Discovering Cyclic Causal Models in Psychological Research

Kyuri Park*,[1]
Supervisor: Dr. Oisín Ryan[1]

[1]Department of Methodology and Statistics, Utrecht University

February 17, 2023

This research report is written as a *half of the thesis* format, including the introduction, background, and methods section along with the references and appendices at the end. The candidate journal for publication is *Psychological Methods*.

Keywords: cyclic causal discovery, causal inference, directed cyclic graph

1 Introduction

A fundamental task in various disciplines of science is to understand the mechanisms, that is, causal relations underlying the phenomena of interest. In psychology, for example, one of the core questions is how psychopathology comes about, with the network theory positing that mental disorder is produced by a system of direct causal interactions between symptoms (Borsboom & Cramer, 2013). In practice, empirical researchers often aim to gain insights into these causal relations by fitting statistical network models to observational data, an approach that can be characterized as a form of causal discovery (Spirtes, Glymour, Scheines, & Heckerman, 2000). However, it has been shown that network models are likely to perform poorly as causal discovery tools; relations in the network may not reflect the direct causal effects that researchers aim to discover, but can instead be produced by unwittingly conditioning on common effects or by failing to account for unobserved confounders (Dablander & Hinne, 2019; Ryan, Bringmann, & Schuurman, 2022).

In the field of causal discovery, using statistical independencies estimated from observational data to infer causal structures is known as *constraint-based* causal discovery (Spirtes & Glymour, 1991). Ryan et al. (2022) suggest that network models could be replaced by purpose-built constraint-based causal discovery methods. However, the most popular and well-studied constraint-based methods assume that causal relationships are *acyclic*; if X causes Y, then Y does not cause X (Glymour, Zhang, & Spirtes, 2019). This is problematic since *cyclic* relationships or *feedback loops*

*Correspondence concerning this paper should be addressed to: Kyuri Park, Department of Methodology and Statistics, Utrecht University, Padualaan 14, 3584 CH Utrecht, The Netherlands. e-mail: k.park@uu.nl

are critical to the theoretical understanding of psychopathology (Borsboom, 2017). For example, Wittenborn, Rahmandad, Rick, and Hosseinichimeh (2016) suggest that several different causal feedback loops, such as *perceived stress* → *negative affect* → *rumination* → *perceived stress* play a key role in sustaining depression. Such theoretical expectations necessitate the use of *cyclic causal discovery* methods.

Although some cyclic causal discovery algorithms have been developed (Mooij & Claassen, 2020), they have not been as well studied as their acyclic counterparts. In part, this is due to practical difficulties in fitting and interpreting cyclic causal models, since the properties which map statistical to causal dependencies are more complex than for acyclic models. While other cyclic causal discovery techniques have recently been applied in psychological settings (Kossakowski, Waldorp, & van der Maas, 2021), these methods require a mix of experimental data from different settings, a disadvantage when compared to constraint-based methods. To our knowledge, little to no research has been done on the applicability of constraint-based cyclic causal discovery methods in psychology, and much remains unknown about the performance of these methods.

Therefore, in this paper, we aim to address the following question: how well do constraint-based cyclic causal discovery methods perform in typical psychological research contexts? We focus on three different constraint-based causal discovery methods, which use statistical independence patterns estimated from observational data. The goal of this study is threefold. First, we provide an accessible overview of different cyclic causal discovery methods. Second, we investigate how well each of these methods works by means of a simulation study. Third, we demonstrate their applicability in practice by testing them on empirical data.

2 Background

In this section, we introduce the basic concepts of graphical models, which are necessary to understand the causal discovery methods that we will study in the remainder of this paper.

2.1 Graphical Models

A graph ($\mathcal{G}$) is a diagram made up of a set of vertices ($\mathcal{V}$) and edges ($\mathcal{E}$), describing connections between vertices, denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. A probabilistic graphical model uses a graph to express the conditional (in)dependencies between random variables, where the vertices represent random variables, and the edges encode conditional dependencies holding among the set of variables (Lauritzen, 1996).

There exist various graphical models, typically differing in the types of edges (e.g., directed vs. undirected), and the corresponding interpretation of edges in terms of statistical relationships. One of the most commonly known graphical models in psychology is the *Pairwise Markov Random Field* (PMRF) – an undirected graph in which edges indicate statistical association between variables after controlling for all other variables – which forms the basis of statistical network models. In PMRF

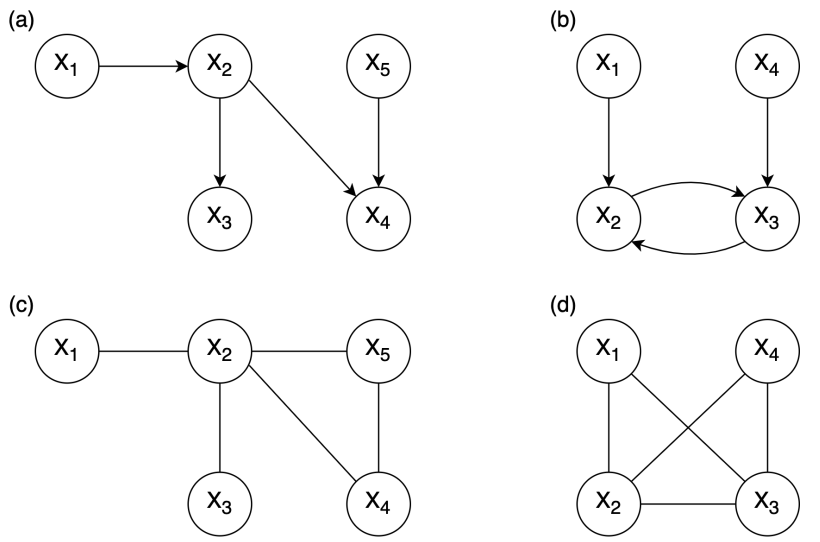
networks, the edges are strictly undirected, and the presence of an edge $A - B$ represents that A and B are statistically dependent conditioning on the set of all other variables in the network (Borsboom et al., 2021).

In a *causal* graphical model, on the other hand, the edges describe *causal* relationships between variables; the edges are typically directed, with $A \rightarrow B$ denoting that intervening on A results in a change in the probability distribution of B (Geiger & Pearl, 1990). One of the simplest causal graphs is a *directed acyclic graph* (DAG), also known as a *Bayesian network*, which consists of directed edges without cycles. When a causal graph contains cycles, we call it *directed cyclic graph* (DCG). Two example causal graphical models are shown in Figure 1. Figure 1a does not contain any cycles, whereas Figure 1b does, hence called a DAG and DCG, respectively.

Causal graphical models also describe patterns of statistical independencies, which can be read off from the graph using Pearl's *d-separation criterion* (Geiger, Verma, & Pearl, 1990). For instance, in Figure 1a, we see a *chain* structure $X_1 \rightarrow X_2 \rightarrow X_3$, which implies that X_1 and X_3 are marginally dependent ($X_1 \not\perp\!\!\!\perp X_3$), but independent conditional on X_2 ($X_1 \perp\!\!\!\perp X_3 \mid X_2$). More formally, we would say X_1 and X_3 are *d-separated* by X_2. A *fork* structure $X_3 \leftarrow X_2 \rightarrow X_4$ implies the same pattern of independencies; X_3 and X_4 are marginally dependent ($X_3 \not\perp\!\!\!\perp X_4$), but independent conditional on X_2 ($X_3 \perp\!\!\!\perp X_4 \mid X_2$). That is, X_3 and X_4 are *d-connected* given an empty set, but *d-separated* by X_2. However, a *collider* structure $X_2 \rightarrow X_4 \leftarrow X_5$ implies a contrasting pattern; here X_2 and X_5 are marginally independent ($X_2 \perp\!\!\!\perp X_5$), but dependent conditional on X_4 ($X_2 \not\perp\!\!\!\perp X_5 \mid X_4$). This distinguishing characteristic of colliders is crucial when identifying the directions of causal relations, as will be shown later in section 2.3.

Figure 1c depicts the PMRF model corresponding to the DAG in Figure 1a, where an additional edge is introduced between $X_2 - X_5$ due to conditioning on the common effect X_4. The spurious edges induced by conditioning on common effects (i.e., colliders) are well-known problems of using PMRF-based statistical network models to infer acyclic causal structures (Dablander & Hinne, 2019). The same problem carries over to the cyclic case, as shown in Figure 1d; two spurious edges are induced in the PMRF network (e.g., $X_1 - X_3$ and $X_2 - X_4$) when conditioning on the colliders X_2 and X_3. This clearly manifests the limitation of using statistical network models for causal discovery. Ryan et al. (2022) discussed this issue in detail and conclude that purpose-built causal discovery methods are likely to outperform statistical network models as causal discovery tools. Hence, in the remainder of the paper, we focus on constraint-based causal discovery methods while examining how they recover the underlying causal structure and under what assumptions they can be expected to work.

