Consistency Guarantees for Greedy Permutation-Based Causal Inference Algorithms

Citation: Solus, L, Wang, Y and Uhler, C. 2021. "Consistency Guarantees for Greedy Permutation-Based Causal Inference Algorithms." Biometrika, 108 (4).

As Published: 10.1093/BIOMET/ASAA104

Publisher: Oxford University Press (OUP)

Persistent URL: https://hdl.handle.net/1721.1/143915

CONSISTENCY GUARANTEES FOR GREEDY PERMUTATION-BASED CAUSAL INFERENCE ALGORITHMS

LIAM SOLUS, YUHAO WANG, AND CAROLINE UHLER

Abstract. Directed acyclic graphical models, or DAG models, are widely used to represent complex causal systems. Since the basic task of learning such a model from data is NP-hard, a standard approach is greedy search over the space of directed acyclic graphs or Markov equivalence classes of directed acyclic graphs. As the space of directed acyclic graphs on p nodes and the associated space of Markov equivalence classes are both much larger than the space of permutations, it is desirable to consider permutation-based greedy searches. Here, we provide the first consistency guarantees, both uniform and high-dimensional, of a greedy permutation-based search. This search corresponds to a simplex-like algorithm operating over the edge-graph of a subpolytope of the permutohedron, called a DAG associahedron. Every vertex in this polytope is associated with a directed acyclic graph, and hence with a collection of permutations that are consistent with the directed acyclic graph ordering. A walk is performed on the edges of the polytope maximizing the sparsity of the associated directed acyclic graphs. We show via simulated and real data that this permutation search is competitive with current approaches.

1. Introduction

Bayesian networks, or DAG models, are widely used to model complex causal systems arising, for example, in computational biology, epidemiology, or sociology [8, 24, 27, 29]. Given a directed acyclic graph, i.e., a DAG, $\mathcal{G} := ([p], A)$ with node set $[p] := \{1, 2, \dots, p\}$ and arrow set A, a DAG model associates to each node $i \in [p]$ of $\mathcal{G}$ a random variable X_i. By the Markov property, $\mathcal{G}$ encodes a set of conditional independence relations $X_i \perp\!\!\!\perp X_{\mathrm{Nd}(i)\setminus \mathrm{Pa}(i)} \,|\, X_{\mathrm{Pa}(i)}$, where $\mathrm{Nd}(i)$ and $\mathrm{Pa}(i)$, respectively, denote the nondesendants and parents of the node i in $\mathcal{G}$. A joint distribution $\mathbb{P}$ on $X_1, \dots, X_p$ is said to satisfy the Markov assumption, or be Markov, with respect to $\mathcal{G}$ if it entails these conditional independence relations. This paper is concerned with the structural learning problem: Suppose we sample from a distribution $\mathbb{P}$ that is Markov with respect to a DAG $\mathcal{G}^*$. If we infer from this data a collection of conditional independence relations $\mathcal{C}$, can we recover the unknown DAG $\mathcal{G}^*$ using $\mathcal{C}$?

In general, this problem is not well-defined since multiple DAGs can encode the same set of conditional independence relations. Any two such DAGs are termed Markov equivalent, and they are said to belong to the same Markov equivalence class. Thus, our goal becomes to identify the Markov equivalence class $\mathcal{M}(\mathcal{G}^*)$ of $\mathcal{G}^*$. The Markov assumption alone is not sufficient to guarantee identifiability, and so

Date: June 9, 2021.

Key words and phrases. causal inference; Bayesian network; DAG; Gaussian graphical model; greedy search; pointwise and high-dimensional consistency; faithfulness; generalized permutohedron; DAG associahedron.

additional identifiability assumptions have been studied, the most prominent being faithfulness [29]. Unfortunately, this assumption has been shown to be restrictive in practice [36]. Hence, it is desirable to develop structure learning algorithms that are consistent under strictly weaker assumptions than faithfulness.

Since the space of all Markov equivalence classes of DAGs on p nodes grows super-exponentially in p [13], one way to perform structure learning is to use greedy approaches. For example, the greedy equivalence search [4, 20] greedily maximizes a score, such as the Bayesian information criterion, over the space of all Markov equivalence classes on p nodes. An alternative approach is to consider algorithms with a reduced search space, such as the space of all $p!$ linear extensions of DAGs; i.e., the permutations of $[p]$. Greedy permutation-based algorithms combine both of these heuristic approaches for DAG model learning. In recent decades, a variety of greedy permutation-based algorithms have been proposed and analyzed. See, for instance, [2, 6, 18, 28, 31]. However, all of these algorithms rely on heuristics for sparse DAG recovery and are therefore not provably consistent, even under the faithfulness assumption.

On the other hand, a non-greedy permutation-based algorithm known as the sparsest permutation algorithm was introduced in [26], and it was shown to be consistent under strictly weaker assumptions than faithfulness. Unfortunately, the sparsest permutation algorithm must generate and score a DAG for each permutation of $\{1, \dots, p\}$, and hence it runs in $\mathcal{O}(p!)$ time no matter the true underlying DAG model. Here, we provide the first consistency guarantees of a greedy permutation-based algorithm for DAG model structure learning. This algorithm is a greedy version of the sparsest permutation algorithm. Unlike its non-greedy predecessor, the proposed algorithm scales to DAG structure discovery with hundreds of variables, since only in the worst case does it have to search over all $p!$ permutations; e.g., when the true model is the complete graph. Such worst-case behavior is to be expected since in general the problem of learning a DAG model is NP-hard [5]. In addition, we show that our greedy sparsest permutation algorithm is consistent under weaker assumptions than standardly assumed, which translates into competitive performance in terms of structure recovery when compared to currently popular algorithms on both simulated and real data.

2. Background

We refer the reader to Section A in the Supplementary Material for graph theory definitions and notation. A fundamental result about DAG models is that the complete set of conditional independence relations implied by the Markov assumption for $\mathcal{G}$ is given by the d-separation relations in $\mathcal{G}$ [19, Section 3.2.2]; i.e., a distribution $\mathbb{P}$ is Markov with respect to $\mathcal{G}$ if and only if $X_A \perp\!\!\!\perp X_B \,|\, X_C$ in $\mathbb{P}$ whenever A and B are d-separated in $\mathcal{G}$ given C. The faithfulness assumption asserts that all conditional independence relations entailed by $\mathbb{P}$ are given by d-separations in $\mathcal{G}$ [29].

Assumption 1 (Faithfulness Assumption)**.** A distribution $\mathbb{P}$ satisfies the faithfulness assumption with respect to a DAG $\mathcal{G} = ([p], A)$ if for any pair of nodes $i, j \in [p]$ and any $S \subset [p]\backslash\{i, j\}$ we have that $i \perp\!\!\!\perp j \,|\, S$ if and only if i is d-separated from j given S in $\mathcal{G}$.

All DAG model learning algorithms assume the Markov assumption, i.e. the forward direction of the faithfulness assumption, and many of the classical algorithms

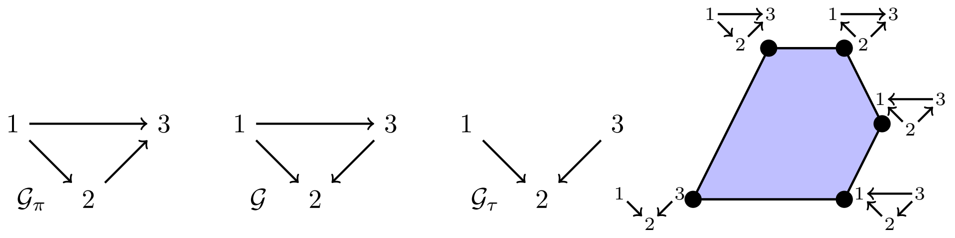


FIGURE 1. For $\mathcal{C} = \{1 \perp\!\!\!\perp 3\}$, we see the polytope $\mathcal{A}_3(\mathcal{C})$, the graphs $\mathcal{G}_\pi$ and $\mathcal{G}_\tau$ for the $\pi = 123$ and $\tau = 132$, and a graph $\mathcal{G}$ Markov equivalent to $\mathcal{G}_\pi$ that is not a minimal independence map of $\mathcal{C}$. π and τ are related by transposing 2 and 3 in π and the arrow $2 \to 3$ in $\mathcal{G}_\pi$ is covered.

also assume the converse. Unfortunately, the faithfulness assumption has been shown to be restrictive in practice [36], and a number of relaxations of this assumption have been suggested [25]. For example, restricted faithfulness is the weakest known sufficient condition for consistency of the popular PC-algorithm [29].

Assumption 2 (Restricted faithfulness assumption)**.** A distribution $\mathbb{P}$ satisfies the restricted faithfulness assumption with respect to a DAG $\mathcal{G} = ([p], A)$ if it satisfies the two conditions:

(1) (Adjacency Faithfulness) For all arrows $i \to j \in A$ we have that $X_i \not\perp\!\!\!\perp X_j \mid X_S$ for all subsets $S \subset [p]\backslash\{i, j\}$.
(2) (Orientation Faithfulness) For all unshielded triples (i, j, k) and all subsets $S \subset [p]\backslash\{i, k\}$ such that i is d-connected to k given S, we have that $X_i \not\perp\!\!\!\perp X_k \mid X_S$.

By sacrificing computation time, some algorithms remain consistent under assumptions that further relax restricted faithfulness, an example being the sparsest permutation algorithm [26]: Let S_p denote the space of all permutations of length p. Given a set of conditional independence relations $\mathcal{C}$ on $[p]$, every permutation $\pi \in S_p$ is associated to a DAG $\mathcal{G}_\pi$ as follows:

$$\pi_i \to \pi_j \in A(\mathcal{G}_\pi) \quad \Leftrightarrow \quad i < j \text{ and } \pi_i \not\perp\!\!\!\perp \pi_j \mid \{\pi_1, \ldots, \pi_j\}\backslash\{\pi_i, \pi_j\}.$$

Examples of the DAGs $\mathcal{G}_\pi$ appear in Figure 1. A DAG $\mathcal{G}_\pi$ is known as a minimal independence map with respect to $\mathcal{C}$, since any DAG $\mathcal{G}_\pi$ satisfies the minimality assumption with respect to $\mathcal{C}$, i.e., any conditional independence relation encoded by a d-separation in $\mathcal{G}_\pi$ is in $\mathcal{C}$ and any proper sub-DAG of $\mathcal{G}_\pi$ encodes a conditional independence relation that is not in $\mathcal{C}$ [23]. The sparsest permutation algorithm searches over all DAGs $\mathcal{G}_\pi$ for $\pi \in S_p$ and returns a DAG that maximizes the score

$$\text{score}(\mathcal{C}; \mathcal{G}) := \begin{cases} -|\mathcal{G}| & \text{if } \mathcal{G} \text{ is Markov with respect to } \mathcal{C}, \\ -\infty & \text{otherwise,} \end{cases}$$

where $|\mathcal{G}|$ denotes the number of arrows in $\mathcal{G}$. In [26], it is shown that the sparsest permutation algorithm is consistent under the sparsest Markov representation assumption, which is strictly weaker than restricted faithfulness.

Assumption 3 (Sparsest Markov representation assumption)**.** A probability distribution $\mathbb{P}$ satisfies the sparsest Markov representation assumption with respect to a DAG $\mathcal{G}$ if it is Markov with respect to $\mathcal{G}$ and $|\mathcal{G}| < |\mathcal{H}|$ for every DAG $\mathcal{H}$ to which $\mathbb{P}$ is Markov and which satisfies $\mathcal{H} \notin \mathcal{M}(\mathcal{G})$.

The downside to the sparsest permutation algorithm is that it requires a search over all $p!$ permutations of the node set $[p]$. A typical approach to accommodate a large search space is to pass to a greedy variant of the algorithm. To this end, we analyze greedy variants of the sparsest permutation algorithm in the coming section.

3. Greedy sparsest permutation algorithm

The sparsest permutation algorithm has a natural interpretation in the setting of discrete geometry. The permutohedron on p elements, denoted $\mathcal{A}_p$, is the convex hull in $\mathbb{R}^p$ of all vectors obtained by permuting the coordinates of $(1, 2, 3, \ldots, p)^T$. The sparsest permutation algorithm can be thought of as searching over the vertices of $\mathcal{A}_p$, since it considers the DAGs $\mathcal{G}_\pi$ for each $\pi \in S_p$. Hence, a natural first step to reduce the size of the search space is to contract all vertices of $\mathcal{A}_p$ that correspond to the same DAG $\mathcal{G}_\pi$. This can be done via the following construction first presented in [21].

Two vertices of the permutohedron $\mathcal{A}_p$ are connected by an edge if and only if the permutations indexing the vertices differ by an adjacent transposition. We associate a conditional independence relation to adjacent transpositions, and hence to each edge of $\mathcal{A}_p$; namely $\pi_i \perp\!\!\!\perp \pi_{i+1} | \{\pi_1, \ldots, \pi_{i-1}\}$ to the edge between

$$(\pi_1, \ldots, \pi_i, \pi_{i+1}, \ldots, \pi_p)^T \quad \text{and} \quad (\pi_1, \ldots, \pi_{i+1}, \pi_i, \ldots, \pi_p)^T.$$

In [21, Section 4], it is shown that given a set of conditional independence relations $\mathcal{C}$ from a joint distribution $\mathbb{P}$ on $[p]$, then contracting all edges in $\mathcal{A}_p$ corresponding to conditional independence relations in $\mathcal{C}$ results in a convex polytope, which we denote by $\mathcal{A}_p(\mathcal{C})$. Note that $\mathcal{A}_p(\emptyset) = \mathcal{A}_p$. Furthermore, if the conditional independence relations in $\mathcal{C}$ form a graphoid, i.e., they satisfy the semigraphoid properties and the intersection property:

(1) if $i \perp\!\!\!\perp j | S$ then $j \perp\!\!\!\perp i | S$,
(2) if $i \perp\!\!\!\perp j | S$ and $i \perp\!\!\!\perp k | \{j\} \cup S$, then $i \perp\!\!\!\perp k | S$ and $i \perp\!\!\!\perp j | \{k\} \cup S$,
(3) if $i \perp\!\!\!\perp j | \{k\} \cup S$ and $i \perp\!\!\!\perp k | \{j\} \cup S$, then $i \perp\!\!\!\perp j | S$ and $i \perp\!\!\!\perp k | S$,

then it was shown in [21, Theorem 7.1] that contracting edges in $\mathcal{A}_p$ that correspond to conditional independence relations in $\mathcal{C}$ is the same as identifying vertices of $\mathcal{A}_p$ that correspond to the same DAG. The semigraphoid properties hold for any distribution. On the other hand, the intersection property holds, for example, for strictly positive distributions. Another example of a graphoid is the set of conditional independence relations $\mathcal{C}$ corresponding to all d-separations in a DAG. In that case $\mathcal{A}_p(\mathcal{C})$ is also called a DAG associahedron [21]. The edge graph of the polytope $\mathcal{A}_p(\mathcal{C})$, where each vertex corresponds to a different DAG, represents a natural search space for a greedy version of the sparsest permutation algorithm.

Through a closer examination of the polytope $\mathcal{A}_p(\mathcal{C})$, we arrive at two greedy versions of the sparsest permutation algorithm: one based on the geometry of $\mathcal{A}_p(\mathcal{C})$ by walking along edges of the polytope and another based on the combinatorial description of the vertices by walking from DAG to DAG. These two greedy versions of the sparsest permutation algorithm are given in Algorithms 1 and 2.

Algorithm 1: The Edge Sparsest Permutation Algorithm

```
Input  : A set of conditional independence relations $\mathcal{C}$ on node set $[p]$ and a
         starting permutation $\pi \in S_p$.
Output: A minimal independence map $\mathcal{G}$.
Compute the polytope $\mathcal{A}_p(\mathcal{C})$ and set $\mathcal{G} := \mathcal{G}_\pi$.
Using a depth-first search approach with root $\mathcal{G}$ along the edges of $\mathcal{A}_p(\mathcal{C})$,
  search for a minimal independence map $\mathcal{G}_\tau$ with $|\mathcal{G}| > |\mathcal{G}_\tau|$. If no such $\mathcal{G}_\tau$
  exists, return $\mathcal{G}$; else set $\mathcal{G} := \mathcal{G}_\tau$ and repeat this step.
```

Both algorithms take as input a set of conditional independence relations $\mathcal{C}$ and an initial permutation $\pi \in S_p$. Beginning at the vertex $\mathcal{G}_\pi$ of $\mathcal{A}_p(\mathcal{C})$, Algorithm 1 walks along an edge of $\mathcal{A}_p(\mathcal{C})$ to any vertex whose corresponding DAG has at most as many arrows as $\mathcal{G}_\pi$. Once it can no longer discover a sparser DAG, the algorithm returns the last DAG it visited, from which we deduce the corresponding Markov equivalence class. Since this algorithm is based on walking along edges of $\mathcal{A}_p(\mathcal{C})$, we call this greedy version the edge sparsest permutation algorithm. The corresponding identifiability assumption can be stated as follows:

Assumption 4 (Edge assumption)**.** A distribution $\mathbb{P}$ satisfies the edge assumption with respect to a DAG $\mathcal{G}$ if it is Markov with respect to $\mathcal{G}$ and if Algorithm 1 returns only DAGs in $\mathcal{M}(\mathcal{G})$.

Algorithm 1 requires computing the polytope $\mathcal{A}_p(\mathcal{C})$. This is inefficient, since an edge walk in a polytope only requires knowing the neighbors of a vertex and not the full polytope. In the following, we overcome this inefficiency by providing a graphical characterization of neighboring DAGs.

