*Submitted to the Annals of Statistics*

# SUPPLEMENT TO "LEARNING HIGH-DIMENSIONAL DIRECTED ACYCLIC GRAPHS WITH LATENT AND SELECTION VARIABLES"

By Diego Colombo, Marloes H. Maathuis, Markus Kalisch and Thomas S. Richardson

*ETH Zürich, ETH Zürich, ETH Zürich and University of Washington*

This supplementary document is organized as follows. Section 1 contains the proofs of the theoretical results presented in Sections 3 and 4 of [1]. Section 2 contains technical Lemmas used to show these results. In Section 3 we briefly explain the Anytime FCI algorithm [5] and our modification of it, called Adaptive Anytime FCI (AAFCI). Section 4 contains pseudocode for all algorithms considered in [1]. Finally, in Section 5 we present two additional examples to illustrate details of Algorithms 4.4 and 4.5 that are used in Steps 2 and 3 of the Really Fast Causal Inference (RFCI) algorithm.

## 1. Proofs.

### 1.1. *Proofs of results in Section 3 of [1].*

Lemma 1.1. *(Cf. Spirtes, Meek and Richardson [4], Lemma 13). Consider the oracle version of FCI, CFCI or SCFCI. Let the distribution of* $\mathbf{V} = \mathbf{X} \dot{\cup} \mathbf{L} \dot{\cup} \mathbf{S}$ *be faithful to a DAG* $\mathcal{G}$*, and let conditional independence information among all variables in* $\mathbf{X}$ *given* $\mathbf{S}$ *be the input to the algorithm. Let* $\mathcal{C}_2$ *be the graph resulting from Step 2 of the algorithm, and let* $X_i, X_j \in \mathbf{X}$ *with* $X_i \notin an(\mathcal{G}, X_j)$*. Then every vertex in D-SEP*$(X_i, X_j)$ *in* $\mathcal{G}$ *belongs to pds*$(\mathcal{C}_2, X_i, X_j)$*.*

Proof. A vertex $X_k$ belongs to D-SEP$(X_i, X_j)$ in $\mathcal{G}$ if and only if there is a sequence of vertices $U = \langle X_i, \dots, X_k \rangle$ in $\mathbf{X} \cap (\text{an}(\mathcal{G}, \{X_i \cup X_j\} \cup \mathbf{S}))$ such that (i) there is an inducing path relative to $\mathbf{X}$ given $\mathbf{S}$ between every pair of consecutive vertices on $U$, and (ii) with the exception of the endpoints every vertex on $U$ is not an ancestor of the vertices preceding and succeeding it in the sequence $U$ nor an ancestor of $\mathbf{S}$ (see [4], page 244). Condition (i) implies that every pair of consecutive vertices of $U$ are adjacent in the FCI-PAG, and hence also in $\mathcal{C}_2$, since the skeleton of $\mathcal{C}_2$ is in general a superset of the skeleton of the FCI-PAG.

If $U$ contains only 2 vertices, it follows that $X_k$ is adjacent to $X_i$ in $\mathcal{C}_2$, and hence $X_k \in \text{pds}(\mathcal{C}_2, X_i, X_j)$. If $U$ contains 3 or more vertices, consider all subsequences of three consecutive vertices. Each such subsequence is either an unshielded triple or a triangle in $\mathcal{C}_2$. By the definition of Possible-D-SEP, it follows that $X_k \in \text{pds}(\mathcal{C}_2, X_i, X_j)$ if all such unshielded triples are oriented as v-structures in $\mathcal{C}_2$. To see that the latter is indeed the case, consider a subsequence $\langle X_l, X_m, X_q \rangle$ of $U$ that is an unshielded triple in $\mathcal{C}_2$. By definition of D-SEP, there exist inducing paths between $X_l$ and $X_m$ and between $X_m$ and $X_q$ in $\mathcal{G}$ relative to $\mathbf{X}$ given $\mathbf{S}$, and $X_m \notin \text{an}(\mathcal{G}, \{X_l, X_q\} \cup \mathbf{S})$. Moreover, by definition of an unshielded triple, there is at least one set $\mathbf{Y}$ such that $X_l \perp\!\!\!\perp X_q | (\mathbf{Y} \cup \mathbf{S})$. Using Lemma 2.5, it follows that $X_m$ cannot belong to any such set $\mathbf{Y}$. Hence, $\langle X_l, X_m, X_q \rangle$ is an unambiguous triple, and is oriented as a v-structure in FCI, CFCI and SCFCI. □

LEMMA 1.2. *Consider the oracle version of FCI$_{path}$, CFCI$_{path}$ or SCFCI$_{path}$. Let the distribution of $\mathbf{V} = \mathbf{X} \dot{\cup} \mathbf{L} \dot{\cup} \mathbf{S}$ be faithful to a DAG $\mathcal{G}$, and let conditional independence information among all variables in $\mathbf{X}$ given $\mathbf{S}$ be the input to the algorithm. Let $\mathcal{C}_2$ be the graph resulting from Step 2 of the algorithm, and let $X_i, X_j \in \mathbf{X}$. If there is a set $\mathbf{Y}$ such that $X_i \perp\!\!\!\perp X_j | (\mathbf{Y} \cup \mathbf{S})$, then $X_i \perp\!\!\!\perp X_j | (\mathbf{Y}' \cup \mathbf{S})$, where $\mathbf{Y}'$ is the intersection of $\mathbf{Y}$ with the set of all vertices that lie on a path between $X_i$ and $X_j$ in $\mathcal{C}_2$.*

PROOF. We first note that the skeleton of $\mathcal{C}_2$ is identical for all mentioned versions of the FCI algorithm. Let $\mathbf{Y}$ satisfy $X_i \perp\!\!\!\perp X_j | (\mathbf{Y} \cup \mathbf{S})$, or in other words, let $\mathbf{Y}$ block all paths between $X_i$ and $X_j$ in the FCI-PAG. Since a path can only be blocked by vertices on the path, and since the skeleton of $\mathcal{C}_2$ is a superset of the skeleton of the FCI-PAG, $\mathbf{Y}'$ also blocks all these paths. □

LEMMA 1.3. *Consider the oracle version of SCFCI or SCFCI$_{path}$. Let the distribution of $\mathbf{V} = \mathbf{X} \dot{\cup} \mathbf{L} \dot{\cup} \mathbf{S}$ be faithful to a DAG $\mathcal{G}$, and let conditional independence information among all variables in $\mathbf{X}$ given $\mathbf{S}$ be the input to the algorithm. Let $\mathcal{C}_3$ be the graph resulting after Step 3 of the given algorithm, and let $\langle X_i, X_j, X_k \rangle$ be an unshielded triple in $\mathcal{C}_3$. Then either $X_j$ is in every set $\mathbf{Y}$ that satisfies $X_i \perp\!\!\!\perp X_k | (\mathbf{Y} \cup \mathbf{S})$, or it is in no such set $\mathbf{Y}$. Hence, $\langle X_i, X_j, X_k \rangle$ is unambiguous.*

PROOF. Let $\langle X_i, X_j, X_k \rangle$ be an unshielded triple in $\mathcal{C}_3$. Then there is a set $\mathbf{S_{ik}}$ such that $X_i \perp\!\!\!\perp X_k | (\mathbf{S_{ik}} \cup \mathbf{S})$. Moreover, in the underlying DAG there exists a d-connecting path $\Pi_{ij}$ between $X_i$ and $X_j$ given $(\mathbf{S_{ik}} \setminus \{X_j\}) \cup \mathbf{S}$, as well as a d-connecting path $\Pi_{jk}$ between $X_j$ and $X_k$ given $(\mathbf{S_{ik}} \setminus \{X_j\}) \cup \mathbf{S}$.

First, suppose that the paths $\Pi_{ij}$ and $\Pi_{jk}$ are both into $X_j$. Then using Lemma 2.5 with $\mathcal{T} = \{\Pi_{ij}, \Pi_{jk}\}$ implies that $X_j$ must be in no set $\mathbf{Y}$ that satisfies $X_i \perp\!\!\!\perp X_k | (\mathbf{Y} \cup \mathbf{S})$. Next, assume that at least one path is out of $X_j$. Again using Lemma 2.5 with $\mathcal{T} = \{\Pi_{ij}, \Pi_{jk}\}$, it follows that $X_j$ is in every set $\mathbf{Y}$ that satisfies $X_i \perp\!\!\!\perp X_k | (\mathbf{Y} \cup \mathbf{S})$. □

Proof of Theorem 3.1 of [1]. Consider one of the oracle versions of $\text{FCI}_{\text{path}}$, CFCI, $\text{CFCI}_{\text{path}}$, SCFCI and $\text{SCFCI}_{\text{path}}$. By Lemmas 1.1 and 1.2, it follows that for every pair of vertices $X_i, X_j \in \mathbf{X}$ for which there exists a set $\mathbf{Y} \subseteq \mathbf{X} \setminus \{X_i, X_j\}$ so that $X_i \perp\!\!\!\perp X_j | (\mathbf{Y} \cup \mathbf{S})$, the algorithm finds a separating set. Hence, the skeleton and the separation sets of the algorithm are correct. This completes the proof for $\text{FCI}_{\text{path}}$, CFCI and $\text{CFCI}_{\text{path}}$, since the orientation rules in Steps 4 and 5 of these algorithms are identical to those of the original FCI algorithm.

We are left to consider the conservative orientation rules of SCFCI and $\text{SCFCI}_{\text{path}}$. Note that the final skeleton of the FCI-PAG is found after Step 3 of the algorithm. Since all edges present in this skeleton correspond to inducing paths relative to $\mathbf{X}$ given $\mathbf{S}$, it follows from Lemma 1.3 that all unshielded triples are unambiguous. Hence, the orientation rules of SCFCI and $\text{SCFCI}_{\text{path}}$ are identical to those of the original FCI algorithm. □

The proof of Lemma 3.1 of [1] is related to the proof of the "Base case" of Theorem 5 of [4]. Condition (a1) is used in Lemma 2.3 in the first part of the proof. Condition (a2) describes conditional dependence relationships that are used in Lemma 2.5 in the second part of the proof, which requires the existence of d-connecting paths between $X_i$ and $X_j$ and between $X_j$ and $X_k$ given $(\mathbf{S}_{ik} \setminus \{X_j\}) \cup \mathbf{S}$.