Figure 1. Example causal graphical models and corresponding PMRF models.



Note. (a) is the example directed acyclic graph (DAG). (b) is the example directed cyclic graph (DCG). (c) is the PMRF (Pairwise Markov Random Field) corresponding to the DAG in (a). (d) is the PMRF corresponding to the DCG in (b).

2.2 Acyclic vs. Cyclic Causal Graphs

The d-separation criterion described above applies to all acyclic graphs, but only applies to graphs with cycles under certain conditions. To understand these conditions, first we need to introduce some graph terminology. In the field of graphical models, we use kinship terminology to describe a graph structure as follows:

$$\text{if } \begin{Bmatrix} A \to B \\ A \leftarrow B \\ A \to \cdots \to B \text{ or } A = B \\ A \leftarrow \cdots \leftarrow B \text{ or } A = B \end{Bmatrix} \text{ in } \mathcal{G} \text{ then } A \text{ is a } \begin{Bmatrix} \text{parent} \\ \text{child} \\ \text{ancestor} \\ \text{descendant} \end{Bmatrix} \text{ of } B \text{ and } \begin{Bmatrix} A \in pa_{\mathcal{G}}(B) \\ A \in ch_{\mathcal{G}}(B) \\ A \in an_{\mathcal{G}}(B) \\ A \in de_{\mathcal{G}}(B) \end{Bmatrix}.$$

Also, when there exists an edge between two vertices $A - B$, A and B are said to be *adjacent*. For example, in Figure 1b, $X_1 \in pa_{\mathcal{G}}\, X_2$, $X_2 \in ch_{\mathcal{G}}\, X_1$, $\{X_1, X_2, X_3, X_4\} \in an_{\mathcal{G}}\, X_3$, $\{X_1, X_2, X_3\} \in de_{\mathcal{G}}\, X_1$, and X_2 is adjacent to X_1 and X_3. With this in place, we can define the *global Markov* condition, which states that d-separation relations represented in causal graphs can be used to read off statistical independence relations such that:

$$\text{if } A \perp\!\!\!\perp_{\mathcal{G}} B \mid C \implies X_A \perp\!\!\!\perp X_B \mid X_C \text{ for all subsets of } A, B, C,$$

where $\perp\!\!\!\perp_{\mathcal{G}}$ refers to d-separation. If causal graphs are *acyclic* (i.e., DAG), then the *global Markov*

condition holds regardless of the functional forms of causal relations and the distributions of variables involved (Lauritzen, 1996). In addition, in DAGs, the *global Markov* condition also entails the *local Markov* condition stating that a variable is independent of its non-descendants given its parents (Lauritzen, 2000). The fact that one Markov property instantly implies the other comes in handy when reading off conditional independencies from a graph.

In contrast to the acyclic case, the situation is not so straightforward in *cyclic* graphs (i.e., DCG). In DCGs, the global Markov property does not always hold. Spirtes (1994), in fact, showed that it holds for DCGs when causal relations are *linear* and error terms are *independent*. Furthermore, the local Markov property may not hold, even when the global Markov property holds. For example, in Figure 1b, the global Markov property is preserved ($X_1 \perp\!\!\!\perp_{\mathcal{G}} X_4 \mid \{X_2, X_3\} \implies X_1 \perp\!\!\!\perp X_4 \mid \{X_2, X_3\}$), but the local Markov property is violated as $X_2 \not\perp\!\!\!\perp_{\mathcal{G}} X_4 \mid X_3$ (i.e., X_2 is *not* independent of its non-descendant X_4 given its parent X_3). This is because X_3 is both a parent of X_2 and a collider ($X_2 \rightarrow X_3 \leftarrow X_4$) at the same time.

Accordingly, we limit the scope of our study to cyclic causal graphs that represent *linear* causal relationships with jointly *independent* error terms, so for which the global Markov condition is satisfied. Additionally, we make use of one more assumption, known as *faithfulness*, which is required for constraint-based causal discovery. The *faithfulness* assumption is essentially the reverse of the global Markov condition, stating that statistical independencies map onto the structure of causal graphs such that:

$$X_A \perp\!\!\!\perp X_B \mid X_C \implies A \perp\!\!\!\perp_{\mathcal{G}} B \mid C.$$

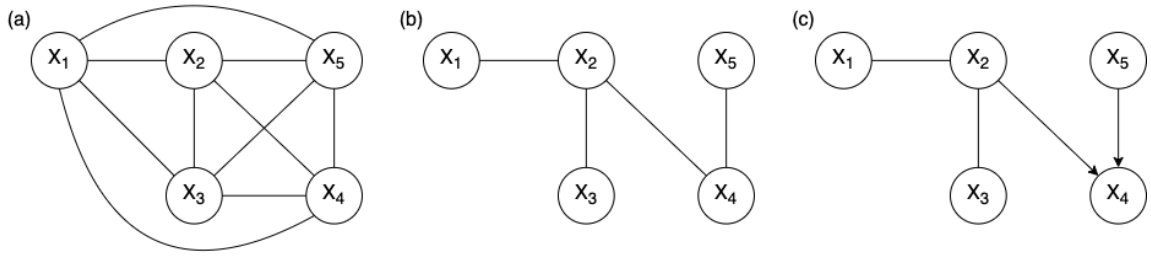
Together with the global Markov property, faithfulness enables us to make inferences about causal relationships represented in graphs by testing for statistical independence among variables (Bongers, Forré, Peters, & Mooij, 2021).

2.3 A Primer on Constraint-Based Causal Discovery

Under the aforementioned assumptions, constraint-based methods seek to recover the underlying causal structure using conditional independencies estimated from observational data. Constraint-based methods typically employ a two-step procedure; first, establishing the *skeleton* – an undirected version of the underlying graph – and second, attempting to assign directions to the edges. In general, constraint-based techniques are unable to uniquely identify the underlying causal graph, but instead return a set of causal graphs that imply the same statistical independence relations.

To develop an intuition for this, we examine how a constraint-based method works for the relatively simple DAG from Figure 1a. We start with a fully-connected graph, as shown in Figure 2a. In the first step, the *skeleton* is estimated by testing for conditional independence; if two variables are independent when conditioning on any subset of the remaining variables (e.g., $X_1 \perp\!\!\!\perp X_3 \mid X_2$, $X_1 \perp\!\!\!\perp X_4 \mid X_2, ...$), then the edge between the two variables is removed (see Figure 2b). In the second step, (some) edges are oriented by searching for *colliders* that induce distinctive patterns of independencies (e.g., X_4 is identified as a collider given $X_2 \perp\!\!\!\perp X_5$ and $X_2 \not\perp\!\!\!\perp X_5 \mid X_4$, thus

Figure 2. Steps of a constraint-based method.



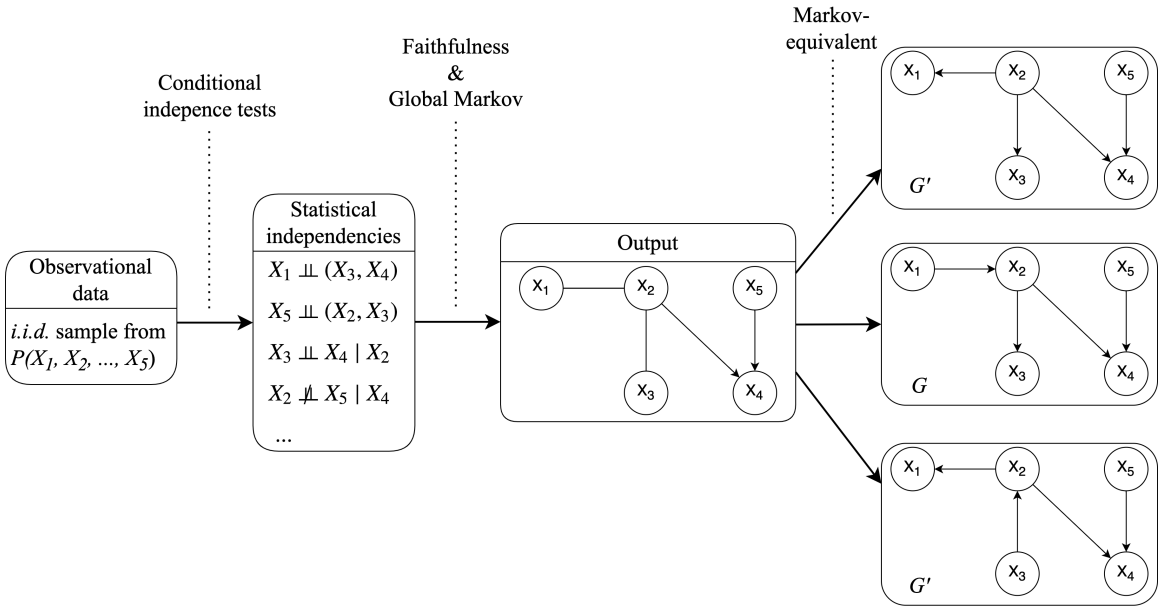
Note. (a) shows the fully-connected graph for the example DAG from Figure 1a, which is the starting point. (b) shows the estimated *skeleton* – an undirected graph of the underlying causal structure – after the first step. (c) shows the resulting graph after the second step, which represents the *Markov equivalence class* of DAGs (i.e., a set of DAGs that entail the same set of conditional independencies).