We say that an arrow $i \to j$ in a DAG $\mathcal{G}$ is covered if $\mathrm{Pa}(i) = \mathrm{Pa}(j) \setminus \{i\}$ and it is trivially covered if $\mathrm{Pa}(i) = \mathrm{Pa}(j) \setminus \{i\} = \emptyset$. For example, the arrows $1 \to 2$ and $2 \to 3$ in the DAG $\mathcal{G}_\pi$ in Figure 1 are both covered, but only the arrow $1 \to 2$ is trivially covered. In addition, we call a sequence of minimal independence maps $(\mathcal{G}_{\pi^1}, \mathcal{G}_{\pi^2}, \dots, \mathcal{G}_{\pi^N})$ a weakly decreasing sequence if $|\mathcal{G}_{\pi^i}| \geq |\mathcal{G}_{\pi^{i+1}}|$ for all $i \in [N-1]$. If $\mathcal{G}_{\pi^{i+1}}$ is produced from $\mathcal{G}_{\pi^i}$ by reversing a covered arrow in $\mathcal{G}_{\pi^i}$, then we refer to this sequence as a weakly decreasing sequence determined by covered arrow reversals. For instance, given the DAGs $\mathcal{G}_\pi$ and $\mathcal{G}_\tau$ from Figure 1, $(\mathcal{G}_\pi, \mathcal{G}_\tau)$ is a weakly decreasing sequence determined by covered arrow reversals. Let $\mathcal{G}_\pi$ and $\mathcal{G}_\tau$ denote two adjacent vertices in a DAG associahedron $\mathcal{A}_p(\mathcal{C})$. Let $\bar{\mathcal{G}}$ denote the skeleton of $\mathcal{G}$; i.e., the undirected graph obtained by undirecting all arrows in $\mathcal{G}$. Then, as noted in [21, Theorem 8.3], $\mathcal{G}_\pi$ and $\mathcal{G}_\tau$ differ by a covered arrow reversal if and only if $\overline{\mathcal{G}_\pi} \subseteq \overline{\mathcal{G}_\tau}$ or $\overline{\mathcal{G}_\tau} \subseteq \overline{\mathcal{G}_\pi}$. In some instances, this fact gives a combinatorial interpretation of all edges of $\mathcal{A}_p(\mathcal{C})$. However, this need not always be true as demonstrated in Example 16 in Section A of the Supplementary Material.

The combinatorial description of some edges of $\mathcal{A}_p(\mathcal{C})$ via covered arrow reversals motivates Algorithm 2, a combinatorial greedy sparsest permutation algorithm. Since this algorithm is based on flipping covered arrows, we call this the triangle sparsest permutation algorithm. Unlike Algorithm 1, this algorithm does not require computing the polytope $\mathcal{A}_p(\mathcal{C})$ and is thus the version run in practice. Similar to Algorithm 1, we specify an identifiability assumption in relation to Algorithm 2.

Algorithm 2: The Triangle Sparsest Permutation Algorithm

```
Input  : A set of conditional independence relations $\mathcal{C}$ on node set $[p]$ and a
         starting permutation $\pi \in S_p$.
Output: A minimal independence map $\mathcal{G}$.

Set $\mathcal{G} := \mathcal{G}_\pi$.
Using a depth-first search approach with root $\mathcal{G}$, search for a minimal
  independence map $\mathcal{G}_\tau$ with $|\mathcal{G}| > |\mathcal{G}_\tau|$ that is connected to $\mathcal{G}$ by a weakly
  decreasing sequence determined by covered arrow reversals. If no such $\mathcal{G}_\tau$
  exists, return $\mathcal{G}$; else set $\mathcal{G} := \mathcal{G}_\tau$ and repeat this step.
```

Assumption 5 (Triangle assumption)**.** A distribution $\mathbb{P}$ satisfies the triangle assumption with respect to a DAG $\mathcal{G}$ if it is Markov with respect to $\mathcal{G}$ and if Algorithm 2 returns only DAGs in $\mathcal{M}(\mathcal{G})$.

In the same way that the sparsest Markov representation assumption is precisely the necessary and sufficient condition under which the sparsest permutation algorithm is consistent, Assumption 4 and Assumption 5, respectively, are defined to be the necessary and sufficient conditions under which Algorithm 1 and Algorithm 2, respectively, are consistent. By associating an identifiability assumption with an algorithm in this way, we can more easily describe which algorithms are consistent for more distributions. It is straightforward to verify that every covered arrow reversal in some minimal independence map $\mathcal{G}_\pi$ with respect to $\mathcal{C}$ corresponds to some edge of the DAG associahedron $\mathcal{A}_p(\mathcal{C})$. Consequently, if a distribution satisfies the triangle assumption then it also satisfies the edge assumption. In Theorem 11 we show that both these assumptions are weaker than the faithfulness assumption, but stronger than the sparsest Markov representation assumption.

4. Consistency Guarantees and Identifiability Implications

4.1. **Consistency of the edge and triangle sparsest permutation algorithms under faithfulness.** In this section, we prove that both Algorithm 1 and Algorithm 2 are pointwise consistent under the faithfulness assumption; i.e., in the oracle-version as $n \to \infty$ the algorithm outputs the true Markov equivalence class. First note that since the triangle assumption implies the edge assumption, it is sufficient to prove pointwise consistency of Algorithm 2. To prove this, we need to show that for given a set of conditional independence relations $\mathcal{C}$ corresponding to d-separations in a DAG $\mathcal{G}^*$, every weakly decreasing sequence determined by covered arrow reversals ultimately leads to a DAG in $\mathcal{M}(\mathcal{G}^*)$. Given two DAGs $\mathcal{G}$ and $\mathcal{H}$, $\mathcal{H}$ is an independence map of $\mathcal{G}$, denoted by $\mathcal{G} \leq \mathcal{H}$, if every conditional independence relation encoded by $\mathcal{H}$ holds in $\mathcal{G}$ (i.e. $\mathrm{CI}(\mathcal{G}) \supseteq \mathrm{CI}(\mathcal{H})$). The following simple result, whose proof is given in the Supplementary Material, reveals the main idea of the proof.

Lemma 6. *A probability distribution $\mathbb{P}$ on the node set $[p]$ is faithful with respect to a DAG $\mathcal{G}$ if and only if $\mathcal{G} \leq \mathcal{G}_\pi$ for all $\pi \in S_p$.*

The goal is to prove that for any pair of DAGs such that $\mathcal{G}_\pi \leq \mathcal{G}_\tau$, there is a weakly decreasing sequence determined by covered arrow reversals such that

$$(\mathcal{G}_\tau = \mathcal{G}_{\pi^0}, \mathcal{G}_{\pi^1}, \mathcal{G}_{\pi^2}, \ldots, \mathcal{G}_{\pi^M} = \mathcal{G}_\pi).$$

Our proof relies heavily on Chickering's consistency proof of greedy equivalence search and, in particular, on his proof of a conjecture known as Meek's conjecture.

Theorem 7. [4, Theorem 4] *Let $\mathcal{G}$ and $\mathcal{H}$ be any pair of DAGs such that $\mathcal{G} \leq \mathcal{H}$. Let r be the number of arrows in $\mathcal{H}$ that have opposite orientation in $\mathcal{G}$, and let m be the number of arrows in $\mathcal{H}$ that do not exist in either orientation in $\mathcal{G}$. There exists a sequence of at most $r + 2m$ arrow reversals and additions in $\mathcal{G}$ with the following properties:*

(1) Each arrow reversal is a covered arrow.
(2) After each reversal and addition, the graph $\mathcal{G}'$ is a DAG and $\mathcal{G}' \leq \mathcal{H}$.
(3) After all reversals and additions $\mathcal{G} = \mathcal{H}$.

In [4], a constructive proof of this result is given via the APPLY-EDGE-OPERATION algorithm. For convenience, we will henceforth refer to this algorithm as the Chickering algorithm. The Chickering algorithm takes in an independence map $\mathcal{G} \leq \mathcal{H}$ and adds an arrow to $\mathcal{G}$ or reverses a covered arrow in $\mathcal{G}$ to produce a new DAG $\mathcal{G}^1$ for which $\mathcal{G} \leq \mathcal{G}^1 \leq \mathcal{H}$. By Theorem 7, repeated applications of this algorithm produces a sequence of graphs

$$\mathcal{G} = \mathcal{G}^0 \leq \mathcal{G}^1 \leq \mathcal{G}^2 \leq \cdots \leq \mathcal{G}^N = \mathcal{H}.$$

We will call any sequence of DAGs produced in this fashion a Chickering sequence from $\mathcal{G}$ to $\mathcal{H}$. A quick examination of the Chickering algorithm reveals that there can be multiple Chickering sequences from $\mathcal{G}$ to $\mathcal{H}$. We are interested in identifying a specific type of Chickering sequence in which the covered arrow reversals and edge additions correspond to steps between minimal independence maps in a weakly decreasing sequence.

Given two DAGs $\mathcal{G}_\pi \leq \mathcal{G}_\tau$, Algorithm 2 proposes that there is a path along the edges of $\mathcal{A}_p(\mathcal{C})$ corresponding to covered arrow reversals taking us from $\mathcal{G}_\tau$ to $\mathcal{G}_\pi$, say $(\mathcal{G}_\tau = \mathcal{G}_{\pi^0}, \mathcal{G}_{\pi^1}, \mathcal{G}_{\pi^2}, \ldots, \mathcal{G}_{\pi^M} = \mathcal{G}_\pi)$, for which $|\mathcal{G}_{\pi^{j-1}}| \geq |\mathcal{G}_{\pi^j}|$ for all $j = 1, \ldots, M$. Recall that we call such a sequence of minimal independence maps satisfying the latter property a weakly decreasing sequence determined by covered arrow reversals. If such a weakly decreasing sequence exists from any $\mathcal{G}_\tau$ to $\mathcal{G}_\pi$, then Algorithm 2 must find it. By definition, such a path is composed of covered arrow reversals and arrow deletions. Since these are precisely the types of moves used in the Chickering algorithm, then we must understand the subtleties of the relationship between independence maps relating the DAGs $\mathcal{G}_\pi$ for a collection of conditional independence relations $\mathcal{C}$ and the skeletal structure of the $\mathcal{G}_\pi$. To this end, we will use the following two definitions: We will denote that two DAGs $\mathcal{G}$ and $\mathcal{H}$ are Markov equivalent by $\mathcal{G} \approx \mathcal{H}$. A minimal independence map $\mathcal{G}_\pi$ with respect to a graphoid $\mathcal{C}$ is called Markov equivalence class-minimal if for all $\mathcal{G} \approx \mathcal{G}_\pi$ and linear extensions τ of $\mathcal{G}$ we have that $\mathcal{G}_\pi \leq \mathcal{G}_\tau$. Notice by [21, Theorem 8.1], it suffices to check only one linear extension τ for each $\mathcal{G}$. The minimal independence map $\mathcal{G}_\pi$ is further called Markov equivalence class-s-minimal if it is class-minimal and $\overline{\mathcal{G}}_\pi \subseteq \overline{\mathcal{G}}_\tau$ for all $\mathcal{G} \approx \mathcal{G}_\pi$ and linear extensions τ of $\mathcal{G}$. We are now ready to state the main proposition that allows us to verify consistency of Algorithm 2 under the faithfulness assumption.

Proposition 8. *Suppose that $\mathcal{C}$ is a graphoid and $\mathcal{G}_\pi$ and $\mathcal{G}_\tau$ are minimal independence maps with respect to $\mathcal{C}$. Then* *(a) if $\mathcal{G}_\pi \approx \mathcal{G}_\tau$ and $\mathcal{G}_\pi$ is class-s-minimal then there exists a weakly decreasing edgewalk from $\mathcal{G}_\pi$ to $\mathcal{G}_\tau$ along $\mathcal{A}_p(\mathcal{C})$. In particular, any Chickering sequence connecting $\mathcal{G}_\tau$ and $\mathcal{G}_\pi$ is a sequence of Markov equivalent minimal independence maps;*

(b) if $\mathcal{G}_\pi \leq \mathcal{G}_\tau$ but $\mathcal{G}_\pi \not\approx \mathcal{G}_\tau$ then there exists a minimal independence map $\mathcal{G}_{\tau'}$ with respect to $\mathcal{C}$ satisfying $\mathcal{G}_{\tau'} \leq \mathcal{G}_\tau$ that is strictly sparser than $\mathcal{G}_\tau$ and is connected to $\mathcal{G}_\tau$ by a weakly decreasing edgewalk along $\mathcal{A}_p(\mathcal{C})$.

The proof of Proposition 8 can be found in the Supplementary Material. We see from Proposition 8 (a) that class-s-minimality is simply a formality required to guarantee that moving between Markov equivalent minimal independence maps via covered arrow reversals is equivalent to moving along the edge graph of the associahedron $\mathcal{A}_p(\mathcal{C})$. Recall that, unlike edgewalks along $\mathcal{A}_p(\mathcal{C})$, not all Chickering sequences are sequences of minimal independence maps. Instead, a single move along an edge of $\mathcal{A}_p(\mathcal{C})$ is given by reversing a covered arrow to produce a Markov equivalent graph, taking a linear extension of that graph, and then computing the associated minimal independence map. For instance, a single edge of the associahedron depicted in Figure 1 corresponds to transforming the DAG $\mathcal{G}_\pi$ into $\mathcal{G}$ and then into $\mathcal{G}_\tau$. Intuitively, by Lemma 6 and Theorem 7, one would expect that we can always move in such a fashion from a minimal independence map to a sparser one that better approximates the sparsest minimal independence map $\mathcal{G}_{\pi^*}$. Proposition 8 (b) says this intuition is correct, and Proposition 8 (a) ensures that once we make it to the Markov equivalence class of the sparsest minimal independence map, moving between elements of the class is equivalent to moving along the edge graph of $\mathcal{A}_p(\mathcal{C})$. These ideas form the basis for the proof of the following theorem.

Theorem 9. *Algorithms 1 and 2 are pointwise consistent under the faithfulness assumption.*

Proof. Since the triangle assumption implies the edge assumption, it suffices to prove consistency of the triangle sparsest permutation algorithm. Suppose that $\mathcal{C}$ is a graphoid that is faithful to the sparsest minimal independence map $\mathcal{G}_{\pi^*}$ with respect to $\mathcal{C}$. By Lemma 6, we know that $\mathcal{G}_{\pi^*} \leq \mathcal{G}_\pi$ for all $\pi \in S_p$. By (b) of Proposition 8, if Algorithm 2 is at a minimal independence map $\mathcal{G}_\tau$ that is not in the same Markov equivalence class as $\mathcal{G}_\pi^*$, then we can take a weakly decreasing edgewalk along $\mathcal{A}_p(\mathcal{C})$ to reach a sparser minimal independence map $\mathcal{G}_{\tau'}$ satisfying $\mathcal{G}_{\pi^*} \leq \mathcal{G}_{\tau'} \leq \mathcal{G}_\tau$. Following repeated applications of Proposition 8 (b), the algorithm eventually returns a minimal independence map in the Markov equivalence class of $\mathcal{G}_{\pi^*}$. In order for the algorithm to verify that it is in the correct Markov equivalence class, it needs to compute a minimal independence map $\mathcal{G}_\tau$ for a linear extension τ for each DAG $\mathcal{G} \approx \mathcal{G}_{\pi^*}$ and check that it is not sparser than $\mathcal{G}_{\pi^*}$. Proposition 8 (a) states that any such minimal independence map $\mathcal{G}_\tau$ is connected to $\mathcal{G}_{\pi^*}$ by a weakly decreasing edgewalk along $\mathcal{A}_p(\mathcal{C})$. In other words, the Markov equivalence class of $\mathcal{G}_{\pi^*}$ is a connected subgraph of the edge graph of $\mathcal{A}_p(\mathcal{C})$, and hence the algorithm can verify that it has reached the sparsest Markov equivalence class and terminate. Thus, Algorithm 2 is pointwise consistent under the faithfulness assumption. □

4.2. **Consistency of Algorithm 2 using the Bayesian information criterion.** We now show that a version of Algorithm 2 that uses the Bayesian information criterion instead of graph sparsity is also consistent under faithfulness. This algorithm is Algorithm 3, and it is constructed in analogy to the methods studied in [31].

Algorithm 3: Triangle Sparsest Permutation Algorithm with Bayesian information criterion

```
Input  : Observations $\hat{X}$, initial permutation $\pi$.
Output: Permutation $\hat{\pi}$ with DAG $\mathcal{G}_{\hat{\pi}}$.
Set $\hat{\mathcal{G}}_\pi := \underset{\mathcal{G} \text{ consistent with permutation } \pi}{\operatorname{argmax}} \mathrm{BIC}(\mathcal{G}; \hat{X})$.
Using a depth-first search approach with root $\pi$, search for a permutation $\tau$
  with $\mathrm{BIC}(\hat{\mathcal{G}}_\tau; \hat{X}) > \mathrm{BIC}(\hat{\mathcal{G}}_\pi; \hat{X})$ that is connected to $\pi$ through a sequence
  of permutations $(\pi_1, \cdots, \pi_k)$ where each permutation $\pi_i$ is produced from
  $\pi_{i-1}$ by first doing a covered arrow reversal $\hat{\mathcal{G}}_{\pi_{i-1}}$ and selecting a linear
  extension $\pi_i$ of the DAG $\hat{\mathcal{G}}_{\pi_{i-1}}$. If no such $\hat{\mathcal{G}}_\tau$ exists, return $\hat{\mathcal{G}}_\pi$; else set
  $\pi := \tau$ and repeat.
```

Theorem 10. *Algorithm 3 is pointwise consistent under the faithfulness assumption.*

Theorem 10 is proven in Section B of the Supplementary Material. It is based on the fact that the Bayesian information criterion is locally consistent. This fact follows from the first line of the proof of [4, Lemma 7], which states that Bayesian scoring is locally consistent. We note that Algorithm 3 differs from the ordering-based search method proposed in [31] in two main ways: First, Algorithm 3 selects each new permutation by a covered arrow reversal in the associated independence maps. Second, it uses a depth-first-search approach instead of greedy hill-climbing. In particular, our search guarantees that any independence map of minimal independence maps $\mathcal{G}_\pi \leq \mathcal{G}_\tau$ are connected by a Chickering sequence. The proof of Theorem 10 then follows since $|\mathcal{G}_\tau| < |\mathcal{G}_\pi|$ if and only if $\mathrm{BIC}(\mathcal{G}_\tau; \hat{X}) > \mathrm{BIC}(\mathcal{G}_\pi; \hat{X})$, for any minimal independence maps $\mathcal{G}_\pi$ and $\mathcal{G}_\tau$. However, since this fact does not hold for arbitrary DAGs satisfying the Markov assumption with respect to a given distribution, the algorithm of [31] lacks known consistency guarantees.

4.3. **Beyond faithfulness.** We now examine the relationships between the edge, triangle, sparsest Markov representation, faithfulness, and restricted faithfulness assumptions. All proofs for this section can be found in Section B of the Supplementary Material.

Theorem 11. *Faithfulness implies Assumption 5, which implies Assumption 4, which implies the sparsest Markov representation assumption. Moreover, all implications are strict.*

It is clear from the definition that restricted faithfulness is a significantly weaker assumption than faithfulness. In [26, Theorem 2.5] it was shown that the sparsest Markov representation assumption is strictly weaker than restricted faithfulness. In the following, we compare restricted faithfulness to the triangle assumption and show that the restricted faithfulness assumption is not weaker than the triangle assumption.