Proof of Lemma 3.1 of [1]. Suppose that $X_j \in \mathbf{S_{ik}}$. Combining assumption (a1) with Lemma 2.3 implies that $X_j \in \mathbf{S_{ik}}$ is an ancestor of $\{X_i, X_k\} \cup \mathbf{S}$ in $\mathcal{G}$.

Suppose that $X_j \notin \mathbf{S_{ik}}$. Let $\Pi_{ij}$ and $\Pi_{jk}$ be two arbitrary d-connecting paths given $\mathbf{S_{ik}} \cup \mathbf{S}$ between $X_i$ and $X_j$ and between $X_j$ and $X_k$, respectively. The existence of such paths follows from assumption (a2). We first show that $\Pi_{ij}$ and $\Pi_{jk}$ are into $X_j$ in $\mathcal{G}$. Suppose, contrary to the claim, that at least one of the paths is out of $X_j$. Then Lemma 2.5 with $\mathcal{T} = \{\Pi_{ij}, \Pi_{jk}\}$ implies that there is a d-connecting path between $X_i$ and $X_k$ given $\mathbf{S_{ik}} \cup \mathbf{S}$, which contradicts assumption (a1). Hence, $\Pi_{ij}$ and $\Pi_{jk}$ are both into $X_j$ in $\mathcal{G}$.

We now prove that $X_j \notin \mathbf{S_{ik}}$ implies that $X_j \notin \text{an}(\mathcal{G}, \{X_i, X_k\} \cup \mathbf{S})$. Suppose first, for a contradiction, that $X_j \in \text{an}(\mathcal{G}, \mathbf{S})$. Then there is a d-connecting path between $X_i$ and $X_k$ given $\mathbf{S_{ik}} \cup \mathbf{S}$ in $\mathcal{G}$ using Lemma 2.5

with $\mathcal{T} = \{\Pi_{ij}, \Pi_{jk}\}$, which contradicts assumption (a1). Next, suppose that $X_j \in \text{an}(\mathcal{G}, X_k)$. Then there is a directed path $\Pi'_{jk}$ from $X_j$ to $X_k$ in $\mathcal{G}$. We distinguish two cases: (i) $X_j \notin \text{an}(\mathcal{G}, \mathbf{S_{ik}} \cup \mathbf{S})$ and (ii) $X_j \in \text{an}(\mathcal{G}, \mathbf{S_{ik}} \cup \mathbf{S})$. In case (i) the path $\Pi'_{jk}$ d-connects $X_j$ and $X_k$ given $\mathbf{S_{ik}} \cup \mathbf{S}$, since all of the vertices on the path are non-colliders and none of them are in $\mathbf{S_{ik}} \cup \mathbf{S}$. Hence, using Lemma 2.5 with $\mathcal{T} = \{\Pi_{ij}, \Pi'_{jk}\}$ implies that there is a d-connecting path between $X_i$ and $X_k$ given $\mathbf{S_{ik}} \cup \mathbf{S}$, which contradicts assumption (a1). In case (ii) there is a d-connecting path between $X_i$ and $X_k$ given $\mathbf{S_{ik}} \cup \mathbf{S}$ (use Lemma 2.5 with $\mathcal{T} = \{\Pi_{ij}, \Pi_{jk}\}$), which again contradicts assumption (a1). By symmetry, it also follows that $X_j \notin \text{an}(\mathcal{G}, X_i)$. □

The proof of Lemma 3.2 of [1] is related to the proof of "Case 5" of Theorem 5 of [4]. Condition (a1) describes the absence of an edge between $X_i$ and $X_k$. Condition (a2) describes conditional dependence relationships between successive vertices $X_h$ and $X_{h+1}$ on the discriminating path that ensure the existence of inducing paths between $X_h$ and $X_{h+1}$ relative to $\mathbf{S}_{ik} \setminus \{X_h, X_{h+1}\}$, which is needed to apply Lemma 2.4.

Proof of Lemma 3.2 of [1]. Let $X_h$ and $X_{h+1}$ be two adjacent vertices on $\pi_{ik}$ between $X_i$ and $X_j$ (not including $X_i$ and $X_j$) and let $X_{h_1}$ be the vertex on $\pi_{ik}$ that is adjacent to $X_i$. We first establish the following results: (d1) there exists a d-connecting path in $\mathcal{G}$ between $X_h$ and $X_{h+1}$ given $(\mathbf{S_{ik}} \setminus \{X_h, X_{h+1}\}) \cup \mathbf{S}$ that is into $X_h$ and into $X_{h+1}$, and there is a d-connecting path between $X_i$ and $X_{h_1}$ given $(\mathbf{S_{ik}} \setminus \{X_{h_1}\}) \cup \mathbf{S}$ that is into $X_{h_1}$, as well as a d-connecting path between $X_l$ and $X_j$ given $(\mathbf{S_{ik}} \setminus \{X_l, X_j\}) \cup \mathbf{S}$ that is into $X_l$; and (d2) $X_h \in \text{an}(\mathcal{G}, \mathbf{S_{ik}})$.

To see that (d1) holds, note that assumption (a2) implies the existence of an inducing path between $X_h$ and $X_{h+1}$ relative to $\mathbf{S_{ik}} \setminus \{X_h, X_{h+1}\}$ given $\mathbf{S}$ in $\mathcal{G}$. It follows from Lemma 2.4 that such an inducing path is into $X_h$ and into $X_{h+1}$, since $X_h \notin \text{an}(\mathcal{G}, \{X_{h+1}\} \cup \mathbf{S})$ and $X_{h+1} \notin \text{an}(\mathcal{G}, \{X_h\} \cup \mathbf{S})$ by assumption (a3). Hence, Lemma 2.2 implies the existence of a d-connecting path $\Pi_{h,h+1}$ in $\mathcal{G}$ given $(\mathbf{S_{ik}} \setminus \{X_h, X_{h+1}\}) \cup \mathbf{S}$ that is into $X_h$ and into $X_{h+1}$. The other claims can be shown analogously.

To see that (d2) holds, suppose that some vertex on $\pi_{ik}$ between $X_i$ and $X_j$ is not an ancestor of $\mathbf{S_{ik}}$ in $\mathcal{G}$. Let $X_w$ be the closest such vertex to $X_i$ on $\pi_{ik}$. By hypothesis, any vertex $X_h$ between $X_i$ and $X_w$ on $\pi_{ik}$ is an ancestor of $\mathbf{S_{ik}}$ in $\mathcal{G}$. By combining this with (d1), it follows by Lemma 2.5 that there exists a d-connecting path $\Pi_{iw}$ between $X_i$ and $X_w$ given $(\mathbf{S_{ik}} \setminus \{X_i, X_w\}) \cup \mathbf{S}$. By hypothesis (a3), $X_w \in \text{an}(\mathcal{G}, X_k)$. Since, by construction $X_w \notin an(\mathcal{G}, \mathbf{S_{ik}})$, and by (a3) $X_w \notin an(\mathcal{G}, \mathbf{S})$, there exists a directed path $\Pi'_{wk}$ d-connecting $X_w$ and $X_k$ given $(\mathbf{S_{ik}} \setminus \{X_w\}) \cup \mathbf{S}$ in $\mathcal{G}$. Since $X_w \notin \mathbf{S_{ik}} \cup \mathbf{S}$, it follows

by Lemma 2.5 using $\mathcal{T} = \{\Pi_{iw}, \Pi'_{wk}\}$ that there exists a d-connecting path between $X_i$ and $X_k$ given $\mathbf{S_{ik}} \cup \mathbf{S}$, which contradicts assumption (a1).

We now prove claims (b1) and (b2) of Lemma 3.2 of [1]:

(b1) Let $X_j \in \mathbf{S_{ik}}$.
We first prove that $X_j \in \text{an}(\mathcal{G}, \{X_k\} \cup \mathbf{S})$. Suppose, contrary to the claim, that $X_j \notin \text{an}(\mathcal{G}, \{X_k\} \cup \mathbf{S})$. Assumption (a2) implies the existence of an inducing path between $X_j$ and $X_k$ relative to $\mathbf{S_{ik}} \setminus \{X_j\}$ given $\mathbf{S}$. Since $X_j \notin \text{an}(\mathcal{G}, \{X_k\} \cup \mathbf{S})$ by hypothesis, such a path must be into $X_j$ by Lemma 2.4. Assumption (a2) also implies the existence of an inducing path between $X_j$ and $X_l$ relative to $\mathbf{S_{ik}} \setminus \{X_j, X_l\}$ given $\mathbf{S}$. One can show that such a path must be into $X_j$ and into $X_l$ by combining Lemma 2.4 with the following two facts: (i) $X_j \notin \text{an}(\mathcal{G}, \{X_l\} \cup \mathbf{S})$ because $X_l \in \text{an}(\mathcal{G}, X_k)$ by assumption (a3) and $X_j \notin \text{an}(\mathcal{G}, \{X_k\} \cup \mathbf{S})$ by hypothesis, and (ii) $X_l \notin \text{an}(\mathcal{G}, \{X_j\} \cup \mathbf{S}\})$ by assumption (a3). Lemma 2.2 now implies the existence of a d-connecting path between $X_j$ and $X_k$ given $(\mathbf{S_{ik}} \setminus \{X_j\}) \cup \mathbf{S}$ that is into $X_j$ and the existence of a d-connecting path between $X_l$ and $X_j$ given $(\mathbf{S_{ik}} \setminus \{X_j, X_l\}) \cup \mathbf{S}$ that is into $X_l$ and into $X_j$. By combining this with (d1), (d2) and $X_j \in \mathbf{S_{ik}}$ by the premise of (b1), Lemma 2.5 implies that $X_i$ and $X_k$ are d-connected given $\mathbf{S_{ik}} \cup \mathbf{S}$, which contradicts assumption (a1). Hence $X_j$ is an ancestor of $\{X_k\} \cup \mathbf{S}$ in $\mathcal{G}$.
We now prove that $X_k \notin \text{an}(\mathcal{G}, \{X_j\} \cup \mathbf{S})$. By assumption (a3), combining that $X_h \in \text{an}(\mathcal{G}, X_k)$ and that $X_h \notin \text{an}(\mathcal{G}, \mathbf{S})$, it follows that $X_k \notin \text{an}(\mathcal{G}, \mathbf{S})$. Hence, we only need to prove that $X_k \notin \text{an}(\mathcal{G}, X_j)$. Assume, contrary to the claim, that $X_k \in \text{an}(\mathcal{G}, X_j)$. Then $X_l \in \text{an}(\mathcal{G}, X_j)$ since $X_l \in \text{an}(\mathcal{G}, X_k)$ by assumption (a3). This is a contradiction to another part of assumption (a3), namely that $X_l \notin \text{an}(\mathcal{G}, \{X_j\} \cup \mathbf{S})$. Hence $X_k \notin \text{an}(\mathcal{G}, X_j)$.