$X_2 \rightarrow X_4 \leftarrow X_5$ is oriented; see Figure 2c). Note that the resulting graph in Figure 2c is not identical to the original true graph $\mathcal{G}$, as the two edges between $X_1 - X_2$ and $X_2 - X_3$ remain undirected. There are in fact three DAGs that are implied by the resulting graph, as shown in Figure 3. These DAGs are called *Markov equivalent*, meaning that they encode the same conditional independencies (i.e., the same d-separation relations hold), and we call such a set of equivalent graphs a *Markov equivalence class*, denoted by $Equiv(\mathcal{G})$. This illustrates a general difficulty in constraint-based approach; there are usually multiple graphs that are consistent with an observed set of statistical independencies.

The problem is exacerbated when cycles are allowed; when variables influence each other and consist of a causal cycle, the whole group of variables involved in a cyclic structure acts like one functional unit, which makes it challenging to learn the causal relations connected to the cyclic structure. For example in Figure 1b, since X_2 and X_3 form a cycle and behave as if they are a single unit, it becomes difficult to tell whether X_1 is a parent of X_2 or X_3, while it is possible to make a weaker statement such that X_1 is an ancestor of X_2 and X_3. Therefore, when there exists a cycle, a constraint-based method often cannot directly identify parental relations but recovers only up to ancestral relations, typically leading to a larger Markov equivalence class. This also means that the estimated skeleton of a cyclic graph represents ancestral relations – hence called *ancestral skeleton*. Although the same principles that we described here for DAGs are adapted and used for DCGs[1], some additional constraints and orientation rules are utilized due to the complications arising from presence of cycles, which we explain in the following section 3.

[1]See Figure 3 for a summary of the constraint-based approach procedure.

Figure 3. Summary of the constraint-based causal discovery procedure.



Note. A constraint-based algorithm starts with performing a series of conditional independence tests on observational (*i.i.d.*: independent and identically distributed) data . Under the faithfulness and global Markov assumption, the algorithm estimates a graph structure based on the observed statistical independence patterns. The output is a *partially directed* graph (as some edges remain undirected). It can represent multiple graphs that are *Markov equivalent*, meaning that they imply the same statistical independence relations. This equivalent set of graphs is called *Markov equivalence class*, and in this example, it consists of three different DAGs including the true DAG (G).

3 Causal Discovery Algorithms

In this paper, we compare the performance of three different constraint-based algorithms for cyclic graphs using a simulation study: *cyclic causal discovery* (CCD) (Richardson, 1996b), *fast causal inference* (FCI) (Mooij & Claassen, 2020), and *cyclic causal inference* (CCI) (Strobl, 2019). All three algorithms learn causal structures from observational data but under slightly different assumptions. The CCD algorithm assumes *causal sufficiency*, which means that all common causes of variables involved have been measured, and hence no unobserved confounding exists. The other two algorithms, FCI and CCI, relax this assumption and account for the possibility of latent confounders. Note that the FCI algorithm was not initially designed for cyclic causal discovery, but Mooij and Claassen (2020) showed that it performs equally well in the cyclic settings, and thus is considered as one of the cyclic causal discovery algorithms. For the sake of simplicity, in the current paper, we exclude the possibility of selection bias. An overview of the assumptions made by each algorithm can be found in Table 1.

Table 1. Overview of cyclic causal discovery algorithms.

	CCD	FCI	CCI
Global Markov condition	✓	✓	✓
Faithfulness	✓	✓	✓
Acyclicity	✗	–[a]	✗
Causal sufficiency	✓	✗	✗
Independent errors	✓	✓	✓

Note. [a] FCI is originally designed assuming acyclicity, but in a recent research it has been proposed that it performs comparably well in the cyclic settings (Mooij & Claassen, 2020).

3.1 CCD Algorithm

The CCD algorithm is considered relatively simple among the three algorithms, as it assumes that there is no unobserved latent confounding (i.e., *causal sufficiency*). Thus, it can be seen as the base causal discovery method for cyclic graphs and the other two algorithms are essentially built on top of the base with additional steps. In what follows, we introduce the type of output generated by the CCD algorithm in detail and trace the algorithm step-by-step using an example.

3.1.1 Output Representation: Partial Ancestral Graph (PAG)

As was the case with DAGs shown in section 2.3, there typically exist multiple directed cyclic graphs (DCG) that imply the same statistical independencies, and so are statistically indistinguishable from one another. To represent a set of equivalent DCGs, the CCD algorithm uses a *partial ancestral graph* (PAG) that characterizes the common features shared by all equivalent DCGs, $Equiv(\mathcal{G})$. As explained in section 2.3, the possibility of cyclic relations makes the causal semantics of edges in PAGs more complicated; directed edges denote causal *ancestry* (i.e., $A \longrightarrow B$ means A is an *ancestor* of B), and three different types of edge-endpoints $\{\circ, >, -\}$ are utilized to represent the ancestral relations in $Equiv(\mathcal{G})$. The interpretation of each edge-endpoint in a PAG is as follows:[2]

1. $A *\!\!\rightarrow B$ is interpreted as B is *not* an ancestor of A in every graph in $Equiv(\mathcal{G})$.
2. $A *\!\!- B$ is interpreted as A is an ancestor of B in every graph in $Equiv(\mathcal{G})$.
3. $A *\!\!-\!\circ B$ is interpreted as the ancestral relation of B with regard to A is undetermined or invariant across graphs in $Equiv(\mathcal{G})$.

Additionally, a solid underlining or dotted underlining can be added in a PAG, which provides additional information regarding the causal directions. If there is a solid underlining $A *\!\!-\!\!* \underline{B} *\!\!-\!\!* C$, then it is interpreted as B is an ancestor of (at least one of) A or C in every graph in $Equiv(\mathcal{G})$. If there is a dotted underlining added to a collider structure $A \longrightarrow \underline{\mathrm{B}} \longleftarrow C$, it indicates that B is *not* a

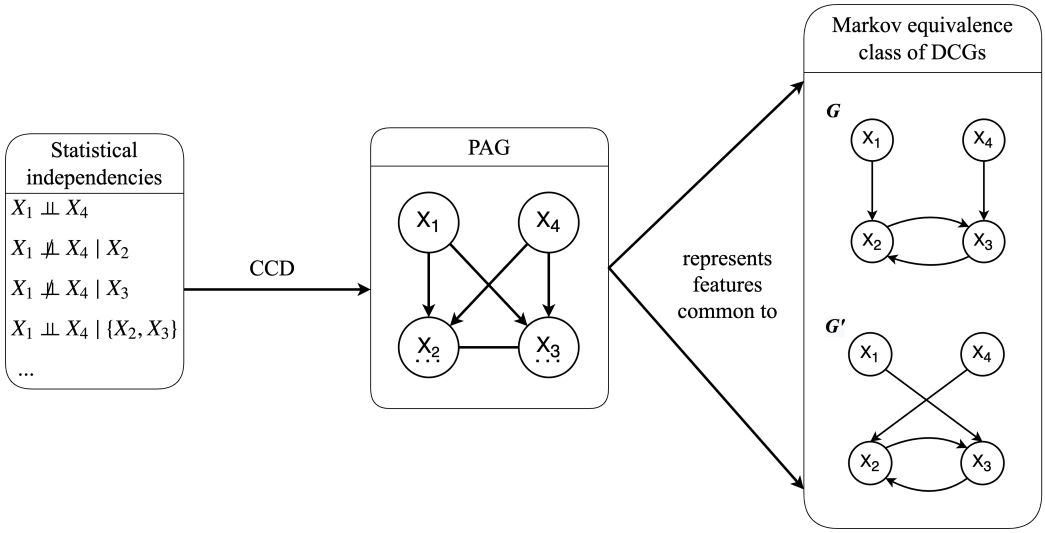
[2] In the description of the semantics for PAGs (Richardson, 1996b), $*$ is used as a *meta-symbol* indicating one of the three possible edge-endpoints. For instance, $A *\!\!- B$ indicates any of the following edges: $A - B$, $A \longrightarrow B$, or $A -\!\circ B$.