Theorem 12. *Let $\mathbb{P}$ be a semigraphoid w.r.t. a DAG $\mathcal{G}$. If $\mathbb{P}$ satisfies the triangle assumption, then $\mathbb{P}$ satisfies adjacency faithfulness with respect to $\mathcal{G}$. However, there exist distributions $\mathbb{P}$ such that $\mathbb{P}$ satisfies the triangle assumption with respect to a DAG $\mathcal{G}$ and $\mathbb{P}$ does not satisfy orientation faithfulness with respect to $\mathcal{G}$.*

```
Algorithm 4: The Greedy Sparsest Permutation Algorithm
Input  : A set of conditional independence relations $\mathcal{C}$ on node set $[p]$, and
         two positive integers $d$ and $r$.
Output: A minimal independence map $\mathcal{G}_\pi$.
1  Set $R := 0$, and $Y := \emptyset$.
2  while $R < r$ do
3      Select a permutation $\pi \in S_n$ and set $\mathcal{G} := \mathcal{G}_\pi$.
4      Using a depth-first search approach with root $\mathcal{G}$, search for a minimal
         independence map $\mathcal{G}_\tau$ with $|\mathcal{G}| > |\mathcal{G}_\tau|$ that is connected to $\mathcal{G}$ by a
         weakly decreasing sequence determined by covered arrow reversals that
         is length at most $d$.
5      if no such $\mathcal{G}_\tau$ exists then
6          set $Y := Y \cup \{\mathcal{G}\}$, $R := R + 1$, and go to step 2.
7      else
8          set $\mathcal{G} := \mathcal{G}_\tau$ and go to step 4.
9      end
10 end
11 Return the sparsest DAG $\mathcal{G}_\pi$ in the collection $Y$.
```

4.4. **The problem of Markov equivalence.** It is important to note that in contrast to for example the PC-algorithm, Algorithms 1 and 2 may need to search over DAGs that belong to the same Markov equivalence class. This is due to the fact that two DAGs in the same Markov equivalence class differ only by a sequence of covered arrow reversals [3, Theorem 2]. Thus, the greedy nature of Algorithm 2 can leave us searching through large portions of Markov equivalence classes until we identify a sparser minimal independence map. In particular, in order for Algorithm 2 to terminate, it must visit all members of the Markov equivalence class $\mathcal{M}(\mathcal{G}^*)$.

To address this problem, Algorithm 4 provides a parametrized alternative that approximates Algorithm 2. We call this algorithm the greedy sparsest permutation algorithm since this is the version that we run in practice; see Section 6. The greedy sparsest permutation algorithm operates exactly like Algorithm 2, with the exception that it bounds the search depth d and number of runs r allowed before the algorithm terminates. Recall that Algorithm 2 searches for a weakly decreasing edge-walk from a minimal independence map $\mathcal{G}_\pi$ to another $\mathcal{G}_\tau$ with $|\mathcal{G}_\pi| > |\mathcal{G}_\tau|$ via a depth-first-search approach. In Algorithm 4, if this search step does not produce a sparser minimal independence map after searching up to and including depth d, the algorithm terminates and returns $\mathcal{G}_\pi$. The computational analysis in [13] suggests that the average Markov equivalence class contains four graphs. This suggests that a search depth of 4 is, on average, sufficient for escaping a Markov equivalence class of minimal independence maps. This intuition is verified via simulations in Section 6.

5. Uniform Consistency

In this section we show that minor adjustments can turn Algorithm 2 into Algorithm 5, which is uniformly consistent in the high-dimensional Gaussian setting.

Algorithm 5: A High-dimensional Greedy Sparsest Permutation Algorithm

```
Input: Observations $\hat{X}$, threshold $\lambda$, and initial permutation $\pi_0$.
Output: Permutation $\hat{\pi}$ together with the DAG $\hat{\mathcal{G}}_{\hat{\pi}}$.
Construct the minimal independence map $\hat{\mathcal{G}}_{\pi_0}$ from the initial permutation
  $\pi_0$ and $\hat{X}$;
Perform Algorithm 2 with constrained conditioning sets, i.e., let $i \to j$ be a
  covered arrow and let $S = \mathrm{pa}(i) = \mathrm{pa}(j) \setminus \{i\}$; perform the edge flip,
  i.e. $i \leftarrow j$, and update the DAG by removing edges $(k, i)$ for $k \in S$ such
  that $|\hat{\rho}_{i,k|(S\cup\{j\}\setminus\{k\})}| \leq \lambda$ and edges $(k, j)$ for $k \in S$ such that
  $|\hat{\rho}_{j,k|(S\setminus\{k\})}| \leq \lambda$.
```

In particular, Algorithm 5 only tests conditioning sets made up of parent nodes of covered arrows. This feature turns out to be critical for high-dimensional consistency. Recently, it was shown that a variant of greedy equivalence search is consistent in the high-dimensional setting [22]. Similarly to this approach, by assuming sparsity of the initial DAG, we obtain uniform consistency of the triangle sparsest permutation algorithm in the high-dimensional setting, i.e., it converges to the data-generating DAG when p scales with n.

Letting the dimension p grow as a function of the sample size n, we write $p = p_n$. Similarly, for the true underlying DAG and the data-generating distribution we let $G^* = G^*_n$ and $\mathbb{P} = \mathbb{P}_n$, respectively. The assumptions under which we will guarantee high-dimensional consistency of Algorithm 5 are as follows:

(1) $\mathbb{P}_n$ is multivariate Gaussian and faithful to the DAG $\mathcal{G}^*_n$ for all n.
(2) The number of nodes p_n scales as $p_n = \mathcal{O}(n^a)$ for some $0 \leq a < 1$.
(3) Given an initial permutation π_0, the maximal degree d_{π_0} of the corresponding minimal independence map $\mathcal{G}_{\pi_0}$ satisfies $d_{\pi_0} = \mathcal{O}(n^{1-m})$ for some $0 < m \leq 1$.
(4) There exists $M < 1$ and $c_n > 0$ such that all non-zero partial correlations $\rho_{i,j|S}$ satisfy $|\rho_{i,j|S}| \leq M$ and $|\rho_{i,j|S}| \geq c_n$ where $c_n^{-1} = \mathcal{O}(n^\ell)$ for some $0 < \ell < m/2$.

Analogous to the conditions needed in [15], assumptions (1), (2), (3), and (4) relate to faithfulness, the scaling of the number of nodes with the number of observations, the maximum degree of the initial DAG, and bounds on the minimal non-zero and maximal partial correlations, respectively. In the Gaussian setting, the conditional independence relation $X_j \perp\!\!\!\perp X_k | X_S$ is equivalent to the partial correlation $\rho_{j,k|S} = \mathrm{corr}(X_j, X_k | X_S)$ equaling zero, and a hypothesis test based on Fischer's z-transform can be used to test whether $X_j \perp\!\!\!\perp X_k | X_S$. Combining these facts, we arrive at the following theorem.

Theorem 13. *Suppose that assumptions (1), (2), (3), and (4) hold and let the threshold* λ *in Algorithm 5 be defined as* $\lambda := c_n/2$. *Then there exists a constant* $c > 0$ *such that Algorithm 5 is consistent, i.e., it returns a DAG* $\hat{\mathcal{G}}_{\hat{\pi}}$ *that is in the same Markov equivalence class as* $\mathcal{G}^*_n$, *with probability at least* $1 - \mathcal{O}\{\exp(-cn^{1-2\ell})\}$, *where* ℓ *is defined to satisfy assumption (4).*

As seen in the proof of Theorem 13, consistent estimation in the high-dimensional setting requires that we initialize the algorithm at a permutation satisfying assumption (3). This assumption corresponds to a sparsity constraint. In the Gaussian

Algorithm 6: A neighbor-based minimum degree algorithm

```
Input: Observations $\hat{X}$, threshold $\lambda$
Output: Permutation $\hat{\pi}$ together with the DAG $\hat{\mathcal{G}}_{\hat{\pi}}$
Set $S := [p]$; construct undirected graph $\hat{G}_S$ with $(i,j) \in \hat{G}_S$ if and only if
    $|\hat{\rho}_{i,j|(S\setminus\{i,j\})}| \geq \lambda$;
while $S \neq \emptyset$ do
    Uniformly draw node $k$ from all nodes with the lowest degree in the
      graph $\hat{G}_S$;
    Construct $\hat{G}_{S\setminus\{k\}}$ by first removing node $k$ and its adjacent edges; then
      update the graph $\hat{G}_{S\setminus\{k\}}$ as follows:
          $\forall i,j \in \mathrm{adj}(\hat{G}_S,k)$: if $(i,j)$ not an edge in $\hat{G}_S$, add $(i,j)$;
                              else $(i,j)$ an edge in $\hat{G}_{S\setminus\{k\}}$ iff $|\hat{\rho}_{i,j|S\setminus\{i,j,k\}}| \geq \lambda$;
          $\forall i,j \notin \mathrm{adj}(\hat{G}_S,k)$: $(i,j)$ an edge in $\hat{G}_{S\setminus\{k\}}$ iff $(i,j)$ an edge in $\hat{G}_S$.
      Set $\hat{\pi}(k) := |S|$ and $S := S \setminus \{k\}$.
Output the minimal independence map $\hat{\mathcal{G}}_{\hat{\pi}}$ constructed from $\hat{\pi}$ and $\hat{X}$.
```

oracle setting the problem of finding a sparsest DAG is equivalent to finding the sparsest Cholesky decomposition of the inverse covariance matrix [26]. Various heuristics have been developed for finding sparse Cholesky decompositions, the most prominent being the minimum degree algorithm [12, 33]. In Algorithm 6 we provide a heuristic for finding a sparse minimal independence map $\mathcal{G}_\pi$ that reduces to the minimum degree algorithm in the oracle setting as shown in Theorem 15. In Algorithm 6, for a subset of nodes $S \subset [p]$ we let G_S denote the vertex-induced subgraph of G with node set S, and for $k \in V$ we let $\mathrm{adj}(G,i)$ denote the nodes $k \in V \setminus \{i\}$ such that $\{i,k\} \in E$. The following theorem states that Algorithm 6 is equivalent to the minimum degree algorithm [33] in the oracle setting.

Theorem 14. *Let the data-generating distribution $\mathbb{P}$ be multivariate Gaussian with precision matrix Θ. Then in the oracle-setting the set of possible output permutations from Algorithm 6 is equal to the possible output permutations of the minimum degree algorithm applied to Θ.*

The following result shows that Algorithm 6 in the non-oracle setting is also equivalent to the minimum degree algorithm in the oracle setting.

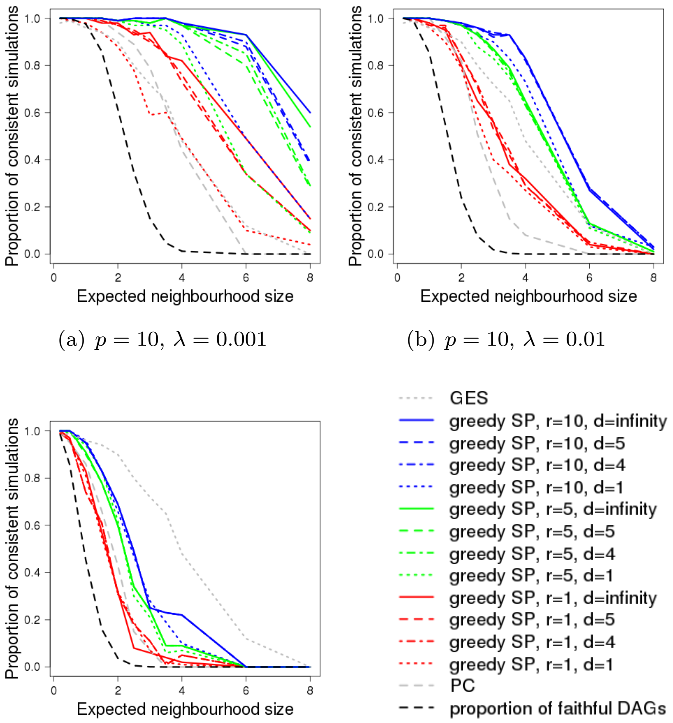
Theorem 15. *Suppose that assumptions (1), (2), and (4) hold, and let the threshold λ in Algorithm 5 be defined as $\lambda := c_n/2$. Then with probability at least $1 - \mathcal{O}\{\exp(-cn^{1-2\ell})\}$ the output permutation from Algorithm 6 is contained in the possible output permutations of the minimum degree algorithm applied to Θ.*

6. Simulations

The simulations presented in this section were done using the `R` library `pcalg` [17], and linear structural equation models with Gaussian noise:

$$(X_1, \ldots, X_p)^T = \{(X_1, \ldots, X_p)A\}^T + \varepsilon,$$

where $\varepsilon \sim \mathcal{N}(0, \mathbb{I}_p)$ with $\mathbb{I}_p$ being the $p \times p$ identity matrix and $A = [a_{ij}]_{i,j=1}^p$ is, without loss of generality, an upper-triangular matrix of edge weights with $a_{ij} \neq 0$



(a) $p = 10$, $\lambda = 0.001$

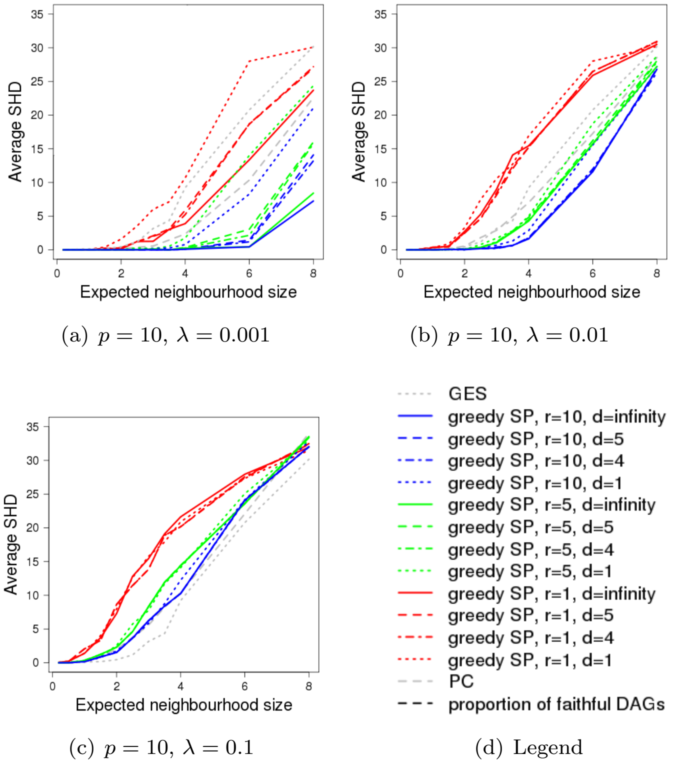
(b) $p = 10$, $\lambda = 0.01$

(c) $p = 10$, $\lambda = 0.1$

(d) Legend

FIGURE 2. Expected neighborhood size versus proportion of consistently recovered Markov equivalence classes based on 100 simulations for each expected neighborhood size on DAGs with $p = 10$ nodes, edge weights sampled uniformly in $[-1, -0.25] \cup [0.25, 1]$, and λ-values 0.1, 0.01 and 0.001. Greedy SP denotes Algorithm 4. When $r = 1$ and $d = \infty$ this is Algorithm 2.

if and only if $i \to j$ is an arrow in the underlying DAG $\mathcal{G}^*$. For each simulation study, we generated 100 realizations of a p-node random Gaussian DAG model on an Erdös-Renyi graph for different values of p and expected neighborhood sizes; i.e., edge probabilities. The edge weights a_{ij} were sampled uniformly in $[-1, -0.25] \cup [0.25, 1]$, ensuring that the edge weights are bounded away from 0. We first analyzed the oracle setting, where we have access to the true underlying covariance matrix Σ. In the remaining simulations, n samples were drawn from the distribution induced by the Gaussian DAG model for different values of n and p. In the oracle setting, the conditional independence relations were computed by thresholding the partial correlations using different thresholds λ. For the simulations with n samples, conditional independence relations were estimated by applying Fisher's z-transform and comparing the derived p-values with a significance level α. In the oracle and low-dimensional settings, the greedy equivalence search, denoted GES in all figures, was simulated using the Bayesian information criterion. In the high-dimensional setting, we used the ℓ_0-penalized maximum likelihood estimation score [22, 35].



(a) $p = 10$, $\lambda = 0.001$

(b) $p = 10$, $\lambda = 0.01$

(c) $p = 10$, $\lambda = 0.1$

(d) Legend

FIGURE 3. Expected neighborhood size versus structural Hamming distance between the true and recovered Markov equivalence classes based on 100 simulations for each expected neighborhood size on DAGs with $p = 10$ nodes, edge weights sampled uniformly in $[-1, -0.25] \cup [0.25, 1]$, and λ-values 0.1, 0.01 and 0.001.

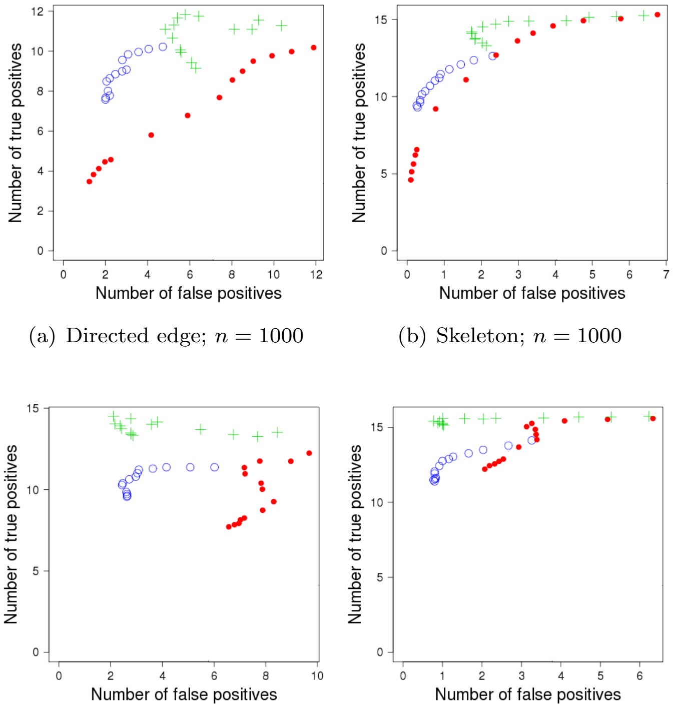
Figure 2 compares the proportion of consistently estimated DAGs in the oracle setting for Algorithm 4 with number of runs $r \in \{1, 5, 10\}$ and depth $d \in \{1, 4, 5, \infty\}$, the PC-algorithm, and the greedy equivalence search. Notice that the instance of Algorithm 4 with parameter settings $r = 1$ and $d = \infty$ is the triangle sparsest permutation algorithm; i.e. Algorithm 2. The number of nodes in these simulations is $p = 10$, and we consider λ-values: $0.1, 0.01$ and 0.001 for the PC-algorithm and Algorithm 4. Note that we only run the greedy equivalence search with $n = 100,000$ samples since there is no oracle version for this algorithm. As expected, increasing the number of runs for Algorithm 4 results in a consistently higher rate of model recovery. In addition, for each fixed number of runs, Algorithm 4 with search depth $d = 4$ performs similarly to $d = \infty$, in line with the observation that the average Markov equivalence class has 4 elements, as discussed in Section 4.4. For this reason, we recommend setting the search depth $d = 4$. Regarding the choice of r, in the low-dimensional setting we have found that choosing r to be of the same magnitude as the number of nodes p produces good estimates. To accelerate computations, for high-dimensional sparse graphs with large p, we used $d = 1$ and $r = 50$; see Figure 6.

