the MEC $\mathcal{E}'$ resulting from the forward phase of GGES. Let $\mathcal{G}$ be any DAG in $\mathcal{E}'$. By assumption, since GETOPERATOR does not find any valid score-increasing INSERT operators, there exists no such operator. Since there exist no score-increasing INSERT operators, the local consistency of the scoring criterion (Def. 1) implies that for every $X \in \mathcal{G}$ and $Y \in \mathbf{Nd}_X^{\mathcal{G}}$, $X \perp\!\!\!\perp Y \mid \mathbf{Pa}_X^{\mathcal{G}}$. Otherwise, if we had some $X \in \mathcal{G}$ and $Y \in \mathbf{Nd}_X^{\mathcal{G}}$ such that $X \not\perp\!\!\!\perp Y \mid \mathbf{Pa}_X^{\mathcal{G}}$, the INSERT$(Y, X, *)$ operator corresponding to $\mathcal{G} \cup \{Y \to X\}$ would result in a score increase. Since $P(\mathbf{v})$ is faithful to some DAG, it satisfies the *composition axiom* of conditional independence [41]: if $X \perp\!\!\!\perp Y \mid \mathbf{Pa}_X^{\mathcal{G}}$ for every $Y \in \mathbf{Nd}_X^{\mathcal{G}}$, then $X \perp\!\!\!\perp \mathbf{Nd}_X^{\mathcal{G}} \mid \mathbf{Pa}_X^{\mathcal{G}}$. Since this is true for every X, $P(\mathbf{v})$ satisfies the local Markov property of $\mathcal{G}$. By the equivalence of the global and local Markov properties (Prop. A.1), this means every d-separation in $\mathcal{G}$ implies a corresponding CI in $P(\mathbf{v})$. Thus, $P(\mathbf{v})$ is Markov with respect to $\mathcal{G}$ and hence to $\mathcal{E}'$.

Turning phase. Next, we show that $P(\mathbf{v})$ is Markov with respect to the MEC $\mathcal{E}''$ resulting from the turning phase of GGES. The turning phase starts with $\mathcal{E}'$, output by the forward phase. We have shown that $P(\mathbf{v})$ is Markov with respect to $\mathcal{E}'$. By construction, each TURN operator applied to the current MEC in the backward phase results in a score increase If any operator resulted in an $\mathcal{E}''$ with respect to which $P(\mathbf{v})$ is not Markov, the consistency of the scoring criterion A.4 implies that it would decrease the score. Therefore, $P(\mathbf{v})$ must be Markov with respect to $\mathcal{E}''$.

Backward phase. Finally, we show that $P(\mathbf{v})$ is both Markov and faithful to the MEC $\mathcal{E}$ resulting from the backward phase of GGES. The argument that $P(\mathbf{v})$ is Markov with respect to $\mathcal{E}$ is similar to that of the turning phase above. It remains to show that $P(\mathbf{v})$ is faithful to $\mathcal{E}$. Let $\mathcal{E}^*$ be the true MEC underlying $P(\mathbf{v})$. Since $P(\mathbf{v})$ is both Markov and faithful with respect to $\mathcal{E}^*$, and $P(\mathbf{v})$ is Markov with respect to $\mathcal{E}$, every d-separation in $\mathcal{E}$ must also hold in $\mathcal{E}^*$. Then, by the Chickering-Meek theorem (Thm. A.3), for any $\mathcal{G} \in \mathcal{E}$ and $\mathcal{H} \in \mathcal{E}^*$, there exists a sequence of covered edge reversals and edge additions that transform $\mathcal{H}$ to $\mathcal{G}$. If this sequence only contains covered edge reversals, then $\mathcal{H}$ and $\mathcal{G}$ are Markov-equivalent (Lemma. A.1), and we are done. Otherwise, let the last edge addition in this sequence add the edge $X \to Y$, resulting in the DAG $\mathcal{G}'$. Since $\mathcal{G}'$ can be transformed to $\mathcal{G}$ by a sequence of covered edge reversals, they are Markov equivalent, and we have $\mathcal{G}' \in \mathcal{E}$ (Lemma. A.1). Moreover, since this sequence of transformations includes only covered edge reversals and edge additions, and $P(\mathbf{v})$ is Markov with respect to $\mathcal{H}$, $P(\mathbf{v})$ is also Markov with respect to $\mathcal{G}' \setminus \{X \to Y\}$ (by Lemma. A.1, covered edge reversals and additions do not create additional d-separations). By the consistency of the scoring criterion, $\mathcal{G}' \setminus \{X \to Y\}$ has a higher score than $\mathcal{G}' \in \mathcal{E}$ since the former has fewer parameters. The corresponding DELETE operator thus results in a score increase, and by assumption, GETOPERATOR is guaranteed to find some score-increasing deletion in this case. Thus, we have a contradiction.

Next, we will prove the correctness of GGES without the phase structure, i.e., when GETOPERATOR returns a score-increasing INSERT or DELETE operator if one exists. The proof is similar to the previous case. GGES terminates at an MEC $\mathcal{E}$ only when GETOPERATOR finds no score-increasing INSERT *or* DELETE operator. By assumption, this implies no such operator exists. If no score-increasing increasing INSERT operator exists, by the same argument as for the forward phase above, the given $P(\mathbf{v})$ must be Markov with respect to $\mathcal{E}$. Given that $P(\mathbf{v})$ is Markov with respect to $\mathcal{E}$, and, furthermore, no score-increasing increasing DELETE operator exists, then by the same argument as for the backward phase above, the given $P(\mathbf{v})$ must be both Markov and faithful to $\mathcal{E}$. □

Proposition 1. Let $\mathcal{E}$ denote an arbitrary CPDAG and let $P(\mathbf{v})$ denote the distribution from which the data $\mathbf{D}$ was generated. Assume, as the number of samples goes to infinity, that there exists a valid score-decreasing INSERT$(X, Y, \mathbf{T})$ operator for $\mathcal{E}$. Then, there exists a DAG $\mathcal{G} \in \mathcal{E}$ such that (1) $Y \perp_d X \mid \mathbf{Pa}_Y^{\mathcal{G}}$ and (2) $Y \perp\!\!\!\perp X \mid \mathbf{Pa}_Y^{\mathcal{G}}$ in $P(\mathbf{v})$.

Proof. The score change of a valid INSERT$(X, Y, \mathbf{T})$ corresponds to picking a DAG $\mathcal{G} \in \mathcal{E}$ and comparing $S(\mathcal{G}, \mathbf{D})$ with $S(\mathcal{G} \cup \{X \to Y\}, \mathbf{D})$. By local consistency (Def. 1), if $Y \not\perp\!\!\!\perp X \mid \mathbf{Pa}_Y^{\mathcal{G}}$, then the score must increase. By contrapositive, we have $Y \perp\!\!\!\perp X \mid \mathbf{Pa}_Y^{\mathcal{G}}$ in $P(\mathbf{v})$. Moreover, since X must be a non-descendant of Y for $\mathcal{G} \cup \{X \to Y\}$ to be a DAG, $Y \perp_d X \mid \mathbf{Pa}_Y^{\mathcal{G}}$ in $\mathcal{G}$. □

We provide the following guarantee for LGES Alg. 1 run with CONSERVATIVEINSERT (Strategy 1).

Proposition C.1 (Partial guarantee on CONSERVATIVEINSERT)**.** *Let LGES* denote the variant of LGES (Alg. 1) that uses* CONSERVATIVEINSERT *instead of* SAFEINSERT *in the forward phase. Let $\mathcal{E}$ denote the equivalence class that results from the forward phase of LGES* initialized to an arbitrary MEC $\mathcal{E}_0$, let $P(\mathbf{v})$ denote the distribution from which the data $\mathbf{D}$ was generated, and let $\mathcal{E}^*$ be the true MEC underlying $P(\mathbf{v})$. Then, as the number of samples goes to infinity,*

1. $skel(\mathcal{E}^*) \subseteq skel(\mathcal{E})$

2. *For any unshielded triplet $(X, Z, Y) \in \mathcal{E}^*$, either X, Y are adjacent in $\mathcal{E}$ or (X, Z, Y) is a collider in $\mathcal{E}^*$ if and only if it is a collider in $\mathcal{E}$.*

Proof. For any variables X, Y adjacent in $\mathcal{E}^*$, since $P(\mathbf{v})$ is faithful to $\mathcal{E}^*$, X, Y are not independent in the data conditional on any set $\mathbf{Z} \subseteq \mathbf{V}$. Hence, for any $\mathcal{G} \in \mathcal{E}$, $X \not\perp\!\!\!\perp Y \mid \mathbf{Pa}_Y^{\mathcal{G}}$. Therefore, we always have $s(\mathcal{G}) < s(\mathcal{G} \cup \{X \to Y\})$ for any $\mathcal{G} \in \mathcal{E}$ such that $X \in \mathbf{Nd}_Y^{\mathcal{G}}$, and all valid INSERT$(X, Y, *)$ operators will result in a score increase. Hence, CONSERVATIVEINSERT will consider all such operators. Since $\hat{\mathcal{E}}$ is a local optimum of the score, any variables that are adjacent in $\mathcal{E}^*$ must also be adjacent in $\mathcal{E}$. Therefore, $skel(\mathcal{E}^*) \subseteq skel(\mathcal{E})$.

Then, consider some unshielded triplet $(X, Z, Y) \in \mathcal{E}^*$. Since $skel(\mathcal{E}^*) \subseteq skel(\mathcal{E})$, (X, Z) and (Y, Z) must be adjacent in $\mathcal{E}$. If (X, Y) are also adjacent in $\mathcal{E}$, we are done. Otherwise, we have an unshielded triplet $(X, Z, Y) \in \mathcal{E}$. Assume (X, Z, Y) is a collider in $\mathcal{E}^*$. Since CONSERVATIVEINSERT finds no score-increasing INSERT operators for $\mathcal{E}$, and X, Y are non-adjacent in $\mathcal{E}$, it must be the case that $\exists \mathcal{G} \in \mathcal{E}$ such that $Y \in \mathbf{Nd}_X^{\mathcal{G}}$ and $X \perp\!\!\!\perp Y \mid \mathbf{Pa}_X^{\mathcal{G}}$ or $X \in \mathbf{Nd}_Y^{\mathcal{G}}$ and $X \perp\!\!\!\perp Y \mid \mathbf{Pa}_Y^{\mathcal{G}}$. Without loss of generality, assume it is the former. Since (X, Z, Y) is a collider in $\mathcal{E}^*$, $X \not\perp\!\!\!\perp Y \mid \mathbf{Z}$ in $P(\mathbf{v})$ for any set $\mathbf{Z}$ containing Z. Therefore, it must be the case that $Z \notin \mathbf{Pa}_X^{\mathcal{G}}$. Hence, $\mathcal{G}$ contains the edge $X \to Z$. Since $Y \in \mathbf{Nd}_X^{\mathcal{G}}$, this further implies that $\mathcal{G}$ contains the edge $Y \to Z$. Therefore, (X, Z, Y) is a collider in $\mathcal{G}$ and hence $\mathcal{E}^*$. Next, assume that (X, Z, Y) is a collider in $\mathcal{E}$. Then, $Z \notin \mathbf{Pa}_X^{\mathcal{G}}$ and $Z \notin \mathbf{Pa}_Y^{\mathcal{G}}$ for all $\mathcal{G} \in \mathcal{E}$. As before, since CONSERVATIVEINSERT finds no score-increasing INSERT operators for $\mathcal{E}$, and X, Y are non-adjacent in $\mathcal{E}$, it must be the case that $\exists \mathcal{G} \in \mathcal{E}$ such that $Y \in \mathbf{Nd}_X^{\mathcal{G}}$ and $X \perp\!\!\!\perp Y \mid \mathbf{Pa}_X^{\mathcal{G}}$ or $X \in \mathbf{Nd}_Y^{\mathcal{G}}$ and $X \perp\!\!\!\perp Y \mid \mathbf{Pa}_Y^{\mathcal{G}}$. Then, conditioning on Z is not needed to separate X, Y in $P(\mathbf{v})$, which implies that (X, Z, Y) is also a collider in $\mathcal{E}^*$.