descendant of a common child of A and C in every graph in $Equiv(\mathcal{G})$. As an example, from the PAG shown in Figure 4, we can read off the following:

1. X_2 and X_3 are not ancestors of X_1 and X_4 in every graph in $Equiv(\mathcal{G})$.
2. X_1 and X_4 are both ancestors of X_2 and X_3 in every graph in $Equiv(\mathcal{G})$.
3. X_2 is an ancestor of X_3 and X_3 is an ancestor of X_2 in every graph in $Equiv(\mathcal{G})$, which may imply presence of a cycle between them.
4. X_2 and X_3 are not descendants of a common child of X_1 and X_4 in every graph in $Equiv(\mathcal{G})$, which indicates that in no graph in $Equiv(\mathcal{G})$ both directed edges (i.e., $X_1 \rightarrow X_2, X_4 \rightarrow X_2$ or $X_1 \rightarrow X_3, X_4 \rightarrow X_3$) are present.

Given all the aforementioned causal ancestral relations represented by the example PAG, we can correspondingly derive the Markov-equivalent DCGs, as shown in Figure 4.

Figure 4. Summary of CCD algorithm operation.



Note. Given the observed statistical independencies, CCD constructs a partial ancestral graph (PAG), which represents the *ancestral* features that are common to every directed cyclic graph (DCG) in a Markov equivalence class. In this example, the Markov equivalence class consists of two different DCGs, including the true graph $\boldsymbol{G}$.

3.1.2 Steps of CCD Algorithm

The CCD algorithm consists of 6 steps. We illustrate each step using the example DCG from Figure 5a.[3] The algorithm starts with a fully-connected PAG with circle endpoints, as shown in Figure 5b, and as it proceeds (some) circles will be replaced by either an arrow head or a tail.[4]

Step 1. Estimate the *ancestral* skeleton – an undirected graph of ancestral relations implied by the underlying structure – based on conditional independencies. When two vertices A and B are *d-separated* given a set S, remove $A \ast\!\!-\!\!\ast B$ and record $S = \mathbf{Sepset}\langle A, B\rangle = \mathbf{Sepset}\langle B, A\rangle$. Since $X_1 \perp\!\!\!\perp X_4 \mid \varnothing$ in our example DCG, $X_1 \circ\!\!-\!\!\circ X_4$ is removed and $\mathbf{Sepset}\langle X_1, X_4\rangle = \mathbf{Sepset}\langle X_4, X_1\rangle = \varnothing$ is recorded, resulting in Figure 5c.

Step 2. Search for collider structures. If $B \notin \mathbf{Sepset}\langle A, C\rangle$ in a triplet $A \ast\!\!-\!\!\ast B \ast\!\!-\!\!\ast C$, identify B as a collider and orient $A \rightarrow B \leftarrow C$. Given that $X_2 \notin \mathbf{Sepset}\langle X_1, X_4\rangle$ and $X_3 \notin \mathbf{Sepset}\langle X_1, X_4\rangle$ in our example, $X_1 \circ\!\!-\!\!\circ X_2 \circ\!\!-\!\!\circ X_4$ and $X_1 \circ\!\!-\!\!\circ X_3 \circ\!\!-\!\!\circ X_4$ are oriented respectively as $X_1 \rightarrow X_2 \leftarrow X_4$ and $X_1 \rightarrow X_3 \leftarrow X_4$, resulting in Figure 5d.

Step 3. Check for additional d-separating relations in each triplet $\langle A, B, C\rangle$ such that:
(i) A is not adjacent to B or C, *(ii)* B and C are adjacent, and *(iii)* $B \notin \mathbf{Sepset}\langle A, C\rangle$. If such triplets exist, orient $B \ast\!\!-\!\!\ast C$ as $B \leftarrow C$. As no additional d-separating relations are found in our example, no further orientations are performed in step 3.

Step 4. Search for **Supsets**, which are d-separating sets including colliders. For each collider structure $A \rightarrow B \leftarrow C$, check if there is any set T including B that d-separates A and C. When such exists, record $T = \mathbf{Supset}\langle A, B, C\rangle$ and add a dotted-underlining $A \rightarrow \underline{B} \leftarrow C$. Since $X_1 \perp\!\!\!\perp X_4 \mid \{X_2, X_3\}$ in our example, $\mathbf{Supset}\langle X_1, X_2, X_4\rangle = \mathbf{Supset}\langle X_1, X_3, X_4\rangle = \{X_2, X_3\}$ is recorded and each of the colliders is dotted-underlined as $X_1 \rightarrow \underline{X_2} \leftarrow X_4$ and $X_1 \rightarrow \underline{X_3} \leftarrow X_4$, resulting in Figure 5e.

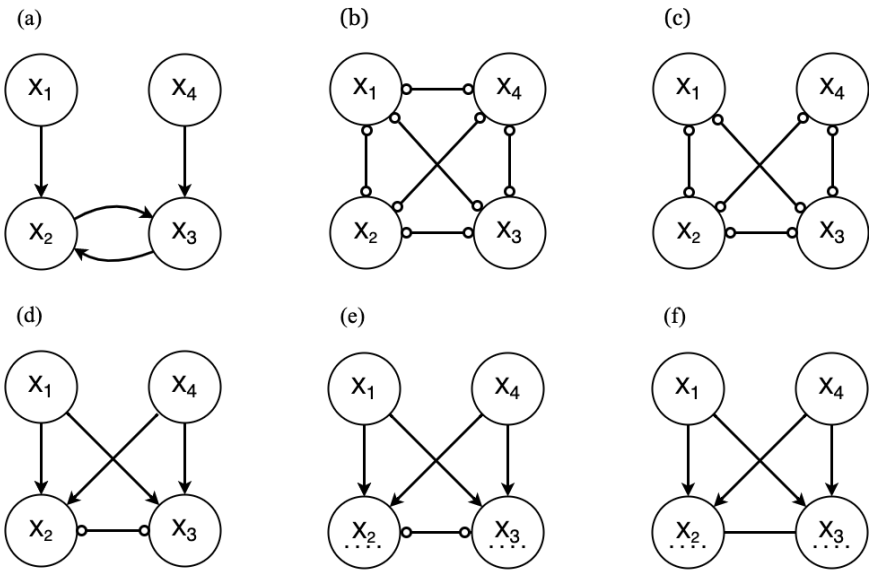
Step 5. Search for quadruplets – four ordered vertices $\langle A, B, C, D\rangle$ – where:
(i) $A \rightarrow \underline{B} \leftarrow C$, *(ii)* $A \rightarrow D \leftarrow C$ or $A \rightarrow \underline{D} \leftarrow C$, and *(iii)* B and D are adjacent. If $D \in \mathbf{Supset}\langle A, B, C\rangle$ in such quadruplets, orient $B \ast\!\!-\!\!\ast D$ as $B \ast\!\!-\!\! D$. Else, orient $B \ast\!\!-\!\!\ast D$ as $B \rightarrow D$. In our example, there is such a quadruplet; *(i)* $X_1 \rightarrow \underline{X_2} \leftarrow X_4$, *(ii)* $X_1 \rightarrow \underline{X_3} \leftarrow X_4$, and *(iii)* X_2 and X_3 are adjacent. Since $X_2 \in \mathbf{Supset}\langle X_1, X_3, X_4\rangle$ and $X_3 \in \mathbf{Supset}\langle X_1, X_2, X_4\rangle$, $X_2 \circ\!\!-\!\!\circ X_3$ is oriented as $X_2 \!-\!\!\circ X_3$, then $X_2 \!-\!\!\circ X_3$ is subsequently oriented as $X_2 - X_3$, resulting in Figure 5f.

