TECHNICAL REPORT
R-134
May, 2025

Less Greedy Equivalence Search

Adiba Ejaz **Elias Bareinboim**
Causal Artificial Intelligence Lab
Columbia University
{adiba.ejaz,eb}@cs.columbia.edu

Abstract

Greedy Equivalence Search (GES) is a classic score-based algorithm for causal discovery from observational data. In the sample limit, it recovers the Markov equivalence class of graphs that describe the data. Still, it faces two challenges in practice: computational cost and finite-sample accuracy. In this paper, we develop Less Greedy Equivalence Search (LGES), a variant of GES that retains its theoretical guarantees while partially addressing these limitations. LGES modifies the greedy step: rather than always applying the highest-scoring insertion, it avoids edge insertions between variables for which the score implies some conditional independence. This more targeted search yields up to a 10-fold speed-up and a substantial reduction in structural error relative to GES. Moreover, LGES can guide the search using prior assumptions, while correcting these assumptions when contradicted by the data. Finally, LGES can exploit interventional data to refine the learned observational equivalence class. We prove that LGES recovers the true equivalence class in the sample limit from observational and interventional data, even with misspecified prior assumptions. Experiments demonstrate that LGES outperforms GES and other baselines in speed, accuracy, and robustness to misspecified assumptions. Our code is available at https://github.com/CausalAILab/lges.

1 Introduction

Causal discovery, the task of learning causal structure from data, is a core problem in the field of causality [48]. The causal structure may be an end in itself to the scientist, or a prerequisite for downstream tasks such as inference, decision-making, and generalization [1, 36]. Causal discovery algorithms have been applied to a range of disciplines that span biology, medicine, climate science, and neuroscience, among others [15, 37, 41, 42].

A hallmark of the field is the algorithm known as Greedy Equivalence Search (GES) [7, 29], which takes as input observational data and finds a *Markov equivalence class* (MEC) of causal graphs that describe the data. In general, the true graph is not uniquely identifiable from observational data, and the MEC is the most informative structure that can be learned. Under standard assumptions in causal discovery, GES is guaranteed to recover the true MEC in the sample limit. In contrast, many causal discovery algorithms—including prominent examples such as max-min hill-climbing [50] and NoTears [54]—lack such large-sample guarantees. Many variants of GES have been developed, including faster, parallelized implementations [37], restricted search over bounded in-degree graphs [8], and Greedy Interventional Equivalence Search (GIES) [19], which can exploit interventional data but is not asymptotically correct [52].

Despite its attractive features, the GES family faces challenges shared across most causal discovery algorithms. For instance, the problem of causal discovery is NP-hard [9], and GES commonly struggles to scale in high-dimensional settings. Moreover, in finite-sample regimes, GES often fails to recover the true MEC. In other words, applying GES in practice is challenging due to both computational complexity (scaling) and sample complexity (accuracy) issues. We refer readers to [50] for an extensive empirical study of GES performance.

At a high-level, GES searches over the space of MECs by inserting and deleting edges to maximise a score that reflects data fit. At each state, it evaluates a set of neighbors—possibly exponentially many—and moves to the *highest-scoring* neighbor that scores more than the current MEC. It continues the search greedily until no higher-scoring neighbors are found. In the sample limit, this strategy is guaranteed to find the global optimum of the score: the true MEC.

In this paper, we first show that GES recovers the true MEC even if it moves to *any* neighbor that scores more than the current state—not necessarily the highest-scoring one (Alg. 3, Thm. 1). This relaxed greedy strategy still finds the global optimum of the score in the sample limit. More importantly, it opens the door to more strategic neighbor selection. While it may seem that choosing the highest-scoring neighbor would yield the best performance in practice, surprisingly, we show that this is not the case; a careful and *less greedy* choice improves both accuracy and runtime. Based on this insight, we develop Less Greedy Equivalence Search (LGES) (Alg. 1), a score-based algorithm for causal discovery from observational and interventional data. LGES can also leverage possibly misspecified prior assumptions, for e.g., a hypothesized graph, to guide the search while correcting misspecified edges. In particular, LGES advances on GES in the following ways:

1. **Faster, more accurate observational learning.** In Sec. 3.2, we introduce two novel strategies, CONSERVATIVEINSERT and SAFEINSERT, which LGES exploits to choose which neighbor to move to at a given state. Empirically, these procedures yield up to a 10-fold reduction in runtime and 2-fold reduction in structural error relative to GES (Experiment 5.1). LGES with SAFEINSERT asymptotically recovers the true MEC (Prop. 2, Cor. 1).
2. **Repairing misspecified causal models.** In Sec. 3.3, we show how LGES can take as input a partially misspecified causal model expressed as prior assumptions about required or forbidden edges and repair it to align with the true MEC. We evaluate performance under varying levels of misspecification and find that LGES is more robust, i.e., able to correct misspecified assumptions, than GES initialized with the same assumptions (Experiment 5.2).
3. **Faster interventional learning.** In Sec. 4, we develop a score-based procedure, $\mathcal{I}$-ORIENT (Alg. 2, Thm. 2), that LGES (or any observational learning algorithm) can use to refine an observational MEC with interventional data. To our knowledge, this is the first asymptotically correct score-based procedure for learning from interventional data that can scale to graphs with more than a hundred nodes. LGES with $\mathcal{I}$-ORIENT is 10x faster than GIES [19] while maintaining competitive accuracy (Experiment 5.3).

Proofs for all results are provided in Appendix C. Experimental details and further experiments with synthetic data and real-world protein signalling data [42] are provided in Appendix D.

2 Background

Notation. Capital letters denote variables (V), small letters denote their values (v), and bold letters denote sets of variables ($\mathbf{V}$) and their values ($\mathbf{v}$). $P(\mathbf{v})$ denotes a probability distribution over a set of variables $\mathbf{V}$. For disjoint sets of variables $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$, $\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}$ denotes that $\mathbf{X}$ and $\mathbf{Y}$ are conditionally independent given $\mathbf{Z}$ and $\mathbf{X} \perp_d \mathbf{Y} \mid \mathbf{Z}$ denotes that $\mathbf{X}$ and $\mathbf{Y}$ are *d-separated* given $\mathbf{Z}$ in the graph in context.

Causal graphs [36, 2]. A causal graph over variables $\mathbf{V}$ is a directed acyclic graph (DAG) with an edge $X \to Y$ denoting that X is a possible cause of Y. The parents of a variable X in a graph $\mathcal{G}$, denoted $\mathbf{Pa}_X^{\mathcal{G}}$, are those variables with a directed edge into X. The superscript will be omitted when clear from context. A given distribution $P(\mathbf{v})$ is said to be *Markov* with respect to a DAG $\mathcal{G}$ if for all disjoint sets $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{V}$, if $\mathbf{X} \perp_d \mathbf{Y} \mid \mathbf{Z}$ in $\mathcal{G}$, then $\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}$ in $P(\mathbf{v})$. If the converse is also true, $P(\mathbf{v})$ is said to be *faithful* to $\mathcal{G}$. In this work, like GES [7], we assume that the system of interest is *Markovian*, i.e. it contains no unobserved confounders, and that there exists a DAG $\mathcal{G}$ with respect to which the given $P(\mathbf{v})$ is both Markov and faithful.

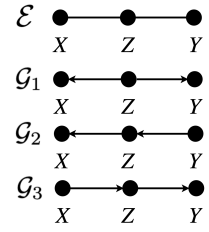


Figure 1: A CPDAG $\mathcal{E}$ and the three DAGs in the MEC it represents, encoding $X \perp_d Y \mid Z$.

Markov equivalence classes [35, 48]. Two causal DAGs $\mathcal{G}, \mathcal{H}$ are said to be Markov equivalent if they encode exactly the same d-separations. The Markov equivalence class (MEC) of a DAG is the set of all graphs that are Markov equivalent to it. A given $P(\mathbf{v})$ may be Markov and faithful with respect to more than DAG. Hence, the target of causal discovery from observational data is the MEC of DAGs with respect to which $P(\mathbf{v})$ is Markov and faithful. An MEC $\mathcal{M}$ is represented by a unique completed partially directed graph (CPDAG). A CPDAG $\mathcal{E}$ for $\mathcal{M}$ has an undirected edge $X - Y$ if $\mathcal{M}$ contains two DAGs $\mathcal{G}_1, \mathcal{G}_2$ with $X \to Y$ in $\mathcal{G}_1$ and $Y \to X$ in $\mathcal{G}_2$. $\mathcal{E}$ has a directed edge $X \to Y$ if $X \to Y$ is in every DAG in $\mathcal{M}$. We frequently refer to an MEC by its representative CPDAG. The adjacencies (neighbours) of a variable X in a CPDAG $\mathcal{E}$, denoted $\mathbf{Adj}_{\mathbf{X}}^{\mathcal{G}}$ ($\mathbf{Ne}_{\mathbf{X}}^{\mathcal{G}}$), comprise those variables connected by any edge (an undirected edge) to X.

Greedy Equivalence Search [7, 29]. Greedy Equivalence Search (GES) is a score-based algorithm for learning MECs from observational data. It searches for the true MEC by maximizing a scoring criterion given m samples of data $\mathbf{D} \sim P(\mathbf{v})$. For example, a popular choice of scoring criterion is the *Bayesian information criterion* (BIC) [45].

GES assumes that the given scoring criterion is decomposable, consistent, and score-equivalent, so that the score of an MEC is the score of any DAG in that MEC (Defs. A.3, A.4, A.5). BIC satisfies each of these conditions for distributions that are Markov and faithful to some DAG and are curved exponential families, for e.g., linear-Gaussian or multinomial models [7, 17, 18]. Moreover, decomposability and consistency imply local consistency ([7, Lemma 7]), the key property needed for the correctness of GES.

Definition 1 (Locally consistent scoring criterion [7, Def. 6])**.** Let $\mathbf{D}$ be a dataset consisting of i.i.d. samples from some distribution $P(\mathbf{v})$. Let $\mathcal{G}$ be any DAG, and let $\mathcal{G}'$ be the DAG that results from adding the edge $X \to Y$ to $\mathcal{G}$. A scoring criterion S is said to be *locally consistent* if, as the number of samples goes to infinity, the following two properties hold:

1. If $X \not\perp\!\!\!\perp Y \mid \mathbf{Pa}_Y^{\mathcal{G}}$ in $P(\mathbf{v})$ then $S(\mathcal{G}, \mathbf{D}) < S(\mathcal{G}', \mathbf{D})$.

2. If $X \perp\!\!\!\perp Y \mid \mathbf{Pa}_Y^{\mathcal{G}}$ in $P(\mathbf{v})$ then $S(\mathcal{G}, \mathbf{D}) > S(\mathcal{G}', \mathbf{D})$.

Example 1. Consider a distribution $P(\mathbf{v})$ whose true MEC is $\mathcal{E}$ and true DAG is $\mathcal{G}_2 \in \mathcal{E}$ as in Fig. 1. Consider $\mathcal{G}_1 \in \mathcal{E}$ (Fig. 1), and let $\mathcal{G}_1^+ = \mathcal{G}_1 \cup \{X \to Y\}$ and $\mathcal{G}_1^- = \mathcal{G}_1 \setminus \{Z \to Y\}$. Since $Y \perp\!\!\!\perp X \mid \mathbf{Pa}_Y^{\mathcal{G}_1}$ in $P(\mathbf{v})$, where $\mathbf{Pa}_Y^{\mathcal{G}_1} = \{Z\}$, $\mathcal{G}_1^+$ has a lower score than $\mathcal{G}_1$. Since $Y \not\perp\!\!\!\perp Z \mid \mathbf{Pa}_Y^{\mathcal{G}_1^-}$ in $P(\mathbf{v})$, where $\mathbf{Pa}_Y^{\mathcal{G}_1^-} = \emptyset$, $\mathcal{G}_1^-$ has a lower score than $\mathcal{G}_1$. □

Given a scoring criterion satisfying the above conditions and data $\mathbf{D} \sim P(\mathbf{v})$ where $P(\mathbf{v})$ is Markov and faithful to some DAG, GES recovers the true MEC in the sample limit [7, Lemma 10]. The PC algorithm [48], a constraint-based method, has similar asymptotic correctness guarantees, but uses conditional independence (CI) tests instead of a score. PC starts with a fully connected graph and removes edges using CI tests. In contrast, GES starts with a fully disconnected graph and proceeds in two phases. In the forward phase, at each state, GES finds the highest-scoring INSERT operator that results in a score increase, applies it, and repeats until no score-increasing INSERT operator exists. At this point, it has found an MEC $\mathcal{E}$ with respect to which $P(\mathbf{v})$ is Markov.