□

We give the following condition, sufficient to guarantee that CONSERVATIVEINSERT returns a score-increasing INSERT operator when one exists.

Proposition C.2 (Conditional guarantee on CONSERVATIVEINSERT)**.** *Let $\mathcal{E}$ denote a Markov equivalence class and let $P(\mathbf{v})$ denote the distribution from which the data $\mathbf{D}$ was generated.*

Assume the following holds.

> ***Assumption.*** *Let $\mathcal{G}, \mathcal{H}$ be two DAGs such that some d-separation encoded in $\mathcal{G}$ does not hold in $\mathcal{H}$. Then, there exists a pair of variables X, Y non-adjacent in $\mathcal{G}$ with $Y \in \mathbf{Nd}_X^{\mathcal{G}}$ such that for every $\mathcal{G}'$ Markov-equivalent to $\mathcal{G}$ with $Y \in \mathbf{Nd}_X^{\mathcal{G}'}$, $X \not\perp_d Y \mid \mathbf{Pa}_X^{\mathcal{G}'}$ in $\mathcal{H}$.*

Then, as the number of samples goes to infinity, CONSERVATIVEINSERT *returns a valid score-increasing* INSERT *operator if and only if one exists.*

Proof. Let $\mathcal{E}^*$ indicate the true MEC underlying $P(\mathbf{v})$. If there exists a valid score-increasing INSERT operator for the current state $\mathcal{E}$, then $P(\mathbf{v})$ is not Markov with respect to $\mathcal{E}$. Since $P(\mathbf{v})$ is faithful to $\mathcal{E}^*$, this implies that there exists some d-separation encoded in $\mathcal{E}$ that does not hold in $\mathcal{E}^*$. By the assumption, this implies that there exists X, Y non-adjacent in $\mathcal{E}$ such that for every $\mathcal{G}$ in $\mathcal{E}$ with $Y \in \mathbf{Nd}_X^{\mathcal{G}}$, $X \not\perp_d Y \mid \mathbf{Pa}_X^{\mathcal{G}}$ in $\mathcal{E}^*$ and hence $X \not\perp\!\!\!\perp Y \mid \mathbf{Pa}_X^{\mathcal{G}}$ in $P(\mathbf{v})$. Therefore, every INSERT$(Y, X, *$ operator results in a score increase for $\mathcal{E}$. Then, CONSERVATIVEINSERT is guaranteed to find a score-increasing INSERT. The reverse direction follows by construction, since CONSERVATIVEINSERT enumerates only valid INSERT operators and returns one only if it increases the score. □

As a corollary of the above and Thm. 1, we can also show that LGES with CONSERVATIVEINSERT instead of SAFEINSERT is guaranteed to recover the true MEC in the sample limit, if the assumption in Prop. C.2 holds. We leave the correctness of this assumption open.

Proposition 2 (Correctness of SAFEINSERT). Let $\mathcal{E}$ denote a Markov equivalence class and let $P(\mathbf{v})$ denote the distribution from which the data $\mathbf{D}$ was generated. Then, as the number of samples goes to infinity, SAFEINSERT returns a valid score-increasing INSERT operator if and only if one exists.

Proof. Assume there exists a valid score-increasing INSERT operator for the given MEC $\mathcal{E}$. Then, $P(\mathbf{v})$ is not Markov with respect to $\mathcal{E}$. Hence, $P(\mathbf{v})$ is not Markov with respect to the $\mathcal{G} \in \mathcal{E}$ chosen by SAFEINSERT. By the equivalence of the global and local Markov properties (Prop. A.1), this implies that there exists $X \in \mathcal{G}$ such that $X \not\perp\!\!\!\perp \mathbf{Nd}_X^{\mathcal{G}} \mid \mathbf{Pa}_X^{\mathcal{G}}$. Since $P(\mathbf{v})$ is faithful to some DAG, it satisfies the *composition* axiom of conditional independence [41]; hence, there exists some $Y \in \mathbf{Nd}_X^{\mathcal{G}}$ such that $X \not\perp\!\!\!\perp Y \mid \mathbf{Pa}_X^{\mathcal{G}}$. By the local consistency and decomposability of the scoring criterion, we have $s(X, \mathbf{Pa}_X^{\mathcal{G}}) < s(X, \mathbf{Pa}_X^{\mathcal{G}} \cup \{Y\})$. Then, SAFEINSERT will find the valid score-increasing INSERT$(Y, X, \mathbf{T})$ operator corresponding to $\mathcal{G} \cup \{Y \to X\}$. The reverse direction is similar. If SAFEINSERT outputs some $(X, Y, \mathbf{T})$, this implies it has found some $X \in \mathcal{G}$ and $Y \in \mathbf{Nd}_X^{\mathcal{G}}$ such $s(X, \mathbf{Pa}_X^{\mathcal{G}}) < s(X, \mathbf{Pa}_X^{\mathcal{G}} \cup \{Y\})$, and hence $X \not\perp\!\!\!\perp Y \mid \mathbf{Pa}_X^{\mathcal{G}}$. This implies that $P(\mathbf{v})$ is not Markov with respect to $\mathcal{G}$ and hence to $\mathcal{E}$, and there exists a valid score-increasing INSERT for $\mathcal{E}$. The INSERT output by SAFEINSERT is a valid score-increasing operator by construction. □

We provide pseudocode for the GETSAFEINSERT procedure in Alg. 5. GETSAFEINSERT generalizes SAFEINSERT; instead of searching for a valid INSERT across all non-adjacencies in $\mathcal{E}$, it searches for a valid INSERT in a subset of the non-adjacencies in $\mathcal{E}$, given by the $candidates$ set. This enables the use of the prioritisation scheme of GETPRIORITYINSERTS.

Proposition C.3 (Correctness of GETPRIORITYINSERTS)**.** *Let $priorityList$ be the list of sets of edges output by* GETPRIORITYINSERTS *(Alg. 6) given a Markov equivalence class $\mathcal{E}$ and prior assumptions $\mathbf{S} = \langle \mathbf{R}, \mathbf{F} \rangle$. Then, the union of all sets of edges in $priorityList$ is equal to the set of variable pairs (X, Y) that are non-adjacent in $\mathcal{E}$.*

Proof. This follows from the fact that GETPRIORITYINSERTS loops over all non-adjacencies in $\mathcal{E}$, and any adjacencies not determined by $\mathbf{S}$ are added to $priorityList[3]$ on line 10. □

Corollary C.1 (Correctness of LGES-0)**.** *Let $\mathcal{E}$ denote the Markov equivalence class that results from LGES-0 (Alg. 8) with* SAFEINSERT*, initialised from an arbitrary MEC $\mathcal{E}_0$ and given prior assumptions $\mathbf{S} = \langle \mathbf{R}, \mathbf{F} \rangle$ using* SAFEINSERT*, and let $P(\mathbf{v})$ denote the distribution from which the data $\mathbf{D}$ was generated. Then, as the number of samples goes to infinity, $\mathcal{E}$ is the Markov equivalence class underlying $P(\mathbf{v})$.*

Proof. This follows from Prop. 2, Prop. C.3, and Thm. 1. In the forward phase, if $P(\mathbf{v})$ is not Markov with respect to the current MEC $\mathcal{E}$, SAFEINSERT will find some score-increasing INSERT$(X, Y, \mathbf{T})$(Prop. 2. Since (X, Y) must be in some set in the priority list returned by GETPRIORITYINSERT (Prop. C.3), some call to GETSAFEINSERT will find a score-increasing INSERT operator. Therefore, each forward step is guaranteed to find a valid score-increasing INSERT operator, if it exists. In the backward phase, since LGES-0 enumerates all valid DELETE operators at each step, it is also guaranteed to find a valid score-increasing DELETE operator when if one exists. Thus, LGES-0 terminates only when there are no score-increasing operators for the current MEC. As such, LGES-0 satisfies the conditions of GGES (Alg. 3) and its correctness follows from Thm. 1. □

Corollary 1 (Correctness of LGES). Let $\mathcal{E}$ denote the Markov equivalence class that results from LGES (Alg. 1) with SAFEINSERT, initialised from an arbitrary MEC $\mathcal{E}_0$ and given prior assumptions $\mathbf{S} = \langle \mathbf{R}, \mathbf{F} \rangle$, and let $P(\mathbf{v})$ denote the distribution from which the data $\mathbf{D}$ was generated. Then, as the number of samples goes to infinity, $\mathcal{E}$ is the Markov equivalence class underlying $P(\mathbf{v})$.

Proof. This follows from Prop. 2, Prop. C.3, and Thm. 1. When LGES terminates, there exists no score-increasing deletion for the current MEC $\mathcal{E}$ by construction. By a similar argument to Cor. C.1, we can further show that there exists no score-increasing insertion for the current MEC $\mathcal{E}$. Thus, LGES satisfies the conditions of GGES (Alg. 3) and its correctness follows from Thm. 1. □

Corollary C.2 (Correctness of LGES+)**.** *Let $\mathcal{E}$ denote the Markov equivalence class that results from LGES+ (Alg. 8) with* SAFEINSERT*, initialised from an arbitrary MEC $\mathcal{E}_0$ and given prior assumptions $\mathbf{S} = \langle \mathbf{R}, \mathbf{F} \rangle$ using* SAFEINSERT*, and let $P(\mathbf{v})$ denote the distribution from which the*

Algorithm 3: Generalized Greedy Equivalence Search (GGES)

Input: Data $\mathbf{D} \sim \mathbf{P}(\mathbf{v})$, initial MEC $\mathcal{E}$, scoring criterion S, initial MEC $\mathcal{E}_0$, list of phases $phases$ in {['forward', 'backward'], ['any']}
Output: MEC $\mathcal{E}$ of $\mathbf{P}(\mathbf{v})$

```
E ← E_0;
foreach phase in phases do
    repeat
        Operate(X, Y, T) ← GetOperator(E, D, S, phase) ;
        E ← E + Operate(X, Y, T) ;
    until no score-increasing operators exist;
return E
```

data $\mathbf{D}$ was generated. Then, as the number of samples goes to infinity, $\mathcal{E}$ is the Markov equivalence class underlying $P(\mathbf{v})$.