Step 6. Search for quadruplets $\langle A, B, C, D\rangle$, where $A \rightarrow \underline{B} \leftarrow C$ while D is adjacent to neither A nor C. If A and D are d-connected given $\mathbf{Supset}\langle A, B, C\rangle \cup D$, then orient $B \ast\!\!-\!\!\circ D$ as $B \rightarrow D$. In our example, no such quadruplets exist; therefore Figure 5f remains the final PAG. As shown in Figure 4, the resulting PAG represents two different DCGs that entail the same conditional independencies.

[3] It is the same as the example DCG that we previously introduced in Figure 1b.

[4] It is important to note, once again, that the algorithm aims to retrieve a PAG for the underlying cyclic graph, and the edges in a PAG represent *ancestral* relations that are common to all directed cyclic graphs (DCG) in an equivalence class.

Figure 5. Trace of CCD algorithm.



Note. (a) shows the true directed cyclic graph, $\boldsymbol{G}$. (b) shows the fully-connected PAG for $\boldsymbol{G}$, which is the starting point of the algorithm. (c) shows the *ancestral* skeleton (i.e., an undirected version of the PAG) estimated in step 1. (d) shows the state of the PAG after step 2, where some of the edges are oriented given the identified colliders. (e) shows the state of the PAG after step 4, where the *Supsets* are identified and the corresponding colliders are dotted-underlined. (f) shows the final state of the PAG after step 5, where an additional edge between X_2 and X_3 is oriented.

3.2 FCI Algorithm

The FCI algorithm is explicitly designed to deal with latent confounding, which commonly exists in psychological research. It was initially designed for learning acyclic causal structures including latent confounders, but Mooij and Claassen (2020) showed that it can be applied to the cyclic case under the more general version of the faithfulness and Markov condition.[5]

3.2.1 Output Representation: Partial Ancestral Graph (PAG)

As with the CCD algorithm, the FCI algorithm can identify the underlying causal graph up to its Markov equivalence class and also employ a PAG to represent common ancestral features of graphs in an equivalence class. A complication arises though, from allowing latent variables; DCGs are not closed under marginalization over latent variables, meaning that there exist infinitely many DCGs of observed variables (O) and latent variables (L) that entail the same independencies (Richardson & Spirtes, 2002). This problem originates from the fact that we do not know how many latent variables are involved and the algorithm has to account for the possibilities of arbitrarily many latent variables (Colombo, Maathuis, Kalisch, & Richardson, 2012).

[5] See Forré and Mooij (2017) for details of the conditions.

In order to represent the presence of latent confounders in the finite space of causal graphs, we introduce a new class of graphs called *directed mixed graphs* (DMG). DMGs are a type of extended DCGs that make use of additional bidirected edges ($\leftrightarrow$) to represent latent confounding (Richardson, 2003); $A \leftrightarrow B$ is interpreted as the presence of latent confounders between A and B. As shown in Figure 6, a resulting PAG from the FCI algorithm thus amounts to a characterization of common features shared by an equivalence class of DMGs. The interpretation of edges in the PAGs output by the FCI algorithm is the same as described in section 3.1.1, except that in the PAGs from FCI, the fully-connected vertices with circle endpoints ($\circ\!\!-\!\!\circ$) may indicate a possible cyclic structure (Mooij & Claassen, 2020). For example, from the PAG shown in Figure 6, we can read off that:

1. X_2 and X_3 are not ancestors of X_1 and X_4 in every graph in $Equiv(\mathcal{G})$.
2. X_2 and X_3 might be part of a cycle in $\mathcal{G}$.

Notice that we are less certain about causal ancestry based on the PAG output by FCI compared to the PAG output by CCD, which is manifested by the increased number of circle ($\circ$) endpoints. This is typically the case as the FCI algorithm accounts for the possibility of latent confounding leading to a greater deal of uncertainty about ancestral relations. In the following, we briefly go over how the PAG in Figure 6 is estimated given the same example DCG used for the CCD algorithm (Figure 1b).

3.2.2 Steps of FCI Algorithm

The FCI algorithm in principle works in a similar way to the CCD algorithm. The FCI algorithm largely consists of three steps; skeleton discovery, collider structure orientation, and further orientation rule application. The first two steps are analogous to CCD's procedure. As with the CCD algorithm, the FCI algorithm starts with a fully-connected PAG with $\circ\!\!-\!\!\circ$ edges between every pair of variables, as shown in Figure 7a.

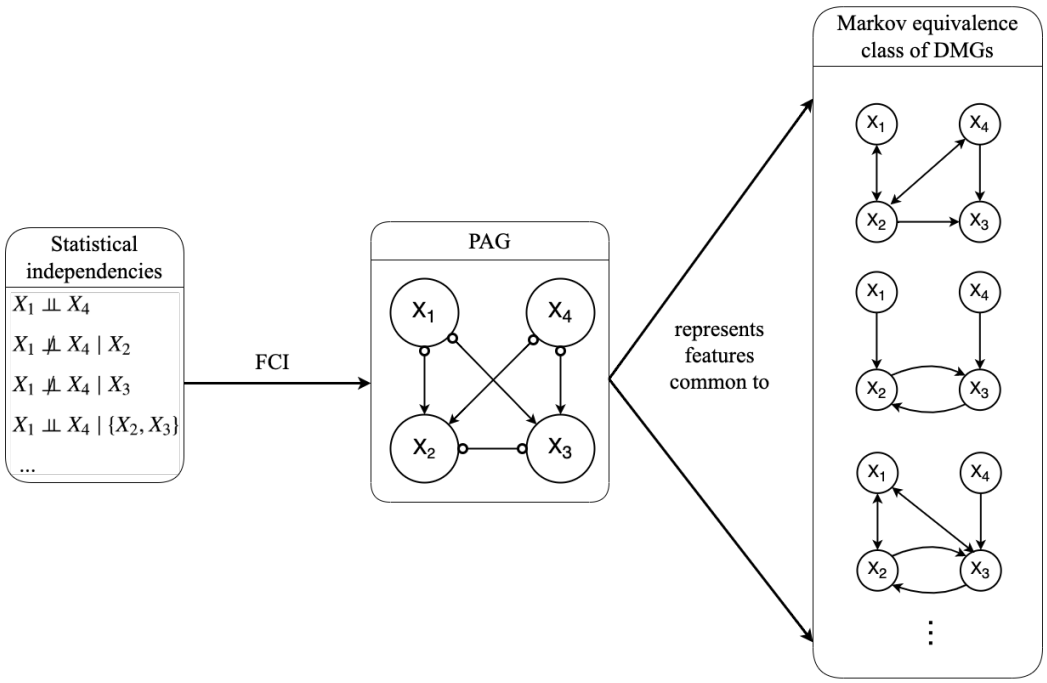
Step 1. Run CCD's skeleton discovery procedure to estimate an ancestral skeleton. This results in Figure 7b.

Step 2. Search for collider structures in the same way as the CCD algorithm. When a collider (B) is identified, orient $A \ast\!\!\rightarrow B \leftarrow\!\!\ast C$ as $A \ast\!\!\rightarrow B \leftarrow\!\!\ast C$. This results in Figure 7c.

Step 3. Execute a set of orientation rules to further orient the edges.[6] In this case, no additional endpoints are oriented, leaving Figure 7c as the final resulting PAG.

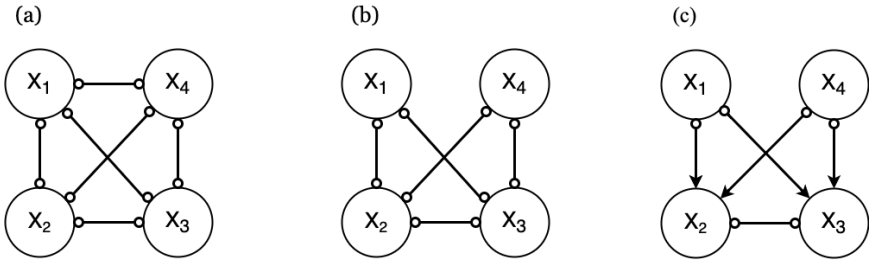
[6] See Appendix B for a complete list of orientation rules (Zhang, 2008).

Figure 6. Summary of FCI algorithm operation.



Note. Given the observed statistical independencies, FCI constructs a partial ancestral graph (PAG), which represents the *ancestral* features that are common to every directed mixed graph (DMG) in a Markov equivalence class. The equivalence class is relatively large, implied by many circle endpoints presented in the PAG (i.e., undetermined directions), which is resulted from the increased uncertainty about causal relations by allowing latent confounders.

Figure 7. Trace of FCI algorithm.



Note. (a) shows the fully-connected PAG, which is the starting point. (b) shows the *ancestral* skeleton estimated in the same way as in the CCD algorithm. (c) shows the state of the PAG after orientation using the collider structures in step 2.