For each run, we also recorded the structural Hamming distance between the true and the recovered Markov equivalence classes. Figure 3 shows the average structural Hamming distance versus the expected neighborhood size of the true DAG. While Figure 2 demonstrates that Algorithm 4 with search depth $d = 4$ and multiple runs learns the true Markov equivalence class at a higher rate than the PC-algorithm and greedy equivalence search when λ is chosen small, Figure 3 shows that, for small values of d and r, when Algorithm 4 learns the wrong DAG it is further off from the true DAG than the PC-algorithm. On the other hand, it appears that this trend only holds for Algorithm 4 with a relatively small search depth and few runs. That is, increasing the value of these parameters ensures that the wrong DAG learned by Algorithm 4 will consistently be closer to the true DAG than that learned by the PC-algorithm.

Recall that Algorithm 4 and the PC-algorithm can be sensitive to wrong conditional independence test results. Each edge and non-edge in the DAG returned by Algorithm 4 or the PC-algorithm is the result of a conditional independence test. To get a sense of their respective sensitivities to wrong conditional independence tests, in Figure 4 we report the number of true positives and false positives for directed edge recovery and skeleton recovery. Each data point in the plots represent the average number of true positives and false positives based on 100 simulated models with $p = 8$ nodes and expected neighborhood size 4 with a fixed parameter setting. For Algorithm 4 and the PC-algorithm, the reported data points correspond to 14 chosen significance levels in the interval $[0.00005, 0.6]$. In practice, we recommend tuning the significance level parameter via stability selection [16] as described in Section 7. Similarly, for greedy equivalence search the reported data points correspond to 14 different choices of the scaling constant c from the interval $[0.125, 100]$ for the ℓ_0-penalization parameter $\lambda_n = c\log(n)$. In Figure 4, we see that Algorithm 4 generally outperforms greedy equivalence search and the PC-algorithm in both directed edge and skeleton recovery with large enough sample size.

As noted in Section 3, we do not consider Algorithm 1 in these simulations. Recall that Algorithm 2 relies on the fact that a, generally strict, subset of the edges of the polytope $\mathcal{A}_p(\mathcal{C})$ have a combinatorial interpretation in terms of their associated minimal independence maps, which makes moving between elements of the search space easier to code. Since we do not have a complete combinatorial characterization of all edges of $\mathcal{A}_p(\mathcal{C})$, implementing Algorithm 1 would require generating a geometric realization of $\mathcal{A}_p(\mathcal{C})$ in a program such as `polymake` [11], recovering the complete edge graph of this embedding, and then implementing our search over this data structure. A natural line of follow-up research is to identify a complete combinatorial interpretation of the edges of $\mathcal{A}_p(\mathcal{C})$ so as to allow for an implementation of Algorithm 1 that does not require computing the entire polytope $\mathcal{A}_p(\mathcal{C})$ and its edge graph. As shown in Theorem 11, an efficient implementation of Algorithm 1 should recover the true DAG at a higher rate than Algorithm 2.

We then compared the recovery performance of Algorithm 4 to the sparsest permutation algorithm, greedy equivalence search, the PC-algorithm and its original version, denoted SGS in Figure 5, and the max-min hill-climbing algorithm [34], which is denoted MMHC in Figure 5. This hybrid method first estimates a skeleton through conditional independence testing and then performs a hill-climbing search to orient the edges. We fixed the number of nodes to be $p = 8$ due to the computational limitations of the sparsest permutation algorithm, and considered sample



(a) Directed edge; $n = 1000$

(b) Skeleton; $n = 1000$

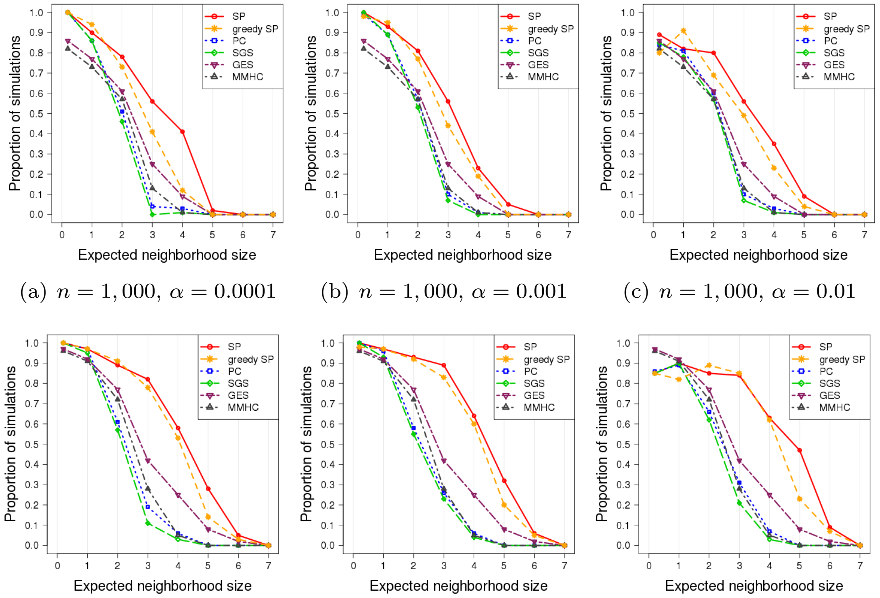
(c) Directed edge; $n = 10000$

(d) Skeleton; $n = 10000$

FIGURE 4. Receiver operating characteristic curves for directed edge recovery and skeleton recovery based on 100 simulations on DAGs with 8 nodes, expected neighborhood size 4 and sample size $n \in \{1000, 10000\}$. The dots denote GES, crosses is greedy SP, and circles is PC.

sizes $n = \{1,000, 10,000\}$. We analyzed the performance of greedy equivalence search using the Bayesian information criterion along with Algorithm 4 and the PC-algorithm for $\alpha = \{0.01, 0.001, 0.0001\}$. Figure 5 shows that the sparsest permutation and greedy sparsest permutation algorithms achieve the best performance among all algorithms. Since for computational reasons the sparsest permutation algorithm cannot be applied to graphs with over 10 nodes, Algorithm 4 is the preferable approach for most applications.

In the remainder of this section, we analyze the performance of Algorithm 5 in the sparse high-dimensional setting. We compared the performance of Algorithm 5 with $d = 1$ and $r = 50$ with methods that have high-dimensional consistency guarantees; namely the PC-algorithm [15] and greedy equivalence search [22, 35]. The initial permutation of Algorithm 6 and its associated minimal independence map were used as a starting point in Algorithm 5, called high-dim greedy SP in Figure 6. To understand the influence of accurately selecting an initial minimal independence map on the performance of Algorithm 5, we also considered the case when the moral graph of the data-generating DAG is given as prior knowledge; these results appear in Figure 6 (d)-(f). Figure 6 compares the skeleton recovery of Algorithm 5 with the PC-algorithm and greedy equivalence search, both without



(a) $n = 1,000$, $\alpha = 0.0001$ (b) $n = 1,000$, $\alpha = 0.001$ (c) $n = 1,000$, $\alpha = 0.01$

(d) $n = 10,000$, $\alpha = 0.0001$ (e) $n = 10,000$, $\alpha = 0.001$ (f) $n = 10,000$, $\alpha = 0.01$

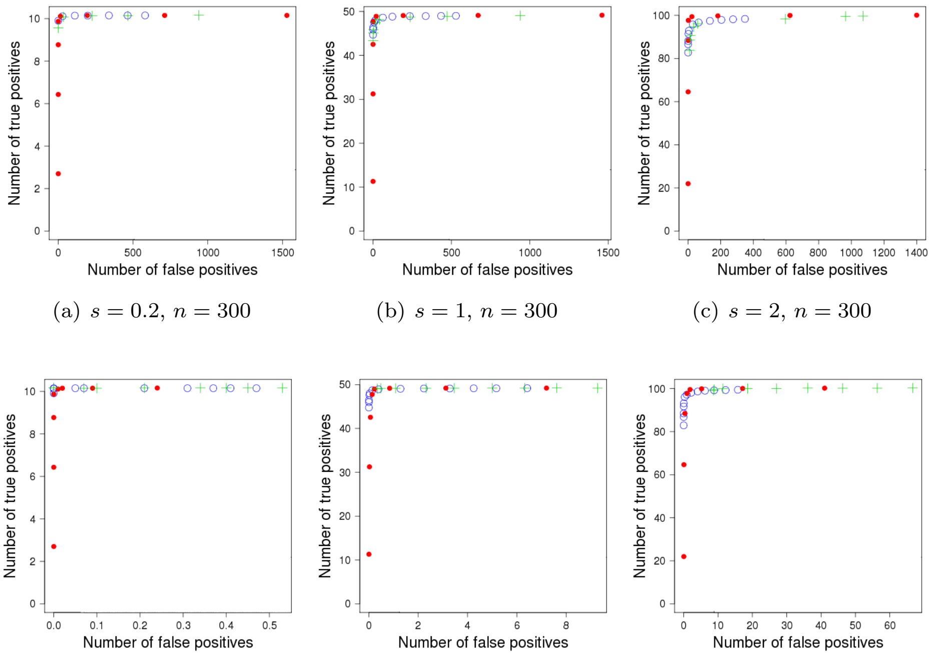
FIGURE 5. Expected neighborhood size versus proportion of consistently recovered skeleta based on 100 simulations for each expected neighborhood size on DAGs with $p = 8$ nodes, sample size $n = 1,000$ and $10,000$, edge weights sampled uniformly in $[-1, -0.25] \cup [0.25, 1]$, and α-values 0.01, 0.001 and 0.0001; we used $r = 10$ and $d = 4$ for the greedy sparsest permutation algorithm.

prior knowledge of the moral graph (subfigures (a)-(c)), and with prior knowledge of the moral graph (subfigures (d)-(f)). We used the ARGES-CIG algorithm [22] to run greedy equivalence search with knowledge of the moral graph.

The number of nodes in our simulations is $p = 100$, the number of samples considered is $n = 300$, and the neighborhood sizes used are $s = 0.2$, 1 and 2. We varied the tuning parameters of each algorithm; namely, the significance level α for the PC-algorithm and Algorithm 5, and the penalization parameter λ_n for greedy equivalence search. We reported the average number of true positives and false positives for each tuning parameter in the plots shown in Figure 6. This figure shows that, unlike the low-dimensional setting, although Algorithm 5 is still comparable to the PC-algorithm and greedy equivalence search in the high-dimensional setting, greedy equivalence search tends to achieve a slightly better performance in some of the settings.

7. An Application to Real Data

In this section, we compare the performance of the greedy sparsest permutation algorithm, i.e., Algorithm 4, with that of the PC-algorithm and greedy equivalence search on the task of gene regulatory network recovery. We consider the perturb-seq data set [7] containing both observational and interventional data from bone-marrow derived dendritic cells. Each data point contains gene expression measurements of 32,777 genes; each interventional data point is sampled from a



(a) $s = 0.2$, $n = 300$

(b) $s = 1$, $n = 300$

(c) $s = 2$, $n = 300$

(d) $s = 0.2$, $n = 300$

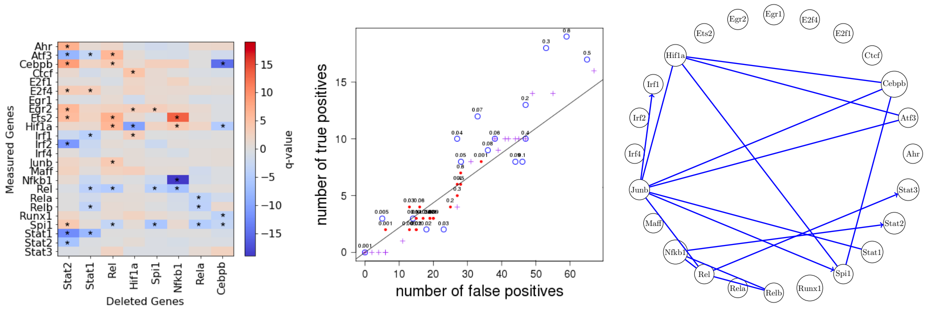
(e) $s = 1$, $n = 300$

(f) $s = 2$, $n = 300$

Figure 6. ROC curves for skeleton recovery for 100 simulations on DAGs with 100 nodes, expected neighborhood size s, sample size n, and edge weights sampled uniformly in $[-1, -0.25] \cup [0.25, 1]$. Figures (a)-(c) are without prior knowledge of the moral graph(d)-(f) are with prior knowledge of the moral graph. Dots denote high-dim GES, crosses denote high-dim greedy SP, and circles denote high-dim PC.

cell where a single gene was targeted for deletion using the CRISPR/Cas9 system. Following preprocessing, the data set consists of 992 observational samples and 13,534 interventional samples over eight gene deletions. As in [7, 37], we focused on learning the DAG structure on 24 genes that are transcription factors known to regulate expression of a variety of different genes, including one another [10].

We used the observational samples to infer the DAG and the interventional samples to evaluate it. In particular, using the interventional data corresponding to a deletion of gene A we identified the genes that are downstream of gene A by testing whether the interventional distribution is significantly different from the observational distribution. For this, we used a Wilcoxon Rank-Sum test with p-value $\alpha = 0.05$, corresponding to a magnitude of at least 3 in the q-value heat map depicted in Figure 7(a); a positive q-value indicates that the gene expression level is increased by the gene deletion, whereas a negative value means that it is decreased. The accuracy of an estimated causal network is evaluated based on the edges adjacent to intervened nodes: an arrow from gene A to gene B in the learned network is considered a true positive if the expression of gene B in the interventional distribution when targeting gene A is significantly different from the observational



(a) Effects of gene deletions

(b) Recovery of gene deletion effects

(c) Network from stability selection

FIGURE 7. (a) Heatmap indicating the effect of each gene deletion on each measured gene; q-values with magnitude at least 3 are marked with “*”. (b) Performance of the causal network learned by Algorithm 4 with $d = 4$ and $r = 20$ (circle) as compared to the PC-algorithm (dot) and GES (cross) in predicting the effect of each intervention; line corresponds to random guessing. (c) The PDAG discovered by Algorithm 4 via stability selection using cutoff 0.6.

distribution, e.g., there is a star in the (A, B)-entry in Figure 7(a), and it is considered a false positive otherwise. Using this metric, Figure 7(b) compares the performance of Algorithm 4 with $d = 4$ and $r = 20$, the PC-algorithm, and greedy equivalence search. Each point in Figure 7(b) corresponds to the number of true positives and false positives in a DAG on the 24 genes learned from the observational data with a fixed parameter setting. The fixed parameter is the significance level of the conditional independence test for Algorithm 4 and the PC-algorithm, and it is the ℓ_0-penalization constant c in the penalty $\lambda_n = c\log(n)$ used in the score function for greedy equivalence search. While the PC-algorithm performs similar to random guessing, the other two algorithms perform better, with the greedy sparsest permutation algorithm, i.e., Algorithm 4, generally outperforming greedy equivalence search.

To get a sense of the corresponding gene regulatory network, in Figure 7(c) we plotted the network constructed from our algorithm using stability selection [16]. We determined the cutoff parameter for stability selection by varying it between 0.5 to 0.8 and found that the resulting network was very robust in the range $[0.6, 0.7]$. The network shown in Figure 7(c) corresponds to a cutoff of 0.6, which is also within the recommended range given in [16]. In the supplementary material, we provide ROC plots using a q-value cutoff of 1 instead of 3, showing that our results and conclusions regarding the comparison of the different algorithms are robust with respect to the selection of q-value cutoff.

8. DISCUSSION

The greedy sparsest permutation algorithm, i.e., Algorithm 4, with parameter choices $d = \infty$ and $r = 1$ is Algorithm 2, which was shown to be consistent under strictly weaker conditions than faithfulness. Algorithm 2 is an approximation of

Algorithm 1, which is further consistent under strictly weaker conditions than Algorithm 2. The fact that Algorithm 2 is consistent under strictly weaker conditions than faithfulness was observed by its performance on simulated data in Section 6. On the other hand, Algorithm 1 was not simulated since we must produce the entire polytope $\mathcal{A}_p(\mathcal{C})$ so as to recover its edge graph. A complete characterization of the edges of $\mathcal{A}_p(\mathcal{C})$ would thus be of use, so that Algorithm 1 can be implemented without computing $\mathcal{A}_p(\mathcal{C})$ and its entire edge graph. Such an implementation would likely recover the true Markov equivalence class more often than any of the algorithms in Section 6. Further perspectives on this could be gained via a characterization of all distributions satisfying Assumption 4 or Assumption 5.

We expect the greedy permutation-based approaches developed in this paper to be useful in a variety of settings. For instance, extensions of Algorithm 4 to the setting where a mix of observational and interventional data is available were presented in [37, 38, 30], and they were implemented using kernel-based conditional independence tests [9, 32] which are better able to deal with non-linear structural equations and non-Gaussian noise. Extensions of Algorithm 4 to the causally insufficient setting are also being developed [1]. In addition, it would be interesting to extend Algorithm 4 so as to accommodate cyclic graphs.

Since passage to a greedy permutation-based algorithm is often motivated by a need to efficiently search through a state space that is super-exponential in size, it would be interesting to compare the computational efficiency of the algorithms discussed in Section 6. Such studies could be conducted using the `CausalDAG` Python package available at https://github.com/uhlerlab/causaldag, which provides an efficient implementation of Algorithm 4.