(b2) Let $X_j \notin \mathbf{S_{ik}}$.
Combining claims (d1) and (d2) with Lemma 2.5, we can infer the existence of a d-connecting path $\Pi_{ij}$ between $X_i$ and $X_j$ given $\mathbf{S_{ik}} \cup \mathbf{S}$ in $\mathcal{G}$. Assumption (a2) implies the existence of a d-connecting path $\Pi_{jk}$ between $X_j$ and $X_k$ given $\mathbf{S_{ik}} \cup \mathbf{S}$ in $\mathcal{G}$. We first show that both paths must be into $X_j$. Suppose that at least one of the paths is out of $X_j$. Then Lemma 2.5 with $\mathcal{T} = \{\Pi_{ij}, \Pi_{jk}\}$ implies the existence of a d-connecting path between $X_i$ and $X_k$ given $\mathbf{S_{ik}} \cup \mathbf{S}$, which contradicts assumption (a1).
We now prove that $X_j \notin \text{an}(\mathcal{G}, \{X_l, X_k\} \cup \mathbf{S})$. First, suppose that $X_j \in \text{an}(\mathcal{G}, X_k)$. Then there exists a directed path $\Pi'_{jk}$ from $X_j$ to $X_k$ in $\mathcal{G}$. We distinguish two cases: (i) $X_j \notin \text{an}(\mathcal{G}, \mathbf{S_{ik}} \cup \mathbf{S})$ and (ii)

$X_j \in \text{an}(\mathcal{G}, \mathbf{S_{ik}} \cup \mathbf{S})$. In case (i), the path $\Pi'_{jk}$ d-connects $X_j$ and $X_k$ given $\mathbf{S_{ik}} \cup \mathbf{S}$ in $\mathcal{G}$, since all of the vertices on this path are non-colliders and none of them are in $\mathbf{S_{ik}} \cup \mathbf{S}$. It then follows from Lemma 2.5 with $\mathcal{T} = \{\Pi_{ij}, \Pi'_{jk}\}$ that there exists a d-connecting path between $X_i$ and $X_k$ given $\mathbf{S_{ik}} \cup \mathbf{S}$, which contradicts assumption (a1). In case (ii), Lemma 2.5 with $\mathcal{T} = \{\Pi_{ij}, \Pi_{jk}\}$ implies the existence of a d-connecting path between $X_i$ and $X_k$ given $\mathbf{S_{ik}} \cup \mathbf{S}$ in $\mathcal{G}$, which again contradicts assumption (a1). Hence, $X_j \notin \text{an}(\mathcal{G}, X_k)$. Next, suppose that $X_j \in \text{an}(\mathcal{G}, X_l)$. Since $X_l \in \text{an}(\mathcal{G}, X_k)$ by assumption (a3), this implies that $X_j \in \text{an}(\mathcal{G}, X_k)$, which contradicts the previous result. Finally, suppose that $X_j \in \text{an}(\mathcal{G}, \mathbf{S})$. Then $X_j \in \text{an}(\mathcal{G}, \mathbf{S_{ik}} \cup \mathbf{S})$ and the same contradiction as in case (ii) above is reached.
The fact that $X_k \notin \text{an}(\mathcal{G}, \{X_j\} \cup \mathbf{S})$ follows as in the second part of the proof of (b1).

□

Proof of Theorem 3.2 of [1]. The skeleton found in Step 1 of the RFCI algorithm satisfies conditions (i) and (ii') in Definition 3.2 of [1] by construction.

The correctness of the v-structures in Step 2 of the algorithm is given by Lemma 3.1 of [1]. While testing both edges in an unshielded triple to check if it could be oriented as a v-structure, some edges could be deleted, leading to an update of the skeleton. Although the skeleton can change during Step 2, at the end of this step all unshielded triples in the graph are labeled as colliders or non-colliders. Moreover, edges are only deleted because of conditional independence relationships which is in accordance with condition (i) of Definition 3.2 of [1].

We now consider the further orientations in Step 3 of the algorithm. Correctness of the orientation rules (R1)-(R3) in Algorithm 4.5 follows from the work of [4], since all unshielded triples have been correctly assessed previously. Correctness of the new orientation rule for discriminating paths follows directly from Lemma 3.2 of [1]. Line 15 of Algorithm 4.5 can remove edges and possible new unshielded triples are labeled as v-structures or not. Finally, correctness of the orientation rules (R5)-(R10) follows from the work of [6], since all unshielded triples have been correctly assessed. By induction on the number of applications of the orientation rules in the repeated loop of Algorithm 4.5, lines 1-31, all further orientations are also correct. □

proof of theorem 3.3 of [1]. We first focus on the skeletons of $\mathcal{C}'$ and $\mathcal{C}''$. Let $X_i, X_j \in \mathbf{X}$, and suppose that Scenario (S1) holds for $X_i$ and $X_j$.

Then $X_i$ and $X_j$ are not d-separated in $\mathcal{G}$ by $\mathbf{Y} \cup \mathbf{S}$ for any subset $\mathbf{Y}$ of $\mathrm{Pds}(i,j)$ or of $\mathrm{Pds}(j,i)$. Hence, there is no subset $\mathbf{Y}$ of $\mathbf{X}$ such that $X_i$ and $X_j$ are d-separated by $\mathbf{Y} \cup \mathbf{S}$ in $\mathcal{G}$. As a result, $X_i$ and $X_j$ are adjacent in both $\mathcal{C}'$ and $\mathcal{C}''$.

Now suppose that Scenario (S3) holds for $X_i$ and $X_j$. Then there is a subset $\mathbf{Y}$ of $\mathrm{Adj}(i,j)$ or of $\mathrm{Adj}(j,i)$ such that $X_i$ and $X_j$ are conditionally independent given $\mathbf{Y} \cup \mathbf{S}$. This conditional independence is found in Step 1 of the FCI and RFCI algorithms. Hence, $X_i$ and $X_j$ are not adjacent in both $\mathcal{C}'$ and $\mathcal{C}''$.

Thus, if for every pair of vertices in $\mathbf{X}$ either (S1) or (S3) holds, then the skeletons of $\mathcal{C}'$ and $\mathcal{C}''$ must be identical. Conversely, if the adjacency of $X_i$ and $X_j$ is different in $\mathcal{C}'$ than in $\mathcal{C}''$, then Scenario (S2) must hold for $X_i$ and $X_j$.

We now show that identical skeletons lead to identical edge orientations. Consider the point in the RFCI algorithm just after its last edge deletion (this could be in Step 1 of the algorithm, or after orienting a v-structure or a discriminating path), and call the corresponding graph $\mathcal{C}'^*$. By assumption the skeleton of $\mathcal{C}'^*$ is identical to the skeleton of $\mathcal{C}''$. Any edges that may be oriented in $\mathcal{C}'^*$ are oriented correctly, since the orientation rules of RFCI are sound. Moreover, after RFCI's last edge deletion, the orientation rules of RFCI lead to the same orientations as those of FCI: RFCI's orientation rules for unshielded triples and discriminating paths are sound and thus lead to the same orientations, even though, for a given triple, RFCI and FCI may use different sepsets; see Lemmas 3.1 and 3.2 of [1]. No additional edges are deleted due to the extra conditional independence tests performed by RFCI since, by hypothesis, $\mathcal{C}'^*$ and $\mathcal{C}''$ have the same skeleton. Since the orientation rules of FCI are complete, it can then be shown inductively that any edge orientation performed by FCI that has not been performed by RFCI prior to the last edge deletion, will be performed subsequently by RFCI.

Hence, if for every pair of vertices in $\mathbf{X}$ either (S1) or (S3) holds, then $\mathcal{C}'$ and $\mathcal{C}''$ are identical. Conversely, if $\mathcal{C}'$ and $\mathcal{C}''$ are not identical, there must be a difference in their skeletons. In that case, the skeleton of $\mathcal{C}'$ must be a superset of the skeleton of $\mathcal{C}''$, since the meaning of an edge is weaker in the output of RFCI than in the output of FCI. □

PROOF OF THEOREM 3.4 OF [1]. The existence of the edge $X_i \ast\!\!\ast X_j$ in the output of RFCI but not in the output of FCI implies that Scenario (S2) holds for $X_i$ and $X_j$, or equivalently, that there are paths $\pi(i,j)$ and $\pi(j,i)$ satisfying conditions (c1)-(c3) that are equivalent to Scenario (S2) (see Section 3.4 of [1]).