Definition 2 (INSERT operator, [7, Def. 12])**.** Given a CPDAG $\mathcal{E}$, non-adjacent nodes X, Y in $\mathcal{E}$, and some $\mathbf{T} \subseteq \mathbf{Ne}_Y^{\mathcal{E}} \setminus \mathbf{Adj}_X^{\mathcal{E}}$, the INSERT$(X, Y, \mathbf{T})$ operator modifies $\mathcal{E}$ by inserting the edge $X \to Y$ and directing the previously undirected edges $T - Y$ for $T \in \mathbf{T}$ as $T \to Y$.

Intuitively, an INSERT$(X, Y, \mathbf{T})$ operator applied to an MEC $\mathcal{E}$ corresponds to choosing a DAG $\mathcal{G} \in \mathcal{E}$ (depending on X, Y, and $\mathbf{T}$), adding the edge $X \to Y$ to $\mathcal{G}$, and computing the MEC of the resulting DAG. In this paper, we focus on the forward phase of GES. Though $P(\mathbf{v})$ is Markov with respect to the MEC $\mathcal{E}$ found in the forward phase, $P(\mathbf{v})$ may not be faithful to $\mathcal{E}$. In the backward phase, GES starts the search with $\mathcal{E}$. finds the highest-scoring DELETE$(X, Y, \mathbf{H})$ operator (Def. A.2) that results in a score increase, applies it, and repeats until no score-increasing DELETE operator exists. At this point, it has found an MEC with respect to which $P(\mathbf{v})$ is both Markov and faithful.

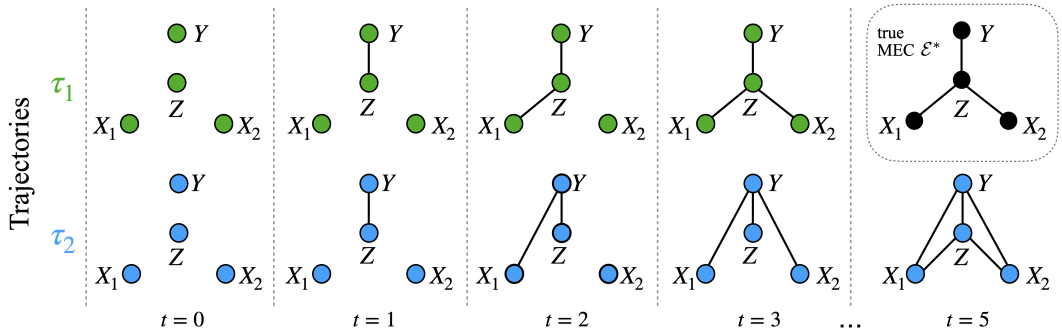


Figure 2: Possible trajectories, τ_1 and τ_2, that GES may take in the forward phase to obtain an MEC with respect to which a given distribution $P(\mathbf{v})$ is Markov. The true MEC is $\mathcal{E}^*$ (top right). In each trajectory, $\mathcal{E}^{(t+1)}$ results from applying some INSERT operator to $\mathcal{E}^{(t)}$.

3 Less Greedy Equivalence Search

3.1 Generalizing GES

To lay the groundwork for our search strategy, we first introduce Generalized GES (GGES) (Alg. 3), which generalizes GES in two ways. Firstly, GGES allows the search to be initialized from an arbitrary MEC $\mathcal{E}_0$, rather than the empty graph. Secondly, in both the forward and backward phases, GGES allows the application of *any* valid score-increasing operator, rather than the highest-scoring one.

At each state $\mathcal{E}$, GGES calls abstract subroutines GETINSERT or GETDELETE, which either return a valid score-increasing operator if one exists or indicate that there is no such operator. The forward and backward phases proceed until no improvements are found.

Theorem 1 (Correctness of GGES)**.** *Let $\mathcal{E}$ denote the Markov equivalence class that results from GGES (Alg. 3) initialised 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 $P(\mathbf{v})$.*

In the next section, we illustrate how GETINSERT can be implemented in a way that yields significant improvements in accuracy and runtime relative to GES.

3.2 An improved forward phase

In practice, the output of GES is known to include adjacencies between many variables that are non-adjacent in the true MEC [30, 50]. Since these adjacencies are introduced by INSERT operators in the forward phase, this motivates a more careful choice of which INSERT operator to apply. Our approach is grounded in the following observation.

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) $Y \not\perp\!\!\!\perp X \mid \mathbf{Pa}_Y^{\mathcal{G}}$ in $P(\mathbf{v})$.

Then, for a variable pair (X, Y), even a single score-decreasing INSERT$(X, Y, \mathbf{T})$ implies that (X, Y) are non-adjacent in the true MEC. However, this does not imply that all INSERT$(X, Y, *)$ are also score-decreasing. GES may thus apply a different INSERT$(X, Y, \mathbf{T}')$, introducing an adjacency not present in the true MEC. The following example shows how such choices can lead GES to MECs that contain many excess adjacencies.

Example 2. Consider a distribution $P(\mathbf{v})$ over $\mathbf{V} = \{X_1, X_2, Y, Z\}$ whose true MEC is given by $\mathcal{E}^*$ in Fig. 2 (top right). GES starts with the empty graph and successively applies the highest-scoring INSERT operator that it finds. Trajectories τ_1 and τ_2 agree until time $t = 1$. Let $\mathcal{E}^{(1)}$ denote the CPDAG common to τ_1 and τ_2 at $t = 1$. At $t = 1$, GES has many INSERT operators it could apply to $\mathcal{E}^{(1)}$. Recall that each INSERT$(\alpha, \beta, \mathbf{T})$ applied to $\mathcal{E}^{(1)}$ corresponds to choosing some DAG $\mathcal{G}$ from

$\mathcal{E}^{(1)}$ and adding $\alpha \to \beta$ to it. The DAG $\mathcal{G}$ is chosen such that for edges $\gamma - \beta$ in $\mathcal{E}^{(1)}$ where α and γ are non-adjacent, $\mathcal{G}$ contains $\gamma \to \beta$ if $\gamma \in \mathbf{T}$ and $\beta \to \gamma$ otherwise.

1. $\alpha = X_1, \beta = Z, \mathbf{T} = \emptyset$. This corresponds to choosing $\mathcal{G}_1 \in \mathcal{E}^{(1)}$ (which already has $Z \to Y$) and adding $X_1 \to Z$ to it (Fig. 3, left). Since $Z \not\perp\!\!\!\perp X_1 \mid \mathbf{Pa}_Z^{\mathcal{G}_1}$, this edge addition increases the score of $\mathcal{G}_1$ (by local consistency, Def. 1) and hence of $\mathcal{E}^{(1)}$. This operator is chosen in trajectory τ_1.
2. $\alpha = X_1, \beta = Y, \mathbf{T} = \{Z\}$. This corresponds to choosing $\mathcal{G}_1 \in \mathcal{E}^{(1)}$ (which already has $Z \to Y$) and adding $X_1 \to Y$ to it (Fig. 3, middle). Since $Y \perp\!\!\!\perp X_1 \mid \mathbf{Pa}_Y^{\mathcal{G}_1}$, this edge addition decreases the score of $\mathcal{G}_1$ and hence of $\mathcal{E}^{(1)}$. This operator is never chosen.
3. $\alpha = X_1, \beta = Y, \mathbf{T} = \emptyset$. This corresponds to choosing $\mathcal{G}_2 \in \mathcal{E}^{(1)}$ (which already has $Y \to Z$) and adding $X_1 \to Y$ to it (Fig. 3, right). Since $Y \not\perp\!\!\!\perp X_1 \mid \mathbf{Pa}_Y^{\mathcal{G}_2}$, this edge addition increases the score of $\mathcal{G}_1$ and hence of $\mathcal{E}^{(1)}$. This operator is chosen in trajectory τ_2.

Which of these INSERT operators scores the highest in practice? We generated 100 linear-Gaussian datasets containing 1000 samples each according to a fixed true DAG in $\mathcal{E}^*$, following the set-up in Sec. 5.1. Then, we computed the scores of $\mathcal{G}_A : \mathcal{G}_1 \cup \{X_1 \to Z\}$, $\mathcal{G}_B : \mathcal{G}_1 \cup \{X_1 \to Y\}$, and $\mathcal{G}_C : \mathcal{G}_2 \cup \{X_1 \to Y\}$ on each dataset. From the fact that $\mathcal{G}_A$ is closer to the true MEC than $\mathcal{G}_C$, it may seem that $\mathcal{G}_A$ would almost always score higher. However, $\mathcal{G}_A$ was the highest-scoring DAG 69% of the time, and $\mathcal{G}_C$ 31% of the time. As expected, $\mathcal{G}_B$ is never the highest-scoring DAG. Hence, GES may often insert an edge between X_1 and Y. Even in the sample limit, it is unknown whether $\mathcal{G}_A$ or $\mathcal{G}_C$ would score higher. For an extended discussion, see Ex. B.1. □

This motivates avoiding edge insertions for variable pairs (X, Y) for which a score-decreasing INSERT is observed. We hypothesize this has two benefits: (1) *accuracy*: it avoids inserting excess adjacencies that the backward phase may fail to remove, and (2): *efficiency*: it stops the enumeration of (X, Y) insertions when a lower-scoring one is found; moreover, reducing excess adjacencies reduces the number of operators that need to be evaluated in subsequent states.

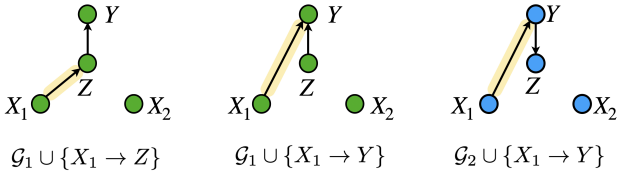


Figure 3: Illustration of some INSERT operators that may be applied to the MEC $\mathcal{E}^{(1)}$ at $t = 1$ in Fig. 2. These operators correspond to various edge additions to the DAGs $\mathcal{G}_1, \mathcal{G}_2 \in \mathcal{E}^{(1)}$, where $\mathcal{G}_1$ orients $Z - Y$ as $Z \to Y$ and $\mathcal{G}_2$ orients $Z - Y$ as $Y \to Z$.

We now formalize two strategies for avoiding such insertions.

Strategy 1 (CONSERVATIVEINSERT)**.** At a given state with CPDAG $\mathcal{E}$, for each non-adjacent pair (X, Y), iterate over valid INSERT$(X, Y, \mathbf{T})$. If any score-decreasing $\mathbf{T}$ is found, stop, discard all INSERT$(X, Y, *)$ operators and continue to the next pair. Among all retained candidates, select the highest-scoring operator that results in a score increase, if any.

The CONSERVATIVEINSERT strategy avoids inserting edges between any variables (X, Y) for which some conditional independence has been found, as evidenced by a score-decreasing INSERT (Prop. 1). While intuitive, it is unclear if this strategy is guaranteed to find a score-increasing INSERT whenever one exists. We prove partial guarantees in Prop. C.1, C.2. Moreover, we introduce a relaxation, SAFEINSERT, that is guaranteed to find a score-increasing INSERT when one exists.

Strategy 2 (SAFEINSERT)**.** At a given state with CPDAG $\mathcal{E}$, pick an arbitrary DAG $\mathcal{G} \in \mathcal{E}$. For each non-adjacent pair (X, Y) in $\mathcal{G}$, if $X \in nd_Y^{\mathcal{G}}$, check if $\mathcal{G}$ has a higher score than $\mathcal{G} \cup \{X \to Y\}$. If not, discard all INSERT$(X, Y, *)$ operators and continue to the next pair. Among all retained candidates, select the highest-scoring operator that results in a score increase, if any.