Acknowledgement

Liam Solus was partially supported by an NSF Mathematical Sciences Postdoctoral Research Fellowship (DMS - 1606407), the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, and Starting Grant (Etableringsbidrag) No. 2019-05195 from The Swedish Research Council (Vetenskapsrådet). Caroline Uhler was partially supported by NSF (DMS-1651995), ONR (N00014-17-1-2147 and N00014-18-1-2765), IBM, a Sloan Fellowship and a Simons Investigator Award. The authors are grateful to Robert Castelo for helpful discussions, as well as two reviewers and an associate editor at Biometrika for their quick and thoughtful comments.

Appendix A. Background Material and an Example

A.1. **Background.**

Here, we provide some definitions from graph theory and causal inference that we will use in the coming proofs. Given a DAG $\mathcal{G} := ([p], A)$ with node set $[p] := \{1, 2, \dots, p\}$ and arrow set A, we associate to the nodes of $\mathcal{G}$ a random vector $(X_1, \dots, X_p)$ with a probability distribution $\mathbb{P}$. An arrow in A is an ordered pair of nodes (i, j) which we will often denote by $i \rightarrow j$. A directed path in $\mathcal{G}$ from node i to node j is a sequence of directed edges in $\mathcal{G}$ of the form $i \rightarrow i_1 \rightarrow i_2 \rightarrow \cdots \rightarrow j$. A path from i to j is a sequence of arrows between i and j that connect the two nodes without regard to direction. The parents of a node i in $\mathcal{G}$ is the collection $\mathrm{Pa}_G(i) := \{k \in [p] : k \rightarrow i \in A\}$, and the ancestors of i, denoted $\mathrm{An}_{\mathcal{G}}(i)$, is the collection of all nodes $k \in [p]$ for which there

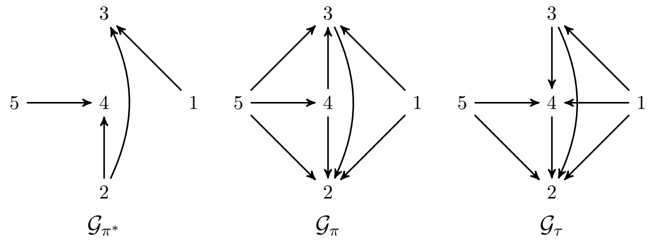


FIGURE 8. An edge of a DAG associahedron that does not correspond to a covered edge flip. The DAG associahedron $\mathcal{A}_p(\mathcal{C})$ is constructed for the conditional independence relations implied by the d-separation statements for $\mathcal{G}_{\pi^*}$ with $\pi^* = 15234$. The DAGs $\mathcal{G}_\pi$ and $\mathcal{G}_\tau$ with $\pi = 15432$ and $\tau = 15342$ correspond to adjacent vertices in $\mathcal{A}_p(\mathcal{C})$, connected by the edge labeled by the transposition of 3 and 4. The arrow between nodes 3 and 4 is not covered in either DAG $\mathcal{G}_\pi$ or $\mathcal{G}_\tau$.

exists a directed path from k to i in $\mathcal{G}$. We do not include i in $\mathrm{An}_\mathcal{G}(i)$. The descendants of i, denoted $\mathrm{De}_\mathcal{G}(i)$, is the set of all nodes $k \in [p]$ for which there is a directed path from i to k in $\mathcal{G}$, and the nondescendants of i is the collection of nodes $\mathrm{Nd}_\mathcal{G}(i) := [p]\backslash(\mathrm{De}_\mathcal{G}(i) \cup \{i\})$. When the DAG $\mathcal{G}$ is understood from context we write $\mathrm{Pa}(i)$, $\mathrm{An}(i)$, $\mathrm{De}(i)$, and $\mathrm{Nd}(i)$, for the parents, ancestors, descendants, and nondescendants of i in $\mathcal{G}$, respectively. The analogous definitions and notation will also be used for any set $S \subset [p]$. If two nodes are connected by an arrow in $\mathcal{G}$ then we say they are adjacent. A triple of nodes (i, j, k) is called unshielded if i and j are adjacent, k and j are adjacent, but i and k are not adjacent. An unshielded triple (i, j, k) forms an immorality if it is of the form $i \to j \leftarrow k$. In any triple, shielded or not, with arrows $i \to j \leftarrow k$, the node j is called a collider. Given disjoint subsets $A, B, C \subset [p]$ with $A \cap B = \emptyset$, we say that A is d-connected to B given C if there exist nodes $i \in A$ and $j \in B$ for which there is a path between i and j such that every collider on the path is in $\mathrm{An}(C) \cup C$ and no non-collider on the path is in C. If no such path exists, we say A and B are d-separated given C.

Example 16. An example of a DAG associahedron containing an edge that does not correspond to a covered arrow reversal in either DAG labeling its endpoints can be constructed as follows: Let $\mathcal{G}_{\pi^*}$ denote the left-most DAG depicted in Figure 8, and let $\mathcal{C}$ denote those conditional independence relations implied by the d-separation statements for $\mathcal{G}_{\pi^*}$. Then for the permutations $\pi = 15432$ and $\tau = 15342$, the DAGs $\mathcal{G}_\pi$ and $\mathcal{G}_\tau$ label a pair of adjacent vertices of $\mathcal{A}_p(\mathcal{C})$ since π and τ differ by the transposition of 3 and 4. This adjacent transposition corresponds to a reversal of the arrow between nodes 3 and 4 in $\mathcal{G}_\pi$ and $\mathcal{G}_\tau$. However, this arrow is not covered in either minimal independence map. We further note that this example shows that not all edges of $\mathcal{A}_p(\mathcal{C})$ can be described by covered arrow reversals even when $\mathcal{C}$ is faithful to the sparsest minimal independence map, $\mathcal{G}_{\pi^*}$.

Appendix B. Proofs for Results on the Pointwise Consistency of the Greedy sparsest permutation algorithms

B.1. **Proof of Lemma 6.** Suppose first that $\mathcal{C}$ is not faithful to $\mathcal{G}$ and take any conditional independence statement $i \perp\!\!\!\perp j \,|\, K$ that is not encoded by the d-separation statements in $\mathcal{G}$. Take π to be any permutation in which $K \prec_\pi i \prec_\pi j \prec_\pi [p]\backslash(K \cup \{i,j\})$. Then $\mathcal{G} \not\leq \mathcal{G}_\pi$ since $i \perp\!\!\!\perp j \,|\, K$ is encoded by the d-separations of $\mathcal{G}_\pi$ but not by the d-separations of $\mathcal{G}$.

Conversely, suppose $\mathbb{P}$ is faithful to $\mathcal{G}$. By [23, Theorem 9, Page 119], we know that $\mathbb{P}$ satisfies the Markov Assumption with respect to $\mathcal{G}_\pi$ for any $\pi \in S_n$. So any conditional independence relation encoded by $\mathcal{G}_\pi$ holds for $\mathbb{P}$, which means it also holds for $\mathcal{G}$. Thus, $\mathcal{G} \leq \mathcal{G}_\pi$. □

To prove that Algorithm 2 is consistent under faithfulness we require a number of lemmas pertaining to the steps of the Chickering algorithm. For the convenience of the reader, we recall the Chickering algorithm in Algorithm 7.

Lemma 17. *Suppose $\mathcal{G} \leq \mathcal{H}$ such that the Chickering algorithm has reached step 5 and selected the arrow $Y \rightarrow Z$ in $\mathcal{G}$ to reverse. If $Y \rightarrow Z$ is not covered in $\mathcal{G}$, then there exists a Chickering sequence*

$$\left(\mathcal{G} = \mathcal{G}^0, \mathcal{G}^1, \mathcal{G}^2, \ldots, \mathcal{G}^N \leq \mathcal{H}\right)$$

in which $\mathcal{G}^N$ is produced by the reversal of $Y \rightarrow Z$, and for all $i = 1, 2, \ldots, N-1$, the DAG $\mathcal{G}^i$ is produced by an arrow addition via step 7 or 8 with respect to the arrow $Y \rightarrow Z$.

Proof. Until the arrow $Y \rightarrow Z$ is reversed, the set $\mathrm{De}_\mathcal{G}(Y)$ and the node choice $D \in \mathrm{De}_\mathcal{G}(Y)$ remain the same. This is because steps 7 and 8 only add parents to Y or Z that are already parents of Y or Z, respectively. Thus, we can always choose the same Y and Z until $Y \rightarrow Z$ is covered. □

```
Algorithm 7: APPLY-EDGE-OPERATION
  Input  : DAGs $\mathcal{G}$ and $\mathcal{H}$ where $\mathcal{G} \leq \mathcal{H}$ and $\mathcal{G} \neq \mathcal{H}$.
  Output: A DAG $\mathcal{G}'$ satisfying $\mathcal{G}' \leq \mathcal{H}$ that is given by reversing an edge in $\mathcal{G}$
          or adding an edge to $\mathcal{G}$.
1 Set $\mathcal{G}' := \mathcal{G}$.
2 While $\mathcal{G}$ and $\mathcal{H}$ contain a node $Y$ that is a sink in both DAGs and for which
    $\mathrm{Pa}_\mathcal{G}(Y) = \mathrm{Pa}_\mathcal{H}(Y)$, remove $Y$ and all incident edges from both DAGs.
3 Let $Y$ be any sink node in $\mathcal{H}$.
4 If $Y$ has no children in $G$, then let $X$ be any parent of $Y$ in $\mathcal{H}$ that is not a
    parent of $Y$ in $\mathcal{G}$. Add the edge $X \rightarrow Y$ to $\mathcal{G}'$ and return $\mathcal{G}'$.
5 Let $D \in \mathrm{De}_\mathcal{G}(Y)$ denote the (unique) maximal element from $\mathrm{De}_\mathcal{G}(Y)$ within
    $\mathcal{H}$. Let $Z$ be any maximal child of $Y$ in $\mathcal{G}$ such that $D$ is a descendant of $Z$
    in $\mathcal{G}$.
6 If $Y \rightarrow Z$ is covered in $\mathcal{G}$, reverse $Y \rightarrow Z$ in $\mathcal{G}'$ and return $\mathcal{G}'$.
7 If there exists a node $X$ that is a parent of $Y$ but not a parent of $Z$ in $\mathcal{G}$,
    then add $X \rightarrow Z$ to $\mathcal{G}'$ and return $\mathcal{G}'$.
8 Let $X$ be any parent of $Z$ that is not a parent of $Y$. Add $X \rightarrow Y$ to $\mathcal{G}'$ and
    return $\mathcal{G}'$.
```

For an independence map $\mathcal{G} \leq \mathcal{H}$, the Chickering algorithm first deletes all sinks in $\mathcal{G}$ that have precisely the same parents in $\mathcal{H}$, and repeats this process for the resulting graphs until there is no sink of this type anymore. This is the purpose of step 2 of the algorithm. If the adjusted graph is $\widetilde{\mathcal{G}}$, the algorithm then selects a sink node in $\widetilde{\mathcal{G}}$, which, by construction, must have fewer parents than the same node in $\mathcal{H}$ and/or some children. The algorithm then adds parents and reverses arrows until this node has exactly the same parents as the corresponding node in $\mathcal{H}$. The following lemma shows that this can be accomplished one sink node at a time. The proof is clear from the statement of the algorithm.

Lemma 18. *Let $\mathcal{G} \leq \mathcal{H}$. If Y is a sink node selectable in step 3 of the Chickering algorithm then we may always select Y each time until it is deleted by step 2.*

We would like to see how the sequence of graphs produced in Chickering's algorithm relates to the DAGs $\mathcal{G}_\pi$ for a set of conditional independence relations $\mathcal{C}$. In particular, we would like to see that if $\mathcal{G}_\pi \leq \mathcal{G}_\tau$ for permutations $\pi, \tau \in S_p$, then there is a sequence of moves given by Chickering's algorithm that passes through a sequence of minimal independence maps taking us from $\mathcal{G}_\pi$ to $\mathcal{G}_\tau$. To do so, we require an additional lemma relating independence maps and minimal independence maps. To state this lemma we need to consider the two steps within Algorithm 7 in which arrow additions occur. We now recall these two steps:

(i) Suppose Y is a sink node in $\mathcal{G} \leq \mathcal{H}$. If Y is also a sink node in $\mathcal{G}$, then choose a parent X of Y in $\mathcal{H}$ that is not a parent of Y in $\mathcal{G}$, and add the arrow $X \rightarrow Y$ to $\mathcal{H}$.

(ii) If Y is not a sink node in $\mathcal{G}$, then there exists an arrow $Y \rightarrow Z$ in $\mathcal{G}$ that is oriented in the opposite direction in $\mathcal{H}$. If $Y \rightarrow Z$ is covered, the algorithm reverses it. If $Y \rightarrow Z$ is not covered, there exists (in $\mathcal{G}$) either
   (a) a parent X of Y that is not a parent of Z, in which case, the algorithm adds the arrow $X \rightarrow Z$.
   (b) a parent X of Z that is not a parent of Y, in which case, the algorithm adds the arrow $X \rightarrow Y$.

Lemma 19. *Let $\mathcal{C}$ be a graphoid and $\mathcal{G}_\pi \leq \mathcal{G}_\tau$ with respect to $\mathcal{C}$. Then the common sink nodes of $\mathcal{G}_\pi$ and $\mathcal{G}_\tau$ all have the same incoming arrows. In particular, the Chickering algorithm needs no instance of arrow additions (i) to move from $\mathcal{G}_\pi$ to $\mathcal{G}_\tau$.*

Proof. Suppose on the contrary that there exists some sink node Y in $\mathcal{G}_\pi$ and there is a parent node X of Y in $\mathcal{G}_\tau$ that is not a parent node of Y in $\mathcal{G}_\pi$. Since Y is a sink in both permutations, then there exists linear extensions $\hat{\pi}$ and $\hat{\tau}$ of the partial orders corresponding to $\mathcal{G}_\pi$ and $\mathcal{G}_\tau$ for which $Y = \hat{\pi}_p$ and $Y = \hat{\tau}_p$. By [21, Theorem 7.4], we know that $\mathcal{G}_\pi = \mathcal{G}_{\hat{\pi}}$ and $\mathcal{G}_\tau = \mathcal{G}_{\hat{\tau}}$. In particular, we know that $X \not\perp\!\!\!\perp Y \mid [p]\backslash\{X, Y\}$ in $\mathcal{G}_{\hat{\tau}}$ and $X \perp\!\!\!\perp Y \mid [p]\backslash\{X, Y\}$ in $\mathcal{G}_{\hat{\pi}}$. However, this is a contradiction, since both of these relations cannot simultaneously hold. □

B.2. **Lemmata for the Proof of Proposition 8.** To prove Proposition 8 we must first prove a few lemmas. Throughout the remainder of this section, we use the following notation: Suppose that $\mathcal{G} \leq \mathcal{H}$ for two DAGs $\mathcal{G}$ and $\mathcal{H}$ and that

$$C = (\mathcal{G}^0 := \mathcal{G}, \mathcal{G}^1, \mathcal{G}^2, \ldots, \mathcal{G}^N := \mathcal{H})$$

is a Chickering sequence from $\mathcal{G}$ to $\mathcal{H}$. We let $\pi^i \in S_p$ denote a linear extension of $\mathcal{G}^i$ for all $i = 0, 1, \ldots, N$. For any DAG $\mathcal{G}$ let $\mathrm{CI}(\mathcal{G})$ denote the collection of conditional independence relations encoded by the d-separation statements in $\mathcal{G}$.

Lemma 20. *Suppose that $\mathcal{G}_\tau$ is a minimal independence map of a graphoid $\mathcal{C}$. Suppose also that $\mathcal{G} \approx \mathcal{G}_\tau$ and that $\mathcal{G}$ differs from $\mathcal{G}_\tau$ only by a covered arrow reversal. If π is a linear extension of $\mathcal{G}$ then $\mathcal{G}_\pi$ is a subDAG of $\mathcal{G}$.*

Proof. Suppose that $\mathcal{G}$ is obtained from $\mathcal{G}_\tau$ by the reversal of the covered arrow $x \to y$ in $\mathcal{G}_\tau$. Without loss of generality, we assume that $\tau = SxyT$ and $\pi = SyxT$ for some disjoint words S and T whose letters are collectively in bijection with the elements in $[p]\backslash\{x, y\}$. So in $\mathcal{G}_\pi$, the arrows going from S to T, x to T, and y to T are all the same as in $\mathcal{G}_\tau$. However, the arrows going from S to x and S to y may be different. So, to prove that $\mathcal{G}_\pi$ is a subDAG of $\mathcal{G}$ we must show that for each letter s in the word S

(1) if $s \to x \notin \mathcal{G}_\tau$ then $s \to x \notin \mathcal{G}_\pi$, and
(2) if $s \to y \notin \mathcal{G}_\tau$ then $s \to y \notin \mathcal{G}_\pi$.

To see this, notice that if $s \to x \notin \mathcal{G}_\tau$, then $s \to y \notin \mathcal{G}_\tau$ since $x \to y$ is covered in $\mathcal{G}_\tau$. Similarly, if $s \to y \notin \mathcal{G}_\tau$ then $s \to x \notin \mathcal{G}_\tau$. Thus, we know that $s \perp\!\!\!\perp x \mid S\backslash s$ and $s \perp\!\!\!\perp y \mid (S\backslash s)x$ are both in the collection $\mathcal{C}$. It then follows from the semigraphoid property (2) given in Section 3 that $s \perp\!\!\!\perp x \mid (S\backslash s)y$ and $s \perp\!\!\!\perp y \mid S\backslash s$ are in $\mathcal{C}$ as well. Therefore, $\mathcal{G}_\pi$ is a subDAG of $\mathcal{G}$. □

Lemma 21. *Let $\mathcal{C}$ be a graphoid and let*

$$C = (\mathcal{G}^0 := \mathcal{G}_\pi, \mathcal{G}^1, \mathcal{G}^2, \ldots, \mathcal{G}^N := \mathcal{G}_\tau)$$

be a Chickering sequence from a minimal independence map $\mathcal{G}_\pi$ of $\mathcal{C}$ to another $\mathcal{G}_\tau$. If, for some index $0 \leq i < N$, $\mathcal{G}^i$ is obtained from $\mathcal{G}^{i+1}$ by deletion of an arrow $x \to y$ in $\mathcal{G}^{i+1}$ then $x \to y$ is not in $\mathcal{G}_{\pi^{i+1}}$.