We prove claim (i) by contradiction. Suppose first that $X_j \in \mathrm{an}(\mathcal{G}, X_i)$. Consider a path $\pi(i,j)$ satisfying conditions (c1)-(c3). Then we have in particular that every collider on $\pi(i,j)$ has a descendant in $\{X_i, X_j\} \cup \mathbf{S}$. Since $X_j \in \mathrm{an}(\mathcal{G}, X_i)$ by assumption, this implies that every collider has a descendant in $\{X_i\} \cup \mathbf{S}$. Moreover, every member of $\mathrm{Adj}(i,j) \cup \mathbf{S}$ on $\pi(i,j)$ is a collider on the path, and there is a member of $(\mathbf{X} \setminus \{X_i, X_j\}) \setminus \mathrm{Adj}(i,j)$ that is a non-collider on $\pi(i,j)$. Let $X_q$ be the closest such non-collider to $X_i$. Then the subpath from $X_i$ to $X_q$ is an inducing path relative to $\mathbf{X}$ given $\mathbf{S}$ (because by construction all vertices in $\mathbf{X}$ on the path are colliders and thus ancestors of $X_i$). This means that $X_q$ is adjacent to $X_i$ in the FCI-PAG. But this is a contradiction, since by construction $X_q \notin \mathrm{Adj}(j,i)$ and vertices that are not adjacent after Step 1 of the FCI algorithm are certainly not adjacent in the FCI-PAG. Hence, $X_j \notin \mathrm{an}(\mathcal{G}, X_i)$. By interchanging $i$ and $j$, it also follows that $X_i \notin \mathrm{an}(\mathcal{G}, X_j)$.

The proof of claim (ii) follows directly from (i) and the soundness of the RFCI algorithm. □

1.2. *Proofs of results in Section 4 of [1].* The proofs of Theorem 4.1 of [1] and 4.2 of [1] rely on the following general Lemma that provides a bound for the probability that an error occurs when testing partial correlations up to a given order.

LEMMA 1.4. *Assume (A1)-(A5), but replace (A3) by the assumption that the maximum order $m_n$ of the partial correlation tests satisfies $m_n = O(n^{1-b})$. Denote by $A(m_n, \alpha_n)$ the event that at least one error occurs when testing all partial correlations among variables in $\mathbf{W} = (W_1, \ldots, W_{p_n})$ up to and including order $m_n$ with tuning parameter $\alpha_n = 2\big(1 - \Phi(n^{1/2} c_n/2)\big)$, where $c_n$ is the lower bound from (A5). Then*

$$\mathbb{P}[A(m_n, \alpha_n)] \leq O(\exp(-Cn^{1-2d})), \tag{1.1}$$

*where $d$ is from (A5).*

PROOF. Let $E_{n,i,j|\mathbf{Y}}$ be the event that an error occurs when testing partial correlation between vertices $W_i$ and $W_j$ given a conditioning set $\mathbf{Y} \subseteq \mathbf{W} \setminus$

$\{W_i, W_j\}$. We have

$$\begin{aligned}\mathbb{P}[A(m_n,\alpha_n)] &\le \mathbb{P}\big[\sum_{i,j,|\mathbf{Y}|\le m_n} E_{n,i,j|\mathbf{Y}}\big]\\ &\le O\big(p_n^{m_n+2}\big)\sup_{i,j,|\mathbf{Y}|\le m_n}\mathbb{P}[E_{n,i,j|\mathbf{Y}}]\\ &\le O\big(p_n^{m_n+2}\big)\Big(\sup_{i,j,|\mathbf{Y}|\le m_n}\mathbb{P}[E^I_{n,i,j|\mathbf{Y}}]+\sup_{i,j,|\mathbf{Y}|\le m_n}\mathbb{P}[E^{II}_{n,i,j|\mathbf{Y}}]\Big)\end{aligned}\tag{1.2}$$

where the first sum is taken over $i\neq j\in\{1,\dots,p_n\}$ and $\mathbf{Y}\subseteq\mathbf{W}\setminus\{W_i,W_j\}$ with $|\mathbf{Y}|\le m_n$, the second inequality follows from the union bound, and the last inequality follows from the split of $E_{n,i,j|\mathbf{Y}}$ into the corresponding type I and type II errors (see (4.1) of [1]):

$$\begin{aligned}E^I_{n,i,j|\mathbf{Y}} &= \{\sqrt{n-|\mathbf{Y}|-3}|\hat{z}_{n;i,j|\mathbf{Y}}|>\Phi^{-1}(1-\alpha_n/2)\mid z_{n;i,j|\mathbf{Y}}=0\},\\ E^{II}_{n,i,j|\mathbf{Y}} &= \{\sqrt{n-|\mathbf{Y}|-3}|\hat{z}_{n;i,j|\mathbf{Y}}|\le\Phi^{-1}(1-\alpha_n/2)\mid z_{n;i,j|\mathbf{Y}}\neq 0\}.\end{aligned}$$

We now bound the two terms of (1.2). Using the definition of $\alpha_n$, we have

$$\begin{aligned}&\sup_{i,j,|\mathbf{Y}|\le m_n}\mathbb{P}\Big[E^I_{n,i,j|\mathbf{Y}}\Big]\\ &=\sup_{i,j,|\mathbf{Y}|\le m_n}\mathbb{P}\Big[|\hat{z}_{n;i,j|\mathbf{Y}}|>\Phi^{-1}(1-\alpha_n/2)(n-|\mathbf{Y}|-3)^{-1/2}\;\Big|\;z_{n;i,j|\mathbf{Y}}=0\Big]\\ &=\sup_{i,j,|\mathbf{Y}|\le m_n}\mathbb{P}\Big[|\hat{z}_{n;i,j|\mathbf{Y}}-z_{n;i,j|\mathbf{Y}}|>\frac{c_n}{2}\big(n/(n-|\mathbf{Y}|-3)\big)^{1/2}\Big]\\ &\le\sup_{i,j,|\mathbf{Y}|\le m_n}\mathbb{P}\Big[|\hat{z}_{n;i,j|\mathbf{Y}}-z_{n;i,j|\mathbf{Y}}|>\frac{c_n}{2}\Big],\end{aligned}$$

Moreover, for large $n$, we have

$$\begin{aligned}&\sup_{i,j,|\mathbf{Y}|\le m_n}\mathbb{P}\Big[E^{II}_{i,j|\mathbf{Y}}\Big]=\\ &=\sup_{i,j,|\mathbf{Y}|\le m_n}\mathbb{P}\Big[|\hat{z}_{n;i,j|\mathbf{Y}}|\le\Phi^{-1}(1-\alpha_n/2)(n-|\mathbf{Y}|-3)^{-1/2}\;\Big|\;z_{n;i,j|\mathbf{Y}}\neq 0\Big]\\ &=\sup_{i,j,|\mathbf{Y}|\le m_n}\mathbb{P}\Big[|\hat{z}_{n;i,j|\mathbf{Y}}|\le\frac{c_n}{2}\big(n/(n-|\mathbf{Y}|-3)\big)^{1/2}\;\Big|\;z_{n;i,j|\mathbf{Y}}\neq 0\Big]\\ &\le\sup_{i,j,|\mathbf{Y}|\le m_n}\mathbb{P}\Big[|\hat{z}_{n;i,j|\mathbf{Y}}-z_{n;i,j|\mathbf{Y}}|\ge c_n-\frac{c_n}{2}\big(n/(n-|\mathbf{Y}|-3)\big)^{1/2}\Big]\\ &\le\sup_{i,j,|\mathbf{Y}|\le m_n}\mathbb{P}\Big[|\hat{z}_{n;i,j|\mathbf{Y}}-z_{n;i,j|\mathbf{Y}}|>\frac{c_n}{3}\Big].\end{aligned}$$

The first inequality in the above display follows from $\inf_{i,j,|\mathbf{Y}|\leq m_n}\{|z_{n;i,j|\mathbf{Y}}| : z_{n;i,j|\mathbf{Y}} \neq 0\} \geq c_n$, which holds because of assumption (A5) and $|g(\rho)| \geq |\rho|$ for every $\rho$, together with the fact that $c_n \geq \frac{c_n}{2}\big(n/(n-|\mathbf{Y}|-3)\big)^{1/2}$ for large $n$. The second inequality follows since $n/(n-|\mathbf{Y}|-3) \to 1$ as $n \to \infty$ since $m_n = O(n^{1-b})$.

Next, using Lemma 2.1 it follows that both $\sup_{i,j,|\mathbf{Y}|\leq m_n} \mathbb{P}[E^I_{n,i,j|\mathbf{Y}}]$ and $\sup_{i,j,|\mathbf{Y}|\leq m_n} \mathbb{P}[E^{II}_{n,i,j|\mathbf{Y}}]$ are bounded above by

$$\begin{aligned}
&\sup_{i,j,|\mathbf{Y}|\leq m_n} \mathbb{P}\Big[|\hat{z}_{n;i,j|\mathbf{Y}} - z_{n;i,j|\mathbf{Y}}| > \frac{c_n}{3}\Big] \\
&\quad \leq O(n-m_n)\Big(\exp\big(-(n-m_n)\frac{c_n^2}{18L^2}\big) + \exp\big(-C(n-m_n)\big)\Big) \\
&\quad \leq O(n-m_n)\exp\big(-C_2 c_n^2(n-m_n)\big),
\end{aligned}$$

where $C_2 = \min(1/(18L^2), C)$ and we used that $c_n^2 < 1$. By plugging this into (1.2) and using assumptions (A2) and (A5), we get

$$\begin{aligned}
\mathbb{P}[A(m_n,\alpha_n)] &\leq O\Big(p_n^{m_n+2}(n-m_n)\exp\Big(-C_2c_n^2(n-m_n)\Big)\Big) \\
&\leq O\Big(n^{a(m_n+2)+1}\exp\Big(-C_2 n^{-2d}(n-m_n)\Big)\Big) \\
&\leq O\Big(\exp\Big(\big(a(m_n+2)+1\big)\log(n) - C_2\big(n^{1-2d} - m_n n^{-2d}\big)\Big)\Big) \\
&\leq O(\exp(-C_3 n^{1-2d})),
\end{aligned}$$

where the final step follows since $n^{1-2d}$ dominates all other terms in the exponential function. □

PROOF OF THEOREM 4.1 OF [1]. Assume (A1)-(A5), suppose that we run the oracle RFCI algorithm, and denote its output by $\mathcal{C}'_n$. Due to assumption (A3), the algorithm only considers conditional independence relationships up to order $q_n = O(n^{1-b})$.