3.3 CCI Algorithm

The CCI algorithm can be seen as a combination of the CCD and FCI algorithms. The algorithm is designed to handle cycles as well as latent confounding simultaneously. However, allowing both cycles and latent confounders comes at a cost; the CCI algorithm has to deal with even greater amount of uncertainty when it comes to learning causal relations and hence more difficult edge-endpoint inferences with more involved orientation rules (see Appendix C for every step of the CCI algorithm in detail).

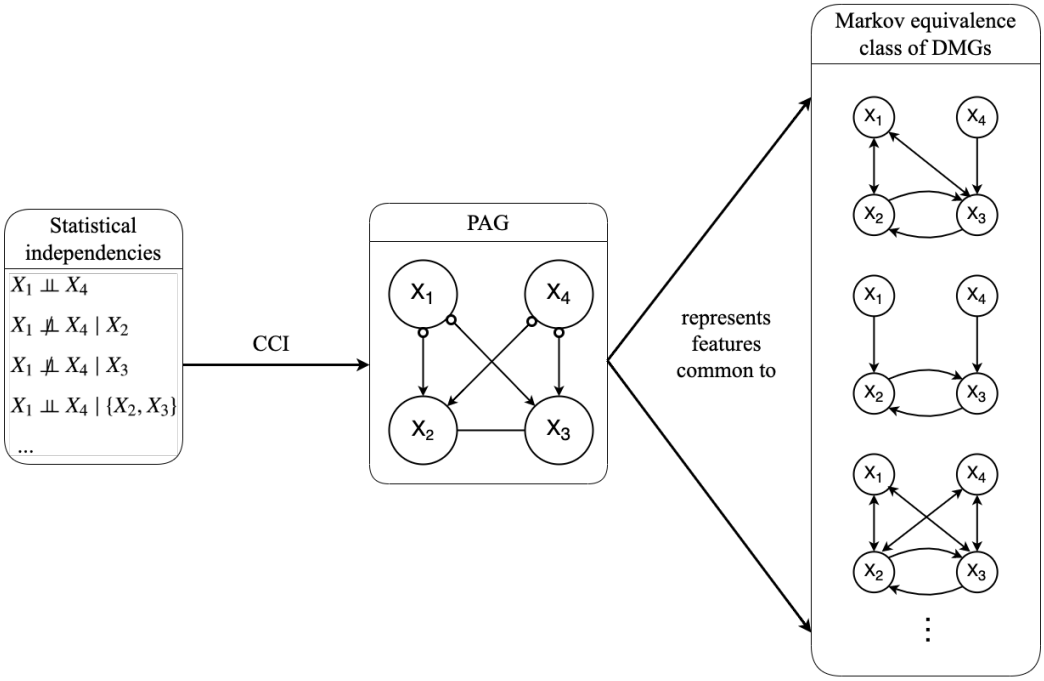
3.3.1 Output Representation: Partial Ancestral Graph (PAG)

As with the other two algorithms, the CCI algorithm outputs a PAG that represents common ancestral features of equivalent graphs. Due to the issue regarding the infinite search space of causal graphs including latent confounders as described in section 3.2.1, the CCI algorithm also uses directed mixed graphs (DMG) where bidirected edges ($\leftrightarrow$) represent presence of latent confounding. Figure 8 summarizes the operation of the CCI algorithm, which is very similar to the one of the FCI algorithm. Each edge-endpoint in PAGs estimated by the CCI algorithm has the same interpretation as described in section 3.1.1. We can, therefore, infer the following given the example PAG in Figure 8:

1. X_2 and X_3 are not ancestors of X_1 and X_4 in every graph in $Equiv(\mathcal{G})$.
2. X_2 is an ancestor of X_3 and X_3 is an ancestor of X_2 in every graph in $Equiv(\mathcal{G})$, which may imply presence of a cycle between them.

Again, notice that the PAG estimated by the CCI algorithm contains more circle endpoints than the PAG estimated by the CCD algorithm, resulting from the fact that the algorithm takes into account the possibility of latent confounding.

Figure 8. Summary of CCI algorithm operation.



Note. Given the observed statistical independencies, CCI constructs a partial ancestral graph (PAG), which represents the *ancestral* features that are common to every directed mixed graph (DMG) in a Markov equivalence class.

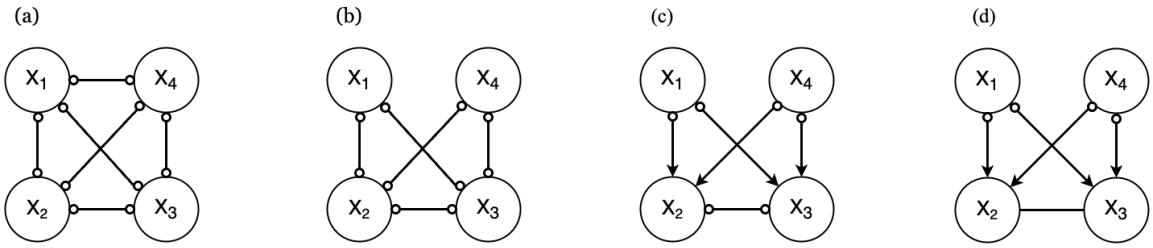
3.3.2 Steps of CCI Algorithm

The CCI algorithm consists of 7 steps in total, in which the first two steps are identical to the other two algorithms (i.e., skeleton discovery and collider structure orientation), and the rest are analogous to the further orientation rules implemented in the CCD and FCI algorithm. In what follows, we briefly illustrate the steps of CCI with the same example DCG (from Figure 1b) that is used throughout the paper.

As with the other two algorithms, it starts with a fully-connected PAG with $\circ\!\!-\!\!\circ$ edges between every pair of variables, as shown in Figure 9a. After running the same skeleton discovery procedure (i.e., step 1), the ancestral skeleton is estimated as in Figure 9b. Upon orienting edges based on identified colliders in the same way as in the other algorithms (i.e., step 2), the CCI algorithm outputs Figure 9c. In the following orientation step (i.e., step 5) utilizing **Supset** as in the step 5 of the CCD algorithm, the edge between X_2 and X_3 is oriented resulting in Figure 9d. Since no

additional edges are oriented in the subsequent steps, Figure 9d remains the final PAG.[7]

Figure 9. Trace of CCI algorithm.



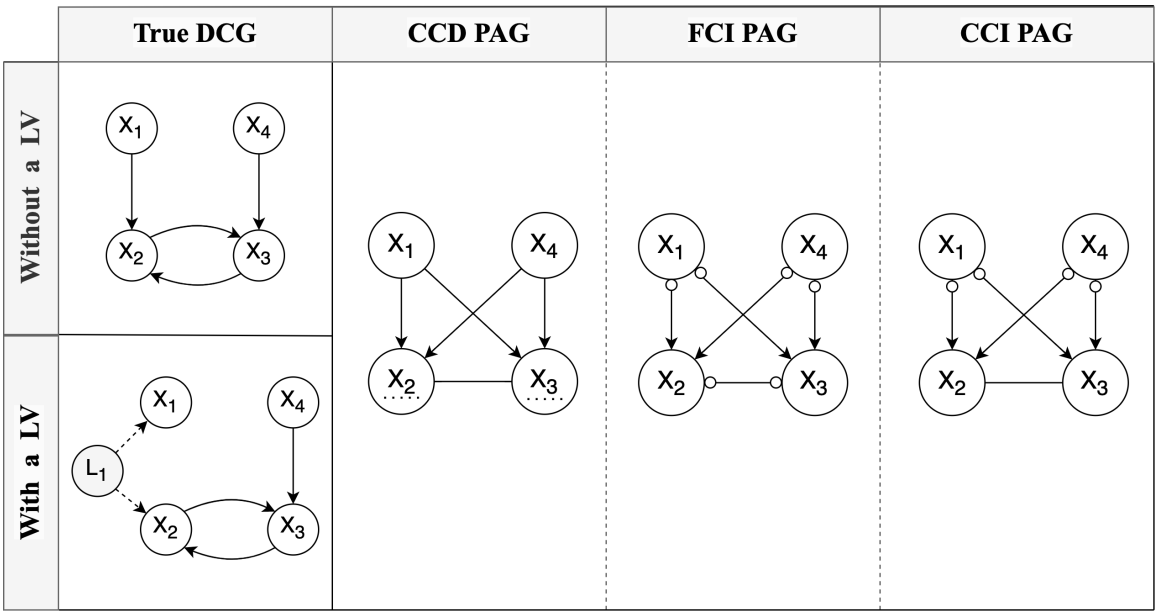
Note. (a) shows the fully-connected PAG, which is the starting point. (b) shows the *ancestral* skeleton estimated in the same ways as in the other algorithms. (c) shows the state of the PAG after orienting the collider structures in step 2. (d) shows the state of the PAG after applying extra orientation rules in step 5. No further orientation is performed in the following steps, leaving (d) as the final PAG.