Proof. The correctness of LGES (Cor. 1) implies that the MEC obtained on line 1 of LGES+ is the highest-scoring MEC. Therefore, the condition on line 7 of LGES+ will never be true, and LGES+ outputs the same MEC as LGES. □

Theorem 2 (Correctness of $\mathcal{I}$-Orient). Let $\mathcal{E}$ denote the Markov equivalence class that results from $\mathcal{I}$-Orient (Alg. 2) given an observational MEC $\mathcal{E}_0$ and interventional targets $\mathcal{I}$, and let $(\mathbf{P_I}(\mathbf{v}))_{\mathbf{I}\in\mathcal{I}}$ denote the family of distributions from which the data $(\mathbf{D_I})_{\mathbf{I}\in\mathcal{I}}$ was generated. Assume that $\mathcal{E}_0$ is the MEC underlying $P_\emptyset(\mathbf{v})$. Then, as the number of samples goes to infinity for each $\mathbf{I} \in \mathcal{I}$, $\mathcal{E}$ is the $\mathcal{I}$-Markov equivalence class underlying $(\mathbf{P_I}(\mathbf{v}))_{\mathbf{I}\in\mathcal{I}}$.

Proof. Let $\mathcal{E}^*$ denote the true $\mathcal{I}$-MEC underlying $(\mathbf{P_I}(\mathbf{v}))_{\mathbf{I}\in\mathcal{I}}$. Since $\mathcal{E}$ only orients undirected edges in $\mathcal{E}_0$, and $\mathcal{E}_0$ has the same skeleton and v-structures as $\mathcal{E}^*$, $\mathcal{E}$ also has the same skeleton and v-structures as $\mathcal{E}^*$. Next, we show that for every variable pair (X, Y) adjacent in $\mathcal{E}^*$ (and hence $\mathcal{E}$) for which there exists some $\mathbf{I} \in \mathcal{I}$ with $X \in \mathbf{I}, Y \notin \mathbf{I}$, this edge is directed in both $\mathcal{E}$ and $\mathcal{E}^*$, and moreover, has the same direction in both.

Consider some edge $(X, Y) \in \mathcal{E}^*$ for which there exists $\mathbf{I} \in \mathcal{I}$ such that $X \in \mathbf{I}, Y \notin \mathbf{I}$. Then, (X, Y) is directed and $\mathcal{I}$-essential in $\mathcal{E}$ [24, Cor. 13]. We will show that $\mathcal{E}^*$ contains $Y \to X$ if and only if for every such $\mathbf{I}$, $s_{\mathbf{D_I}}(y) > s_{\mathbf{D_I}}(y, x)$.

$\Longrightarrow$ Assume $\mathcal{E}^*$ contains $Y \to X$. If $X \to Y \in \mathcal{E}^*$, then $X \to Y$ in every DAG $\mathcal{G} \in \mathcal{E}^*$. This implies that in every $\mathcal{G} \in \mathcal{E}^*$, there are no directed paths from $Y \to X$ in $\mathcal{G}$ and hence $\mathcal{G}_{\overline{\mathbf{I}}}$. Moreover, since all edges into X are removed in $\mathcal{G}_{\overline{\mathbf{I}}}$, there are no directed paths from $X \to Y$ in $\mathcal{G}_{\overline{\mathbf{I}}}$. Since $P_{\mathbf{I}}(\mathbf{v})$ is Markov with respect to $\mathcal{G}_{\overline{\mathbf{I}}}$, this implies $X \perp\!\!\!\perp Y$ in $P_{\mathbf{I}}(\mathbf{v})$. Let $\mathcal{H}$ denote the empty graph over variables $\mathbf{V}$. Since $X \perp\!\!\!\perp Y$ in $P_{\mathbf{I}}(\mathbf{v})$, by the local consistency of the scoring criterion, $\mathcal{H}$ has a higher score than $\mathcal{H} \cup \{X \to Y\}$. By the decomposability of the scoring criterion, this implies $s_{\mathbf{D_I}}(y) > s_{\mathbf{D_I}}(y, x)$. Since $\mathbf{I}$ was arbitrary, this must be true for each $\mathbf{I} \in \mathcal{I}$ such that $X \in \mathbf{I}, Y \notin \mathbf{I}$.

$\Longleftarrow$ Assume that $s_{\mathbf{D_I}}(y) > s_{\mathbf{D_I}}(y, x)$ for some $\mathbf{I} \in \mathcal{I}$. Let $\mathcal{H}$ denote the empty graph over variables $\mathbf{V}$. Then, since $s_{\mathbf{D_I}}(y) > s_{\mathbf{D_I}}(y, x)$, the decomposability of the scoring criterion implies that $\mathcal{H}$ has a higher score than $\mathcal{H} \cup \{X \to Y\}$. If $Y \not\perp\!\!\!\perp X$ in $P_{\mathbf{I}}(\mathbf{v})$, the local consistency of the scoring criterion would imply that $\mathcal{H}$ has a lower score than $\mathcal{H} \cup \{X \to Y\}$. By contrapositive, it must be true that $Y \perp\!\!\!\perp X$ in $P_{\mathbf{I}}(\mathbf{v})$. Since $P_{\mathbf{I}}(\mathbf{v})$ is faithful to $\mathcal{G}_{\overline{\mathbf{I}}}$ for some $\mathcal{G} \in \mathcal{E}^*$, this must imply that X, Y are non-adjacent in $\mathcal{G}_{\overline{\mathbf{I}}}$. Since X, Y are adjacent in $\mathcal{E}^*$, and $X \in \mathbf{I}, Y \notin \mathbf{I}$, this implies that $\mathcal{G}$ and $\mathcal{E}^*$ contain $Y \to X$. This further implies that if the supposition is true for some $\mathbf{I} \in \mathcal{I}$ with $X \in \mathbf{I}, Y \notin \mathbf{I}$, it must be true for all of them.

The argument to show that $\mathcal{E}^*$ contains $X \to Y$ if and only if for every $\mathbf{I}$ with $X \in \mathbf{I}, Y \notin \mathbf{I}$, $s_{\mathbf{D_I}}(y) < s_{\mathbf{D_I}}(y, x)$ is analogous.

Moreover, since these statements are true for each $\mathbf{I} \in \mathcal{I}$, they are also true when comparing the sum over sum over all such $\mathbf{I}$: i.e., $\sum_{\mathbf{I}\in\mathcal{I}, X\in\mathbf{I}, Y\notin\mathbf{I}} s_{\mathbf{D_I}}(y)$ vs $\sum_{\mathbf{I}\in\mathcal{I}, X\in\mathbf{I}, Y\notin\mathbf{I}} s_{\mathbf{D_I}}(y, x)$.

Any edge that is directed in $\mathcal{E}$ is either (a) already directed in $\mathcal{E}_0$, in which case it is similarly directed in $\mathcal{E}^*$, (b) oriented on lines 4 or 7 of $\mathcal{I}$-ORIENT, in which case it is similarly directed in $\mathcal{E}^*$ by the above argument, or (c) oriented by the Meek rules on lines 5 or 8, in which case it is a consequence of edges directed due to (a) and (b), in which case it is also similarly directed in $\mathcal{E}^*$. Moreover, the edges directed in $\mathcal{E}^*$ are also due to (a) their being directed in $\mathcal{E}_0$, (b) there existing some $\mathbf{I} \in \mathcal{I}$ which contains exactly one endpoint of that edge, or (c) their being a consequence by the Meek rules of these two edge types. Therefore, edges directed in $\mathcal{E}^*$ are also similarly directed in $\mathcal{E}$, since $\mathcal{I}$-ORIENT directs each such edge type. We thus have that $\mathcal{E} = \mathcal{E}^*$. $\square$

Algorithm 4: GETCONSERVATIVEINSERT

Input: MEC $\mathcal{E}$, DAG $\mathcal{G} \in \mathcal{E}$, data $\mathbf{D} \sim P(\mathbf{v})$, edge insertion candidates $candidates$, scoring criterion S

Output: A valid score-increasing INSERT operator for $\mathcal{E}$ from the adjacencies in $candidates$, or $\emptyset$ if none is found.

$\Delta S_{max} \leftarrow -\infty$;
$(X_{max}, Y_{max}, \mathbf{T}_{max}) \leftarrow \emptyset$;
foreach (X, Y) *in candidates* **do**
    if $X \in \mathbf{Nd}^{\mathcal{G}}_Y$ *and* $s(Y, \mathbf{Pa}^{\mathcal{G}}_Y) < s(Y, \mathbf{Pa}^{\mathcal{G}}_Y \cup \{X\})$ **then**
        $\Delta S_{xy} \leftarrow -\infty$;
        $(\hat{X}, \hat{Y}, \hat{\mathbf{T}}) \leftarrow \emptyset$;
        foreach *valid* $\mathbf{T} \subseteq \mathbf{Ne}^{\mathcal{E}}_Y \setminus \mathbf{Adj}^{\mathcal{E}}_X$ **do**
            $\Delta S \leftarrow s\big(Y, (\mathbf{Ne}^{\mathcal{E}}_Y \cap \mathbf{Adj}^{\mathcal{E}}_X) \cup \mathbf{T} \cup \mathbf{Pa}^{\mathcal{E}}_Y \cup \{X\}\big) - s\big(Y, (\mathbf{Ne}^{\mathcal{E}}_Y \cap \mathbf{Adj}^{\mathcal{E}}_X) \cup \mathbf{T} \cup \mathbf{Pa}^{\mathcal{E}}_{Y,}\big)$;
            if $\Delta S > \Delta S_{xy}$ **then**
                $\Delta S_{xy} \leftarrow \Delta S$;
                $(\hat{X}, \hat{Y}, \hat{\mathbf{T}}) \leftarrow (X, Y, \mathbf{T})$;
            else if $\Delta S < 0$ **then**
                $\Delta S_{xy} \leftarrow -\infty$;
                break;
        if $\Delta S_{xy} > \Delta S_{max}$ **then**
            $\Delta S_{max} \leftarrow \Delta S_{xy}$;
            $(X_{max}, Y_{max}, \mathbf{T}_{max}) \leftarrow (\hat{X}, \hat{Y}, \hat{\mathbf{T}})$;
return $(X_{max}, Y_{max}, \mathbf{T}_{max})$

Algorithm 5: GETSAFEINSERT

Input: MEC $\mathcal{E}$, DAG $\mathcal{G} \in \mathcal{E}$, data $\mathbf{D} \sim P(\mathbf{v})$, edge insertion candidates $candidates$, scoring criterion S

Output: A valid score-increasing INSERT operator for $\mathcal{E}$ from the adjacencies in $candidates$, or $\emptyset$ if none is found.