Now suppose that we run the sample version of the RFCI algorithm with tuning parameter $\alpha_n$, and denote its output by $\hat{\mathcal{C}}_n(\alpha_n)$. On the event $A^C(q_n,\alpha_n)$, where $A(q_n,\alpha_n)$ is defined as in Lemma 1.4, the sample version coincides with the oracle version in terms of the skeleton, the stored sepsets, and therefore also the edge orientations. Hence,

$$\mathbb{P}[\hat{\mathcal{C}}_n(\alpha_n) = \mathcal{C}'_n] \geq 1 - \mathbb{P}[A(q_n,\alpha_n)] \geq 1 - O(\exp(-Cn^{1-2d})) \to 1$$

as $n \to \infty$, where the last inequality and the constant $C$ come from Lemma 1.4. □

PROOF OF THEOREM 4.2 OF [1]. Assume (A1)-(A5) with (A3) replaced by (A3'). Suppose that we run the oracle version of FCI, $\text{FCI}_{\text{path}}$, CFCI, $\text{CFCI}_{\text{path}}$, SCFCI or $\text{SCFCI}_{\text{path}}$. By Theorem 3.1 of [1], the outputs of all these algorithms are identical, and we denote it by $\mathcal{C}_n$. Due to assumption (A3') and the fact that our alternative definition of Possible-D-SEP (see Definition 3.4 of [1]) and the conservative orientation of v-structures in CFCI and SCFCI can only decrease the size of Possible-D-SEP when compared to the original FCI algorithm, each algorithm only considers conditional independence relationships up to order $r_n = O(n^{1-b})$.

Now suppose that we run the sample version of one of these algorithms, using tuning parameter $\alpha_n$. We denote its output by $\mathcal{C}_n^*(\alpha_n)$. On the event $A^C(r_n, \alpha_n)$, where $A(r_n, \alpha_n)$ is defined as in Lemma 1.4, the sample version coincides with the corresponding oracle version in terms of the skeleton, the stored sepsets, and therefore also the edge orientations. Hence,

$$\mathbb{P}[\mathcal{C}_n^*(\alpha_n) = \mathcal{C}_n] \geq 1 - \mathbb{P}[A(r_n, \alpha_n)] \geq 1 - O(\exp(-Cn^{1-2d})) \to 1 \tag{1.3}$$

as $n \to \infty$, where the last inequality and the constant $C$ come from Lemma 1.4. □

## 2. Useful Lemmas.

LEMMA 2.1. *Assume (A4) and (A5). Then for any $\gamma \in (0, 2)$ there is a constant $0 < C < \infty$ such that*

$$\sup_{i,j,|\mathbf{Y}|\leq m_n} \mathbb{P}[|\hat{z}_{n;i,j|\mathbf{Y}} - z_{n;i,j|\mathbf{Y}}| > \gamma] \leq O(n - m_n)\left( \exp\Big( -(n - m_n)\frac{\gamma^2}{2L^2}\Big) + \exp\big( -C(n - m_n)\big)\right),$$

*where $W_i$ and $W_j$ are in $\mathbf{W}$, $\mathbf{Y} \subseteq \mathbf{W} \setminus \{W_i, W_j\}$, $L = \frac{4}{4-(1+M)^2}$ and $M$ is from (A5).*

PROOF. This is a slightly weaker version of Lemma 3 in [2], which is sufficient for the proof of Lemma 1.4. □

LEMMA 2.2. *Let $\mathcal{G} = (\mathbf{X} \dot\cup \mathbf{L} \dot\cup \mathbf{S}, \mathbf{E})$ be a DAG and let $\{X_i, X_k\} \subseteq \mathbf{X}$ with $X_i \neq X_k$. If there is an inducing path $\Pi_{ik}$ in $\mathcal{G}$ between $X_i$ and $X_k$ relative to $\mathbf{X} \setminus \{X_i, X_k\}$ given $\mathbf{S}$ that is into (out of) $X_i$ and into [out of] $X_k$, then for every subset $\mathbf{Z}$ of $\mathbf{X} \setminus \{X_i, X_k\}$ there is a path $\Gamma_{ik}$ in $\mathcal{G}$ that d-connects $X_i$ and $X_k$ given $\mathbf{Z} \cup \mathbf{S}$ that is into (out of) $X_i$ and into [out of] $X_k$.*

PROOF. This is a reformulation of Lemmas 4, 5 and 7 in [4]. □

LEMMA 2.3. *(Spirtes, Meek and Richardson [4], page 247) Let* $\mathcal{G} = (\mathbf{X} \dot{\cup} \mathbf{L} \dot{\cup} \mathbf{S}, \mathbf{E})$ *be a DAG and let* $\{X_i, X_j, X_k\} \subseteq \mathbf{X}$*, with* $X_i$*,* $X_j$ *and* $X_k$ *three distinct vertices. If* $X_k$ *is in the minimal d-separating set for* $X_i$ *and* $X_j$*, then* $X_k \in an(\mathcal{G}, \{X_i, X_j\} \cup \mathbf{S})$*.*

PROOF. See [4], page 247. □

LEMMA 2.4. *Let* $\mathcal{G} = (\mathbf{X} \dot{\cup} \mathbf{L} \dot{\cup} \mathbf{S}, \mathbf{E})$ *be a DAG and let* $\{X_i, X_j\} \subseteq \mathbf{X}$ *with* $X_i \neq X_j$*. Moreover, assume that there is an inducing path* $\Pi_{ij}$ *between* $X_i$ *and* $X_j$ *relative to some* $\mathbf{X}' \subseteq \mathbf{X}$ *given* $\mathbf{S}$*. If* $X_i \notin an(\mathcal{G}, \{X_j\} \cup \mathbf{S})$*, then* $\Pi_{ij}$ *must be into* $X_i$*.*

PROOF. The proof follows directly from the contraposition of Lemma 11 of [4]. □

Finally, we frequently make use of Lemma 3.3.1 of [3] which says that one can think about d-connectivity of a sequence of paths in a DAG in the same way as d-connectivity of a sequence of edges, by simply replacing "edge" by "path" in the definition of d-connection.

LEMMA 2.5. *(Spirtes, Glymour and Scheines [3], Lemma 3.3.1, pages 385-386) In a directed graph* $\mathcal{G}$ *over* $\mathbf{V}$*, if* $X_i$ *and* $X_j$ *are not in* $\mathbf{Z}$*, there is a sequence* $\sigma$ *of distinct vertices in* $\mathbf{V}$ *from* $X_i$ *to* $X_j$*, and there is a set* $\mathcal{T}$ *of paths such that*

*(i) for each pair of adjacent vertices* $X_v$ *and* $X_w$ *in* $\sigma$ *there is a unique path in* $\mathcal{T}$ *that d-connects* $X_v$ *and* $X_w$ *given* $\mathbf{Z} \setminus \{X_v, X_w\}$*, and*

*(ii) if a vertex* $X_q$ *in* $\sigma$ *is in* $\mathbf{Z}$*, then the paths in* $\mathcal{T}$ *that contain* $X_q$ *as an endpoint collide at* $X_q$ *and*

*(iii) if for three vertices* $X_v$*,* $X_w$*,* $X_q$ *occurring in that order in* $\sigma$ *the d-connecting paths in* $\mathcal{T}$ *between* $X_v$ *and* $X_w$*, and* $X_w$ *and* $X_q$ *collide at* $X_w$ *then* $X_w$ *has a descendant in* $\mathbf{Z}$*,*

*then there is a path* $\Pi_{ij}$ *in* $\mathcal{G}$ *that d-connects* $X_i$ *and* $X_j$ *given* $\mathbf{Z}$*. In addition, if all of the edges in all of the paths in* $\mathcal{T}$ *that contain* $X_i$ *are into (out of)* $X_i$ *then* $\Pi_{ij}$ *is into (out of)* $X_i$*, and similarly for* $X_j$*.*

PROOF. See [3], page 386. □

Note that Lemma 2.5 allows the paths in $\mathcal{T}$ to intersect one another, so that $\Pi_{ij}$ is not in general formed by simply concatenating the paths in $\mathcal{T}$ (since that might contain loops).

**3. The Anytime FCI and the Adaptive Anytime FCI algorithms.** The Anytime FCI algorithm was introduced by Spirtes [5]. It can be viewed as a modification of the FCI algorithm where the conditional independence oracle always returns "dependent" when the conditioning set is larger than some pre-specified cut-off $K$. Or, in other words, the Anytime FCI algorithm is a modification of the FCI algorithm that only performs conditional independence tests up to and including order $K$ when finding the initial and final skeletons (see Steps 1 and 3 of Algorithm 3.1 in [1]).

Thus, Anytime FCI performs fewer conditional independence tests than FCI. Therefore like RFCI, it is typically faster but may be less informative than FCI. In fact, the output of Anytime FCI is an RFCI-PAG, where condition (ii') of Definition 3.2 of [1] can be strengthened as follows "The presence of an edge between two vertices $X_i$ and $X_j$ in $\mathcal{C}$ implies that $X_i \not\perp\!\!\!\perp X_j | (\mathbf{Y} \cup \mathbf{S})$ for all subsets $\mathbf{Y} \subseteq \mathbf{X}$ with $|\mathbf{Y}| \leq K$". So, just as for RFCI, the skeleton of the output of Anytime FCI is generally a supergraph of the skeleton of the output of FCI, and the output of Anytime FCI can describe more than one Markov equivalence class. More details about the interpretation of the output of Anytime FCI can be found in [5].

When introducing Anytime FCI, Spirtes [5] focused on speeding up the step where the initial skeleton is found (Step 1 of Algorithm 3.1 in [1]). However, we found that not this step, but the step in which the final skeleton is found (Step 3 of Algorithm 3.1 in [1]) causes infeasible computing times for high-dimensional sparse graphs. We therefore propose a slight modification of Anytime FCI, called Adaptive Anytime FCI (AAFCI) (see Algorithm 4.6), in which the cut-off $K$ is set adaptively to the maximum size of the conditioning sets that are considered when determining the initial skeleton in FCI. Thus, Step 1 of AAFCI is identical to Step 1 of FCI (and RFCI), and only Step 3 of the FCI algorithm is changed as follows: instead of considering all subsets of the Possible D-SEP sets, AAFCI only considers all subsets up to and including size $K = \max_i(|\text{adj}(\mathcal{C}_1, X_i)| - 1)$, where $\mathcal{C}_1$ is the graph resulting from Step 1. Another advantage of AAFCI is that the cut-off parameter $K$ no longer needs to be pre-specified, but is set adaptively in such a way that the informativeness of the output of AAFCI is comparable to that of FCI and RFCI. It is clear that AAFCI is faster than FCI. It is typically slower than RFCI however, since Steps 2 and 3 of RFCI only consider subsets of adjacency sets, which are small for sparse graphs, while AAFCI considers subsets of Possible D-SEP, of which there can be many of size $K$ even for sparse graphs.