Proof. Let $\pi^{i+1} = SxTyR$ be a linear extension of $\mathcal{G}^{i+1}$ for some disjoint words S, T, and R whose letters are collectively in bijection with the elements in $[p]\backslash\{x, y\}$. Since $\mathcal{G}_{\pi^*} \leq \mathcal{G}^i \leq \mathcal{G}^{i+1}$ then

$$\mathcal{C} \supseteq \mathrm{CI}(\mathcal{G}_\pi) \supseteq \mathrm{CI}(\mathcal{G}^i) \supseteq \mathrm{CI}(\mathcal{G}^{i+1}).$$

We claim that $x \perp\!\!\!\perp y \mid ST \in \mathrm{CI}(\mathcal{G}^i) \subseteq \mathcal{C}$. Therefore, $x \to y$ cannot be an arrow in $\mathcal{G}_{\pi^{i+1}}$.

First, since $\mathcal{G}^i$ is obtained from $\mathcal{G}^{i+1}$ by deleting the arrow $x \to y$, then π^{i+1} is also a linear extension of $\mathcal{G}^i$. Notice, there is no directed path from y to x in $\mathcal{G}^i$, and so it follows that x and y are d-separated in $\mathcal{G}^i$ by $\mathrm{Pa}_{\mathcal{G}^i}(y)$. Therefore, $x \perp\!\!\!\perp y \mid \mathrm{Pa}_{\mathcal{G}^i}(y) \in \mathrm{CI}(\mathcal{G}^i)$. Notice also that $\mathrm{Pa}_{\mathcal{G}^i}(y) \subset ST$ and any path in $\mathcal{G}^i$ between x and y lacking colliders uses only arrows in the subDAG of $\mathcal{G}^i$ induced by the vertices $S \cup T \cup \{x, y\} = [p]\backslash R$. Therefore, $x \perp\!\!\!\perp y \mid ST \in \mathrm{CI}(\mathcal{G}^i)$ as well. It follows that $x \perp\!\!\!\perp y \mid ST \in \mathcal{C}$, and so, by definition, $x \to y$ is not an arrow of $\mathcal{G}_{\pi^{i+1}}$. □

Lemma 22. *Suppose that $\mathcal{C}$ is a graphoid and $\mathcal{G}_\pi$ is a minimal independence map with respect to $\mathcal{C}$. Let*

$$C = (\mathcal{G}^0 := \mathcal{G}_\pi, \mathcal{G}^1, \mathcal{G}^2, \ldots, \mathcal{G}^N := \mathcal{G}_\tau)$$

be a Chickering sequence from $\mathcal{G}_\pi$ to another minimal independence map $\mathcal{G}_\tau$ with respect to $\mathcal{C}$. Let i be the largest index such that $\mathcal{G}^i$ is produced from $\mathcal{G}^{i+1}$ by deletion *of an arrow, and suppose that for all $i+1 < k \leq N$ we have $\mathcal{G}_{\pi^k} = \mathcal{G}^k$. Then $\mathcal{G}_{\pi^{i+1}}$ is a proper subDAG of $\mathcal{G}^{i+1}$.*

Proof. By Lemma 20, we know that $\mathcal{G}_{\pi^{i+1}}$ is a subDAG of $\mathcal{G}^{i+1}$. This is because π^{i+1} is a linear extension of $\mathcal{G}^{i+1}$ and $\mathcal{G}^{i+1} \approx \mathcal{G}^{i+2} = \mathcal{G}_{\pi^{i+2}}$ and $\mathcal{G}^{i+1}$ differs from $\mathcal{G}^{i+2}$ only by a covered arrow reversal. By Lemma 21, we know that the arrow deleted in $\mathcal{G}^{i+1}$ to obtain $\mathcal{G}^i$ is not in $\mathcal{G}_{\pi^{i+1}}$. Therefore, $\mathcal{G}_{\pi^{i+1}}$ is a proper subDAG of $\mathcal{G}$. □

Using these lemmas, we can now give a proof of Proposition 8.

B.3. **Proof of Proposition 8.** To see that (a) holds, notice since $\mathcal{G}_\pi \approx \mathcal{G}_\tau$ then by the transformational characterization of Markov equivalence given in [3, Theorem 2], we know there exists a Chickering sequence

$$C := (\mathcal{G}^0 := \mathcal{G}_\pi, \mathcal{G}^1, \mathcal{G}^2, \dots, \mathcal{G}^N := \mathcal{G}_\tau)$$

for which $\mathcal{G}^0 \approx \mathcal{G}^1 \approx \cdots \approx \mathcal{G}^N$ and $\mathcal{G}^i$ is obtained from $\mathcal{G}^{i+1}$ by the reversal of a covered arrow in $\mathcal{G}^{i+1}$ for all $0 \leq i < N$. Furthermore, since $\mathcal{G}_\pi$ is class-s-minimal, and by Lemma 20, we know that for all $0 \leq i \leq N$

$$\overline{\mathcal{G}}^i \supseteq \overline{\mathcal{G}}_{\pi^i} \supseteq \overline{\mathcal{G}}_\pi.$$

However, since $\mathcal{G}^i \approx \mathcal{G}_\pi$ and $\mathcal{G}_{\pi^i}$ is a subDAG of $\mathcal{G}^i$, then $\mathcal{G}^i = \mathcal{G}_{\pi^i}$ for all i. Thus, the desired weakly decreasing edgewalk along $\mathcal{A}_p(\mathcal{C})$ is

$$(\mathcal{G}_\pi = \mathcal{G}_{\pi^0}, \mathcal{G}_{\pi^1}, \dots, \mathcal{G}_{\pi^{N-1}}, \mathcal{G}_{\pi^N} = \mathcal{G}_\tau).$$

To see that (b) holds, suppose that $\mathcal{G}_\pi \leq \mathcal{G}_\tau$ but $\mathcal{G}_\pi \not\approx \mathcal{G}_\tau$. Since $\mathcal{G}_\pi \leq \mathcal{G}_\tau$, there exists a Chickering sequence from $\mathcal{G}_\pi$ to $\mathcal{G}_\tau$ that uses at least one arrow addition. By Lemmas 17 and 18 we can choose this Chickering sequence such that it resolves one sink at a time and, respectively, reverses one covered arrow at a time. We denote this Chickering sequence by

$$C := (\mathcal{G}^0 := \mathcal{G}_\pi, \mathcal{G}^1, \mathcal{G}^2, \dots, \mathcal{G}^N := \mathcal{G}_\tau).$$

Let i denote the largest index for which $\mathcal{G}^i$ is obtained from $\mathcal{G}^{i+1}$ by deletion of an arrow. Then by our choice of Chickering sequence we know that $\mathcal{G}^k$ is obtained from $\mathcal{G}^{k+1}$ by a covered arrow reversal for all $i < k < N$. Moreover, $\pi^i = \pi^{i+1}$, and so $\mathcal{G}_{\pi^i} = \mathcal{G}_{\pi^{i+1}}$. Furthermore, by Lemma 20 we know that $\mathcal{G}_{\pi^k}$ is a subDAG of $\mathcal{G}^k$ for all $i < k \leq N$.

Suppose now that there exists some index $i+1 < k < N$ such that $\mathcal{G}_{\pi^k}$ is a proper subDAG of $\mathcal{G}^k$. Without loss of generality, we pick the largest such index. It follows that for all indices $k < \ell \leq N$, $\mathcal{G}_{\pi^\ell} = \mathcal{G}^\ell$ and that

$$\mathcal{G}^{k+1} \approx \mathcal{G}^{k+2} \approx \cdots \approx \mathcal{G}^N = \mathcal{G}_\tau.$$

Thus, by [3, Theorem 2], there exists a weakly decreasing edgewalk from $\mathcal{G}_\tau$ to $\mathcal{G}^{k+1}$ on $\mathcal{A}_p(\mathcal{C})$. Since we chose the index k maximally then $\mathcal{G}^k$ is obtained from $\mathcal{G}^{k+1}$ by a covered arrow reversal. Therefore, $\mathcal{G}_{\pi^k}$ and $\mathcal{G}_{\pi^{k+1}}$ are connected by an edge of $\mathcal{A}_p(\mathcal{C})$ indexed by a covered arrow reversal. Since $|\mathcal{G}^k| = |\mathcal{G}^{k+1}| = |\mathcal{G}_{\pi^{k+1}}|$ and $\mathcal{G}_{\pi^k}$ is a proper subDAG of $\mathcal{G}^k$, then the result follows.

On the other hand, suppose that for all indices $i+1 < k \leq N$, we have $\mathcal{G}_{\pi^k} = \mathcal{G}^k$. Then this is precisely the conditions of Lemma 22, and so it follows that $\mathcal{G}_{\pi^{i+1}}$ is a proper subDAG of $\mathcal{G}^{i+1}$. Since $\mathcal{G}^{i+1}$ is obtained from $\mathcal{G}^{i+2}$ by a covered arrow reversal, the result follows. □

B.4. **Proof of Theorem 10.** The proof is composed of two parts. We first prove that for any permutation π, in the limit of large n, $\hat{\mathcal{G}}_\pi$ is a minimal independence map of $\mathcal{G}_\pi$. We prove this by contradiction. Suppose $\hat{\mathcal{G}}_\pi \neq \mathcal{G}_\pi$. Since the Bayesian information criterion is a consistent scoring function [14], in the limit of large n, $\hat{\mathcal{G}}_\pi$ is an independence map of the distribution. Since $\hat{\mathcal{G}}_\pi$ and $\mathcal{G}_\pi$ share the same permutation and $\mathcal{G}_\pi$ is a minimal independence map, then $\mathcal{G}_\pi \subset \hat{\mathcal{G}}_\pi$. Suppose now that there exists $(i,j) \in \hat{\mathcal{G}}_\pi$ such that $(i,j) \notin \mathcal{G}_\pi$. Since $\mathcal{G}_\pi$ is a minimal independence map, we obtain that $i \perp\!\!\!\perp j \,|\, \mathrm{Pa}_{\mathcal{G}_\pi}(j)$. In Lemma 7 of [4], it is shown that Bayesian scoring is locally consistent, and it follows from the first sentence of the proof therein that the Bayesian information criterion is also locally consistent. Since the Bayesian information criterion is locally consistent, it follows that $\mathrm{BIC}(\mathcal{G}_\pi, \hat{X}) > \mathrm{BIC}(\hat{\mathcal{G}}_\pi, \hat{X})$.

Now we prove that for any two permutations τ and π where $\mathcal{G}_\tau$ is connected to $\mathcal{G}_\pi$ by precisely one covered arrow reversal, in the limit of large n,

$$\mathrm{BIC}(\mathcal{G}_\tau; \hat{X}) > \mathrm{BIC}(\mathcal{G}_\pi; \hat{X}) \Leftrightarrow |\mathcal{G}_\tau| < |\mathcal{G}_\pi|,$$

and

$$\mathrm{BIC}(\mathcal{G}_\tau; \hat{X}) = \mathrm{BIC}(\mathcal{G}_\pi; \hat{X}) \Leftrightarrow |\mathcal{G}_\tau| = |\mathcal{G}_\pi|.$$

It suffices to prove

$$|\mathcal{G}_\tau| = |\mathcal{G}_\pi| \Rightarrow \mathrm{BIC}(\mathcal{G}_\tau; \hat{X}) = \mathrm{BIC}(\mathcal{G}_\pi; \hat{X}) \tag{B.1}$$

and

$$|\mathcal{G}_\tau| < |\mathcal{G}_\pi| \Rightarrow \mathrm{BIC}(\mathcal{G}_\tau; \hat{X}) > \mathrm{BIC}(\mathcal{G}_\pi; \hat{X}). \tag{B.2}$$

Eq. B.1 is easily seen to be true using [3, Theorem 2] as $\mathcal{G}_\pi$ and $\mathcal{G}_\tau$ are equivalent. For Eq. B.2, by Theorem 7, since $\mathcal{G}_\tau \leq \mathcal{G}_\pi$ there exists a Chickering sequence from $\mathcal{G}_\tau$ to $\mathcal{G}_\pi$ with at least one edge addition and several covered arrow reversals. For the covered arrow reversals, the Bayesian information criterion remains the same since the involved DAGs are equivalent. For the edge additions, the score necessarily decreases in the limit of large n due to the increase in the number of parameters. This follows from the consistency of the Bayesian information criterion and the fact that DAGs before and after edge additions are both independence maps of $\mathbb{P}$. In this case, the path taken in the triangle sparsest permutation algorithm using the Bayesian information criterion is the same as in the original triangle sparsest permutation algorithm. Since the triangle sparsest permutation algorithm is consistent, it follows that the triangle sparsest permutation algorithm with the Bayesian information criterion is also consistent. □

B.5. **Proof of Theorem 11.** It is quick to see that

$$\begin{aligned}
\text{faithfulness} \quad &\Longrightarrow \quad \text{triangle assumption} \\
\text{triangle assumption} \quad &\Longrightarrow \quad \text{edge assumption} \\
\text{edge assumption} \quad &\Longrightarrow \quad \text{sparsest Markov representation assumption.}
\end{aligned}$$

The first implication is given by Theorem 9, and the latter three are immediate consequences of the definitions of the triangle, edge, and sparsest Markov representation assumptions. Namely, the triangle, edge, and sparsest Markov representation assumptions are each defined to be precisely the condition in which Algorithm 2, Algorithm 1, and the sparsest permutation algorithm are, respectively, consistent. The implications then follow since each of the algorithms is a refined version of the

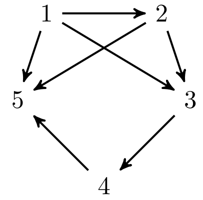


FIGURE 9. A sparsest DAG w.r.t. the conditional independence relations $\mathcal{C}$ given in the proof of Theorem 11.

preceding one in this order. Hence, we only need to show the strict implications. For each statement we identify a collection of conditional independence relations satisfying the former identifiability assumption but not the latter. For the first implication consider the collection of conditional independence relations

$$\begin{aligned}\mathcal{C} = \{1 \perp\!\!\!\perp 5 \,|\, \{2,3\}, \quad & 2 \perp\!\!\!\perp 4 \,|\, \{1,3\}, \quad 3 \perp\!\!\!\perp 5 \,|\, \{1,2,4\}, \\ & 1 \perp\!\!\!\perp 4 \,|\, \{2,3,5\}, \quad 1 \perp\!\!\!\perp 4 \,|\, \{2,3\}\}.\end{aligned}$$

The sparsest DAG $\mathcal{G}_{\pi^*}$ with respect to $\mathcal{C}$ is shown in Figure 9. To see that $\mathcal{C}$ satisfies the triangle assumption with respect to $\mathcal{G}_{\pi^*}$, we can use computer evaluation. To see that it is not faithful with respect to $\mathcal{G}_\pi^*$, notice that $1 \perp\!\!\!\perp 5 \,|\, \{2,3\}$ and $1 \perp\!\!\!\perp 4 \,|\, \{2,3,5\}$ are both in $\mathcal{C}$, but they are not implied by $\mathcal{G}_\pi^*$. We also remark that $\mathcal{C}$ is not a semigraphoid since the semigraphoid property (2) given in Section 3 applied to the conditional independence relations $1 \perp\!\!\!\perp 5 \,|\, \{2,3\}$ and $1 \perp\!\!\!\perp 4 \,|\, \{2,3,5\}$ implies that $1 \perp\!\!\!\perp 5 \,|\, \{2,3,4\}$ should be in $\mathcal{C}$.

For the second implication consider the collection of conditional independence relations

$$\mathcal{D} = \{1 \perp\!\!\!\perp 2 \,|\, \{4\}, \quad 1 \perp\!\!\!\perp 3 \,|\, \{2\}, \quad 2 \perp\!\!\!\perp 4 \,|\, \{1,3\}\}$$

and initialize Algorithm 2 at the permutation $\pi := 1423$. A sparsest DAG $\mathcal{G}_{\pi^*}$ with respect to $\mathcal{D}$ is given in Figure 10(a), and the initial minimal independence map $\mathcal{G}_\pi$ is depicted in Figure 10(b). Notice that the only covered arrow in $\mathcal{G}_\pi$ is $1 \to 4$, and reversing this covered arrow produces the permutation $\tau = 4123$; the corresponding DAG $\mathcal{G}_\tau$ is shown in Figure 10(c). The only covered arrows in $\mathcal{G}_\tau$ are $4 \to 1$ and $4 \to 2$. Reversing $4 \to 1$ returns us to $\mathcal{G}_\pi$, which we already visited, and reversing $4 \to 2$ produces the permutation $\sigma = 2143$; the associated DAG $\mathcal{G}_\sigma$ is depicted in Figure 10(d). Since the only DAGs connected to $\mathcal{G}_\pi$ and $\mathcal{G}_\tau$ via covered arrow flips have at least as many edges as $\mathcal{G}_\pi$ and $\mathcal{G}_\tau$, then Algorithm 2 is inconsistent, and so the triangle assumption does not hold for $\mathcal{C}$. On the other hand, we can verify

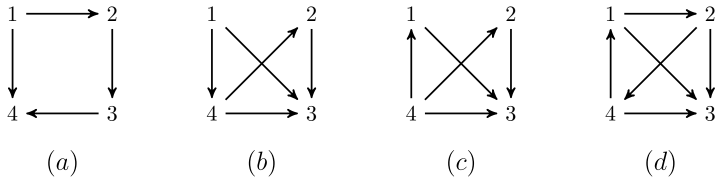


FIGURE 10. The four minimal independence maps with respect to the conditional independence relations $\mathcal{D}$ described in the proof of Theorem 11.

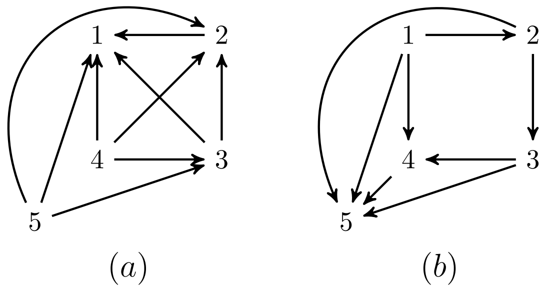


FIGURE 11. The initial minimal independence map and the sparsest minimal independence map with respect to the conditional independence relations $\mathcal{E}$ described in the proof of Theorem 11.

computationally that Algorithm 1 is consistent with respect to $\mathcal{D}$, meaning that the edge assumption holds.