$\Delta S_{max} \leftarrow -\infty$;
$(X_{max}, Y_{max}, \mathbf{T}_{max}) \leftarrow \emptyset$;
foreach (X, Y) *in candidates* **do**
    if $X \in \mathbf{Nd}^{\mathcal{G}}_Y$ *and* $s(Y, \mathbf{Pa}^{\mathcal{G}}_Y) < s(Y, \mathbf{Pa}^{\mathcal{G}}_Y \cup \{X\})$ **then**
        foreach *valid* $\mathbf{T} \subseteq \mathbf{Ne}^{\mathcal{E}}_Y \setminus \mathbf{Adj}^{\mathcal{E}}_X$ **do**
            $\Delta S \leftarrow s\big(Y, (\mathbf{Ne}^{\mathcal{E}}_Y \cap \mathbf{Adj}^{\mathcal{E}}_X) \cup \mathbf{T} \cup \mathbf{Pa}^{\mathcal{E}}_Y \cup \{X\}\big) - s\big(Y, (\mathbf{Ne}^{\mathcal{E}}_Y \cap \mathbf{Adj}^{\mathcal{E}}_X) \cup \mathbf{T} \cup \mathbf{Pa}^{\mathcal{E}}_{Y,}\big)$;
            if $\Delta S > \Delta S_{max}$ **then**
                $\Delta S_{max} \leftarrow \Delta S$;
                $(X_{max}, Y_{max}, \mathbf{T}_{max}) \leftarrow (X, Y, \mathbf{T})$;
return $(X_{max}, Y_{max}, \mathbf{T}_{max})$

Algorithm 6: GETPRIORITYINSERTS

Input: MEC $\mathcal{E}$, prior assumptions $\mathbf{S} = \langle \mathbf{R}, \mathbf{F} \rangle$

$priorityList \leftarrow [\{\} \times 4]$;
foreach (X, Y) *non-adjacent in* $\mathcal{E}$ **do**
  if $X - *Y \in \mathbf{R}$ **then**
    Add (X, Y) to $priorityList[1]$; `// required`
  else if $Y \rightarrow X \in \mathbf{R}$ **then**
    Add (X, Y) to $priorityList[2]$; `// weakly required`
  else if $X \rightarrow Y \in \mathbf{F}$ *or* $X - Y \in \mathbf{F}$ **then**
    Add (X, Y) to $priorityList[4]$; `// forbidden`
  else
    Add (X, Y) to $priorityList[3]$; `// ambivalent`
return $priorityList$

Algorithm 7: PDAGTODAG

Input: MEC $\mathcal{E}$

$\mathcal{E}' \leftarrow \mathcal{E}$;
while *there exists an undirected edge in* $\mathcal{E}$ **do**
  Choose a vertex V in $\mathcal{E}'$ such that (i) V is a sink node in $\mathcal{E}'$ (no outgoing directed edges), and (ii) all undirected neighbors of V are pairwise adjacent in $\mathcal{E}'$ (form a clique);
foreach *undirected edge* (U, V) *in* $\mathcal{E}$ **do**
  Orient (U, V) as $U \rightarrow V$ in $\mathcal{E}$;
Remove V from $\mathcal{E}'$;
return $\mathcal{E}$

Algorithm 8: Less Greedy Equivalence Search-0 (LGES-0)

Input: Data $\mathbf{D} \sim \mathbf{P}(\mathbf{v})$, scoring criterion S, prior assumptions $\mathbf{S} = \langle \mathbf{R}, \mathbf{F} \rangle$, initial MEC $\mathcal{E}_0$, insertion strategy $GetInsert$ in $\{$GETSAFEINSERT, GETCONSERVATIVEINSERT$\}$
Output: MEC $\mathcal{E}$ of $\mathbf{P}(\mathbf{v})$

$\mathcal{E} \leftarrow \mathcal{E}_0$; `// allows initialisation if preferred by user`
repeat
  $\mathcal{G} \leftarrow$ some DAG in $\mathcal{E}$; `// forward phase`
  $priorityList \leftarrow$ GETPRIORITYINSERTS$(\mathcal{E}, \mathcal{G}, \mathbf{S})$;
  foreach *candidates in* $priorityList$ **do**
    $(X_{max}, Y_{max}, \mathbf{T}_{max}) \leftarrow GetInsert(\mathcal{E}, \mathcal{G}, \mathbf{D}, candidates, S)$;
    if $(X_{max}, Y_{max}, \mathbf{T}_{max})$ *is found* **then**
      $\mathcal{E} \leftarrow \mathcal{E} +$ INSERT$(X_{max}, Y_{max}, \mathbf{T}_{max})$;
      break ; `// no need to check lower priority`
until *no improving insertions exist*;
repeat
  $\mathcal{E} \leftarrow \mathcal{E}+$ highest-scoring TURN$(X, Y, \mathbf{T})$; `// turning phase`
until *no score-increasing reversals exist*;
repeat
  $\mathcal{E} \leftarrow \mathcal{E}+$ highest-scoring DELETE$(X, Y, \mathbf{T})$; `// backward phase`
until *no score-increasing deletions exist*;
return $\mathcal{E}$

Algorithm 9: Less Greedy Equivalence Search+ (LGES+)

Input: Data $\mathbf{D} \sim \mathbf{P}(\mathbf{v})$, scoring criterion S, prior assumptions $\mathbf{S} = \langle \mathbf{R}, \mathbf{F} \rangle$, initial MEC $\mathcal{E}_0$, insertion strategy $GetInsert$ in {GETSAFEINSERT, GETCONSERVATIVEINSERT}

Output: MEC $\mathcal{E}$ of $\mathbf{P}(\mathbf{v})$

$\mathcal{E} \leftarrow$ LGES($\mathbf{D}, S, \mathbf{S}, \mathcal{E}_0, GetInsert$) ;

$\mathcal{D} \leftarrow$ all deletions valid for $\mathcal{E}$;

while $|\mathcal{D}| > 0$ **do**

  $(X_{max}, Y_{max}, \mathbf{T}_{max}) \leftarrow$ highest-scoring deletion in $\mathcal{D}$;

  $\mathcal{E}' \leftarrow \mathcal{E} +$ DELETE$(X_{max}, Y_{max}, \mathbf{T}_{max})$;

  $\mathcal{E}' \leftarrow$ LGES*($\mathbf{D}, S, \mathbf{S}, \mathcal{E}_0, GetInsert, X_{max}, Y_{max}$);

  `// LGES`* `is a modified version of LGES that excludes edge insertions between the given` X, Y

  if $S(\mathcal{E}', \mathbf{D}) > S(\mathcal{E}, \mathbf{D})$ **then**

    $\mathcal{E} \leftarrow \mathcal{E}'$;

    $\mathcal{D} \leftarrow$ all deletions valid for $\mathcal{E}$;

  else

    $\mathcal{D} \leftarrow \mathcal{D} \setminus \{(X_{max}, Y_{max}, \mathbf{T}_{max})\}$;

return $\mathcal{E}$

D Experiments

Compute details. All experiments were run on a shared compute cluster with 2x Intel Xeon Platinum 8480+ CPUs (112 cores total, 224 threads) at up to 3.8 GHz, and 210 MiB L3 cache.

D.1 Learning from observational data

Baseline details We ran the PC algorithm using significance level $\alpha = 0.05$ for conditional independence tests, with the null hypothesis of independence. Since NoTears often outputs cyclic graphs, we post-processed the output graph by greedily removing the lowest-weight edges until it was acyclic, following [37]. This was done so that we could convert the output to a valid CPDAG for comparison with other algorithms. We ran NoTears with default parameters from the `causalnex` library, which uses a weight threshold of $w = 0$; note that performance may vary depending on the choice of parameters, particularly the weight threshold and the acyclicity penalty parameter. However, since the implementation of NoTears is very compute-heavy, we were not able to tune these parameters. For fair comparison, we implemented all GES variants, including XGES, LGES, and GIES, by modifying the code provided by [20].[5] Discrepancies in runtime of XGES between our results and the results of [36] may result from the latter's use of an optimized C++ implementation, which includes an implementation of the search operators that they used in XGES but not in GES.

D.1.1 Synthetic data details for Experiment 5.1

For the results shown in Fig. 4, we draw Erdős–Rényi graphs with p variables and $\{2p$ edges in expectation (ER2), for $p \in \{5, 10, 15, 20, 25, 35, 50, 75, 100, 150\}$. For each p, we sample 50 graphs and generate linear-Gaussian data for each graph. Following [37], we draw weights from $\mathcal{U}([-2, -0.5] \cup [0.5, 2])$. In some cases, the resulting weight matrix was (almost) singular, in which case each row of weights was ℓ_1 normalized. We draw noise means from $\mathcal{N}(0, 1)$ and noise variances from $\mathcal{U}([0.1, 0.5])$. We obtain $n = 10^4$ samples per dataset via `sampler` [20].

D.1.2 Further baselines and metrics for Experiment 5.1

In Fig. D.1.1, we present additional baselines and accuracy metrics for the setting in Sec. 5.1, including the particular types of structural errors (excess adjacencies, missing adjacencies, incorrect orientations). As in the case of SHD, LGES outperforms other algorithms across these metrics, with CONSERVATIVEINSERT outperforming SAFEINSERT.