Since AAFCI is a special version of Anytime FCI, soundness follows directly from [5]. AAFCI can be shown to be consistent in sparse high-

dimensional settings under the same assumptions as for RFCI (see Theorem 4.1 in [1]). Moreover, Theorem 3.3 of [1] can be extended to include AAFCI as well.

The modifications of FCI that are proposed in Section 3.1 of [1] can be applied to AAFCI as well, leading to the following algorithms: $\text{AAFCI}_{\text{path}}$, CAAFCI, $\text{CAAFCI}_{\text{path}}$, SCAAFCI, and $\text{SCAAFCI}_{\text{path}}$. Sample versions of all modifications of AAFCI can be obtained by replacing all steps with conditional independence decisions by conditional independence tests, as in Section 4.1 of [1].

## 4. Pseudocode of the algorithms.

**Algorithm 4.1** Obtaining an initial skeleton

**Require:** Conditional independence information among all variables in $\mathbf{X}$ given $\mathbf{S}$

Form the complete graph $\mathcal{C}$ on the vertex set $\mathbf{X}$ with edges $\circ\!\!-\!\!\circ$;
Let $\ell = -1$;
**repeat**
  Let $\ell = \ell + 1$;
  **repeat**
    **for all** vertices $X_i$ in $\mathcal{C}$ **do**
      Compute $\text{adj}(\mathcal{C}, X_i)$
    **end for**
    Select a (new) ordered pair of vertices $(X_i, X_j)$ that are adjacent in $\mathcal{C}$ and satisfy $|\text{adj}(\mathcal{C}, X_i) \setminus \{X_j\}| \geq \ell$;
    **repeat**
      Choose a (new) set $\mathbf{Y} \subseteq \text{adj}(\mathcal{C}, X_i) \setminus \{X_j\}$ with $|\mathbf{Y}| = \ell$;
      **if** $X_i$ and $X_j$ are conditionally independent given $\mathbf{Y} \cup \mathbf{S}$ **then**
        Delete edge $X_i \circ\!\!-\!\!\circ X_j$ from $\mathcal{C}$;
        Let $\text{sepset}(X_i, X_j) = \text{sepset}(X_j, X_i) = \mathbf{Y}$;
      **end if**
    **until** $X_i$ and $X_j$ are no longer adjacent in $\mathcal{C}$ or all $\mathbf{Y} \subseteq \text{adj}(\mathcal{C}, X_i) \setminus \{X_j\}$ with $|\mathbf{Y}| = \ell$ have been considered
  **until** all ordered pairs of adjacent vertices $(X_i, X_j)$ in $\mathcal{C}$ with $|\text{adj}(\mathcal{C}, X_i) \setminus \{X_j\}| \geq \ell$ have been considered
**until** all pairs of adjacent vertices $(X_i, X_j)$ in $\mathcal{C}$ satisfy $|\text{adj}(\mathcal{C}, X_i) \setminus \{X_j\}| \leq \ell$
Form a list $\mathfrak{M}$ of all unshielded triples $\langle X_k, \cdot, X_m \rangle$ (i.e., the middle vertex is left unspecified) in $\mathcal{C}$ with $k < m$;
**return** $\mathcal{C}$, sepset, $\mathfrak{M}$.

**Algorithm 4.2** Orienting v-structures in the FCI algorithm

**Require:** Initial skeleton ($\mathcal{C}$), separation sets (sepset) and unshielded triple list ($\mathfrak{M}$)

**for all** elements $\langle X_i, X_j, X_k \rangle$ of $\mathfrak{M}$ **do**
  **if** $X_j \notin \text{sepset}(X_i, X_k)$ **then**
    Orient $X_i \ast\!\!-\!\!\circ X_j \circ\!\!-\!\!\ast X_k$ as $X_i \ast\!\!\rightarrow X_j \leftarrow\!\!\ast X_k$ in $\mathcal{C}$;
  **end if**
**end for**
**return** $\mathcal{C}$, sepset.

**Algorithm 4.3** Obtaining the final skeleton in the FCI algorithm

**Require:** Partially oriented graph ($\mathcal{C}$) and separation sets (sepset)

**for all** vertices $X_i$ in $\mathcal{C}$ **do**
  Compute $\text{pds}(\mathcal{C}, X_i, \cdot)$ as defined in Definition 3.3 of [1];
  **for all** vertices $X_j \in \text{adj}(\mathcal{C}, X_i)$ **do**
    Let $\ell = -1$;
    **repeat**
      Let $\ell = \ell + 1$;
      **repeat**
        Choose a (new) set $\mathbf{Y} \subseteq \text{pds}(\mathcal{C}, X_i, \cdot) \setminus \{X_j\}$ with $|\mathbf{Y}| = \ell$;
        **if** $X_i$ and $X_j$ are conditionally independent given $\mathbf{Y} \cup \mathbf{S}$ **then**
          Delete edge $X_i \ast\!\!-\!\!\ast X_j$ from $\mathcal{C}$;
          Let $\text{sepset}(X_i, X_j) = \text{sepset}(X_j, X_i) = \mathbf{Y}$;
        **end if**
      **until** $X_i$ and $X_j$ are no longer adjacent in $\mathcal{C}$ or all $\mathbf{Y} \subseteq \text{pds}(\mathcal{C}, X_i, \cdot) \setminus \{X_j\}$ with $|\mathbf{Y}| = \ell$ have been considered
    **until** $X_i$ and $X_j$ are no longer adjacent in $\mathcal{C}$ or $|\text{pds}(\mathcal{C}, X_i, \cdot) \setminus \{X_j\}| < \ell$
  **end for**
**end for**
Reorient all edges in $\mathcal{C}$ as $\circ\!\!-\!\!\circ$;
Form a list $\mathfrak{M}$ of all unshielded triples $\langle X_k, \cdot, X_m \rangle$ in $\mathcal{C}$ with $k < m$;
**return** $\mathcal{C}$, sepset, $\mathfrak{M}$.

**Algorithm 4.4** Orienting v-structures in the RFCI algorithm

**Require:** Initial skeleton ($\mathcal{C}$), separation sets (sepset) and unshielded triple list ($\mathfrak{M}$) from Algorithm 4.1

Let $\mathfrak{L}$ be an empty list;
**while** $\mathfrak{M}$ is non-empty **do**
  Choose an unshielded triple $\langle X_i, X_j, X_k \rangle$ from $\mathfrak{M}$;
  **if** both ($X_i$ and $X_j$) and ($X_j$ and $X_k$) are conditionally dependent given $(\text{sepset}(X_i, X_k) \setminus \{X_j\}) \cup \mathbf{S}$ **then**
    Add $\langle X_i, X_j, X_k \rangle$ to $\mathfrak{L}$;
  **else**
    **for** $r \in \{i, k\}$ **do**
      **if** $X_r$ and $X_j$ are conditionally independent given $(\text{sepset}(X_i, X_k) \setminus \{X_j\}) \cup \mathbf{S}$ **then**
        Find a minimal separating set $\mathbf{Y} \subseteq \text{sepset}(X_i, X_k) \setminus \{X_k\}$ for $X_r$ and $X_j$;
        Let $\text{sepset}(X_r, X_j) = \text{sepset}(X_j, X_r) = \mathbf{Y}$;
        Add to $\mathfrak{M}$ all triples $\left\langle X_{\min(r,j)}, \cdot, X_{\max(r,j)} \right\rangle$ that form a triangle in $\mathcal{C}$;
        Delete from $\mathfrak{M}$ and $\mathfrak{L}$ all triples containing $(X_r, X_j)$: $\langle X_r, X_j, \cdot \rangle$, $\langle X_j, X_r, \cdot \rangle$, $\langle \cdot, X_j, X_r \rangle$ and $\langle \cdot, X_r, X_j \rangle$;
        Delete edge $X_r \ast\!\!-\!\!\ast X_j$ from $\mathcal{C}$;
      **end if**
    **end for**
  **end if**
  Remove $\langle X_i, X_j, X_k \rangle$ from $\mathfrak{M}$;
**end while**
**for all** elements $\langle X_i, X_j, X_k \rangle$ of $\mathfrak{L}$ **do**
  **if** $X_j \notin \text{sepset}(X_i, X_k)$ and both edges $X_i \ast\!\!-\!\!\ast X_j$ and $X_j \ast\!\!-\!\!\ast X_k$ are present in $\mathcal{C}$ **then**
    Orient $X_i \ast\!\!-\!\!\circ X_j \circ\!\!-\!\!\ast X_k$ as $X_i \ast\!\!\rightarrow X_j \leftarrow\!\!\ast X_k$ in $\mathcal{C}$;
  **end if**
**end for**
**return** $\mathcal{C}$, sepset.