3.4 Overview of Algorithms

Figure 10 shows the resulting PAGs of each algorithm given two different example DCGs; one without a latent variable and the other one with a latent variable. All three algorithms output the same PAG given the two different DCGs. Suppose the underlying causal structure is the DCG without a latent variable (top left in Figure 10). Even though all three PAGs depict correct ancestral features of the true DCG, the PAG output by CCD is by far the most informative (i.e., no circle endpoints), thus representing the smallest equivalent set of graphs. However, suppose the underlying causal structure is the DCG with a latent variable (bottom left in Figure 10). Then, the PAG output by CCD contains errors (i.e., X_1 is not an ancestor of X_2 and X_3), while the other two PAGs estimated by the FCI and CCI algorithm correctly represent the ancestral features of the true DCG. Additionally, in this case, it can be seen that the PAG output by CCI is more informative than the output by FCI, as it correctly identify the mutual ancestral relationship between X_2 and X_3. To get a better idea of the overall performance of each algorithm under different conditions, we perform a simulation study in the following section. We assess which method works better in terms of accuracy and certainty in estimating causal structures and investigate which factors (e.g., density, presence of latent confounders, sample size) influence their performance.

[7] For a detailed review of each step, see Appendix C.

Figure 10. Overview of algorithms.



Note. LV = latent variable; DCG = directed cyclic graph; PAG = partial ancestral graph; CCD = cyclic causal discovery (algorithm); FCI = fast causal inference (algorithm); CCI = cyclic causal inference (algorithm).

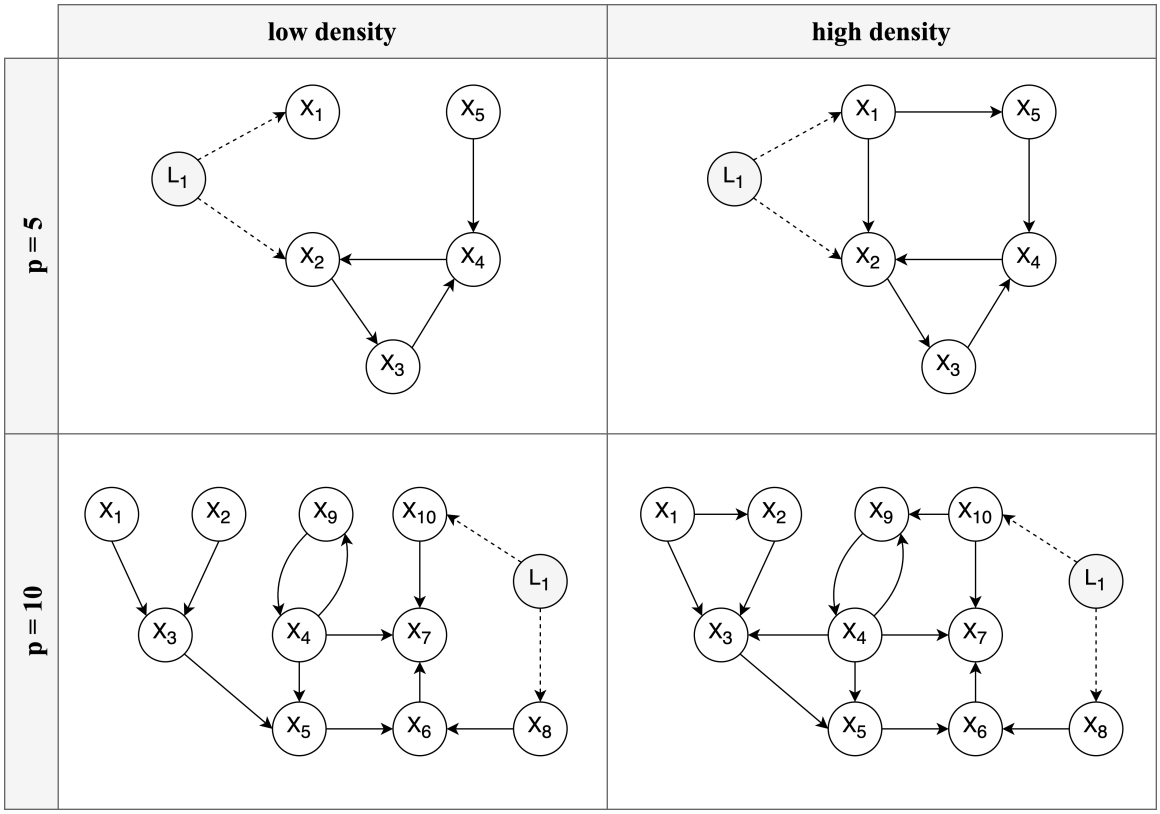
4 Simulation

To evaluate the performance of the considered algorithms, we conduct a simulation study. In this section, we discuss the simulation design, data generating process, and evaluation metrics in detail.

4.1 Simulation Design

We test each algorithm under different conditions by varying the number of variables (rows of Figure 11) and the number of edges – the density (columns of Figure 11). We also evaluate the effect of an unobserved confounder by adding a latent variable (L_1 in Figure 11). Lastly, we vary the sample size ranging from small to moderately large, $n \in \{50, 150, 500, 1000, 5000\}$, for every simulated cyclic model. Thus, it leads to a $2 \times 2 \times 2 \times 5$ design; number of variables $\times$ density $\times$ latent confounder (presence/absence) $\times$ sample size.

Figure 11. Simulation settings.



Note. We vary the number of variables: $p \in \{5, 10\}$, the density: high / low, the influence of a latent confounder (L_1): absence / presence, and the sample size: $n \in \{150, 500, 1000\}$, which results in a $2 \times 2 \times 2 \times 3$ simulation design.

4.2 Data Generation

As illustrated above, we simulate data from different cyclic models, all of which are characterized by *linear* relations and *independent Gaussian* error terms. These types of models are often used in psychological research, and for such cyclic models, the global Markov property – the necessary condition for constraint-based causal discovery – also holds as shown in section 2.2.

To generate data, we first define a coefficient matrix $\mathbf{B}$ and sample the error terms (ε) from independent Gaussian distributions. After drawing the values of ε, we generate observations of $\mathbf{X}$ by solving the following equation: $\mathbf{X} = (\mathbf{I} - \mathbf{B})^{-1}\boldsymbol{\varepsilon}$, where $\mathbf{I}$ denotes the identity matrix. Note that this data generation scheme is possible provided that $(\mathbf{I} - \mathbf{B})$ is invertible, which is the case when the eigenvalues of $\mathbf{B}$ are smaller than one in absolute value, $|\lambda| < 1$ (Eberhardt, Hoyer, & Scheines, 2010). While this is guaranteed if $\mathbf{B}$ defines an acyclic model, for cyclic models, this does not always hold. To satisfy this condition, cyclic relations need to be not too strong such that the dynamical system converges to equilibrium (Rothenhäusler, Heinze, Peters, & Meinshausen, 2015). Therefore, when defining $\mathbf{B}$ matrix, we randomly pick values that are deemed reasonable (i.e., restricting the strength of cyclic relations rather small) and check the eigenvalues to ensure

that the aforementioned condition is met. When violated, we re-scale the parameters and check the eigenvalues again. This fitting process is repeated iteratively until the system reaches equilibrium.