[5] `https://github.com/juangamella/ges`

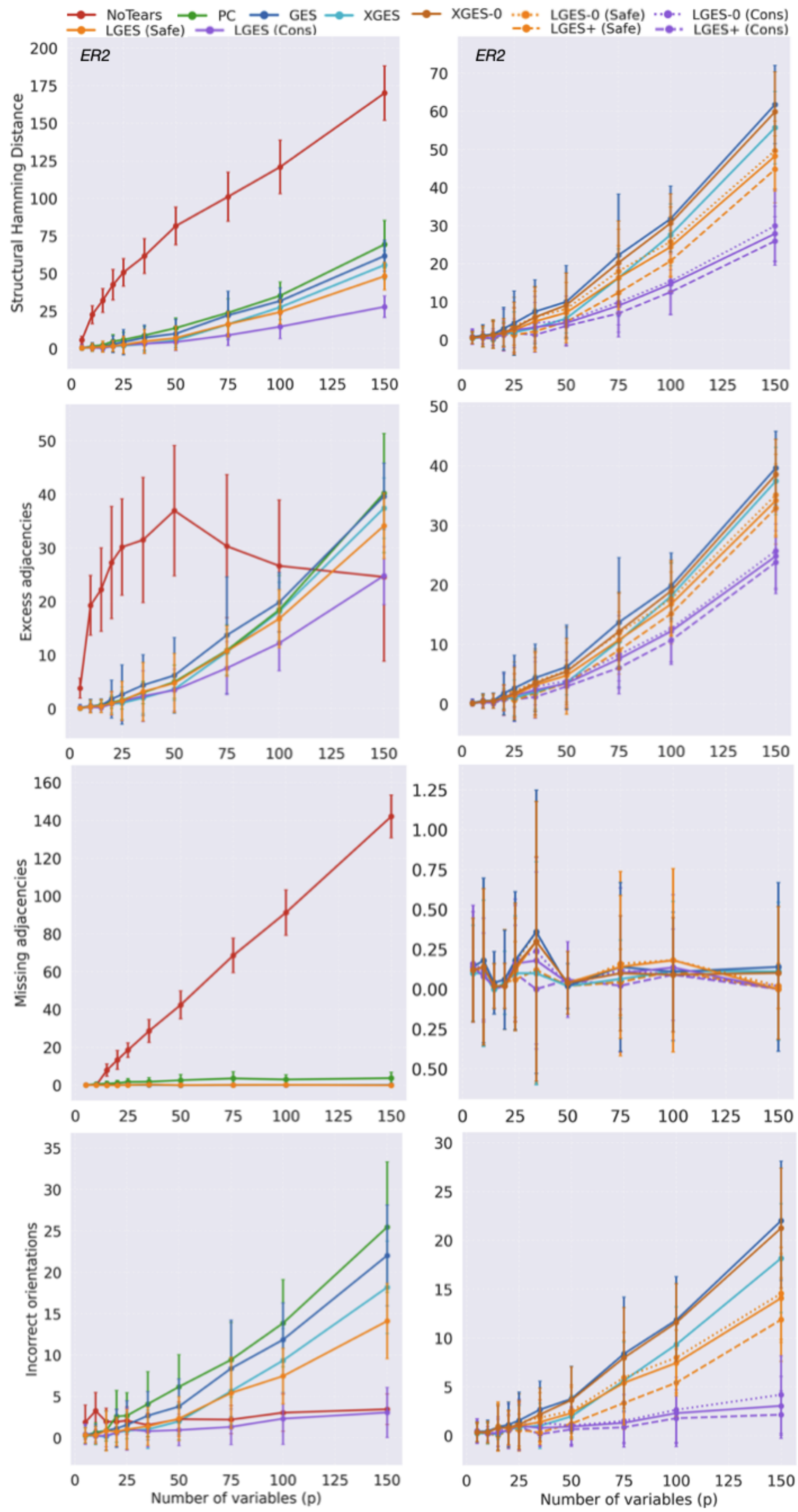


Figure D.1.1: Structural error of algorithms on 50 simulated observational datasets from Erdős–Rényi graphs with p variables and $2p$ edges in expectation, given $n = 10^4$ samples and no prior knowledge (Sec D.1.2). **Lower is better** (more accurate / faster) across all plots. **(Left)** Common baselines. GES-style algorithms outperform PC and NoTears in accuracy. **(Right)** GES variants. LGES-0, LGES, and LGES+ are the less greedy variants of GES, XGES-0, and XGES respectively. Each less greedy algorithm improves on its greedy counterpart in accuracy.

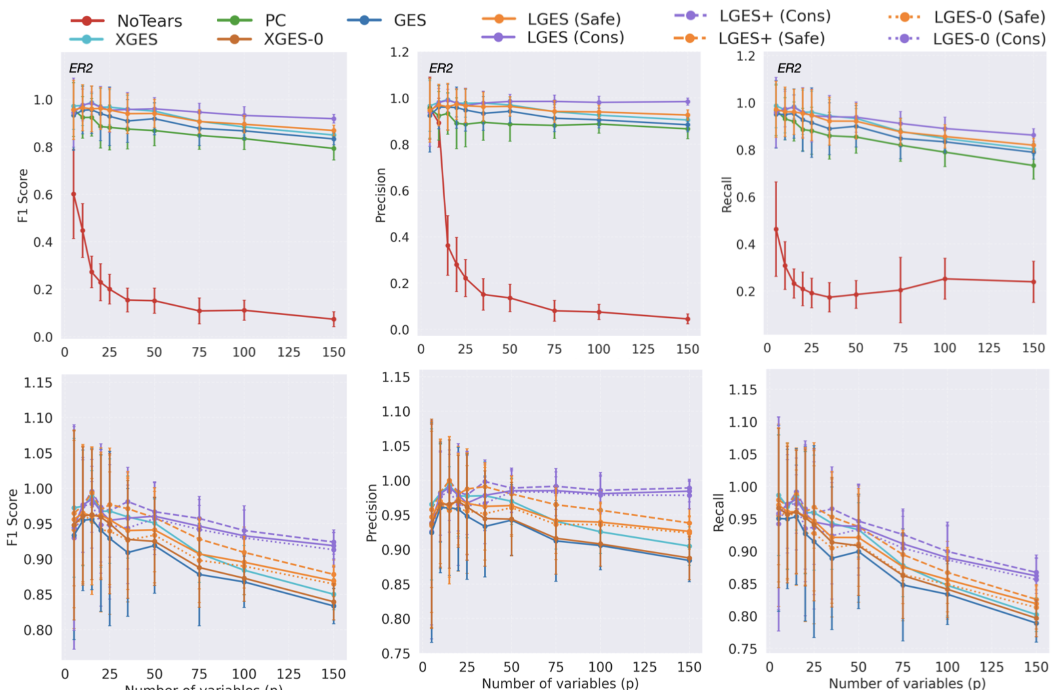


Figure D.1.2: Binary classification accuracy of algorithms on 50 simulated observational datasets from Erdős–Rényi graphs with p variables and $2p$ edges in expectation, given $n = 10^4$ samples and no prior knowledge (Sec. D.1.3). **Higher is better** across all plots. An MEC is said to contain an edge (X, Y) if it contains either $X - Y$ or $X \to Y$ (but not $Y \to X$). **(Top)** Common baselines. GES-style algorithms outperform PC and NoTears in accuracy. **(Bottom)** GES variants. LGES-0, LGES, and LGES+ are the less greedy variants of GES, XGES-0, and XGES respectively. Each less greedy variant improves on its greedy counterpart in accuracy.

The behaviour of NoTears is more variable. It misses significantly more adjacencies than all other methods—approximately linearly many in the number of variables. It also includes many more excess adjacencies than the other algorithms up to $p = 50$, after which the number of excess adjacencies begins to decline, ultimately approaching that of LGES for $p = 150$; however, this could be explained by the increasing number of adjacencies that are also missed by NoTears. We show in the next section how, as a result, the F1 score of NoTears is low across all graph sizes (Fig. D.1.2).

D.1.3 Precision, recall, and F1 score for Experiment 5.1

Next, in addition to structural error, we evaluate accuracy by considering causal discovery as a binary classification task. We use the task definition provided in [36]: an MEC $\mathcal{E}$ is said to contain an edge (X, Y) if it contains either $X \to Y$ or $X - Y$ (but not $Y \to X$). The results are shown in Fig. D.1.2. For graphs with $p < 20$ nodes, we find that GES and LGES have similar F1 scores. They both outperform PC, which in turn substantially outperforms NoTears. For $p > 20$, LGES+ (Conservative) dominates, followed by LGES (Conservative). For large p, LGES (Conservative) achieves a substantially higher F1 score than GES and other algorithms except LGES+. For instance, for $p = 150$, LGES (Conservative) achieves an F1 score just under 0.95, whereas GES's F1 score is slightly under 0.85.

D.1.4 Further experiments with varying edge densities

In Fig. D.1.3, we provide results for select baselines on Erdős–Rényi graphs with p variables and $\{1, 3\} \cdot p$ edges in expectation following the set-up in Experiment 5.1. The relative performance of algorithms is similar to the ER2 case (Fig. D.1.1), though PC and GES achieve similar accuracy on ER3 graphs. LGES+ (Conservative) is the most accurate overall, achieving $2\times$ less structural error than GES on ER1 and ER3 graphs. For instance, graphs with 150 variables and 450 edges in expectation, LGES (Conservative) achieves SHD ≈ 40, and LGES+ (Conservative) ≈ 30. All less

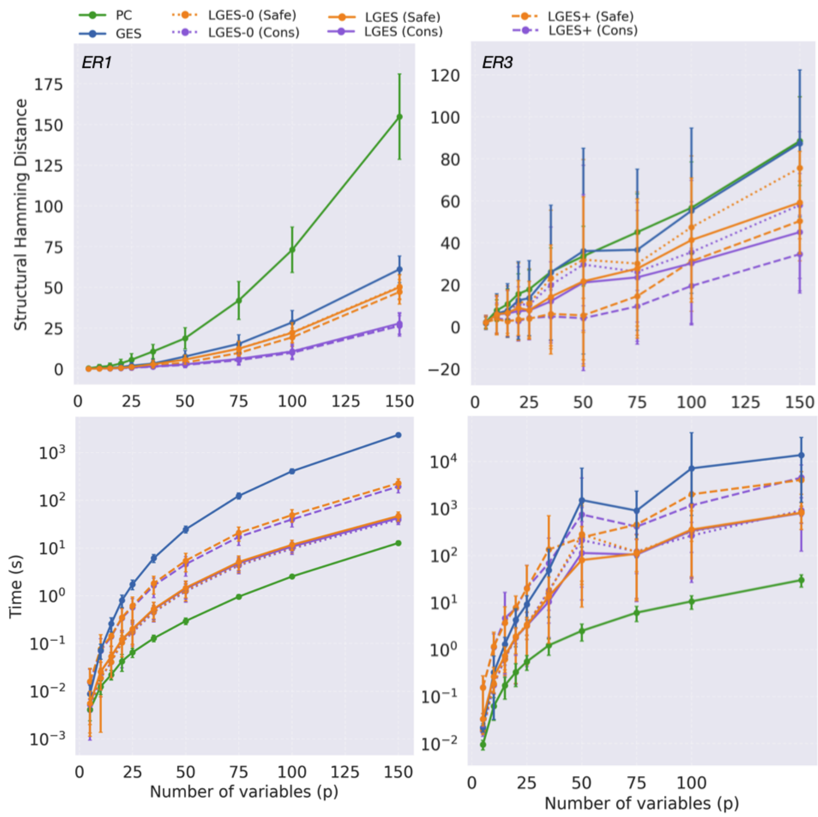


Figure D.1.3: Performance of algorithms on 50 simulated observational datasets from Erdős–Rényi graphs with p variables and **(left)** p edges and **(right)** $3p$ edges in expectation, given $n = 10^4$ samples and no prior knowledge (Sec. D.1.4). **Lower is better** (more accurate / faster) across all plots. See Fig. 4 for graphs with $2p$ edges in expectation. LGES (Conservative) is the fastest GES variant across edge densities, whereas LGES+ (Conservative) is the most accurate overall algorithm across edge densities.

greedy algorithms are faster and more accurate than GES, with LGES-0 and LGES being up to $10\times$ faster.

D.1.5 Further experiments with smaller datasets

To investigate performance in the small-sample setting, we conduct experiments with $n = 10^3$ for ER2 graphs with $p \in \{10, 25, 50, 100\}$ variables. The results are summarized in Fig. D.1.4.