Finally, for the last implication consider the collection of conditional independence relations

$$\mathcal{E} = \{1 \perp\!\!\!\perp 3 \,|\, \{2\}, \quad 2 \perp\!\!\!\perp 4 \,|\, \{1,3\}, \quad 4 \perp\!\!\!\perp 5\},$$

and the initial permutation $\pi = 54321$. The initial DAG $\mathcal{G}_\pi$ and a sparsest DAG $\mathcal{G}_{\pi^*}$ are depicted in Figures 11(a) and (b), respectively. It is not hard to check that any DAG $\mathcal{G}_\tau$ that is edge adjacent to $\mathcal{G}_\pi$ is a complete graph. Thus, the sparsest Markov representation assumption holds for $\mathcal{E}$ but not the edge assumption. □

B.6. **Proof of Theorem 12.** Let $\mathbb{P}$ be a semigraphoid, and let $\mathcal{C}$ denote the conditional independence relations entailed by $\mathbb{P}$. Suppose for the sake of contradiction that Algorithm 2 is consistent with respect to $\mathcal{C}$, but $\mathbb{P}$ fails to satisfy adjacency faithfulness with respect to a sparsest DAG $\mathcal{G}_\pi^*$. Then there exists some conditional independence relation $i \perp\!\!\!\perp j \,|\, S$ in $\mathcal{C}$ such that $i \to j$ is an arrow of $\mathcal{G}_\pi^*$. Now let π be any permutation respecting the concatenated ordering $iSjT$ where $T = [p] \setminus (\{i,j\} \cup S)$. Then our goal is to show that any covered arrow reversal in $\mathcal{G}_\pi$ that results in a minimal independence map $\mathcal{G}_\tau$ with strictly fewer edges than $\mathcal{G}_\pi$ must satisfy the condition that $i \to j$ is not an arrow in $\mathcal{G}_\tau$.

First, we consider the possible types of covered arrows that may exist in $\mathcal{G}_\pi$. To list these, it will be helpful to look at the diagram depicted in Figure 12. Notice first that we need not consider any trivially covered arrows, since such edge reversals do not decrease the number of arrows in the minimal independence maps. Any edge $i \to S$ or $i \to T$ is trivially covered, so the possible cases of non-trivially covered arrows are exactly the covered arrows given in Figure 13. In this figure,

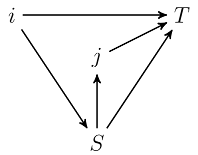


FIGURE 12. This diagram depicts the possible arrows between the node sets $\{i\}, \{j\}, S$, and T for the minimal independence map $\mathcal{G}_\pi$ considered in the proof of Theorem 12.

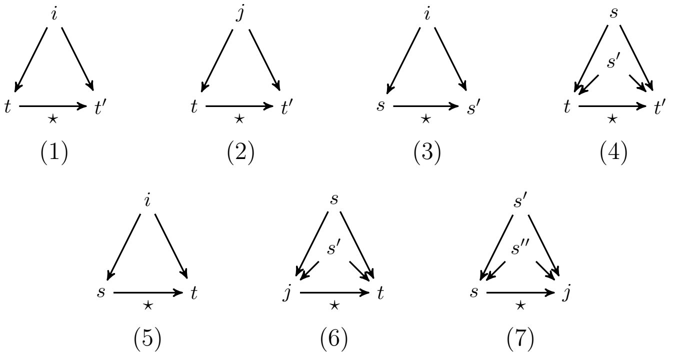


Figure 13. The possible non-trivially covered arrows between the node sets $\{i\}, \{j\}, S$, and T for the minimal independence map $\mathcal{G}_\pi$ considered in the proof of Theorem 12 are labeled with the symbol $\star$. Here, we take $s, s', s'' \in S$ and $t, t' \in T$.

each covered arrow to be considered is labeled with the symbol $\star$. Notice that the claim is trivially true for cases (1) – (4); i.e., any covered arrow reversal resulting in edge deletions produces a minimal independence map $\mathcal{G}_\tau$ for which $i \to j$ is not an arrow of $\mathcal{G}_\tau$.

Case (5) is also easy to see. Recall that $\pi = is_1 \cdots s_k j t_1 \cdots t_m$ where $S := \{s_1, \ldots, s_k\}$ and $T := \{t_1, \ldots, t_k\}$, and that reversing the covered arrow in case (5) results in an edge deletion. Since $s \to t$ is covered, then there exists a linear extension τ of $\mathcal{G}_\pi$ such that s and t are adjacent in τ. Thus, either j precedes both s and t or j follows both s and t in τ. Recall also that by [21, Theorem 7.4] we known $\mathcal{G}_\tau = \mathcal{G}_\pi$. Thus, reversing the covered arrow $s \to t$ in $\mathcal{G}_\tau = \mathcal{G}_\pi$ does not add in $i \to j$.

To see the claim also holds for cases (6) and (7), we utilize the semigraphoid property (2) given in Section 3. It suffices to prove the claim for case (6). So suppose that reversing the $\star$-labeled edge $j \to t$ from case (6) results in a minimal independence map with fewer arrows. We simply want to see that $i \to j$ is still a non-arrow in this new DAG. Assuming once more that $\pi = is_1 \cdots s_k j t_1 \cdots t_m$, by [21, Theorem 7.4] we can, without loss of generality, pick $t := t_1$. Thus, since $i \perp\!\!\!\perp j \mid S$ and $j \to t$ is covered, then $i \perp\!\!\!\perp t \mid S \cup \{j\}$. By the semigraphoid property (2), we then know that $i \perp\!\!\!\perp j \mid S \cup \{t\}$. Thus, the covered arrow reversal $j \leftarrow t$ produces a permutation $\tau = is_1 \cdots s_k t_1 j t_2 \cdots t_m$, and so $i \to j$ is not an arrow in $\mathcal{G}_\tau$. This completes all cases of the proof.

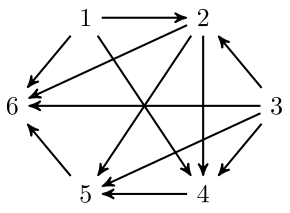


Figure 14. A sparsest DAG for the conditional independence relations $\mathcal{C}$ considered in the proof of Theorem 12.

To complete the proof, we provide an example of a distribution $\mathbb{P}$ that satisfies the triangle assumption but not orientation faithfulness. Consider any probability distribution entailing the conditional independence relations

$$\mathcal{C} = \{1 \perp\!\!\!\perp 3, \quad 1 \perp\!\!\!\perp 5 \,|\, \{2,3,4\}, \quad 4 \perp\!\!\!\perp 6 \,|\, \{1,2,3,5\}, \quad 1 \perp\!\!\!\perp 3 \,|\, \{2,4,5,6\}\}.$$

For example, $\mathcal{C}$ can be faithfully realized by a regular Gaussian. From left-to-right, we label these conditional independence relations as c_1, c_2, c_3, c_4. For the collection $\mathcal{C}$, a sparsest DAG $\mathcal{G}_{\pi^*}$ is depicted in Figure 14. Note that since there is no equally sparse or sparser DAG that is Markov with respect to $\mathbb{P}$ then $\mathbb{P}$ satisfies the sparsest Markov representation assumption with respect to $\mathcal{G}_{\pi^*}$. Notice also that the conditional independence relation c_4 does not satisfy the orientation faithfulness assumption with respect to $\mathcal{G}_{\pi^*}$. Moreover, if $\mathcal{G}_\pi$ entails c_4, then the subDAG on the nodes $\pi_1, \ldots, \pi_5$ forms a complete graph. Thus, by [3, Theorem 2], we can find a sequence of covered arrow reversals preserving edge count such that after all covered arrow reversals, $\pi_5 = 6$. Then transposing the entries $\pi_5\pi_6$ produces a permutation τ in which c_3 holds. Therefore, the number of arrows in $\mathcal{G}_\pi$ is at least the number of arrows in $\mathcal{G}_\tau$. Even more, $\mathcal{G}_\tau$ is an independence map of $\mathcal{G}_{\pi^*}$, i.e., $\mathcal{G}_{\pi^*} \leq \mathcal{G}_\tau$. So by Proposition 8, there exists a weakly decreasing edge walk determined by covered arrow reversals along $\mathcal{A}_p(\mathcal{C})$ taking us from $\mathcal{G}_\tau$ to $\mathcal{G}_\pi^*$. Thus, we conclude that $\mathbb{P}$ satisfies the triangle assumption, but not orientation faithfulness. □

Appendix C. Proofs for Results on the Uniform Consistency of the Greedy sparsest permutation algorithm

C.1. **Lemmata for the Proof of Theorem 13.** To prove Theorem 13, we require a pair of lemmas, the first of which shows that the conditioning sets in the triangle sparsest permutation algorithm can be restricted to parent sets of covered arrows.

Lemma 23. *Suppose that the data-generating distribution $\mathbb{P}$ is faithful to $\mathcal{G}^*$. Then for any permutation π and any covered arrow $i \to j$ in $\mathcal{G}_\pi$ it holds that*

(a) $i \perp\!\!\!\perp k | (S' \cup \{j\}) \setminus \{k\}$ if and only if $i \perp\!\!\!\perp k | (S \cup \{j\}) \setminus \{k\}$,
(b) $j \perp\!\!\!\perp k | S' \setminus \{k\}$ if and only if $j \perp\!\!\!\perp k | S \setminus \{k\}$,

for all $k \in S$, where S is the set of common parent nodes of i and j, and $S' = \{a : a <_\pi \max_\pi(i,j)\}$.

Proof. Let $\mathrm{Pa}_{\mathcal{G}_\pi}(j)$ be the set of parent nodes of node j in the DAG $\mathcal{G}_\pi$. Let $k \in S$ and let $\mathbb{P}_1$ denote the joint distribution of (X_i, X_j, X_k) conditioned on $S \setminus \{k\}$ and $\mathbb{P}_2$ the joint distribution of (X_i, X_j, X_k) conditioned on S'. Then the claimed statements boil down to

(a) $j \perp\!\!\!\perp k$ under distribution $\mathbb{P}_1 \Leftrightarrow j \perp\!\!\!\perp k$ under distribution $\mathbb{P}_2$;
(b) $i \perp\!\!\!\perp k | j$ under distribution $\mathbb{P}_1 \Leftrightarrow i \perp\!\!\!\perp k | j$ under distribution $\mathbb{P}_2$.

Note that

$$\mathbb{P}_1(X_i, X_j, X_k) := \mathbb{P}(X_i, X_j, X_k | X_{S\setminus\{k\}}) = \mathbb{P}(X_i, X_j | X_S)\mathbb{P}_1(X_k).$$

Similarly, the Markov assumption of $\mathbb{P}$ with respect to G_π implies that

$$\mathbb{P}_2(X_i, X_j, X_k) = \mathbb{P}(X_i, X_j | X_{S'})\mathbb{P}_2(X_k) = \mathbb{P}(X_i, X_j | X_S)\mathbb{P}_2(X_k).$$

Hence, $\mathbb{P}_1(X_j | X_k) = \mathbb{P}_2(X_j | X_k)$, $\mathbb{P}_1(X_i | X_j, X_k) = \mathbb{P}_2(X_i | X_j, X_k)$. This completes the proof since $X_a \perp\!\!\!\perp X_b | X_C$ under some distribution $\tilde{\mathbb{P}}$ if and only if $\tilde{\mathbb{P}}(X_a | X_b = z_1, X_C) = \tilde{\mathbb{P}}(X_a | X_b = z_2, X_C)$ for all z_1 and z_2 in the sample space. □

The second lemma we require was first proven in [15, Lemma 3] and is here restated for the sake of completeness.

Lemma 24. [15, Lemma 3] *Suppose that assumption (4) holds, and let $z_{i,j|S}$ be the z-transform of the partial correlation coefficient $\rho_{i,j|S}$. Then*

$$\mathbb{P}[|\hat{z}_{i,j|S} - z_{i,j|S}| > \gamma] \leq \mathcal{O}(n - |S|) * \Phi, \text{ where}$$
$$\Phi = \left[\exp\left\{(n-4-|S|)\log\left(\frac{4-(\gamma/L)^2}{4+(\gamma/L)^2}\right)\right\} + \exp\{-C_2(n-|S|)\}\right],$$

where C_2 is some constant such that $0 < C_2 < \infty$ and

$$L = 1/(1-(1+M)^2/4),$$

in which M is defined such that it satisfies assumption (4).

Provided with Lemmas 23 and 24, we can then prove Theorem 13.

C.2. **Proof of Theorem 13.** For any initial permutation π_0, we let L_{π_0} denote the set of tuples (i,j,S) used for partial correlation testing in the estimation of the initial permtuation DAG $\mathcal{G}_{\pi_0}$. That is,

$$L_{\pi_0} := \Big\{(i,j,S) : S = \{k : \pi_0(k) \leq \max(\pi_0(i), \pi_0(j))\} \setminus \{i,j\}\Big\}.$$

Given a DAG $\mathcal{G}$ and a node i we let $\mathrm{adj}(\mathcal{G}, i)$ denote the collection of nodes that share an arrow with node i in G. We then let K_{π_0} denote the collection of tuples (i,j,S) that will be used in the partial correlation testing done in step (2) of Algorithm 5; i.e.

$$K_{\pi_0} := \bigcup_{(i,j)\in\overline{\mathcal{G}}_{\pi_0}} \Big\{(k,l,S) : k \in \{i,j\},\ l \in \mathrm{adj}(\mathcal{G}_{\pi_0}, i) \cap \mathrm{adj}(\mathcal{G}_{\pi_0}, j),\ S \neq \emptyset, \text{ and}$$
$$S \subseteq \{\mathrm{adj}(\mathcal{G}_{\pi_0}, i) \cap \mathrm{adj}(\mathcal{G}_{\pi_0}, j)\} \cup \{i,j\}\Big\}.$$

It follows from Lemma 23, that when flipping a covered edge $i \to j$ in a minimal independence map $\mathcal{G}_{\tilde{\pi}}$, it is sufficient to calculate the partial correlations $\rho_{a,b|C}$ where

$$(a,b,C) \in \Big\{(a,b,C) : a = i,\ b \in \mathrm{Pa}_i(\mathcal{G}_{\tilde{\pi}}),\ C = \mathrm{Pa}_i(\mathcal{G}_{\tilde{\pi}}) \cup \{j\} \setminus \{b\}\Big\} \cup$$
$$\Big\{(a,b,C) : a = j,\ b \in \mathrm{Pa}_i(\mathcal{G}_{\tilde{\pi}}),\ C = \mathrm{Pa}_i(\mathcal{G}_{\tilde{\pi}}) \setminus \{b\}\Big\}.$$

In particular, we have that $(a,b,C) \in K_{\tilde{\pi}}$.

Because of the skeletal inclusion $\overline{\mathcal{G}}_{\tilde{\pi}} \subseteq \overline{\mathcal{G}}_{\pi_0}$, it follows that $K_{\tilde{\pi}} \subseteq K_{\pi_0}$ and hence $(a,b,C) \in K_{\pi_0}$. In addition, for all partial correlations $\rho_{a,b|C}$ used for constructing the initial DAG $\mathcal{G}_{\pi_0}$, we know that $(a,b,C) \in L_{\pi_0}$. Therefore, for all partial correlations (a,b,C) used in the algorithm, we have:

$$(a,b,C) \in K_{\pi_0} \cup L_{\pi_0}.$$

Let $E_{i,j|S}$ be the event where an error occurs when doing partial correlation testing of $i \perp\!\!\!\perp j|S$, and suppose that α is the significance level when testing this partial correlation. Then we see that $E_{i,j|S}$ corresponds to:

$$(n-|S|-3)^{1/2}|\hat{z}_{i,j|S}| > \Phi^{-1}(1-\alpha/2), \qquad \text{when } z_{i,j|S} = 0;$$
$$(n-|S|-3)^{1/2}|\hat{z}_{i,j|S}| \leq \Phi^{-1}(1-\alpha/2), \qquad \text{when } z_{i,j|S} \neq 0.$$

Choosing $\alpha_n = 2(1-\Phi(n^{1/2}c_n/2))$ it follows under assumption (4) that

$$\mathbb{P}[E_{i,j|S}] \leq \mathbb{P}[|\hat{z}_{i,j|S} - z_{i,j|S}| > (n/(n-|S|-3))^{1/2}c_n/2].$$

Now, by (2) we have that $|S| \leq p = \mathcal{O}(n^a)$. Hence it follows that

$$\mathbb{P}[E_{i,j|S}] \leq \mathbb{P}[|\hat{z}_{i,j|S} - z_{i,j|S}| > c_n/2].$$

Then, Lemma 24 together with the fact that $\log(\frac{4-\delta^2}{4+\delta^2}) \sim -\delta^2/2$ as $\delta \to 0$, imply that

$$\mathbb{P}[E_{i,j|S}] \leq \mathcal{O}(n-|S|)\exp\{-c'(n-|S|)c_n^2\} \leq \mathcal{O}\left(\exp(\log n - cn^{1-2\ell})\right) \tag{C.1}$$

for some constants $c, c' > 0$. Since the DAG estimated using Algorithm 5 is not consistent when at least one of the partial correlation tests is not consistent, then the probability of inconsistency can be estimated as follows:

$$\begin{aligned}\mathbb{P}[\text{an error occurs in Algorithm 5}] &\leq \mathbb{P}\left(\bigcup_{i,j,S\in K_{\hat{\pi}}\cup L_{\hat{\pi}}} E_{i,j|S}\right) \\ &\leq |K_{\hat{\pi}} \cup L_{\hat{\pi}}| \left(\sup_{i,j,S\in K_{\hat{\pi}}\cup L_{\hat{\pi}}} \mathbb{P}(E_{i,j|S})\right).\end{aligned} \tag{C.2}$$

Next note that assumption (3) implies that the size of the set $\mathrm{adj}(\mathcal{G}_{\pi_0}, i) \cup \mathrm{adj}(\mathcal{G}_{\pi_0}, j)$ is at most d_{π_0}. Therefore, $|K_{\pi_0}| \leq p^2 \cdot d_{\pi_0} \cdot 2^{d_{\pi_0}}$ and $|L_{\pi_0}| \leq p^2$. Thus, we see that

$$|K_{\hat{\pi}} \cup L_{\hat{\pi}}| \leq |K_{\hat{\pi}}| + |L_{\hat{\pi}}| \leq (2^{d_{\pi_0}} \cdot d_{\pi_0} + 1)p^2.$$

Therefore, the left-hand-side of inequality (C.2) is upper-bounded by

$$(2^{d_{\pi_0}} \cdot d_{\pi_0} + 1)p^2 \left(\sup_{i,j,S\in K_{\hat{\pi}}\cup L_{\hat{\pi}}} \mathbb{P}(E_{i,j|S})\right).$$

Combining this observation with the upper-bound computed in (C.1), we obtain that the left-hand-side of (C.2) is upper-bounded by

$$\begin{gathered}(2^{d_{\pi_0}} \cdot d_{\pi_0} + 1)p^2\mathcal{O}(\exp(\log n - cn^{1-2l})) \leq \\ \mathcal{O}(\exp(d_{\pi_0}\log 2 + 2\log p + \log d_{\pi_0} + \log n - cn^{1-2\ell})).\end{gathered}$$

By assumptions (3) and (4) it follows that $n^{1-2\ell}$ dominates all terms in this bound. Thus, we conclude that

$$\mathbb{P}[\text{estimated DAG is consistent}] \geq 1 - \mathcal{O}(\exp(-cn^{1-2\ell})).$$

□

The proof of Theorem 14 is based on the following lemma.