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.

Example 3. (Ex. 2 continued). Let $\mathcal{E}^{(1)}, \mathcal{G}_1$, and $\mathcal{G}_2$ be as in Ex. 2. Assume GES is at $\mathcal{E}^{(1)}$ and SAFEINSERT picks the DAG $\mathcal{G}_1 \in \mathcal{E}$. Then, $\mathcal{G}_1 \cup \{X_1 \to Y\}$ has a lower score than $\mathcal{G}_1$

Algorithm 1: 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 $\mathcal{E}_0$
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 $\mathcal{E}$;
  $priorityList \leftarrow$ GETPRIORITYINSERTS$(\mathcal{E}, \mathcal{G}, \mathbf{S})$;
  foreach $candidates$ *in* $priorityList$ **do**
    $(X_{max}, Y_{max}, \mathbf{T}_{max}) \leftarrow$ GETSAFEINSERT$(\mathcal{E}, \mathcal{G}, \mathbf{D}, candidates, S)$;
    if $(X_{max}, Y_{max}, \mathbf{T}_{max})$ *is found* **then**
      break ; `// no need to check lower priority`
  $\mathcal{E} \leftarrow \mathcal{E} +$ INSERT$(X_{max}, Y_{max}, \mathbf{T}_{max})$;
until *no improving insertions exist*;
repeat
  $\mathcal{E} \leftarrow \mathcal{E}+$ the highest-scoring DELETE$(X, Y, \mathbf{T})$ that results in a score increase
until *no improving deletions exist*;
return $\mathcal{E}$

since $X_1 \perp\!\!\!\perp Y \mid \mathbf{Pa}_Y^{\mathcal{G}_1}$ in $P(\mathbf{v})$, where $\mathbf{Pa}_Y^{\mathcal{G}_1} = \{Z\}$. SAFEINSERT thus does not consider any INSERT$(X_1, Y, *)$ operators. In contrast, assume SAFEINSERT picks the DAG $\mathcal{G}_2 \in \mathcal{E}$. Then, $\mathcal{G}_2 \cup \{X_1 \to Y\}$ has a higher score than $\mathcal{G}_2$, and SAFEINSERT may still consider INSERT$(X_1, Y, *)$ operators. However, CONSERVATIVEINSERT will not consider any INSERT$(X, Y, *)$ operators, since INSERT$(X_1, Y, \{Z\})$, corresponding to $\mathcal{G}_1 \cup \{X_1 \to Y\}$, results in a lower score than $\mathcal{E}^{(1)}$. □

Later, in Sec. 5.1, we compare the two aforementioned strategies, and show how both achieve substantial gains in accuracy and runtime over GES.

3.3 Repairing misspecified causal models

Often, researchers have prior assumptions about the underlying graph. Existing methods that leverage such assumptions for causal discovery often assume the assumptions are correct, even if contradicted by the data, and lack theoretical guarantees on the learned MEC [5, 12, 21, 34, 44]. We show how Generalized GES (GGES) can leverage such assumptions during the search process, while correcting them if inconsistent with the data.

We assume we are given prior assumptions in the form of a set of required and forbidden edges $\mathbf{S} = \langle \mathbf{R}, \mathbf{F} \rangle$ that may be either directed or undirected. A natural strategy, which we refer to as GES-INIT, initializes the search to an MEC consistent with the assumptions and then proceeds greedily, as in standard GES.[1] This approach is sound in the large-sample limit (Thm. 1), even when the assumptions are misspecified.

However, given finite samples, this strategy may result in excess adjacencies. If the prior assumptions require adjacencies that don't exist in the true MEC, GES-INIT includes them by default in the initialisation and may fail to remove them later. Moreover, such initialisation precludes the use of insertion strategies (e.g., from Sec. 3.2) that would avoid introducing such excess adjacencies.

We instead propose a strategy that uses prior assumptions to *prioritize* operators, and not to initialize the search. Specifically, for each non-adjacent pair (X, Y), we rank it into one of four categories based on the constraint set $\mathbf{S} = \langle \mathbf{R}, \mathbf{F} \rangle$ using the procedure GETPRIORITYINSERTS (Alg. 6). Insertions for higher-priority adjacencies are considered first, but only applied if they increase the score. For example, if SAFEINSERT finds no score-increasing insertions for the current MEC, then the remaining adjacencies in $\mathbf{R}$ (if any) are redundant, and will not be inserted. In contrast, the initialisation strategy inserts all adjacencies in $\mathbf{R}$ by default.

Next, in Sec. 3.4, we incorporate this prioritization scheme into a novel algorithm, combining it with the search strategy of Sec. 3.2 to enable a less greedy search. In Sec. 5.2, we empirically demonstrate the benefit of this prioritization-based strategy.

[1] This was empirically evaluated in [11]. However, its correctness was not considered.

Algorithm 2: $\mathcal{I}$-ORIENT

Input: Intervention targets $\mathcal{I}$, data $(\mathbf{D_I})_{\mathbf{I}\in\mathcal{I}} \sim (\mathbf{P_I(v)})_{\mathbf{I}\in\mathcal{I}}$, observational MEC $\mathcal{E}$, scoring criterion S

Output: $\mathcal{I}$-MEC $\mathcal{E}$ of $(\mathbf{P_I(v)})_{\mathbf{I}\in\mathcal{I}}$

foreach $X \in \mathcal{E}$ *and* $Y \in ne_X^{\mathcal{E}}$ **do**
  $\Delta S \leftarrow \sum_{\mathbf{I}\in\mathcal{I},\ X\in\mathbf{I},Y\notin\mathbf{I}} s_{\mathbf{D_I}}(y, x) - s_{\mathbf{D_I}}(y)$;
  if $\Delta S > 0$ **then**
    Orient edge $X - Y$ as $X \to Y$ in $\mathcal{E}$;
    Apply Meek's rules in $\mathcal{E}$ to propagate orientations [29] ;
  else if $\Delta S < 0$ **then**
    Orient edge $X - Y$ as $X \leftarrow Y$ in $\mathcal{E}$;
    Apply Meek's rules in $\mathcal{E}$ to propagate orientations [29] ;
return $\mathcal{E}$

3.4 The Less Greedy Equivalence Search algorithm

Finally, we introduce the main result of this work: the algorithm Less Greedy Equivalence Search (LGES, Alg. 1). LGES modifies the forward phase of GES based on our insights in the previous sections, while using the same search strategy as GES in the backward phase.

Using Thm. 1 and Prop. 2, we can show that LGES recovers the true MEC in the sample limit, even given a misspecified set of prior assumptions.

Corollary 1 (Correctness of LGES)**.** *Let $\mathcal{E}$ denote the Markov equivalence class that results from LGES (Alg. 1) 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 $P(\mathbf{v})$.*

Remark 1. While standard LGES uses SAFEINSERT, LGES can also be run with CONSERVATIVEINSERT. Since we only have partial guarantees on CONSERVATIVEINSERT (Prop. C.1, C.2), it remains open whether this variant of LGES is asymptotically correct.

4 Score-based learning from interventional data

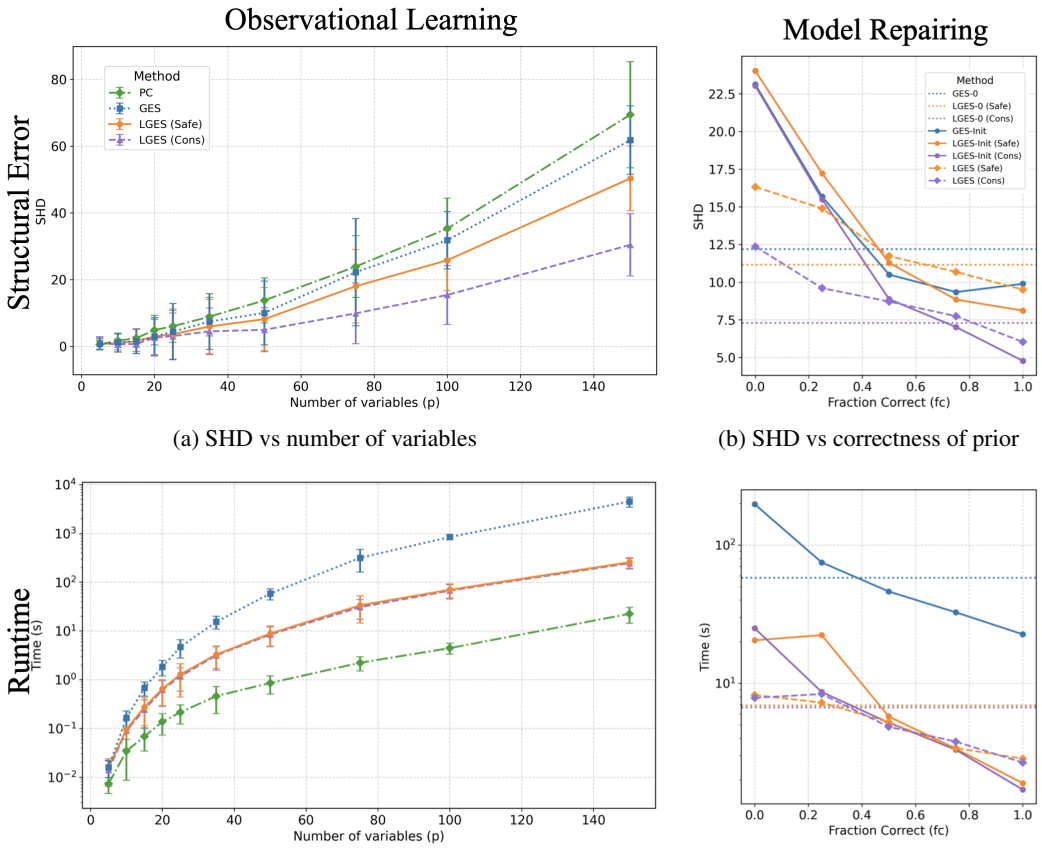
If interventional data is available, LGES can use it to further orient edges in the learned observational MEC. We now develop a score-based procedure that enables LGES to do so. Unlike existing score-based methods, which are often inconsistent or computationally infeasible even on moderately sized graphs [19, 52], our approach scales while preserving soundness.

Following [19], we assume soft unconditional interventions, including hard (do) interventions as a special case. These set the distribution of a variable X to some fixed $P^*(x)$, thereby removing the influence of its parents. Let $\mathcal{I}$ denote a family of interventional targets, i.e., subsets $\mathbf{I} \subseteq \mathbf{V}$, with the empty intervention $\theta \in \mathcal{I}$ producing the observational distribution. We observe data from distributions $(\mathbf{P_I(v)})_{\mathbf{I}\in\mathcal{I}}$. As in the observational case, we assume there exists a DAG $\mathcal{G}$ such that these distributions are $\mathcal{I}$-Markov (Def. A.7) and faithful to the corresponding intervention graphs $(\mathcal{G}_{\overline{\mathbf{I}}})_{\mathbf{I}\in\mathcal{I}}$, obtained by removing edges into any intervened variable $V \in \mathbf{I}$ [13, 36].

Just as observational data identifies an MEC, interventional data identifies an $\mathcal{I}$-MEC, a typically smaller equivalence class encoding constraints on both the observational and interventional data [19, Def. 7]. Two DAGs are $\mathcal{I}$-Markov equivalent iff they are observationally Markov equivalent and if their intervention graphs have the same adjacencies across all interventions in $\mathcal{I}$ [19, Thm. 10].

To recover the $\mathcal{I}$-MEC, we introduce $\mathcal{I}$-ORIENT (Alg. 2), which orients undirected edges in the observational MEC using scores from interventional data.