As in previous experiments, LGES (Safe and Conservative) outperforms GES in runtime and accuracy across all graph sizes, with CONSERVATIVEINSERT yielding higher accuracy than SAFEINSERT. Interestingly, the PC algorithm outperforms LGES for the case of $p = 100$, suggesting PC's usefulness for settings where the number of samples is small compared with the number of variables.

D.1.6 Further experiments with larger graphs

We scaled up Experiment 5.1 to graphs with $p \in \{175, 250\}$ variables. We ran LGES (Safe and Conservative) and PC. We were unable to continue running XGES, GES, and NoTears due to time and compute constraints; for instance, GES took over 10^4 seconds without terminating for $p = 175$ for a single trial. The results are summarized in Table D.1.1.

LGES both with SAFEINSERT and with CONSERVATIVEINSERT is substantially more accurate (in terms of SHD and F1 score) than PC, with CONSERVATIVEINSERT achieving less than half the structural error of PC. While PC is faster than LGES, LGES is much faster than GES, taking only $\approx$5-6 minutes for $p = 175$ and $\approx$15-20 minutes for $p = 250$. Recall that for $p = 150$, GES was already approaching a runtime of 10^4 seconds (>2 hours) (Fig. 4a).

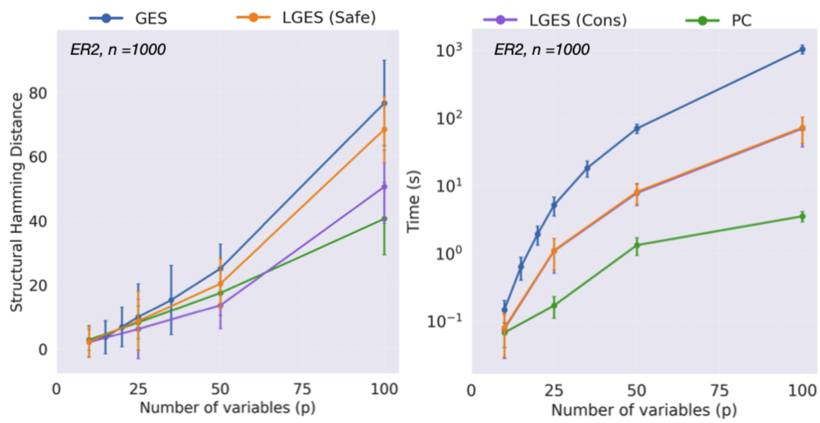


Figure D.1.4: Performance of algorithms on 50 simulated observational datastets from Erdős–Rényi graphs with p variables and $2p$ edges in expectation, given the smaller dataset of 10^3 samples and no prior knowledge (Sec D.1.5). In terms of accuracy, LGES (Conservative) outperforms other algorithms except for the case of $p = 100$, when PC is more accurate.

Metric	Method	$p = 175$	$p = 250$
SHD	LGES-0 (Conservative)	42.04 ± 11.51	83.06 ± 14.42
	LGES (Conservative)	**39.457 ± 10.44**	**82.80 ± 14.73**
	LGES-0 (Safe)	70.62 ± 11.68	135.24 ± 14.31
	LGES (Safe)	67.65 ± 12.31	133.22 ± 14.07
	PC	91.34 ± 17.17	166.20 ± 24.34
Runtime	LGES-0 (Conservative)	315.83 ± 73.23	989.37 ± 205.53
	LGES (Conservative)	326.44 ± 73.166	1019.13 ± 198.12
	LGES-0 (Safe)	337.45 ± 78.43	1058.11 ± 221.32
	LGES (Safe)	348.23 ± 80.57	1095.71 ± 209.10
	PC	**29.20 ± 9.50**	**85.67 ± 23.52**

Table D.1.1: Performance of algorithms (mean ± std) on 50 simulated observational datasets from large Erdős–Rényi graphs with p variables and $2p$ edges in expectation, given $n = 10^4$ samples and no prior knowledge (Sec. D.1.6). **Lower is better** (more accurate / faster) across all metrics. Other algorithms are omitted as they exceeded the 10^4 second bound on runtime for these graph sizes, prohibiting repeated trials. Among these, PC is the fastest, but makes twice as many structural errors as LGES (Conservative).

D.2 Learning with prior knowledge

In Fig. D.2.1, we present more detailed plots of runtime and SHD for the setting in Sec. 5.2, including background knowledge comprising $m' = m/2$ and $m' = 3m/4$ edges for a ground truth graph with m edges. Resutls for $m' = m/2$ follow a very similar trend to those with $m' = 3m/4$ discussed in Sec. 5.2. Initialisation outperforms prioritization in LGES when the expert knowledge is mostly correct (`fc` $\geq$ `0.75`), but prioritization does better otherwise.

D.3 Learning from interventional data

In this section, we further discuss the performance of algorithms given interventional data in Sec. 5.3. We observed that LGES (Safe and Conservative) followed by $\mathcal{I}-$ORIENT has higher SHD from the ground truth than LGIES, Safe and Conservative respectively. We hypothesize that this is because, in our experiments, LGES only uses 10^4 observational samples during the MEC learning phase. It uses the interventional samples only to orient edges using $\mathcal{I}$-ORIENT. In contrast, LGIES uses $10^4 + k \cdot 10^3$ samples in the MEC learning phase given k interventions, since it uses interventional

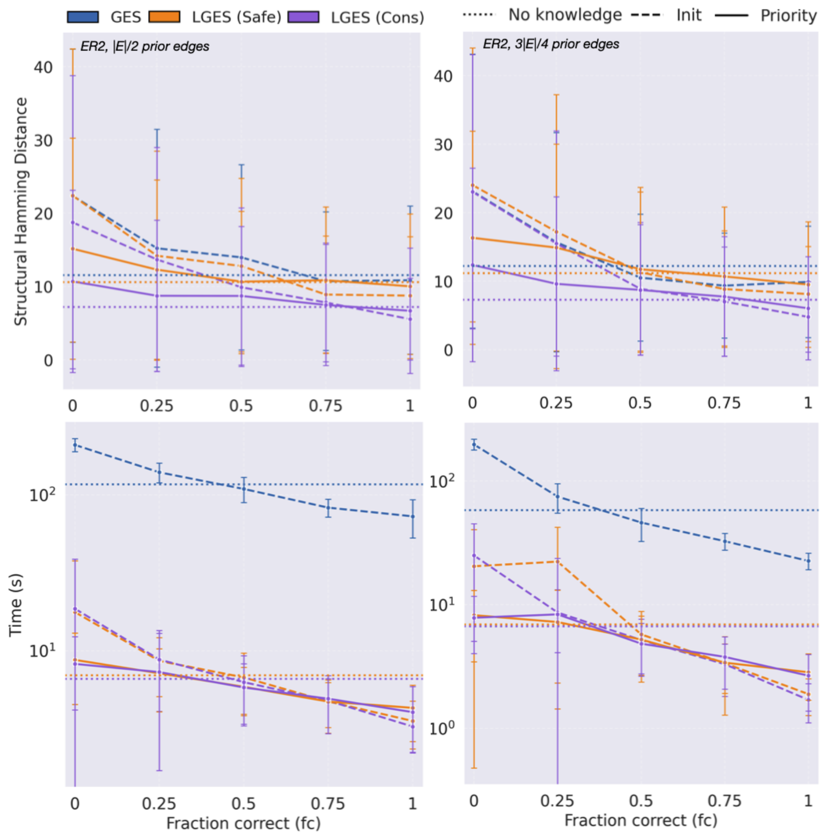


Figure D.2.1: Performance of algorithms on 50 simulated observational datasets from Erdős–Rényi graphs with 50 variables and 100 edges in expectation given $n = 10^3$ samples (Experiment 5.2, Sec. D.2). **Lower is better** (more accurate / faster) across all plots. We vary the correctness of the prior knowledge on the x-axis; higher `fc` indicates a higher fraction of knowledge that is correct. **(Left)** Algorithms are given $m' = m/2$ required edges as knowledge, for a true graph containing m edges. **(Right)** Similar, but with $m' = 3m/4$. LGES' prioritization strategy is more robust to misspecification in the knowledge ($\leq 75\%$ correct) than initialization with the same knowledge

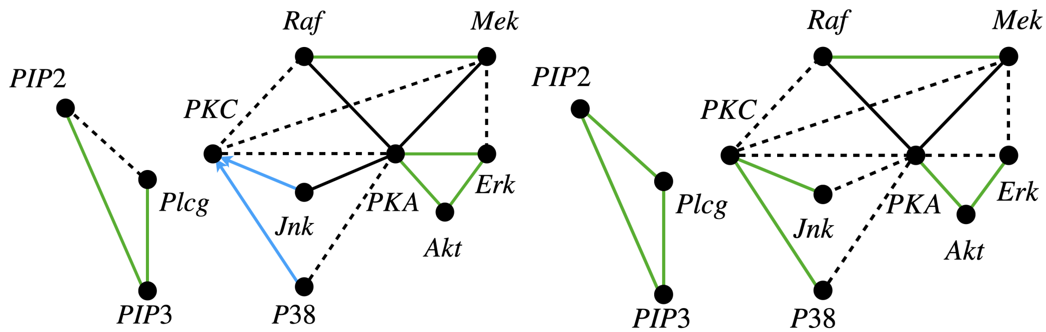
data throughtout learning, and we generate 10^3 interventional samples per target. Moreover, since we choose $k = p/10$ (where p is the number of variables), LGIES is making use of much more data than LGES for learning the MEC. Although GIES is known not to be asymptotically correct [58], this further motivates research into an asymptotically correct causal discovery algorithm that can make use of interventional data throughout learning.

D.4 Real-world protein signaling data

Sachs dataset. We compare GES and LGES on a real-world protein signaling dataset [48]. The observational dataset consists of 853 measurements of 11 phospholipids and phosphoproteins. We compare the output of our methods with the gold-standard inferred graph [48, Fig. 3]), containing 11 variables and 17 edges. The dataset is continuous but violates the linear-Gaussian assumption.[6] We test our methods both on the original continuous dataset and on a discretized version (3 categories per variable corresponding to low, medium, and high concentration) from the `bnlearn` repository.[7]

Results. The learned graphs provided in Fig. D.4.1. All algorithms output the same MEC and thus have the same accuracy in both settings. With discrete data, each algorithm had an SHD of 9 edges from the reference MEC, all of which were missing adjacencies. With continuous data, each

[6] `https://www.bnlearn.com/research/sachs05/`
[7] `https://www.bnlearn.com/book-crc-2ed/`



(a) Continuous data (b) Discretized data

Figure D.4.1: Comparison between the reference MEC and the learned MEC for $n = 853$ observational samples from the Sachs protein-signaling dataset [48]. For both continuous and discretized data, the three algorithms (GES, LGES (SAFEINSERT) and LGES (CONSERVATIVEINSERT)) return the same MEC, so only one learned graph is shown per panel. Green solid lines indicate edges correctly recovered by the algorithms. Blue solid lines indicate edges misoriented by the algorithms. Black dashed lines indicate edges missed by the algorithms. **(a)** Continuous data: nine edges are missed and two are misoriented: $Jnk \to Pkc$ and $P38 \to PKC$, both undirected in the reference MEC. **(b)** Discretised data: nine edges are missed and none are misoriented.

algorithm had an SHD of 11 edges from the reference MEC, 9 of which were missing adjacencies and 2 of which were incorrect orientations.