**Algorithm 4.5** Orientation rules for RFCI-algorithm

**Require:** Partially oriented graph ($\mathcal{C}$) and separation sets (sepset) from Algorithm 4.4

1: **repeat**
2: Orient as many edge marks as possible in $\mathcal{C}$ by applying rules (R1)-(R3) of [6];
3: **while** a triangle between three vertices $\langle X_l, X_j, X_k \rangle$ exists such that $X_j \circ\!\!-\!\!* X_k$, $X_l \leftrightarrow\!\!* X_j$ and $X_l \rightarrow X_k$ in $\mathcal{C}$ **do**
4: Find a shortest discriminating path for $\langle X_l, X_j, X_k \rangle$
5: **if** a discriminating path $\pi$ for $\langle X_l, X_j, X_k \rangle$ exists between $X_k$ and say $X_i$ **then**
6: **for all** pairs of vertices $(X_r, X_q)$ that are adjacent on $\pi$ **do**
7: Let $\ell = -1$;
8: **repeat**
9: Let $\ell = \ell + 1$;
10: **repeat**
11: Choose a (new) subset $\mathbf{Y} \subseteq \text{sepset}(X_i, X_k) \setminus \{X_r, X_q\}$ with $|\mathbf{Y}| = \ell$;
12: **if** $X_r$ and $X_q$ are conditionally independent given $\mathbf{Y} \cup \mathbf{S}$ **then**
13: Let $\text{sepset}(X_r, X_q) = \text{sepset}(X_q, X_r) = \mathbf{Y}$;
14: Create a list $\mathfrak{M}$ of all triples $\langle X_r, \cdot, X_q \rangle$ with $r < q$ that form a triangle in $\mathcal{C}$;
15: Delete edge $X_r *\!\!-\!\!* X_q$ from $\mathcal{C}$;
16: Run Algorithm 4.4 with input $\{\mathcal{C}, \text{sepset}, \mathfrak{M}\}$ and update $\mathcal{C}$ and sepset;
17: **end if**
18: **until** $X_r$ and $X_q$ are no longer adjacent in $\mathcal{C}$ or all $Y \subseteq \text{sepset}(X_i, X_k) \setminus \{X_r, X_q\}$ with $|Y| = \ell$ have been considered
19: **until** $|\text{sepset}(X_i, X_k) \setminus \{X_r, X_q\}| < \ell$
20: **end for**
21: **if** all of the edges between adjacent vertices on $\pi$ are present in $\mathcal{C}$ **then**
22: **if** $X_j \in \text{sepset}(X_i, X_k)$ **then**
23: Orient $X_j \circ\!\!-\!\!* X_k$ as $X_j \rightarrow X_k$ in $\mathcal{C}$;
24: **else**
25: Orient the triple $X_l \leftrightarrow\!\!* X_j \circ\!\!-\!\!* X_k$ as $X_l \leftrightarrow X_j \leftrightarrow X_k$ in $\mathcal{C}$;
26: **end if**
27: **end if**
28: **end if**
29: **end while**
30: Orient as many edge marks as possible in $\mathcal{C}$ by applying rules (R5)-(R10) of [6];
31: **until** $\mathcal{C}$ remains unchanged throughout lines 2-30 above
32: **return** $\mathcal{C}$, sepset.

**Algorithm 4.6** The AAFCI algorithm

**Require:** Conditional independence information among all variables in $\mathbf{X}$ given $\mathbf{S}$

1: Use Algorithm 4.1 to find an initial skeleton ($\mathcal{C}$), separation sets (sepset) and unshielded triple list ($\mathfrak{M}$);
2: Use Algorithm 4.2 to orient v-structures (update $\mathcal{C}$);
3: Use Algorithm 4.7 to find the final skeleton (update $\mathcal{C}$ and sepset);
4: Use Algorithm 4.2 to orient v-structures (update $\mathcal{C}$);
5: Use rules (R1)-(R10) of [6] to orient as many edge marks as possible (update $\mathcal{C}$);
6: **return** $\mathcal{C}$, sepset.

**Algorithm 4.7** Obtaining the final skeleton in the AAFCI algorithm

**Require:** Partially oriented graph ($\mathcal{C}$) and separation sets (sepset)

1: Set cut-off $K = \max_i(|\text{adj}(\mathcal{C}, X_i)| - 1)$;
2: **for all** vertices $X_i$ in $\mathcal{C}$ **do**
3: Compute $\text{pds}(\mathcal{C}, X_i, \cdot)$ as defined in Definition 3.3 of [1];
4: **for all** vertices $X_j \in \text{adj}(\mathcal{C}, X_i)$ **do**
5: Let $\ell = -1$;
6: **repeat**
7: Let $\ell = \ell + 1$;
8: **repeat**
9: Choose a (new) set $\mathbf{Y} \subseteq \text{pds}(\mathcal{C}, X_i, \cdot) \setminus \{X_j\}$ with $|\mathbf{Y}| = \ell$;
10: **if** $X_i$ and $X_j$ are conditionally independent given $\mathbf{Y} \cup \mathbf{S}$ **then**
11: Delete edge $X_i \ast\!\!-\!\!\ast X_j$ from $\mathcal{C}$;
12: Let $\text{sepset}(X_i, X_j) = \text{sepset}(X_j, X_i) = \mathbf{Y}$;
13: **end if**
14: **until** $X_i$ and $X_j$ are no longer adjacent in $\mathcal{C}$ or all $\mathbf{Y} \subseteq \text{pds}(\mathcal{C}, X_i, \cdot) \setminus \{X_j\}$ with $|\mathbf{Y}| = \ell$ have been considered
15: **until** $X_i$ and $X_j$ are no longer adjacent in $\mathcal{C}$ or $|\text{pds}(\mathcal{C}, X_i, \cdot) \setminus \{X_j\}| < \ell$ or $\ell = K$
16: **end for**
17: **end for**
18: Reorient all edges in $\mathcal{C}$ as $\circ\!\!-\!\!\circ$;
19: Form a list $\mathfrak{M}$ of all unshielded triples $\langle X_k, \cdot, X_m \rangle$ in $\mathcal{C}$ with $k < m$;
20: **return** $\mathcal{C}$, sepset, $\mathfrak{M}$.

**5. Further examples.** In this section, we give two examples that explain certain steps in Algorithms 4.4 and 4.5 of the RFCI algorithm. First, Example 1 shows why it is necessary to find a *minimal* separating set in line 9 of Algorithm 4.4. Second, Example 2 shows why one needs to consider conditional independence given *all subsets of* $\text{sepset}(X_i, X_k) \setminus \{X_i, X_j\}$ in lines 10-17 of Algorithm 4.5.

EXAMPLE 1. *Consider the DAG $\mathcal{G}$ depicted in Figure 1 containing observed variables $\mathbf{X} = \{X_a, \ldots, X_k\}$. latent variables $\mathbf{L} = \{L_1, \ldots, L_4\}$ and no selection variables ($\mathbf{S} = \emptyset$).*

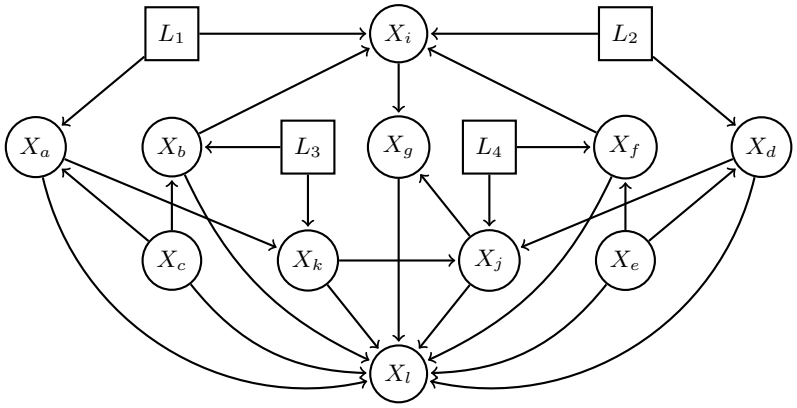


FIG 1. *Underlying DAG for Example 1, which illustrates why it is necessary to find a* minimal *separating set in line 9 of Algorithm 4.4.*

*We focus our analysis on the vertices $X_i$, $X_j$, $X_l$ and $X_k$. Vertices $X_l$ and $X_k$, as well as $X_l$ and $X_j$, are adjacent in the underlying DAG, and will remain so in the RFCI-PAG. Vertices $X_l$ and $X_i$ are conditionally independent given $\mathbf{X} \setminus \{X_i, X_l\} = adj(\mathcal{G}, X_l)$. This conditional independence is found in Step 1 of the RFCI algorithm, and hence the edge between $X_l$ and $X_i$ is removed. Vertices $X_i$ and $X_k$ are adjacent after Step 1, since the path $\langle X_i, L_1, X_a, X_c, X_b, L_3, X_k \rangle$ cannot be blocked by a subset of the adjacency set of $X_i$ or of $X_k$ in the graph after Step 1. Similarly, $X_i$ and $X_j$ are adjacent after Step 1, since the path $\langle X_i, L_2, X_d, X_e, X_f, L_4, X_j \rangle$ cannot be blocked by a subset of the adjacency set of $X_i$ or of $X_j$ in the graph after Step 1.*

*Hence, the skeleton after Step 1 of the RFCI algorithm contains the structure shown in Figure 2. In Step 2 of the RFCI algorithm, all unshielded triples are considered as specified in Algorithm 4.4. The list $\mathfrak{M}$ that is used as input for this algorithm contains $\langle X_i, X_k, X_l \rangle$ and $\langle X_i, X_j, X_l \rangle$.*

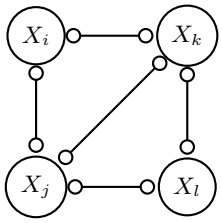


FIG 2. *Sub-skeleton after Step 1 of the RFCI algorithm for the underlying DAG in Figure 1.*

*We now show that incorrect conclusions would be reached if lines 9 and 10 in Algorithm 4.4 were to be replaced by "Let* $sepset(X_r, X_j) = sepset(X_j, X_r) = sepset(X_i, X_k) \setminus \{X_k\}$*". Thus, suppose that Algorithm 4.4 is carried out with this modification.*

*First consider* $\langle X_i, X_j, X_l \rangle$*. Since* $X_g \in sepset(X_i, X_l)$*, we have* $X_i \not\perp\!\!\!\perp X_j | (sepset(X_i, X_l) \setminus \{X_j\})$*. Moreover,* $X_j \not\perp\!\!\!\perp X_l | (sepset(X_i, X_l) \setminus \{X_i\})$*, since* $X_j$ *and* $X_l$ *are adjacent in the underlying DAG. Hence,* $\langle X_i, X_j, X_l \rangle$ *is added to* $\mathfrak{L}$ *and lines 6-16 of Algorithm 4.4 are skipped.*