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 $(\mathbf{P_I(v)})_{\mathbf{I}\in\mathcal{I}}$ 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 $(\mathbf{P_I(v)})_{\mathbf{I}\in\mathcal{I}}$.*



(a) SHD vs number of variables

(b) SHD vs correctness of prior

(c) Runtime vs number of variables

(d) Runtime vs correctness of prior

Figure 4: Performance of algorithms on observational data from Erdős–Rényi graphs with p variables and $2p$ edges. LGES is our proposed method. **Lower is better** (more accurate / faster) across all plots. The time axis uses a log scale. SHD denotes the structural Hamming distance between the true and estimated CPDAGs. Left column: without prior assumptions, $n = 10^4$ samples. Right column: with prior assumptions, $n = 10^3$ samples, $p = 50$ variables with prior assumptions on $m/2$ edges for a true graph with m edges . Error bars denote one standard deviation across 50 random seeds; some bars are limited for legibility. Detailed figures and additional baselines in Sec D.

5 Experiments

5.1 Learning from observational data

Synthetic data and baselines. We draw Erdős–Rényi graphs with p variables and $2p$ or $3p$ edges in expectation (denoted ER2 and ER3 respectively), for p up to 500. For each p, we sample 50 graphs and generate linear-Gaussian data for each graph. Following [31], we draw weights drawn from $\mathcal{U}([-2, -0.5] \cup [0.5, 2])$, and noise variances from $\mathcal{U}([0.1, 0.5])$. We obtain samples of size $n \in \{500, 1000, 10000\}$ via `sempler` [16]. We run (1) GES with a turning phase, known to improve performance [7, 19]; (2) LGES with ConservativeInsert, (3) LGES with SafeInsert, (4) PC [48], and (5) NoTears [54]. [2] Our GES implementations share as much code as possible.

Results. In this section, we present results for ER2 graphs, p up to 150, and $n = 10^4$; see Sec. D.1 for additional results (including precision and recall metrics), which follow a similar trend. LGES both with SafeInsert and with ConservativeInsert outperforms GES in runtime and accuracy as measured by Structural Hamming Distance (SHD) [50] between the estimated and true CPDAGs (Figs. 4c, 4a). Both variants of LGES are up to an order of magnitude faster than GES, making significantly fewer scoring operations under the hood (Fig. D.1.1h). In terms of SHD, LGES with

[2] All implementations in Python. LGES, GES: `https://github.com/juangamella/ges`, PC: `causal-learn` [55], NoTears: `causal-nex` [3]

CONSERVATIVEINSERT is up to 2 times more accurate than GES, for instance, resulting in only ≈ 30 incorrect edges on average in graphs with 150 variables and 300 edges in expectation. The difference in accuracy is due to excess adjacencies and incorrect orientations; missing adjacencies almost never occur. PC, though fast, is less accurate than even GES. NoTears has much worse accuracy than other methods (for e.g., average SHD ≈ 125 on graphs with 100 variables), though its runtime appears to scale better (Figs. D.1.1c, D.1.1d).

5.2 Repairing misspecified causal models

Synthetic data and baselines. We study the effect of the correctness of prior assumptions when the available data is limited ($n = 1000$) on ER2 graphs with up to 50 variables, with data generated as in Sec. 5.1. For a true DAG $\mathcal{G}$ with m edges, we generate prior assumptions on $m' \in \{m/2,\ 3m/4\}$ required edges as follows. We vary the fraction `fc` of the chosen m' edges that is 'correct', with $c \cdot m'$ edges chosen correctly from those in $\mathcal{G}$ and the remaining chosen incorrectly from those not in $\mathcal{G}$. We compare GES-0 and LGES-0 (no initialisation); LGES (only priority insertions); and GES-INIT and LGES-INIT (only initialisation). We evaluate both variants of LGES across all settings.

Results. In Figs. 4b and 4d, we show results for $m' = m/2$, with additional results in Sec. D.2. LGES with CONSERVATIVEINSERT outperforms GES and GES-INIT across all levels of prior correctness in terms of time and SHD. When the prior is mostly accurate (`fc` $\in \{0.75, 1\}$), LGES-INIT performs marginally better than LGES, with both outperforming LGES-0 in runtime and marginally in accuracy. With more misspecification (`fc` $\in \{0.5, 0.25, 0.0\}$), LGES with CONSERVATIVEINSERT outperforms all other methods that use the prior assumptions, as well as GES-0, run without these assumptions. Thus, our prioritization strategy (Sec. 3.3) can leverage prior assumptions but still be robust to misspecification.

5.3 Learning from interventional data

Synthetic data and baselines. We follow a similar set-up as Sec. 5.1 with 10^4 observational samples. For a graph on p variables, we randomly construct $|\mathcal{I}| = p/10$ interventions and generate 10^3 samples for each. We compare LGES with SAFEINSERT / CONSERVATIVEINSERT and $\mathcal{I}$-ORIENT (denoted LGIES) against GIES [19].[3]

Results. LGES is up to 10x faster than GIES (Fig. 5). In terms of accuracy, LGES with CONSERVATIVEINSERT and GIES attain competitive SHD from the true $\mathcal{I}$-MEC (Fig. D.3.1).

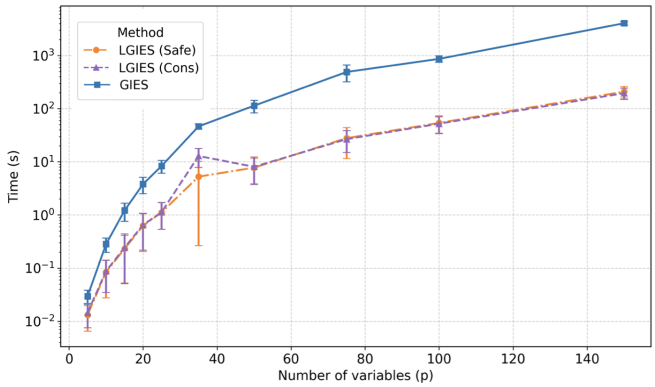


Figure 5: Runtime of LGIES and GIES on interventional data from Erdős–Rényi graphs with p variables and $2p$ edges. The time axis uses a log scale. We generate $n = 10^4$ observational samples and $n = 10^3$ samples per intervention. Error bars denote one standard deviation across 50 random seeds.

6 Conclusions

In this paper, we introduced LGES (Alg. 1), a novel and asymptotically correct algorithm for causal discovery from both observational and interventional data. LGES significantly improves on GES in terms of runtime and accuracy using a more careful and less greedy search strategy, avoiding edge insertions between variables for which it finds a witness of conditional independence. It can also leverage prior assumptions to guide the search, and remains more accurate than GES even when those assumptions are misspecified. We also developed $\mathcal{I}$-ORIENT (Alg. 2), a score-based and theoretically sound (Thm. 2) algorithm for orienting edges in the learned MEC using interventional data. Together, LGES and $\mathcal{I}$-ORIENT are significantly faster than GIES and achieve comparable accuracy. A limitation of our approach is that it uses interventional data only after the observational

[3] `https://github.com/juangamella/gies`

MEC is learned. A natural direction for future work is to incorporate interventional data during the search, while preserving LGES's theoretical guarantees.

Acknowledgements

This research is supported in part by the NSF, ONR, AFOSR, DoE, Amazon, JP Morgan, and The Alfred P. Sloan Foundation.

References

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

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

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

[37] Jseoph 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

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

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] 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.

[55] 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.

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 [35])**.** 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, [7, 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 DELETE$(X, Y, \mathbf{T})$ operator modifies $\mathcal{E}$ by

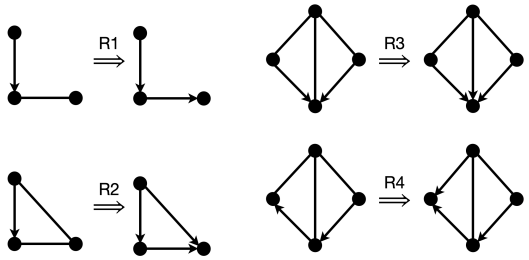


Figure A.1.1: Meek orientation rules for completing partially directed acyclic graphs

deleting the edge between X and Y, and for each $T \in \mathbf{T}$, directing any undirected edges $X - T$ as $X \to T$ and any $Y - T$ as $Y \to T$.

Definition A.3 (Decomposable scoring criterion [7, 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 [7, 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 [7, 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 [19, 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 [19, 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 $(\mathbf{P}_{\mathbf{I}}(\mathbf{v}))_{\mathbf{I} \in \mathcal{I}}$ a corresponding family of interventional distributions. We say $(\mathbf{P}_{\mathbf{I}}(\mathbf{v}))_{\mathbf{I} \in \mathcal{I}}$ 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 $(\mathbf{P}_{\mathbf{I}}(\mathbf{v}))_{\mathbf{I} \in \mathcal{I}}$ 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 [51, 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 [35])**.** 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 [35])**.** 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 [24, 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 [6, 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 [6, 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 [6, 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 [7, 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 [48] and the Sparsest Permutations (SP) algorithm [38] 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 [46]. Score-based algorithms such as GES [7, 28] 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 [50] 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 [50] for an extensive empirical analysis, who found, for instance, that GES outperforms PC in accuracy across various sample sizes [50, Tables 4, 5] .

Commonly, causal discovery algorithms struggle with scaling in high-dimensional settings. This has motivated variants such as Parallel-PC [25] and Fast Greedy Equivalence Search (FGES) [37], 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 [54] are in principle more scalable, but lack theoretical guarantees and show brittle performance even in simulated settings [32, 39].

A.2.2 Repairing misspecified causal models

The task of repairing a partially misspecified causal model using data can be understood as an example of causal discovery with background knowledge [11]. Such background knowledge may be provided by a domain expert or even a large language model [27]. Many algorithms under this umbrella—including tiered-FCI [44] and the K2 algorithm [12]—assume that the background knowledge is correct, i.e., consistent with the ground truth. However, if the expert is imperfect, such methods necessarily fail to recover the true MEC. The approach in [33] allows some misspecification of the expert knowledge, i.e. missing or excess adjacencies, but no incorrect orientations. However, their approach does not guarantee recovery of the true MEC. Other approaches treat knowledge a 'soft' prior [5, 21, 34] to guide the search, but lack theoretical guarantees on the output graph. One exception is the Sparsest Permutations (SP) algorithm [38], 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 [20]. Interventional data can help significantly refine observational MECs [19], and is becoming increasingly available, for e.g., in biological settings due to advances in single-cell technologies [14, 43]. 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) [19] and the CI-based Interventional Greedy Sparsest Permutations (I-GSP) [52]. However, in [52], it was shown that GIES is inconsistent, i.e.. not guaranteed to recover the true interventional MEC in the sample limit.

B Discussion and examples

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 [51, 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 [47, 53] and its interventional variants [22, 23, 26], guaranteed to recover the true equivalence class in the sample limit. While there has been progress towards score-based approaches [4, 10, 40, 49], finding algorithms that are asymptotically correct remains an open problem.

There is a fundamental theoretical challenge to generalizing our approach to non-Markovian settings. GES relies on a transformational characterization of Markovian causal DAGs that are (a) Markov equivalent (Thm. A.2) or (b) such that all d-separations that hold in one also hold in the other (Thm. A.3). While the former has been generalized to non-Markovian causal DAGs, the latter remains an open problem [53].

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 [38].

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

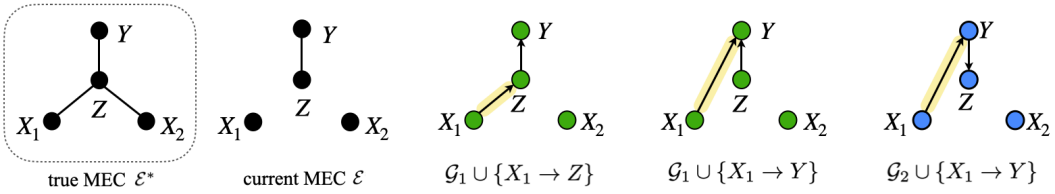


Figure B.0.1: Figs. 2, 3 partially reproduced for convenience. GES is given a distribution $P(\mathbf{v})$ whose true MEC is represented by $\mathcal{E}^*$ (left). GES is currently at MEC $\mathcal{E}$, evaluating which INSERT operator to apply next. Each INSERT corresponds to picking some $\mathcal{G} \in \mathcal{E}$ and adding some edge to it.

as BIC satisfy these criteria as long as the model is a *curved exponential family* [7, 18]. 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 [42]. 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.