E Frequently Asked Questions

Q1. What is the difference between score-based and constraint-based causal discovery?

Answer. Constraint-based and score-based approaches to causal discovery solve the same problem but in different ways. Constraint-based approaches such as PC [54] and Sparsest Permutations [44] use statistical tests, usually for conditional independence, to learn a Markov equivalence class from data. Score-based approaches such as Greedy Equivalence Search [9, 35] instead attempt to maximize a score (for e.g., the Bayesian Information Criterion or BIC [51]) that reflects the fit between graph and data. There is no general claim about which of these methods is superior; we refer readers to [56] for an extensive empirical analysis, who found, for instance, that GES outperforms PC in accuracy across various sample sizes [56, Tables 4, 5].

Q2. I thought causal discovery was only one problem, but it seems the paper claims to be solving three different tasks: observational observational learning, interventional learning, learning with prior knowledge. Can you elaborate on these tasks (why they are different, how they relate, etc.)?

Answer. The most well-studied problem in causal discovery is that of learning causal graphs from only observational data. However, algorithms for this task have a few limitations. Firstly, they are computationally expensive and often fail to produce quality estimates of the true Markov equivalence class from finite samples. This motivates using prior knowledge in the search to produce better quality estimates of the true Markov equivalence class and do so faster. Secondly, algorithms for learning from observational data only identify a Markov equivalence of graphs, since this is the most informative structure that can be learned from observational data. However, MECs can be quite large, and thus uninformative for downstream tasks such as causal inference. When interventional data is available, more edges in the true graph become identifiable. This motivates using interventional data to identify a smaller and more informative interventional Markov equivalence class—the task of interventional learning. When no prior assumptions or interventional data are available, these tasks collapse to observational learning.

Q3. If GES is asymptotically consistent, why bother to consider LGES?

Answer. While GES is guaranteed to recover the true Markov equivalence class given infinite samples, it faces two challenges. First, computational tractability: structure learning is an NP-hard problem, and GES commonly struggles to scale in high-dimensional settings. Second, data is often limited in practice, which results in GES failing to recover the true MEC. LGES improves on GES in both of these aspects; it is up to 10 times faster and 2 times more accurate (Experiment 5.1). Moreover, while GES and LGES can both incorporate prior assumptions to guide the search, LGES is more robust to misspecification in the assumptions (Experiment 5.2).

Q4. How well does LGES scale?

Answer. LGES can scale to graphs with hundreds of variables and is up to 10 times faster than GES (Sec. D.1). Both variants of LGES (Safe and Conservative) terminate in less than 20 minutes hours on graphs with 250 variables (Sec. D.1.6) and have substantially better accuracy than other baselines (including PC and NoTears). D.1.6). LGES also outperforms these baselines in settings with dense graphs (Sec. D.1.4).

Q5. LGES may be asymptotically correct, but how well does it perform with finite samples?

Answer. We conduct an extensive empirical analysis of how LGES performs compared with baselines in finite-sample settings. Both variants of LGES (SAFEINSERT and CONSERVATIVEINSERT) have substantially better accuracy than GES, PC, and NoTears in experiments with $n = 10^4$ samples and up to 500 variables (Experiments 5.1, D.1). For instance, in graphs with 150 variables and 300 edges in expectation, LGES with CONSERVATIVEINSERT only makes $\approx$30 structural errors on average, which is twice as accurate as GES, which makes $\approx$60 structural errors. LGES also outperforms GES in smaller sample settings ($n \in \{500, 1000\}$) (Experiment D.1.5).

Q6. What is the difference between SAFEINSERT and CONSERVATIVEINSERT?

Answer. LGES can use either the SAFEINSERT or the CONSERVATIVEINSERT strategy to select INSERT operators in the forward phase; both result in improved accuracy and runtime relative to GES. We show that LGES with SAFEINSERT is asymptotically guaranteed to recover the true MEC (Cor. 1). However, it remains open whether the same is true of LGES with CONSERVATIVEINSERT, though we provide partial guarantees (Prop. C.1, C.2). Both strategies result in similar runtime, though CONSERVATIVEINSERT consistently has greater accuracy than SAFEINSERT across our experiments (Sec. 5, D).

Q7. How is causal discovery with prior knowledge different from initializing a causal discovery task to the hypothesized model?

Answer. Causal discovery with knowledge is the more general problem of using possibly misspecified prior knowledge in the process of causal discovery to aid the search. Initializing the search to a tentative model is one way to achieve this. However, initialization is not robust to misspecification in the assumptions and can result in worse runtime and accuracy than our approach of guiding the search using prior knowledge (Sec. 3.3, Experiment 5.2).

Q8. What is the difference between Greedy Interventional Equivalence Search (GIES) [24] and LGES + $\mathcal{I}$-ORIENT?

Answer. GIES and LGES are both score-based algorithms for learning from a combination of observational and interventional data. However, LGES has a few primary advantages over GIES. First, LGES with SAFEINSERT is guaranteed to recover the true interventional MEC (Cor. 1, Thm. 2) in the sample limit whereas GIES is not [58]. Second, both variants of LGES are up to 10 times faster than GIES, and LGES with CONSERVATIVEINSERT has accuracy competitive with GIES (Experiment 5.3). However, LGES uses interventional data to only to orient edges in a learned Markov equivalence class (in the $\mathcal{I}$-ORIENT procedure, Alg. 2), whereas GIES uses a combination of observational and experimental data throughout. We additionally introduce LGIES, which incoporates less greedy insertion into GIES and is thus both faster and more accurate than GIES (Experiment 5.3).

Q9. Can your work be combined with other causal discovery algorithms?

Answer. Several components of our approach are modular and can be combined with other causal discovery algorithms. For one, other algorithms in the GES family like FGES [43] and SGES [12] can be easily modified to use the SAFEINSERT or CONSERVATIVEINSERT strategies that we introduce; a simple extension would investigate the resulting changes in accuracy and runtime. For another, the $\mathcal{I}$-ORIENT procedure can be used to refine the observational MEC output by any causal discovery algorithm (not necessarily LGES) using interventional data.

Acknowledgements

This research is supported in part by the NSF, ONR, AFOSR, DoE, Amazon, JP Morgan, and The Alfred P. Sloan Foundation. We thank Jonghwan Kim and the anonymous reviewers for their insightful comments.

References

[1] Nicos Angelopoulos and James Cussens. Bayesian learning of Bayesian networks with informative priors. *Annals of Mathematics and Artificial Intelligence*, 54(1):53–98, November 2008. ISSN 1573-7470. doi: 10.1007/s10472-009-9133-x. URL https://doi.org/10.1007/s10472-009-9133-x.

[2] E. Bareinboim and J. Pearl. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113:7345–7352, 2016.

[3] Elias Bareinboim, Juan D. Correa, Duligur Ibeling, and Thomas Icard. *On Pearl's Hierarchy and the Foundations of Causal Inference*, page 507–556. Association for Computing Machinery, New York, NY, USA, 1 edition, 2022. ISBN 9781450395861. URL https://doi.org/10.1145/3501714.3501743.

[4] M. J. Bayarri, James O. Berger, Woncheol Jang, Surajit Ray, Luis R. Pericchi, and Ingmar Visser. Prior-based Bayesian information criterion. *Statistical Theory and Related Fields*, 3(1):2–13, January 2019. ISSN 2475-4269. doi: 10.1080/24754269.2019.1582126. URL https://doi.org/10.1080/24754269.2019.1582126. Publisher: Taylor Francis _eprint: https://doi.org/10.1080/24754269.2019.1582126.

[5] Paul Beaumont, Ben Horsburgh, Philip Pilgerstorfer, Angel Droth, Richard Oentaryo, Steven Ler, Hiep Nguyen, Gabriel Azevedo Ferreira, Zain Patel, and Wesley Leong. CausalNex, October 2021. URL https://github.com/quantumblacklabs/causalnex.

[6] Alexis Bellot, Junzhe Zhang, and Elias Bareinboim. Scores for learning discrete causal graphs with unobserved confounders. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(10):11043–11051, Mar. 2024. doi: 10.1609/aaai.v38i10.28980. URL https://ojs.aaai.org/index.php/AAAI/article/view/28980.

[7] Robert Castelo and Arno Siebes. Priors on network structures. biasing the search for bayesian networks. *International Journal of Approximate Reasoning*, 24(1):39–57, 2000. ISSN 0888-613X. doi: https://doi.org/10.1016/S0888-613X(99)00041-9. URL https://www.sciencedirect.com/science/article/pii/S0888613X99000419.

[8] David Maxwell Chickering. A transformational characterization of equivalent bayesian network structures. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, page 87–98, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603859.

[9] David Maxwell Chickering. Optimal structure identification with greedy search. *J. Mach. Learn. Res.*, 3(null):507–554, March 2003. ISSN 1532-4435. doi: 10.1162/153244303321897717. URL https://doi.org/10.1162/153244303321897717.

[10] David Maxwell Chickering and Christopher Meek. Selective greedy equivalence search: finding optimal bayesian networks using a polynomial number of score evaluations. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, UAI'15, page 211–219, Arlington, Virginia, USA, 2015. AUAI Press. ISBN 9780996643108.

[11] David Maxwell Chickering, David Heckerman, and Christopher Meek. Large-sample learning of bayesian networks is np-hard. *J. Mach. Learn. Res.*, 5:1287–1330, December 2004. ISSN 1532-4435.

[12] Max Chickering. Statistically efficient greedy equivalence search. In Jonas Peters and David Sontag, editors, *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 241–249. PMLR, 03–06 Aug 2020. URL https://proceedings.mlr.press/v124/chickering20a.html.

[13] Tom Claassen and Ioan G. Bucur. Greedy equivalence search in the presence of latent confounders. In James Cussens and Kun Zhang, editors, *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 443–452. PMLR, 01–05 Aug 2022. URL https://proceedings.mlr.press/v180/claassen22a.html.

[14] Anthony C. Constantinou, Zhigao Guo, and Neville K. Kitson. The impact of prior knowledge on causal structure learning. *Knowledge and Information Systems*, 65(8):3385–3434, August 2023. ISSN 0219-3116. doi: 10.1007/s10115-023-01858-x. URL https://doi.org/10.1007/s10115-023-01858-x.

[15] Gregory F. Cooper and Edward Herskovits. A bayesian method for the induction of probabilistic networks from data. *Mach. Learn.*, 9(4):309–347, October 1992. ISSN 0885-6125. doi: 10.1023/A:1022649401552. URL https://doi.org/10.1023/A:1022649401552.