*When considering* $\langle X_i, X_k, X_l \rangle$*, Algorithm 4.4 finds in line 8 that* $X_i$ *and* $X_k$ *are conditionally independent given* $sepset(X_i, X_l) \setminus \{X_k\} = \mathbf{X} \setminus \{X_i, X_l, X_k\}$*. Next, in our modified lines 9 and 10, it stores* $\mathbf{X} \setminus \{X_i, X_l, X_k\}$ *in* $sepset(X_i, X_k)$ *and* $sepset(X_k, X_i)$*. In line 11, the triple* $\langle X_i, X_j, X_k \rangle$ *is added to* $\mathfrak{M}$*. In line 13, the edge between* $X_i$ *and* $X_k$ *is deleted.*

*Now consider* $\langle X_i, X_j, X_k \rangle$*, the triple that was just added to* $\mathfrak{M}$*. Since* $X_g \in (sepset(X_i, X_k) \setminus \{X_j\})$*, we have* $X_i \not\perp\!\!\!\perp X_j | (sepset(X_i, X_k) \setminus \{X_j\})$*. Moreover,* $X_j \not\perp\!\!\!\perp X_k | (sepset(X_i, X_k) \setminus \{X_j\})$*, since* $X_j$ *and* $X_k$ *are adjacent in the underlying DAG. Hence, this triple is added to* $\mathfrak{L}$ *and lines 6-16 of Algorithm 4.4 are skipped.*

*Finally, in lines 19-23 of Algorithm 4.4, all triples in* $\mathfrak{L}$ *are oriented. Since, under our modified algorithm,* $X_j \in sepset(X_i, X_k) = \mathbf{X} \setminus \{X_i, X_l, X_k\}$*,* $\langle X_i, X_j, X_k \rangle$ *is marked as a non-v-structure. This implies that* $X_j \in an(\mathcal{G}, \{X_i, X_k\} \cup \mathbf{S})$ *(see Definition 3.2 of [1]), which is not the case in the underlying DAG in Figure 1.*

EXAMPLE 2. *Consider the DAG depicted in Figure 3 containing observed variables* $\mathbf{X} = \{X_a, \ldots, X_p\}$*, latent variables* $\mathbf{L} = \{L_{ab}, L_{ap}, L_{bi}, L_{bp}, L_{cd}, L_{cl}, L_{ep}, L_{fl}, L_{gj}, L_{ij}, L_{ip}, L_{jk}, L_{kl}\}$ *and no selection variables (*$\mathbf{S} = \emptyset$*). In order to save space and improve readability of the figure, the latent variables are represented as dashed bi-directed edges, i.e.,* $X_a \dashleftarrow\dashrightarrow X_b$ *is shorthand for* $X_a \leftarrow L_{ab} \rightarrow X_b$*, where* $L_{ab}$ *is a latent variable.*

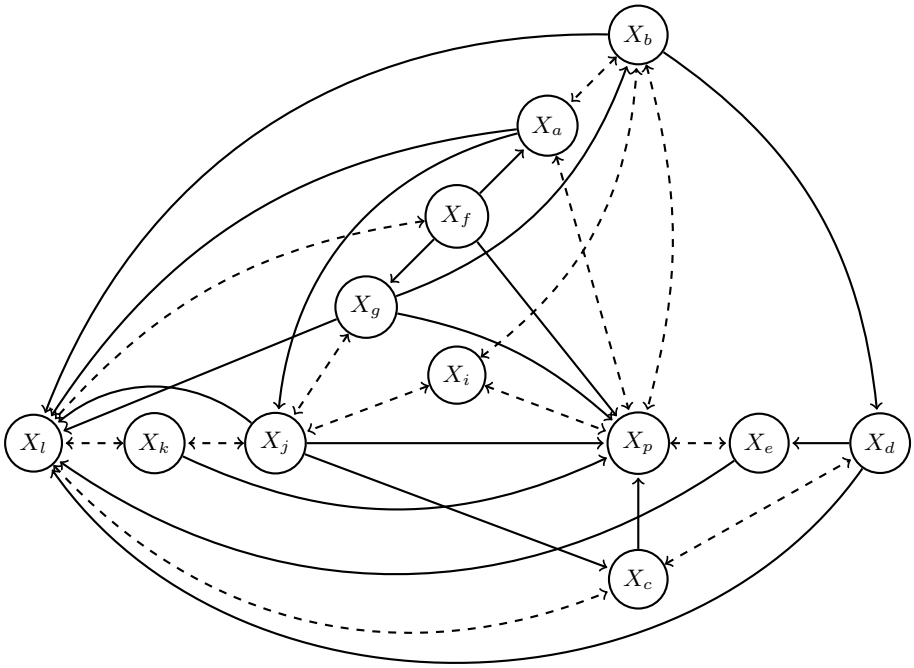


FIG 3. *Underlying graph for Example 2, which illustrates why one needs to consider conditional independence given all subsets of sepset$(X_i, X_k) \setminus \{X_i, X_j\}$ in lines 10-17 of Algorithm 4.5.*

*We focus our analysis on the vertices $X_b$, $X_j$, $X_k$, $X_l$ and $X_p$. Vertices $X_l$ and $X_p$ are conditionally independent given $\mathbf{X} \setminus \{X_l, X_p, X_i\} = adj(\mathcal{G}, X_l)$. This conditional independence is found in Step 1 of the RFCI algorithm, and hence the edge between $X_l$ and $X_p$ is removed. Vertices $X_b$ and $X_j$ are adjacent after Step 1 since the path $\langle X_b, X_a, X_f, X_g, X_j \rangle$ cannot be blocked by a subset of the adjacency set of $X_b$ or of $X_j$ in the graph after Step 1. Moreover, this edge cannot be deleted in Step 2 of the RFCI algorithm using Algorithm 4.4, since no unshielded triple present in the list $\mathfrak{M}$ tests the conditional independence between $X_j$ and $X_b$ given $\{X_a, X_f, X_g\}$, which is the minimal separating set.*

*While orienting the edge marks in Step 3, the RFCI algorithm searches for shortest discriminating paths in line 4 of Algorithm 4.5. Among others, it detects the structure in Figure 4, which is a shortest discriminating path $\pi$ for $X_b$ between $X_l$ and $X_p$.*

*We now show that incorrect conclusions are reached when one only checks*

*conditional independence given* $sepset(X_l, X_p) \setminus \{X_l, X_p\}$ *in lines 10-17 of Algorithm 4.5, instead of given all subsets of* $sepset(X_l, X_p) \setminus \{X_l, X_p\}$. *Thus, suppose that Algorithm 4.5 is carried out with this simplification.*

*Consider all pairs of edges adjacent on* $\pi$, *as specified in line 6 of Algorithm 4.5, and note that* $sepset(X_l, X_p) = \mathbf{X} \setminus \{X_l, X_p, X_i\}$. *We clearly have that* $X_l \not\perp\!\!\!\perp X_k | (sepset(X_l, X_p) \setminus \{X_k\})$, $X_k \not\perp\!\!\!\perp X_j | (sepset(X_l, X_p) \setminus \{X_j, X_k\})$, *and* $X_b \not\perp\!\!\!\perp X_p | (sepset(X_l, X_p) \setminus \{X_b\})$. *Moreover,* $X_j \not\perp\!\!\!\perp X_b | (sepset(X_l, X_p) \setminus \{X_j, X_b\})$, *since the path* $\langle X_j, X_c, X_d, X_b \rangle$ *d-connects* $X_j$ *and* $X_b$ *given* $sepset(X_l, X_p) \setminus \{X_j, X_b\}$.

*Hence, the modified Algorithm 4.5 would not delete any edges between pairs of adjacent vertices on* $\pi$. *As a result, orientation rule (R4) would be applied to* $\pi$ *on line 21 of Algorithm 4.5. Since* $X_b \in sepset(X_l, X_p)$, *the edge* $X_b \circ\!\!\rightarrow X_p$ *would be oriented as* $X_b \rightarrow X_p$. *This would imply that* $X_b \in an(\mathcal{G}, \{X_p\} \cup \mathbf{S})$ *(see Definition 3.2 of [1]), which is not the case in the underlying DAG in Figure 3. The original version of Algorithm 4.5 will not make this error since* $X_j$ *and* $X_b$ *are d-separated given* $\{X_a, X_f, X_g\} \subset \mathbf{X} \setminus \{X_i, X_l, X_p\} = sepset(X_l, X_p)$.

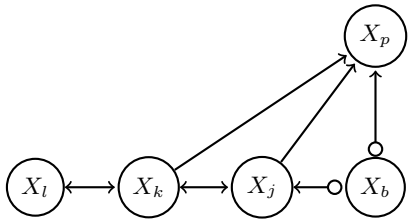


FIG 4. *Subgraph in Step 3 of the RFCI algorithm for the underlying graph in Figure 3, showing a minimal discriminating path* $\pi$ *for* $X_b$ *between* $X_l$ *and* $X_p$.

## REFERENCES

[1] COLOMBO, D., MAATHUIS, M. H., KALISCH, M. and RICHARDSON, T. S. (2012). Learning high-dimensional directed acyclic graphs with latent and selection variables. *Submitted to Ann. Statist.*

[2] KALISCH, M. and BÜHLMANN, P. (2007). Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *J. Mach. Learn. Res.* **8** 613–636.

[3] SPIRTES, P., GLYMOUR, C. and SCHEINES, R. (2000). *Causation, Prediction, and Search*, Second ed. *Adaptive Computation and Machine Learning*. MIT Press, Cambridge.

[4] SPIRTES, P., MEEK, C. and RICHARDSON, T. S. (1999). An algorithm for causal inference in the presence of latent variables and selection bias. In *Computation, Causation and Discovery* 211-252. MIT Press.

[5] Spirtes, P. (2001). An anytime algorithm for causal inference. In *Proc. of the Eighth International Workshop on Artificial Intelligence and Statistics* 213–221. Morgan Kaufmann, San Francisco.

[6] Zhang, J. (2008). On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence* **172** 1873–1896.

ETH Zürich
Seminar for Statistics
Rämistrasse 101
8092 Zürich
Switzerland
E-mail: colombo@stat.math.ethz.ch
maathuis@stat.math.ethz.ch
kalisch@stat.math.ethz.ch

Department of statistics
University of Washington
Seattle, Washington 98195
E-mail: thomasr@u.washington.edu