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.0.1, 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.0.1, 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 additional results

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 [7, Lemma 9, 10].

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 GETINSERT 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 [35]: 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 $\mathcal{E}'$.

Backward phase. Next, we show that $P(\mathbf{v})$ is both Markov and faithful to the MEC $\mathcal{E}$ resulting from the backward phase of GGES. First, we show that $P(\mathbf{v})$ is Markov with respect to $\mathcal{E}$. The backward 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 DELETE 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 $\mathcal{E}$.

Finally, we 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, GETDELETE is guaranteed to find some score-increasing operator in this case. Thus, we have a contradiction.

□

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 [35]; 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 1 (Correctness of LGES). Let $\mathcal{E}$ denote the Markov equivalence class that results from LGES (Alg. 1) 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. 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. Since the backward step enumerates over all possible valid DELETE, each backward step is also guaranteed to find a valid score-increasing DELETE operator, if it exists. Thus, LGES satisfies the conditions of GGES (Alg. 3) and its correctness follows from Thm. 1. □

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 $\mathcal{E}$ [19, Cor. 13]. We will show that $\mathcal{E}^*$ contains $Y \to X$ if and only if for every such $\mathbf{I}$, $s_{\mathbf{D_I}}(y) > s_{\mathbf{D_I}}(y, x)$.

$\implies$ 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 $X \in \mathbf{I}, Y \notin \mathbf{I}$.

$\impliedby$ 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., $\sum_{\mathbf{I}\in\mathcal{I}, X\in\mathbf{I}, Y\notin\mathbf{I}} s_{\mathbf{D}_\mathbf{I}}(y)$ vs $\sum_{\mathbf{I}\in\mathcal{I}, X\in\mathbf{I}, Y\notin\mathbf{I}} s_{\mathbf{D}_\mathbf{I}}(y, x)$.

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 $\mathcal{E} = \mathcal{E}^*$. □

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 $\mathcal{E}_0$
Output: MEC $\mathcal{E}$ of $\mathbf{P}(\mathbf{v})$

$\mathcal{E} \leftarrow \mathcal{E}_0$;
/* forward phase */
repeat
  $(X, Y, \mathbf{T}) \leftarrow$ GETINSERT$(\mathcal{E}, \mathbf{D}, S)$;
  $\mathcal{E} \leftarrow \mathcal{E} +$ INSERT$(X, Y, \mathbf{T})$;
until *no improving insertions exist*;
/* backward phase */
repeat
  $(X, Y, \mathbf{H}) \leftarrow$ GETDELETE$(\mathcal{E}, \mathbf{D}, S)$;
  $\mathcal{E} \leftarrow \mathcal{E} +$ DELETE$(X, Y, \mathbf{H})$;
until *no improving deletions exist*;
return $\mathcal{E}$

Algorithm 4: GETCONSERVATIVEINSERT

Input: MEC $\mathcal{E}$, 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**
  $\Delta S_{xy} \leftarrow -\infty$;
  $(\hat{X}, \hat{Y}, \hat{\mathbf{T}}) \leftarrow \emptyset$;
  foreach *valid* $\mathbf{T} \subseteq \mathbf{Ne}_Y^{\mathcal{E}} \setminus \mathbf{Adj}_X^{\mathcal{E}}$ **do**
    $\Delta S \leftarrow s\big(Y, (\mathbf{Ne}_Y^{\mathcal{E}} \cap \mathbf{Adj}_X^{\mathcal{E}}) \cup \mathbf{T} \cup \mathbf{Pa}_Y^{\mathcal{E}} \cup \{X\}\big) - s\big(Y, (\mathbf{Ne}_Y^{\mathcal{E}} \cap \mathbf{Adj}_X^{\mathcal{E}}) \cup \mathbf{T} \cup \mathbf{Pa}_{Y,}^{\mathcal{E}}\big)$;
    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 $X \in \mathbf{Nd}_Y^{\mathcal{G}}$ *and* $s(Y, \mathbf{Pa}_Y^{\mathcal{G}}) < s(Y, \mathbf{Pa}_Y^{\mathcal{G}} \cup \{X\})$ **then**
    foreach *valid* $\mathbf{T} \subseteq \mathbf{Ne}_Y^{\mathcal{E}} \setminus \mathbf{Adj}_X^{\mathcal{E}}$ **do**
      $\Delta S \leftarrow s\big(Y, (\mathbf{Ne}_Y^{\mathcal{E}} \cap \mathbf{Adj}_X^{\mathcal{E}}) \cup \mathbf{T} \cup \mathbf{Pa}_Y^{\mathcal{E}} \cup \{X\}\big) - s\big(Y, (\mathbf{Ne}_Y^{\mathcal{E}} \cap \mathbf{Adj}_X^{\mathcal{E}}) \cup \mathbf{T} \cup \mathbf{Pa}_{Y,}^{\mathcal{E}}\big)$;
      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 $Y \to X \in \mathbf{R}$ **then**
    Add (X, Y) to $priorityList[2]$; `// weakly required`
  else if $X \to Y \in \mathbf{F}$ *or* $X - Y \in \mathbf{F}$ **then**
    Add (X, Y) to $priorityList[4]$; `// forbidden`
  else
    Add (X, Y) to $priorityList[3]$; `// ambivalent`
return $priorityList$

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

D.1.1 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 [31]. 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.

D.1.2 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 [31], 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 `sempler` [16].

D.1.3 Further metrics for Experiment 5.1

In Fig. D.1.1, we present additional metrics and results for the setting in Sec. 5.1, including the particular types of structural errors (excess adjacencies, missing adjacencies, incorrect orientations) and the number of scoring operations conducted by the various algorithms. As in the case of SHD and runtime, LGES outperforms GES and PC across these metrics, with CONSERVATIVEINSERT outperforming SAFEINSERT.

The behaviour of NoTears is more variable. It misses significantly more adjacencies than all other methods—approximately linearly many in the number of variables (Fig. D.1.1f). 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 $p = 150$ (Fig. D.1.1e); 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.1.2f).

D.1.4 Precision, recall, and F1 score for Experiment 5.1

Finally, we evaluate performance by considering causal discovery as a binary classification task. We use the task definition provided in [30]: 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 with CONSERATIVEINSERT dominates, followed by LGES with SAFEINSERT. For large p, LGES with CONSERATIVEINSERT achieves a substantially higher F1 score than GES and other algorithms. For instance, for $p = 150$, LGES with CONSERATIVEINSERT achieves an F1 score > 0.9, whereas GES's F1 score is slightly under 0.85.

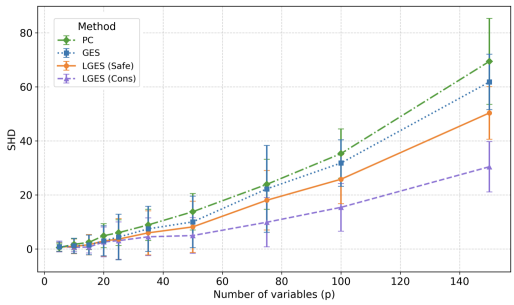
D.1.5 Additional experiments with denser graphs

In Fig. D.1.3, we provide results for Erdős–Rényi graphs with p variables and $3p$ edges in expectation (ER3), with a similar set-up as in Experiment 5.1. We ran all baselines except NoTears due to its heavy compute usage. The relative performance of algorithms is similar to the ER2 case (Fig. D.1.1). Both variants of LGES are up to 10x faster than GES, and substantially more accurate than GES and PC. This shows that our search strategy improves on GES across various edge densities, not limited to sparse or to dense graphs. A noteworthy difference from the ER2 case is in the performance of the PC algorithm. The PC algorithm now includes fewer excess adjacencies than LGES and GES (Fig. D.1.3c), but misses many more adjacencies (Fig. D.1.3d). As a result, PC has a substantially lower F1 score than all other algorithms, though its accuracy in terms of SHD is similar to GES (Fig. D.1.3i).

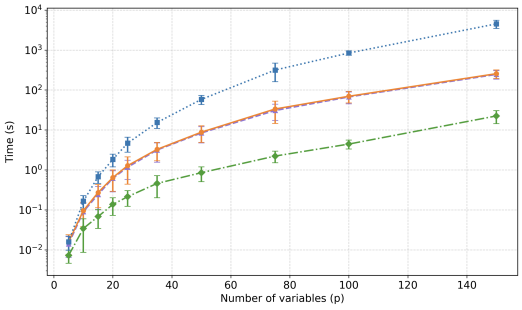
D.1.6 Additional experiments with small datasets

Our experiments in the previous sections were conducted with $n = 10^4$ samples. We conduct additional experiments with smaller sample sizes, i.e. $n \in \{500, 1000\}$ for ER2 graphs with $p \in \{10, 25, 50, 100\}$ variables. The results are summarized in Fig. D.1.4, with additional details in Table D.1.1.

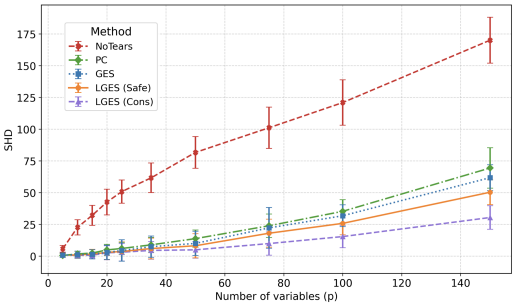
- For smaller graphs ($p \in \{10, 25\}$), GES and LGES with SAFEINSERT perform comparably across all metrics. LGES with CONSERVATIVEINSERT performs comparably to GES and LGES with SAFEINSERT for sample size $n = 500$, but outperforms them both for $n = 1000$, making fewer structural errors and achieving a higher F1 score.
- For larger graphs ($p \in \{10, 25\}$), LGES with SAFEINSERT begins to outperform GES marginally. Notably, LGES with CONSERVATIVEINSERT significantly outperforms GES, achieving > 10 fewer structural errors for both sample sizes.
- Interestingly the PC algorithm tends to outperform the other three algorithms in terms of SHD and excess adjacencies, though it tends to under-insert edges, missing substantially more adjacencies than all the other algorithms. LGES with CONSERVATIVEINSERT also



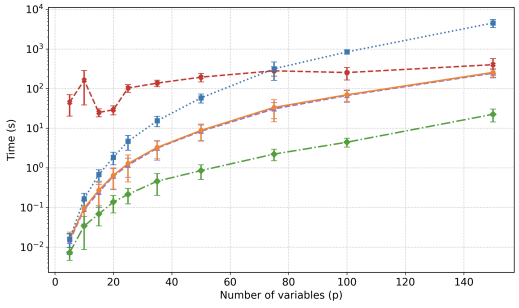
(a) SHD vs number of variables (NoTears omitted)



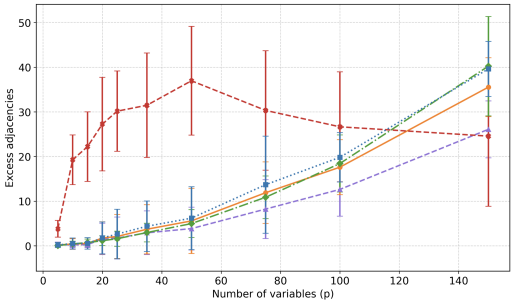
(b) Runtime vs number of variables (NoTears omitted)



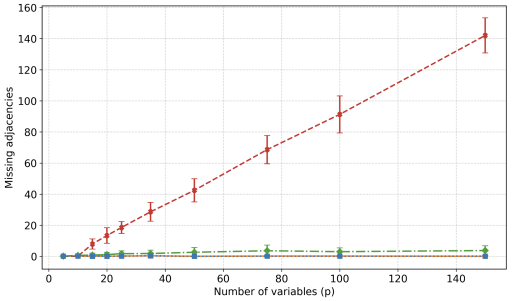
(c) SHD vs number of variables



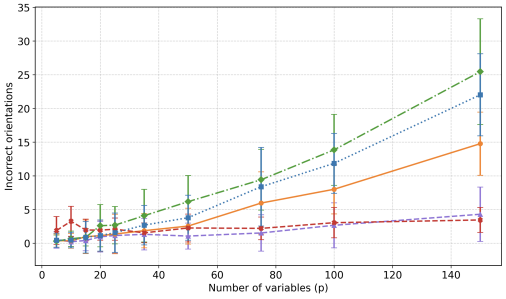
(d) Runtime vs number of variables



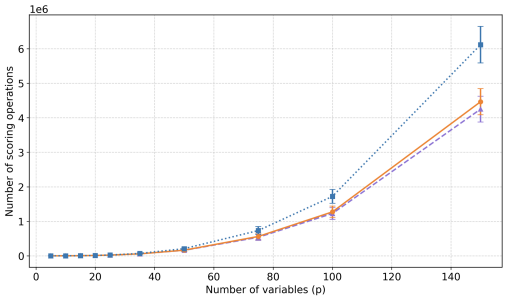
(e) Excess adjacencies vs number of variables.