[16] Juan Correa and Elias Bareinboim. A calculus for stochastic interventions:causal effect identification and surrogate experiments. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(06):10093–10100, Apr. 2020. doi: 10.1609/aaai.v34i06.6567. URL https://ojs.aaai.org/index.php/AAAI/article/view/6567.

[17] Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P Fulco, Livnat Jerby-Arnon, Nemanja D Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, Britt Adamson, Thomas M Norman, Eric S Lander, Jonathan S Weissman, Nir Friedman, and Aviv Regev. Perturb-seq: Dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. *Cell*, 167(7):1853–1866.e17, December 2016. doi: 10.1016/j.cell.2016.11.038.

[18] D. Dor and M. Tarsi. A simple algorithm to construct a consistent extension of a partially oriented graph. 1992. URL https://www.semanticscholar.org/paper/A-simple-algorithm-to-construct-a-consistent-of-a-Dor-Tarsi/3cee18b1965fd94bd98a1e7f2155250276424925.

[19] Julien Dubois, Hiroyuki Oya, J. Michael Tyszka, Matthew Howard, Frederick Eberhardt, and Ralph Adolphs. Causal mapping of emotion networks in the human brain: Framework and initial findings. *Neuropsychologia*, 145:106571, 2020. ISSN 0028-3932. doi: https://doi.org/10.1016/j.neuropsychologia.2017.11.015. URL https://www.sciencedirect.com/science/article/pii/S0028393217304281. The Neural Basis of Emotion.

[20] Juan L. Gamella, Armeen Taeb, Christina Heinze-Deml, and Peter Bühlmann. Characterization and greedy learning of gaussian structural causal models under unknown interventions. *arXiv preprint arXiv:2211.14897*, 2022.

[21] Dan Geiger, David Heckerman, Henry King, and Christopher Meek. Stratified exponential families: Graphical models and model selection. *The Annals of Statistics*, 29(2):505–529, 2001. ISSN 00905364, 21688966. URL http://www.jstor.org/stable/2674112.

[22] Uzma Hasan and Md Osman Gani. Optimizing data-driven causal discovery using knowledge-guided search, 2024. URL https://arxiv.org/abs/2304.05493.

[23] Dominique M. A. Haughton. On the choice of a model to fit data from an exponential family. *The Annals of Statistics*, 16(1):342–355, 1988. ISSN 00905364, 21688966. URL http://www.jstor.org/stable/2241441.

[24] Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *J. Mach. Learn. Res.*, 13(1):2409–2464, August 2012. ISSN 1532-4435.

[25] Yangbo He, Jinzhu Jia, and Bin Yu. Counting and exploring sizes of markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 16(79):2589–2609, 2015. URL http://jmlr.org/papers/v16/he15a.html.

[26] David Heckerman and Dan Geiger. Learning bayesian networks: a unification for discrete and gaussian domains. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, page 274–284, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603859.

[27] A. Jaber, M. Kocaoglu, K. Shanmugam, and E. Bareinboim. Causal discovery from soft interventions with unknown targets: Characterization and learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9551–9561, Vancouver, Canada, Jun 2020. Curran Associates, Inc.

[28] Hyunchai Jeong, Adiba Ejaz, Jin Tian, and Elias Bareinboim. Testing causal models with hidden variables in polynomial delay via conditional independencies. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(25):26813–26822, April 2025. ISSN 2159-5399. doi: 10.1609/aaai.v39i25.34885. URL http://dx.doi.org/10.1609/aaai.v39i25.34885.

[29] Murat Kocaoglu, Amin Jaber, Karthikeyan Shanmugam, and Elias Bareinboim. Characterization and learning of causal graphs with latent variables from soft interventions. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/c3d96fbd5b1b45096ff04c04038fff5d-Paper.pdf.

[30] Steffen L Lauritzen, A Philip Dawid, Birgitte N Larsen, and H. G. G Leimer. Independence Properties of Directed Markov Fields. *Networks*, 20(5):491–505, 1990. ISSN 10970037. doi: 10.1002/net.3230200503.

[31] Thuc Duy Le, Tao Hoang, Jiuyong Li, Lin Liu, Huawen Liu, and Shu Hu. A fast pc algorithm for high dimensional causal discovery with multi-core pcs. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 16(5):1483–1495, September 2019. ISSN 1545-5963. doi: 10.1109/TCBB.2016.2591526. URL https://doi.org/10.1109/TCBB.2016.2591526.

[32] Adam Li, Amin Jaber, and Elias Bareinboim. Causal discovery from observational and interventional data across multiple environments. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 16942–16956. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/368cba57d00902c752eaa9e4770bbbbe-Paper-Conference.pdf.

[33] Stephanie Long, Alexandre Piché, Valentina Zantedeschi, Tibor Schuster, and Alexandre Drouin. Causal discovery with language models as imperfect experts, 2023. URL https://arxiv.org/abs/2307.02390.

[34] Christopher Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, page 403–410, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603859.

[35] Christopher Meek. Graphical Models: Selecting causal and statistical models. 1 1997. doi: 10.1184/R1/22696393.v1. URL https://kilthub.cmu.edu/articles/thesis/Graphical_Models_Selecting_causal_and_statistical_models/22696393.

[36] Achille Nazaret and David Blei. Extremely greedy equivalence search. In Negar Kiyavash and Joris M. Mooij, editors, *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence*, volume 244 of *Proceedings of Machine Learning Research*, pages 2716–2745. PMLR, 15–19 Jul 2024. URL https://proceedings.mlr.press/v244/nazaret24a.html.

[37] Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. On the role of sparsity and dag constraints for learning linear dags. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[38] Ignavier Ng, Biwei Huang, and Kun Zhang. Structure learning with continuous optimization: A sober look and beyond. In Francesco Locatello and Vanessa Didelez, editors, *Proceedings of the Third Conference on Causal Learning and Reasoning*, volume 236 of *Proceedings of Machine Learning Research*, pages 71–105. PMLR, 01–03 Apr 2024. URL https://proceedings.mlr.press/v236/ng24a.html.

[39] Chris J. Oates, Jessica Kasza, Julie A. Simpson, and Andrew B. Forbes. Brief report: Repair of partly misspecified causal diagrams. *Epidemiology*, 28(4):pp. 548–552, 2017. ISSN 10443983, 15315487. URL https://www.jstor.org/stable/26512182.

[40] Rodney T. O'Donnell, Lloyd Allison, and Kevin B. Korb. Learning hybrid bayesian networks by mml. In Abdul Sattar and Byeong-ho Kang, editors, *AI 2006: Advances in Artificial Intelligence*, pages 192–203, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-49788-2.

[41] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, CA, 1988.

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

[43] Joseph Ramsey, Madelyn Glymour, Ruben Sanchez-Romero, and Clark Glymour. A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International Journal of Data Science and Analytics*, 3:121–129, 2017. doi: 10.1007/s41060-016-0032-z. URL https://doi.org/10.1007/s41060-016-0032-z.

[44] Garvesh Raskutti and Caroline Uhler. Learning directed acyclic graph models based on sparsest permutations. *Stat*, 7(1):e183, 2018. doi: https://doi.org/10.1002/sta4.183. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sta4.183. e183 sta4.183.

[45] Alexander G. Reisach, Christof Seiler, and Sebastian Weichwald. Beware of the simulated dag! causal discovery benchmarks may be easy to game. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

[46] Thomas Richardson and Peter Spirtes. Ancestral graph Markov models. *The Annals of Statistics*, 30(4):962 – 1030, 2002. doi: 10.1214/aos/1031689015. URL https://doi.org/10.1214/aos/1031689015.

[47] Jakob Runge. Causal network reconstruction from time series: From theoretical assumptions to practical estimation. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(7):075310, 07 2018. ISSN 1054-1500. doi: 10.1063/1.5025050. URL https://doi.org/10.1063/1.5025050.

[48] Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721): 523–529, 2005.

[49] Reuben A. Saunders, William E. Allen, Xingjie Pan, Jaspreet Sandhu, Jiaqi Lu, Thomas K. Lau, Karina Smolyar, Zuri A. Sullivan, Catherine Dulac, Jonathan S. Weissman, and Xiaowei Zhuang. A platform for multimodal in vivo pooled genetic screens reveals regulators of liver function. *bioRxiv*, 2024. doi: 10.1101/2024.11.18.624217. URL https://www.biorxiv.org/content/early/2024/11/21/2024.11.18.624217.

[50] Richard Scheines, Peter Spirtes, Clark Glymour, Christopher Meek, and Thomas Richardson. The tetrad project: Constraint based aids to causal model specification. *Multivariate Behavioral Research*, 33(1):65–117, 1998.

[51] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978. ISSN 00905364, 21688966. URL http://www.jstor.org/stable/2958889.

[52] Rajen D. Shah and Jonas Peters. The hardness of conditional independence testing and the generalised covariance measure. *The Annals of Statistics*, 48(3), June 2020. ISSN 0090-5364. doi: 10.1214/19-aos1857. URL http://dx.doi.org/10.1214/19-AOS1857.

[53] Peter Spirtes, Christopher Meek, and Thomas Richardson. Causal inference in the presence of latent variables and selection bias. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, page 499–506, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603859.

[54] Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, Cambridge, MA, 2nd edition, 2000.

[55] Sofia Triantafillou and Ioannis Tsamardinos. Score based vs constraint based causal learning in the presence of confounders. 2016. URL http://www.its.caltech.edu/~fehardt/UAI2016WS/papers/Triantafillou.pdf.

[56] Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, October 2006. ISSN 1573-0565. doi: 10.1007/s10994-006-6889-7. URL https://doi.org/10.1007/s10994-006-6889-7.

[57] Thomas Verma and Judea Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, UAI '90, page 255–270, USA, 1990. Elsevier Science Inc. ISBN 0444892648.

[58] Yuhao Wang, Liam Solus, Karren Dai Yang, and Caroline Uhler. Permutation-based causal inference algorithms with interventions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5824–5833, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[59] Marcel Wienöbst, Max Bannach, and Maciej Liśkiewicz. Polynomial-Time Algorithms for Counting and Sampling Markov Equivalent DAGs, December 2020. URL http://arxiv.org/abs/2012.09679. arXiv:2012.09679 [cs].

[60] Jiji Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16):1873–1896, 2008. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2008.08.001. URL https://www.sciencedirect.com/science/article/pii/S0004370208001008.

[61] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. Dags with no tears: continuous optimization for structure learning. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 9492–9503, Red Hook, NY, USA, 2018. Curran Associates Inc.

[62] Yujia Zheng, Biwei Huang, Wei Chen, Joseph Ramsey, Mingming Gong, Ruichu Cai, Shohei Shimizu, Peter Spirtes, and Kun Zhang. Causal-learn: Causal discovery in python. *Journal of Machine Learning Research*, 25(60):1–8, 2024.