Lemma 25. *Let $\mathbb{P}$ be a distribution on $[p]$ that is faithful to a DAG $\mathcal{G}$, and let $\mathbb{P}_S$ denote the marginal distribution on $S \subset [p]$. Let G_S be the undirected graphical model corresponding to $\mathbb{P}_S$, i.e., the edge $\{i,j\}$ is in G_S if and only if $\rho_{i,j|(S\backslash\{i,j\})} \neq 0$. Then $G_{S\backslash\{k\}}$ can be obtained from G_S as follows:*

1. *for all $i,j \in \mathrm{adj}(G_S, k)$, if $\{i,j\}$ is not an edge in G_S, then add $\{i,j\}$. Otherwise, $\{i,j\}$ is an edge of $G_{S\backslash\{k\}}$ if and only if $|\rho_{i,j|S\backslash\{i,j,k\}}| \neq 0$.*
2. *for all $i,j \notin \mathrm{adj}(G_S, k)$, $\{i,j\}$ is an edge of $G_{S\backslash\{k\}}$ if and only if $\{i,j\}$ is an edge in G_S.*

Proof. First, we prove:

$$\text{For } i,j \notin \text{adj}(G_S,k): \quad (i,j) \text{ is an edge in } G_{S\setminus\{k\}} \text{ iff } (i,j) \text{ is an edge in } G_S.$$

Suppose at least one of i or j are not adjacent to node k in G_S. Without loss of generality, we assume i is not adjacent to k in G_S; this implies that $\rho_{i,k|S\setminus\{i,k\}} = 0$. To prove the desired result we must show that

$$\rho_{i,j|S\setminus\{i,j\}} = 0 \Leftrightarrow \rho_{i,j|S\setminus\{i,j,k\}} = 0.$$

To show this equivalence, first suppose that $\rho_{i,j|S\setminus\{i,j\}} = 0$ but $\rho_{i,j|S\setminus\{i,j,k\}} \neq 0$. This implies that there is a path P between i and j through k such that nodes i and j are d-connected given $S \setminus \{i,j,k\}$ and d-separated given $S \setminus \{i,j\}$. This implies that k is a non-collider along P. Define P_i as the path connecting i and k in the path P and P_j the path connecting j and k in P. Then the nodes i and j are d-connected to k given $S\setminus\{i,k\}$ and $S\setminus\{j,k\}$ respectively, by using P_i and P_j. Since j is not on P_i, clearly i and k are also d-connected given $S\setminus\{i,j,k\}$ through P_i, and the same holds for j.

Conversely, suppose that $\rho_{i,j|S\setminus\{i,j,k\}} = 0$ but $\rho_{i,j|S\setminus\{i,j\}} \neq 0$. Then there exists a path P that *d*-connects nodes i and j given $S\setminus\{i,j\}$, while i and j are d-separated given $S \setminus \{i,j,k\}$. Thus, one of the following must occur:

(1) k is a collider on the path P, or
(2) Some node $\ell \in \text{an}(S \setminus \{i,j\}) \setminus \text{an}(S \setminus \{i,j,k\})$ is a collider on P.

For case (2), there must exist a path: $\ell \to \cdots \to k$ that d-connects ℓ and k given $S \setminus \{i,j,k\}$ and $\ell \notin S$. Such a path exists since ℓ is an ancestor of k and not an ancestor of all other nodes in $S \setminus \{i,j,k\}$. So in both cases i and k are also d-connected given $S \setminus \{i,j,k\}$ using a path that does not containing the node j. Hence, i and k are also d-connected given $S \setminus \{i,k\}$, a contradiction.

Next, we prove for $i,j \in \text{adj}(G_S,k)$, if (i,j) is not an edge in G_S, then (i,j) is an edge in $G_{S\setminus\{k\}}$. Since $i \in \text{adj}(G_S,k)$, there exists a path P_i that d-connects i and k given $S \setminus \{i,k\}$, and similar for j. Using the same argument as the above, i and j are also d-connected to k using P_i and P_j, respectively, given $S\setminus\{i,j,k\}$. Defining P as the path that combines P_i and P_j, then k must be a non-collider along P as otherwise i and j would be d-connected given $S\setminus\{i,j\}$, in which case i and j would also be d-connected given $S \setminus \{i,j,k\}$, and (i,j) would be an edge in $G_{S\setminus\{k\}}$. □

C.3. **Proof of Theorem 14.** In the oracle setting, there are two main differences between Algorithm 6 and the minimum degree algorithm. First, Algorithm 6 uses partial correlation testing to construct a graph, while the minimum degree algorithm uses the precision matrix Θ. The second difference is that Algorithm 6 only updates based on the partial correlations of neighbors of the tested nodes.

Let Θ_S denote the precision matrix of the marginal distribution over the variables $\{X_i : i \in S\}$. Since the marginal distribution is Gaussian, the (i,j)-th entry of Θ_S is nonzero if and only if $\rho_{i,j|S\setminus\{i,j\}} \neq 0$. Thus, to prove that Algorithm 6 and the minimum degree algorithm are equivalent, it suffices to show the following: Let G_S be an undirected graph with edges corresponding to the nonzero entries of Θ_S. Then for any node k, the graph $G_{S\setminus\{k\}}$ constructed as defined in Algorithm 6 has edges corresponding to the nonzero entries of $\Theta_{S\setminus\{k\}}$. To prove that this is indeed the case, note that by Lemma 25, if G_S is already estimated then nodes i and j are connected in $G_{S\setminus\{k\}}$ if and only if $\rho_{i,j|S\setminus\{i,j,k\}} \neq 0$. Finally, since the marginal

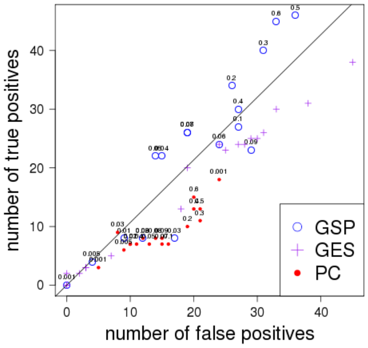


FIGURE 15. Performance of the causal network learned by Algorithm 4 with $d = 4$ and $r = 20$ as compared to the PC-algorithm and GES in predicting the effect of each intervention using a q-value cutoff of 1; line corresponds to random guessing.

distribution over S is multivariate Gaussian, the (i, j)-th entry of $\Theta_{S\setminus\{k\}}$ is non-zero if and only if $\rho_{i,j|S\setminus\{i,j,k\}} \neq 0$. □

C.4. **Proof of Theorem 15.** Let $\mathbb{P}^{\text{oracle}}(\hat{\pi})$ denote the probability that $\hat{\pi}$ is output by Algorithm 6 in the oracle-setting, and let $N_{\hat{\pi}}$ denote the number of partial correlation tests that had to be performed. Then $N_{\hat{\pi}} \leq \mathcal{O}(pd_{\hat{\pi}}^2)$, where $d_{\hat{\pi}}$ is the maximum degree of the corresponding minimal independence map $\mathcal{G}_{\hat{\pi}}$. Therefore, using the same arguments as in the proof of Theorem 13, we obtain:

$$\begin{aligned}
&\mathbb{P}[\hat{\pi} \text{ is generated by Algorithm 6}] \\
&\quad \geq \mathbb{P}^{\text{oracle}}(\hat{\pi})\mathbb{P}[\text{all hypothesis tests for generating } \hat{\pi} \text{ are consistent}] \\
&\quad \geq \mathbb{P}^{\text{oracle}}(\hat{\pi})\left(1 - \mathcal{O}(pd_{\hat{\pi}}^2)\sup_{(i,j,S)\in N_{\hat{\pi}}} \mathbb{P}(E_{i,j|S})\right), \\
&\quad \geq \mathbb{P}^{\text{oracle}}(\hat{\pi})\left(1 - \mathcal{O}(\exp(2\log d_{\hat{\pi}} + \log p + \log n - c'n^{1-2\ell}))\right), \\
&\quad \geq \mathbb{P}^{\text{oracle}}(\hat{\pi})\left(1 - \mathcal{O}(\exp(-cn^{1-2\ell}))\right).
\end{aligned}$$

Let Π denote the set of all possible output permutations of the minimum degree algorithm applied to Θ. Then

$$\begin{aligned}
&\mathbb{P}[\text{Algorithm 6 outputs a permutation in } \Pi] \\
&\quad \geq \sum_{\hat{\pi}\in\Pi} \mathbb{P}[\hat{\pi} \text{ is output by Algorithm 6}], \\
&\quad \geq 1 - \mathcal{O}(\exp(-cn^{1-2\ell})),
\end{aligned}$$

which completes the proof. □

APPENDIX D. ADDITIONAL FIGURES FOR EXPERIMENTS

In this section, we present an additional figure supporting our experimental findings in Section 7. Figure 15 shows the resulting receiver operating characteristic curves for the greedy sparsest permutation algorithm, the PC-algorithm as well as greedy equivalence search when using a q-value of 1 to identify true positive / false positive edges. More specifically, we consider an arrow from gene A to gene B in the learned network as a true positive if the magnitude of the corresponding q-value is larger than 1, and a false positive otherwise. The random guessing line was adjusted accordingly. Our greedy sparsest permutation algorithm outperforms the PC-algorithm and greedy equivalence search, which both perform similar to random guessing.

Appendix E. Computational Times for Simulations

To test the computational efficiency of our greedy sparsest permutation algorithm, we compared its run time to the PC-algorithm and greedy equivalence search in the setting $p = 8, s = 4$ and $n = 1000$, which is the setting considered in Figures 4 (a)-(b) in the main paper. For a fair comparison, selected the hyperparameters of each algorithm so that the resulting graphs have a similar sparsity, namely 0.001 for our greedy sparsest permutation search, 0.01 for the PC-algorithm and $\lambda_n = 1/2\log(n)$ for greedy equivalence search. The R implementation of our greedy sparsest permutation algorithm used in this paper took 0.42 seconds for one run, while it took 0.08 seconds for the PC-algorithm and 0.02 seconds for greedy equivalence search (using the pcalg package in R). While our implementation of the greedy sparsest permutation algorithm should be seen mainly as a proof-of-concept, in the meantime, a faster implementation of the greedy sparsest permutation algorithm has been developed and is available as a python package at `https://github.com/uhlerlab/causaldag`.

Finally, we note that the moves used in Algorithm 4 are a strict subset of the moves used by the algorithm of [31]. Moreover, this subset explicitly excludes moves that are guaranteed not to improve the value of the score function. Therefore, it seems likely that Algorithm 4 performs with efficiency comparable or favorable to the algorithm of [31], which was already shown to be more efficient than the greedy equivalence search.

References

[1] Bernstein, D. I., Saeed, B., Squires, C., & Uhler, C. (2019). Ordering-based causal structure learning in the presence of latent variables. *Preprint available at arXiv:1910.09014.* 20

[2] Bouckaert, R. R. (1992). Optimizing causal orderings for generating DAGs from data. *Proceedings of the Eighth International Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc.* 2

[3] Chickering, D. M. (1995). A transformational characterization of equivalent Bayesian network structures. *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc.* 10, 25, 26, 30

[4] Chickering, D. M. (2002). Optimal structure identification with greedy search. *Journal of Machine Learning Research: 507-554.* 2, 7, 9, 26

[5] Learning Bayesian networks is NP-complete. *In D. Fisher and H. Lenz (Eds.), Learning from data: Artificial intelligence and statistics V: 121–130, Springer-Verlag.* 2

[6] Cooper, G. F. & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning 9.4: 309-347.* 2

[7] A. Dixit, O. Parnas, B. Li, J. Chen, C. P. Fulco, L. Jerby-Arnon, N. D. Marjanovic, D. Dionne, T. Burks, R. Raychowdhury, B. Adamson, T. M. Norman, E. S. Lander, J. S. Weissman, N. Friedman and A. Regev. (2016). Perturb-seq: dissecting molecular circuits with scalable single-cell RNA profiling of pooled genetic screens. *Cell 167.7: 1853-1866.* 17, 18

[8] Friedman, N., Linial, M., Nachman, I., & Peter, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology 7: 601–620.* 1

[9] Fukumizu, K., Gretton, A., Sun, X., & Schölkopf, B. (2008). Kernel measures of conditional dependence. *Advances in Neural Information Processing Systems.* 20

[10] M. Garber, N. Yosef, A Goren, R Raychowdhury, A. Thielke, M. Guttman, J. Robinson, B. Minie, N. Chevrier, Z. Itzhaki, R. Blecher-Gonen, C. Bornstein, D. Amann-Zalcenstein, A. Weiner, D. Friedrich, J. Meldrim, O. Ram, C. Chang, A. Gnirke, S. Fisher, N. Friedman, B. Wong, B. E. Bernstein, C. Nusbaum, N. Hacohen, A. Regev, and I. Amit. (2012). A high throughput chromatin immunoprecipitation approach reveals principles of dynamic gene regulation in mammals *Mol. Cell.* 447.5: 810-822. 18

[11] Gawrilow, E. and Joswig, M. (1997). Polymake: a framework for analyzing convex polytopes. *Polytopes, combinatorics and computation* (Oberwolfach, 1997), 43-73, DMV Sem., 29, Birkhäuser, Basel, 2000. MR1785292. 15

[12] George, A. (1973). Nested dissection of a regular finite element mesh. *SIAM J Numer Anal. 10.2: 345-363.* 12

[13] Gillispie, S. B. & Perlman, M. D. (2001). Enumerating Markov equivalence classes of acyclic digraph models. *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc.* 2, 10

[14] Haughton, Dominique M. A. (1988). On the choice of a model to fit data from an exponential family. *The Annals of Statistics* 16.1: 342-355. 26

[15] Kalisch, M. & Bühlmann, P. (2007). Estimating high-dimensional DAGs with the PC-algorithm. *Journal of Machine Learning Research 8 (2007): 613-636.* 11, 16, 31

[16] Meinshausen, N. and Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72.4 (2010): 417-473.* 15, 19

[17] Kalisch, M., Mächler, M., Colombo, D., Maathuis, M. H., & Bühlmann, P. (2011). Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software 47: 1-26.* 12

[18] Larrañaga, P., Kuijpers, C. M. H., Murga, R. H., & Yurramendi, Y. (1996). Learning Bayesian network structures by searching for the best ordering with genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 26.4: 487-493.* 2

[19] Lauritzen, S. L. (1996). Graphical Models *Oxford University Press.* 2

[20] Meek, C. (1997). Graphical Models: Selecting Causal and Statistical Models. *Diss. PhD thesis, CMU* 2

[21] Mohammadi, F., Uhler, C., Wang, C., & Yu, J. (2016). Generalized permutohedra from probabilistic graphical models. *SIAM Journal on Discrete Mathematics 32.1: 64-93.* 4, 5, 7, 23, 29

[22] Nandy, P., Hauser, A., & Maathuis, M. H. (2018). High-dimensional consistency in score-based and hybrid structure learning. *The Annals of Statistics 46.6A: 3151-3183.* 11, 13, 16, 17

[23] Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems. *Morgan Kaufman, San Mateo.* 3, 22

[24] Pearl, J. (2000). Causality: Models, Reasoning, and Inference. *Cambridge University Press, Cambridge.* 1

[25] Ramsey, J., Zhang, J., & Spirtes, P. L. (2006). Adjacency-faithfulness and conservative causal inference. *Proceedings of the Twenty-second Annual Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc.* 3

[26] Raskutti, G. & Uhler, C. (2018). Learning DAG models based on sparsest permutations. *Stat 7.1: e183.* 2, 3, 9, 12

[27] Robins, J. M., Hernán, M. A., & Brumback, B. (2000). Marginal structural models and causal inference in epidemiology. *Epidemiology 11.5: 550-560.* 1

[28] Singh, M. & Valtorta, M. (1993). An algorithm for the construction of Bayesian network structures from data. *Proceedings of the Ninth International Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc.* 2

[29] Spirtes, P., Glymour, C. M., & Scheines, R. (2001). Causation, Prediction, and Search. *MIT Press, Cambridge.* 1, 2, 3

[30] Squires, C., Wang, Y., & Uhler, C. (2019). Permutation-based causal structure learning with unknown intervention targets. *Preprint available at arXiv:1910.09007.* 20

[31] Teyssier, M. and Koller, D. (2005). Ordering-based search: A simple and effective algorithm for learning Bayesian networks. *Proceedings of the Twenty-first Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc.* 2, 8, 9, 35

[32] Tillman, R. E., Gretton, A., & Spirtes, P. (2009). Nonlinear directed acyclic structure learning with weakly additive noise model. *Advances in Neural Information Processing Systems 23.* 20

[33] Tinney, W. F. & Walker, J. W. (1967). Direct solutions of sparse network equations by optimally ordered triangular factorization. *Proceedings of the IEEE 55.11: 1801-1809.* 12

[34] Tsamardinos, I., Brown, L. E., & Aliferis, C. F. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning 65.1: 31-78.* 15

[35] Van de Geer, S. & Bühlmann, P. (2013). ℓ_0-penalized maximum likelihood for sparse DAGs. *The Annals of Statistics 41.2: 536–567.* 13, 16

[36] Uhler, C., Raskutti, G., Bühlmann, P., & Yu, B. (2013). Geometry of the faithfulness assumption in causal inference. *The Annals of Statistics 41.2: 436-463.* 2, 3

[37] Wang, Y., Solus, L., Yang, K. D., & Uhler, C. (2017). Permutation-based causal inference algorithms with interventions. *Neural Information Processing 31.* 18, 20

[38] Yang, K. D., Katcoff, A., & Uhler, C. (2018). Characterizing and learning equivalence classes of causal DAGs under interventions. *Proceedings of Machine Learning Research 80 (2018):5537-5546.* 20

Institutionen för Matematik, KTH, SE-100 44 Stockholm, Sweden
Email address: solus@kth.se

Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China, and Shanghai Qi Zhi Institute, Shanghai, China.
Email address: yuhaow@tsinghua.edu.cn

Department of Electrical Engineering and Computer Science, and Institute for Data, Systems and Society, Massachusetts Institute of Technology, Cambridge, MA, USA.
Email address: cuhler@mit.edu