(f) Missing adjacencies vs number of variables.

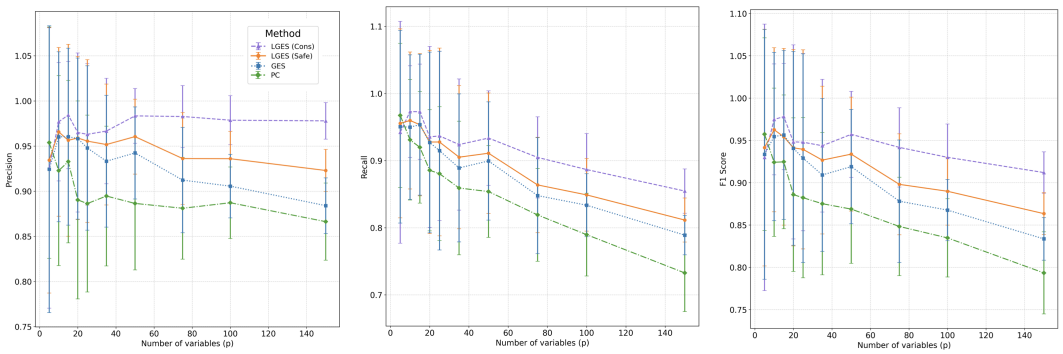


(g) Incorrect orientations vs number of variables.



(h) Scoring operations vs number of variables.

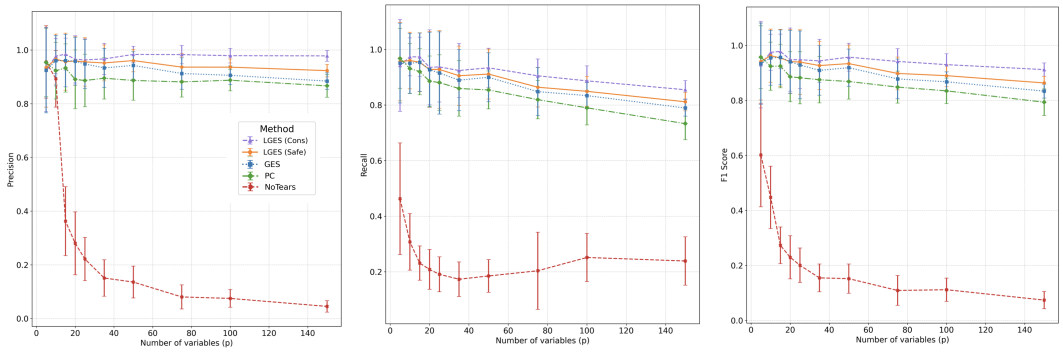
Figure D.1.1: Performance of algorithms on observational data from Erdős–Rényi graphs with p variables and $2p$ edges and $n = 10^4$ samples (Experiment 5.1, Sec. D.1.3). LGES is our proposed method. **Lower is better** (more accurate / faster) across all plots. The time axis uses a log scale. SHD denotes the structural Hamming distance between the true and estimated CPDAGs. Error bars denote one standard deviation across 50 random seeds. NoTears is omitted from the first row for clarity.



(a) Precision vs number of variables (NoTears omitted)

(b) Recall vs number of variables (NoTears omitted)

(c) F1 score vs number of variables (NoTears omitted)



(d) Precision vs number of variables

(e) Recall vs number of variables

(f) F1 score vs number of variables

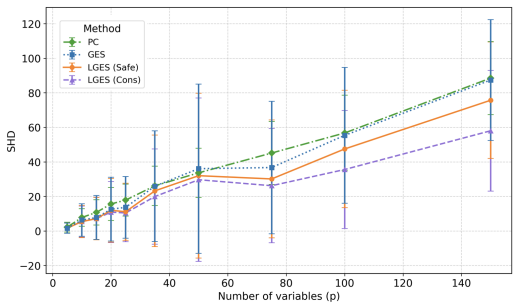
Figure D.1.2: Binary classification accuracy of algorithms on observational data from Erdős–Rényi graphs with p variables and $2p$ edges and $n = 10^4$ samples (Experiment D.1.4). LGES is our proposed method. **Higher is better** across all plots. An MEC is said to contain an edge (X, Y) if it contains either $X - Y$ or $X \to Y$ (but not $Y \to X$) (Sec. D.1.4). Error bars denote one standard deviation across 50 random seeds. NoTears is omitted from the first row for clarity.

generally makes fewer misorientations than PC. As a result, LGES with CONSERVATIVEINSERT tends to have comparable or better F1 score than the PC algorithm, except for the somewhat extreme case of $n = 100, p = 500$.

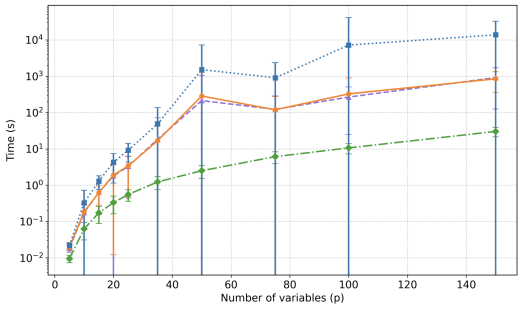
D.1.7 Additional experiments with larger graphs

We scaled up Experiment 5.1 to graphs with $p \in \{175, 250, 500\}$ variables. We ran LGES both with SAFEINSERT and with CONSERVATIVEINSERT and PC for 50 random seeds for $p \in \{175, 250\}$. The results are summarized in Table D.1.2. We were unable to continue running 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. 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 ≈5-6 minutes for $p = 175$ and ≈15-20 minutes for $p = 250$. Recall that for $p = 150$, GES was already approaching a runtime of 10^4 seconds (>2 hours) (Fig. D.1.1b).

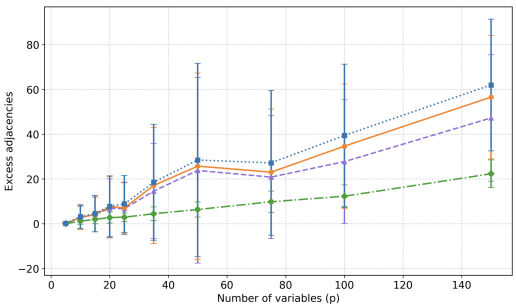
Finally, we also ran LGES for $p = 500$ for 5 random seeds. LGES with SAFEINSERT had an average SHD of 490.75±16.18 and an average runtime of 13657.00±1378.95 seconds (≈3.8 hours). LGES with CONSERVATIVEINSERT had an average SHD of 360.00±22.64 and an average runtime of 12214.55±1162.79 second (≈3.4 hours). Thus, LGES remains feasible even for graphs with 500 variables.



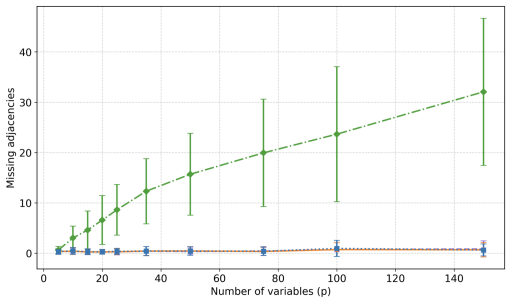
(a) SHD vs number of variables



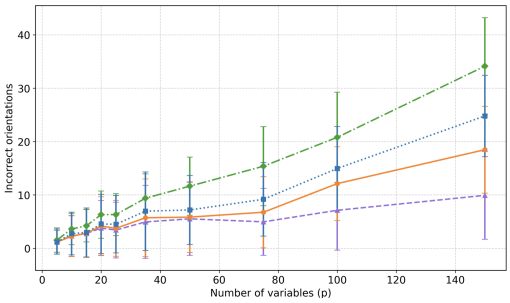
(b) Runtime vs number of variables



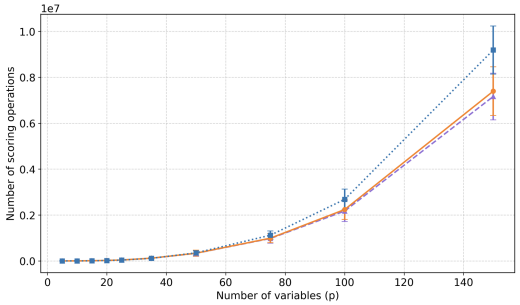
(c) Excess adjacencies vs number of variables.



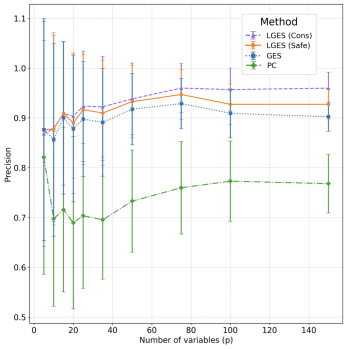
(d) Missing adjacencies vs number of variables.



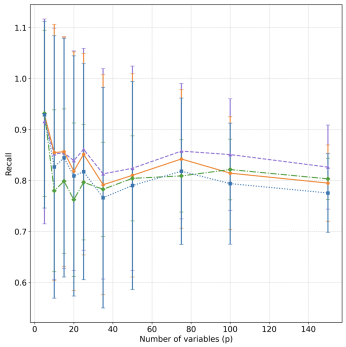
(e) Incorrect orientations vs number of variables.



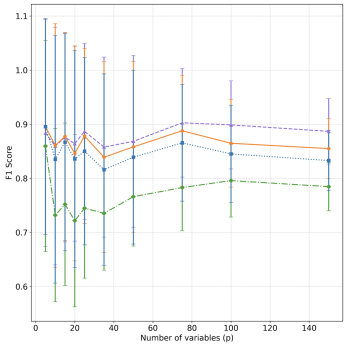
(f) Scoring operations vs number of variables.



(g) Precision vs number of variables.



(h) Recall vs number of variables.



(i) F1 score vs number of variables.

Figure D.1.3: Performance of algorithms on observational data from Erdős–Rényi graphs with p variables and $3p$ edges and $n = 10^4$ samples (Experiment D.1.5). LGES is our proposed method. **Lower is better** (more accurate / faster) across all metrics **except precision, recall, and F1 score**, for which higher is better. The time axis uses a log scale. SHD denotes the structural Hamming distance between the true and estimated CPDAGs. For the binary classification metrics (last row), an MEC is said to contain an edge (X, Y) if it contains either $X - Y$ or $X \to Y$ (but not $Y \to X$) (Sec. D.1.4). Error bars denote one standard deviation across 50 random seeds.

Metric	Method	$\mathbf{n = 500}$				$\mathbf{n = 1000}$			
		$p = 10$	$p = 25$	$p = 50$	$p = 100$	$p = 10$	$p = 25$	$p = 50$	$p = 100$
SHD	PC	8.58 ± 5.00	36.35 ± 7.22	20.47 ± 7.27	46.96 ± 10.84	2.84 ± 3.24	8.30 ± 5.15	17.50 ± 7.04	40.61 ± 11.21
	GES	9.14 ± 5.91	65.68 ± 22.98	32.37 ± 8.01	100.93 ± 13.71	2.46 ± 4.85	10.02 ± 10.33	25.12 ± 7.58	76.68 ± 13.32
	LGES (Safe)	9.06 ± 6.38	66.09 ± 23.70	28.19 ± 8.12	97.28 ± 14.98	2.44 ± 4.86	10.18 ± 10.67	21.34 ± 7.91	71.45 ± 13.16
	LGES (Cons)	9.38 ± 6.35	65.55 ± 23.81	22.26 ± 14.47	75.17 ± 14.86	2.02 ± 4.79	6.98 ± 9.40	13.78 ± 7.97	53.41 ± 13.45
False Positives	PC	2.80 ± 1.60	16.35 ± 3.86	4.88 ± 2.29	17.20 ± 6.21	0.44 ± 0.75	1.24 ± 1.11	4.54 ± 2.34	16.55 ± 6.49
	GES	4.88 ± 3.64	48.68 ± 20.71	21.19 ± 5.51	76.24 ± 10.55	1.08 ± 2.18	6.12 ± 6.53	15.76 ± 5.04	55.64 ± 10.44
	LGES (Safe)	4.90 ± 3.77	48.96 ± 21.17	19.58 ± 5.80	75.02 ± 11.23	1.04 ± 2.18	6.20 ± 6.66	14.28 ± 5.23	53.82 ± 9.88
	LGES (Cons)	5.10 ± 3.75	48.57 ± 21.13	16.88 ± 10.78	61.13 ± 10.78	0.80 ± 2.11	4.58 ± 6.24	10.68 ± 5.27	43.68 ± 8.94
False Negatives	PC	3.42 ± 2.73	14.88 ± 4.18	7.02 ± 4.11	12.02 ± 5.78	1.20 ± 1.52	3.08 ± 2.62	5.10 ± 3.71	8.25 ± 5.67
	GES	1.46 ± 1.25	9.02 ± 2.30	0.33 ± 0.60	0.83 ± 1.49	0.26 ± 0.66	0.36 ± 0.79	0.32 ± 0.71	0.16 ± 0.47
	LGES (Safe)	1.36 ± 1.09	9.02 ± 2.40	0.30 ± 0.55	0.72 ± 1.39	0.26 ± 0.66	0.44 ± 0.92	0.26 ± 0.63	0.27 ± 0.65
	LGES (Cons)	1.44 ± 1.06	9.06 ± 2.26	0.53 ± 1.09	1.00 ± 1.74	0.28 ± 0.66	0.32 ± 0.79	0.22 ± 0.58	0.43 ± 1.29
Wrong Orientations	PC	2.36 ± 1.84	5.12 ± 2.09	8.56 ± 4.10	17.74 ± 6.80	1.20 ± 1.81	3.98 ± 2.77	7.86 ± 4.43	15.82 ± 5.92
	GES	2.80 ± 2.44	7.98 ± 2.67	10.86 ± 4.33	23.87 ± 5.26	1.12 ± 2.47	3.54 ± 3.74	9.04 ± 3.79	20.89 ± 5.55
	LGES (Safe)	2.80 ± 2.80	8.11 ± 2.90	8.30 ± 3.97	21.54 ± 5.51	1.14 ± 2.47	3.54 ± 3.87	6.80 ± 3.75	17.36 ± 5.01
	LGES (Cons)	2.84 ± 2.80	7.91 ± 3.25	4.84 ± 4.06	13.04 ± 4.97	0.94 ± 2.44	2.08 ± 3.05	2.88 ± 3.17	9.30 ± 5.81
F1 Score	PC	0.60 ± 0.22	0.33 ± 0.10	0.81 ± 0.07	0.78 ± 0.06	0.87 ± 0.14	0.85 ± 0.10	0.83 ± 0.06	0.81 ± 0.05
	GES	0.65 ± 0.19	0.29 ± 0.10	0.76 ± 0.06	0.66 ± 0.05	0.90 ± 0.17	0.84 ± 0.14	0.80 ± 0.06	0.72 ± 0.05
	LGES (Safe)	0.66 ± 0.20	0.29 ± 0.10	0.79 ± 0.06	0.67 ± 0.05	0.90 ± 0.17	0.84 ± 0.15	0.83 ± 0.06	0.74 ± 0.04
	LGES (Cons)	0.65 ± 0.19	0.30 ± 0.10	0.83 ± 0.08	0.73 ± 0.05	0.92 ± 0.16	0.89 ± 0.13	0.89 ± 0.06	0.80 ± 0.04

Table D.1.1: Accuracy of algorithms (mean $\pm$ std) on observational data from Erdős–Rényi graphs with p variables and $2p$ edges and small datasets with $n \in \{500, 1000\}$ samples (Experiment D.1.6). LGES is our proposed method. **Lower is better** (more accurate / faster) **across all metrics except the F1 score**, for which higher is better. SHD denotes the structural Hamming distance between the true and estimated CPDAGs. For the F1 score, an MEC is said to contain an edge (X, Y) if it contains either $X - Y$ or $X \to Y$ (but not $Y \to X$) (Sec. D.1.4). See Fig. D.1.4 for heatmaps representing the above data. We draw 50 random seeds per n, p.

Metric	Method	$p = 175$	$p = 250$
SHD	LGES (Cons)	$\mathbf{42.04 \pm 11.51}$	$\mathbf{83.06 \pm 14.42}$
	LGES (Safe)	70.62 ± 11.68	135.24 ± 14.31
	PC	91.34 ± 17.17	166.20 ± 24.34
F1 Score	LGES (Cons)	$\mathbf{0.90 \pm 0.03}$	$\mathbf{0.86 \pm 0.02}$
	LGES (Safe)	0.84 ± 0.03	0.79 ± 0.02
	PC	0.78 ± 0.04	0.73 ± 0.04
Runtime	LGES (Cons)	315.83 ± 73.23	989.37 ± 205.53
	LGES (Safe)	337.45 ± 78.43	1058.11 ± 221.32
	PC	$\mathbf{29.20 \pm 9.50}$	$\mathbf{85.67 \pm 23.52}$

Table D.1.2: Performance of algorithms (mean $\pm$ std) on observational data from large Erdős–Rényi graphs with p variables and $2p$ edges and $n = 10^4$ samples (Experiment D.1.7). LGES is our proposed method. **Lower is better** (more accurate / faster) **across all metrics except the F1 score**, for which higher is better. SHD denotes the structural Hamming distance between the true and estimated CPDAGs. For the F1 score, an MEC is said to contain an edge (X, Y) if it contains either $X - Y$ or $X \to Y$ (but not $Y \to X$) (Sec. D.1.4). We draw 50 random seeds per p. GES and NoTears are omitted as they required too much compute to scale to these graph sizes.

D.2 Repairing misspecified causal models

D.2.1 Detailed metrics for Experiment 5.2

In Fig. D.2.1, we present more detailed plots of runtime and SHD for the setting in Sec. 5.2. We use GES-0 and LGES-0 to denote the corresponding algorithms run without any prior assumptions. LGES with CONSERVATIVEINSERT outperforms all other algorithms across all levels of correctness of the prior assumptions, with the exception of LGES-0 with CONSERVATIVEINSERT when the background knowledge is more misspecified ($\texttt{fc} \leq 0.5$). Even when all assumptions are incorrect, LGES with CONSERVATIVEINSERT achieves better runtime and marginally lower SHD than GES-0.

For both SAFEINSERT and CONSERVATIVEINSERT, we find that the guided search strategy introduced in Sec. 3.3 is more robust to misspecified assumptions ($\texttt{fc} \leq 0.5$) than simply initialising the search using the prior assumptions. This confirms our hypothesis in Sec. 3.3 that incorrect edges included by default harm the quality of the search.

The runtime of LGES both with CONSERVATIVEINSERT and with SAFEINSERT improves noticably when given correct prior assumptions ($\texttt{fc} \geq 0.75$) relative to LGES-0, though the accuracy improves only marginally. For $\texttt{fc} \geq 0.75$, the initialization strategy benefits LGES more than the guided search strategy. This suggests a tradeoff between robustness to incorrect prior assumptions and advantage gained from these assumptions when correct.

D.2.2 Additional experiments varying the number of prior edge assumptions

The results in Fig. 4 are for experiments conducted with $m' = m/2$ edges in the set of prior assumptions for a ground truth graph containing m edges. We conduct additional experiments with $m' = 3m/4$, increasing the number of edges included in the prior assumptions. The results are shown in Fig. D.2.1, and follow a similar trend as in Experiment 5.2 discussed previously.

D.3 Learning from interventional data

D.3.1 Detailed metrics for Experiment 5.3

In Fig. D.3.1, we present more detailed plots of runtime and SHD for the setting in Sec. 5.3. We refer to our approach, LGES followed by $\mathcal{I}$-ORIENT, as LGIES. We find that LGIES both with SAFEINSERT and with CONSERVATIVEINSERT is up to an order of magnitude faster than GIES. The accuracy of LGIES with CONSERVATIVEINSERT is comparable to that of GIES. However, LGIES with SAFEINSERT has larger SHD from the ground truth than GIES and LGES with CONSERVATIVEINSERT. This is possibly 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, GIES uses $10^4 + k \cdot 10^3$ samples in the MEC learning phase given k interventional, since it also uses interventional data during this phase and we generate 10^3 interventional samples per target. Moreover, since we choose $k = p/10$ (where p is the number of variables), GIES is making use of much more data than LGES for learning the MEC. Although GIES is known not to be asymptotically sound [52], this suggests that LGIES may benefit from a way of incorporating interventional data in the MEC learning phase.

D.4 Real-world protein signaling data

Sachs dataset. We compare GES and LGES on a real-world protein signaling dataset [42]. 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 [42, Fig. 3]), containing 11 variables and 17 edges. The dataset is continuous but violates the linear-Gaussian assumption.[4] 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.[5]

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 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 [48] and Sparsest Permutations [38] use statistical tests, usually for conditional independence, to learn a

[4] `https://www.bnlearn.com/research/sachs05/`

[5] `https://www.bnlearn.com/book-crc-2ed/`

Markov equivalence class from data. Score-based approaches such as Greedy Equivalence Search [7, 29]instead attempt to maximize a score (for e.g., the Bayesian Information Criterion or BIC [45]) that reflects the fit between graph and data. There is no general claim about which of these methods is superior; we refer readers to [50] for an extensive empirical analysis, who found, for instance, that GES outperforms PC in accuracy across various sample sizes [50, 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 learning, interventional learning, and model repairing. 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 assumptions in the search to produce better quality estimates of the true Markov equivalence class and do so faster—the task of model repairing. 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 (SAFEINSERT and CONSERVATIVEINSERT) terminate in less than 4 hours on graphs with 500 variables and have substantially better accuracy than other baselines (including PC and NoTears) on these large graphs (Experiments D.1.3, D.1.7). LGES also outperforms these baselines in settings with dense graphs (Experiment D.1.5).

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.1.6).

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 model repairing different from initializing a causal discovery task to the hypothesized model?

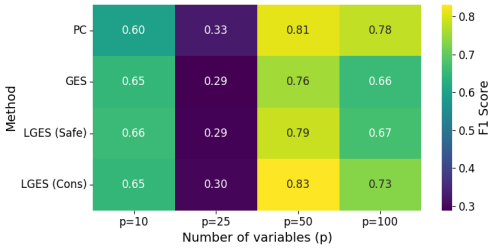
Answer. Model repairing is the more general problem of using possibly misspecified prior assumptions in the process of causal discovery to aid the search. Initializing the search to a tentative model is one way to achieve this. However, initialization is not robust to misspecification in the assumptions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

Q8. What is the difference between Greedy Interventional Equivalence Search (GIES) [19] and LGES?

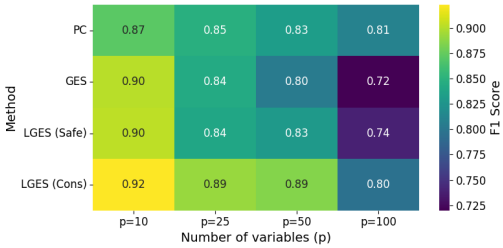
Answer. GIES and LGES are both score-based algorithms for learning from a combination of observational and interventional data. However, LGES has a few primary advantages over GIES. First, LGES with SAFEINSERT is guaranteed to recover the true interventional MEC (Cor. 1, Thm. 2) in the sample limit whereas GIES is not [52]. 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. An interesting future direction is to incorporate interventional data in all phases of LGES while maintaining its asymptotic guarantees, since this may lead to improved accuracy.

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

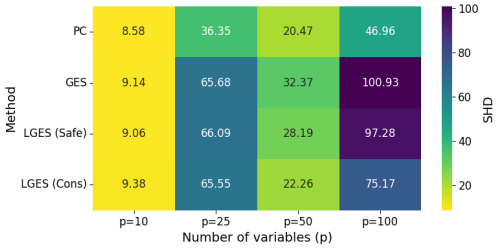
Answer. Several components of our approach are modular and can be combined with other causal discovery algorithms. For one, algorithms in the GES family like FGES [37] and GIES [19] can be easily modified to use the SAFEINSERT or CONSERVATIVEINSERT strategies that we introduce; a simple extension would 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.



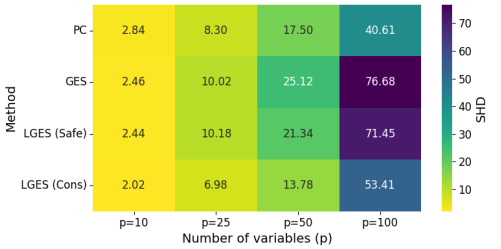
(a) F1 score ($n = 500$)



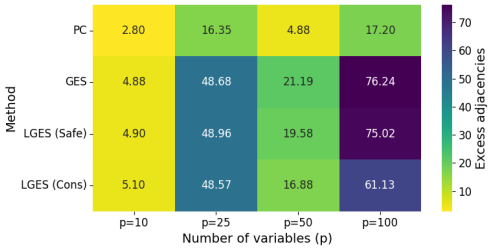
(b) F1 score ($n = 1000$)



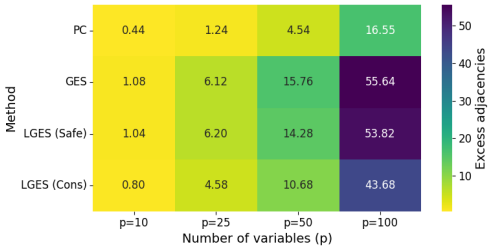
(c) SHD ($n = 500$)



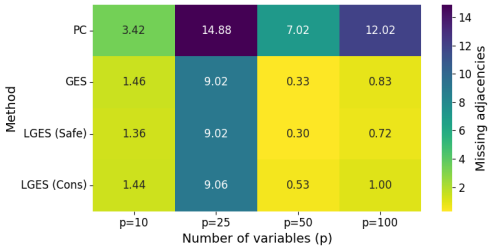
(d) SHD ($n = 1000$)



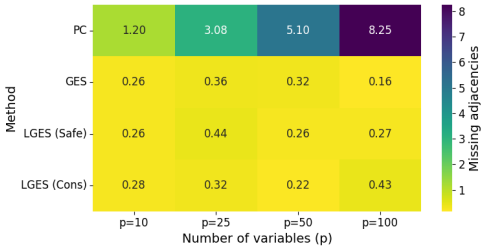
(e) Excess adjacencies ($n = 500$)



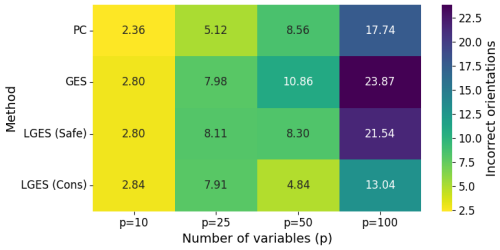
(f) Excess adjacencies ($n = 1000$)



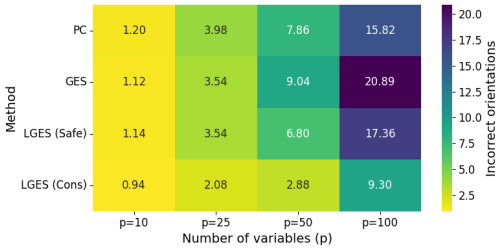
(g) Missing adjacencies ($n = 500$)



(h) Missing adjacencies ($n = 1000$)

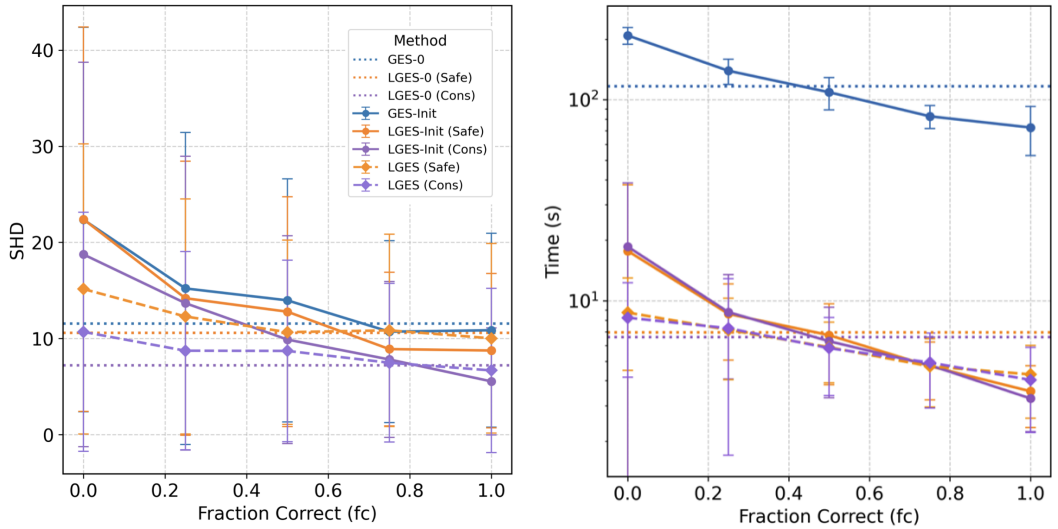


(i) Incorrect orientations ($n = 500$)



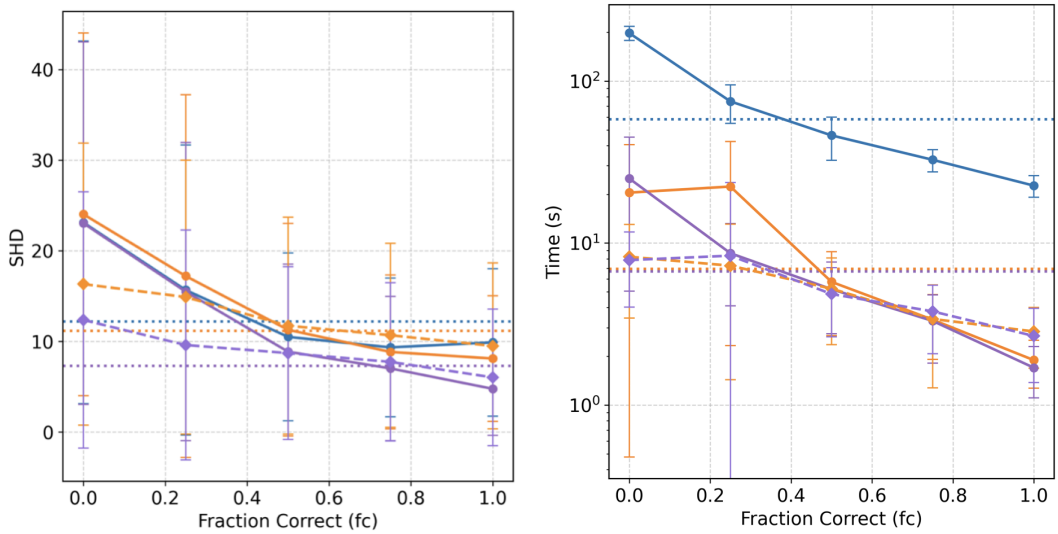
(j) Incorrect orientations ($n = 1000$)

Figure D.1.4: Accuracy of algorithms on observational data from Erdős–Rényi graphs with p variables and $2p$ edges and small datasets with $n \in \{500, 1000\}$ samples (Experiment D.1.6). LGES is our proposed method. **Lighter is better** across all heatmaps. SHD denotes the structural Hamming distance between the true and estimated CPDAGs. For the F1 score, an MEC is defined as containing an edge (X, Y) if it contains either $X - Y$ or $X \to Y$ (but not $Y \to X$) (Sec. D.1.4). Numbers denote averages across 50 random seeds. More details in Table D.1.1.



(a) SHD, varying correctness of $m/2$ prior edges

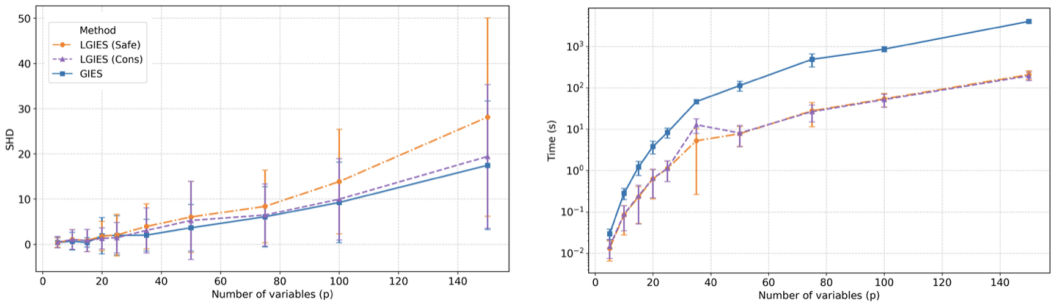
(b) Time, varying correctness of $m/2$ prior edges



(c) SHD, varying correctness of $3m/4$ prior edges

(d) Time, varying correctness of $3m/4$ prior edges

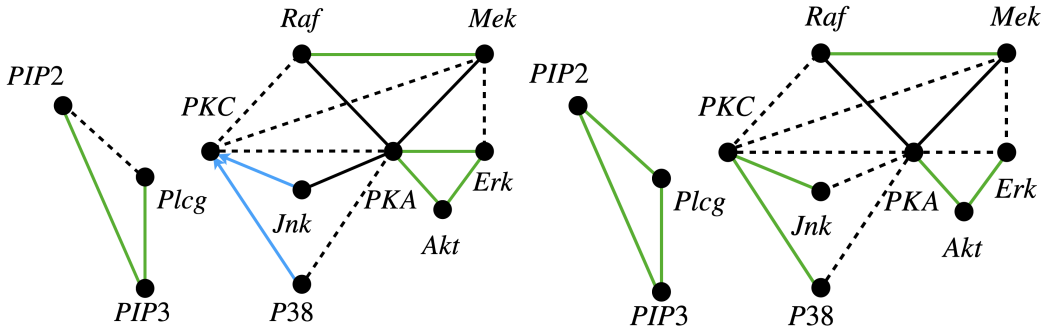
Figure D.2.1: Performance of algorithms given prior assumptions and observational data from Erdős–Rényi graphs with 50 variables and 100 edges in expectation and $n = 10^3$ samples (Experiment 5.2, Sec. D.2). We vary the correctness of the prior assumptions on the x-axis. LGES is our proposed method. In the upper panel, we include $m/2$ prior edge assumptions, where m is the number of edges in the ground truth DAG; in the lower panel, we increase this to $3m/4$. **Lower is better** (more accurate / faster) across all plots. The time axis uses a log scale. Error bars denote one standard deviation across 50 random seeds. Error bars for SHD are clipped to ± 20.



(a) SHD vs number of variables

(b) Runtime vs number of variables

Figure D.3.1: Performance of LGIES and GIES on interventional data from Erdős–Rényi graphs with p variables and $2p$ edges (Experiment 5.3). We generate $n = 10^4$ observational samples and $n = 10^3$ samples per intervention. LGES is our proposed method. **Lower is better** (more accurate / faster) across all plots. The time axis uses a log scale. SHD denotes the structural Hamming distance between the true and estimated interventional CPDAGs. Error bars denote one standard deviation across 50 random seeds.



(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 [42]. 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.