4.3 Evaluation Metrics

For each simulated model, we construct a correct ancestral graph and assess the performance of each algorithm using both *local* and *global* evaluation metrics; at a local level, we look at the individual edge-endpoints and at a global level, we look at the graph structure as a whole. As the local metrics, we utilize *precision*, *recall*, and *uncertainty rate*.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} \ + \ \text{False Positive}}$$

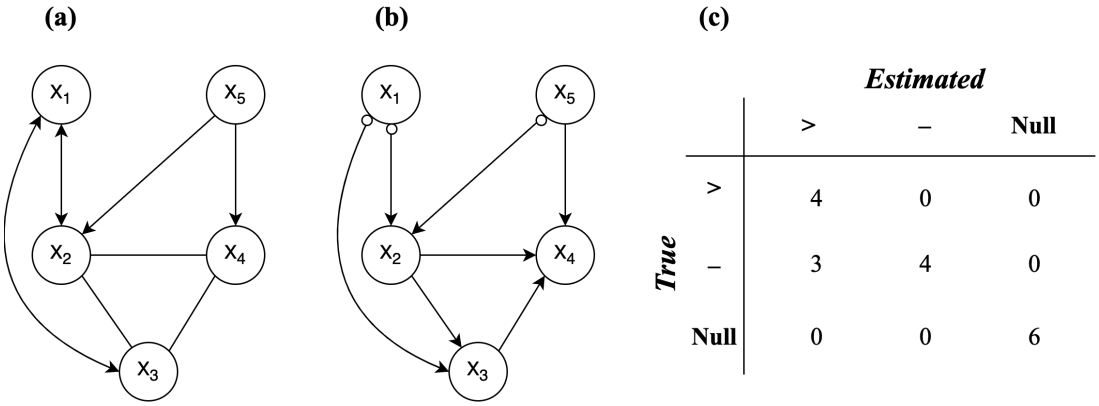
$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} \ + \ \text{False Negative}}$$

$$\text{Uncertainty rate} = \frac{\text{Number of circle endpoints } (\circ)}{\text{Total number of edge-endpoints}}$$

Precision reflects the prediction accuracy (i.e., out of all predicted cases, how many are correct), and recall reflects the retrieval rate (i.e., out of all true cases, how many are retrieved). There are in total four possibilities for each edge-endpoint in a resulting graph: no edge-endpoint (null), arrow head ($>$), arrow tail ($-$), and circle ($\circ$). Given that circle endpoints imply an algorithm is unsure of the direction of causal relations, the uncertainty rate is defined as a proportion of the circle endpoints occurred in an output. Suppose Figure 12b is the estimated PAG output, then the uncertainty rate is calculated as $\frac{3}{20}$ (total number of edge-endpoints $= \binom{5}{2} \times 2 = 20$). For the other endpoints, we calculate the precision and recall. Figure 12a displays the ancestral graph correctly representing all ancestral relations given the example cyclic graph (5-variable low density case with a latent variable) from the top left panel of Figure 11. Presuming that Figure 12b is the estimated PAG output, we can construct a confusion matrix of estimated versus true edge-endpoints. Based on the confusion matrix shown in Figure 12c, we can compute the precision and recall for each type of endpoint. For example, for the arrow head ($>$), they are computed as: $precision = \frac{4}{4+3+0}$ and $recall = \frac{4}{4+0+0}$.

As the global metric, we use *structural Hamming distance* (SHD) (de Jongh & Druzdzel, 2009). SHD quantifies the level of differences between two graphs by counting the number of edge insertions, deletions, and direction changes required to move from one graph (estimated graph $\hat{\mathcal{G}}$) to the other (true graph $\mathcal{G}$). It can be formulated as: $SHD = A + D + C$, where A, D, and C represent, respectively, the number of added edges, deleted edges, and direction changes. Thus, the smaller the SHD value is, the more similar $\hat{\mathcal{G}}$ is to $\mathcal{G}$, indicating that an algorithm recovers the true graph well. For instance, the value of SHD for the example PAG output from Figure 12b – provided that the true ancestral graph is Figure 12a – is 6, which is calculated by summing: 0 (A) + 0 (D) + 6 (C).

Figure 12. Example evaluation metrics.



Note. (a) depicts the correct ancestral graph of the example DCG shown in the top left panel of Figure 11 (i.e., 5-variable low density case with a latent variable). (b) depicts an example estimated PAG output. (c) is the confusion matrix of estimated versus true edge-endpoint for three different types of endpoints. The true endpoints are presented in rows, and the estimated endpoints are presented in columns. There are in total four possible edge-endpoints that can occur in an output: arrow head (>), arrow tail (−), null (no endpoint), and circle (∘). The circle endpoints, however, are not counted toward the calculation of *precision* and *recall* but are used for calculating the *uncertainty rate*.

References

Bollen, K. A., & Pearl, J. (2013). Eight Myths About Causality and Structural Equation Models. In S. L. Morgan (Ed.), *Handbook of Causal Analysis for Social Research* (pp. 301–328). Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-94-007-6094-3_15

Bongers, S., Forré, P., Peters, J., & Mooij, J. M. (2021). Foundations of structural causal models with cycles and latent variables. *The Annals of Statistics*, *49*(5), 2885 – 2915. https://doi.org/10.1214/21-AOS2064

Borsboom, D. (2017). A network theory of mental disorders. *World Psychiatry*, *16*(1), 5–13. https://doi.org/10.1002/wps.20375

Borsboom, D., & Cramer, A. O. (2013). Network analysis: An integrative approach to the structure of psychopathology. *Annual Review of Clinical Psychology*, *9*(1), 91-121. (PMID: 23537483) https://doi.org/10.1146/annurev-clinpsy-050212-185608

Borsboom, D., Deserno, M. K., Rhemtulla, M., Epskamp, S., Fried, E. I., McNally, R. J., ... Waldorp, L. J. (2021, August). Network analysis of multivariate data in psychological science. *Nature Reviews Methods Primers*, *1*(1), 1–18. https://doi.org/10.1038/s43586-021-00055-w

Briganti, G., Scutari, M., & McNally, R. J. (2022). A tutorial on bayesian networks for psychopathology researchers. *Psychological Methods*. https://doi.org/10.1037/met0000479

Colombo, D., Maathuis, M. H., Kalisch, M., & Richardson, T. S. (2012). Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, *40*(1), 294 – 321. Retrieved from `https://doi.org/10.1214/11-AOS940` https://doi.org/10.1214/11-AOS940

Constantin, M., & Cramer, A. O. J. (2022). Sample size recommendations for estimating cross-sectional network models. *OSF*. https://doi.org/10.17605/OSF.IO/ZKAXU

Dablander, F., & Hinne, M. (2019, May). Node centrality measures are a poor substitute for causal inference. *Scientific Reports*, *9*(1), 6846. https://doi.org/10.1038/s41598-019-43033-9

de Jongh, M., & Druzdzel, M. J. (2009). A comparison of structural distance measures for causal bayesian network models. *Recent Advances in Intelligent Information Systems, Challenging Problems of Science, Computer Science series*, 443–456.

Eberhardt, F., Hoyer, P., & Scheines, R. (2010, 13–15 May). Combining experiments to discover linear cyclic models with latent variables. In Y. W. Teh & M. Titterington (Eds.), *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (Vol. 9, pp. 185–192). PMLR.

Forré, P., & Mooij, J. M. (2017). Markov Properties for Graphical Models with Cycles and Latent Variables. *arXiv preprint arXiv:1710.08775*. https://doi.org/10.48550/arXiv.1710.08775

Geiger, D., & Pearl, J. (1990). On the logic of causal models. In R. D. Shachter, T. S. Levitt, L. N. Kanal, & J. F. Lemmer (Eds.), *Machine intelligence and pattern recognition* (Vol. 9, pp. 3–14). North-Holland. https://doi.org/10.1016/B978-0-444-88650-7.50006-8

Geiger, D., Verma, T., & Pearl, J. (1990). d-Separation: From Theorems to Algorithms. In

M. Henrion, R. D. Shachter, L. N. Kanal, & J. F. Lemmer (Eds.), *Machine Intelligence and Pattern Recognition* (Vol. 10, pp. 139–148). North-Holland. https://doi.org/10.1016/B978-0-444-88738-2.50018-X

Glymour, C., Zhang, K., & Spirtes, P. (2019). Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, *10*, 524. https://doi.org/10.3389/fgene.2019.00524

Haslbeck, J. M. B., Ryan, O., Robinaugh, D. J., Waldorp, L. J., & Borsboom, D. (2021, November). Modeling psychopathology: From data models to formal theories. *Psychological Methods*. https://doi.org/10.1037/met0000303

Kossakowski, J., Waldorp, L. J., & van der Maas, H. L. J. (2021). The search for causality: A comparison of different techniques for causal inference graphs. *Psychological Methods*, *26*(6), 719–742. https://doi.org/10.1037/met0000390

Lauritzen, S. (1996). *Graphical models* (Vol. 17). Clarendon Press.

Lauritzen, S. (2000). *Graphical models for causal inference.* Complex Stochastic Systems. London/Boca Raton: Chapman and Hall/CRC Press.

Mooij, J. M., & Claassen, T. (2020, Aug). Constraint-based causal discovery using partial ancestral graphs in the presence of cycles. In J. Peters & D. Sontag (Eds.), *Proceedings of the 36th conference on uncertainty in artificial intelligence (uai)* (Vol. 124, pp. 1159–1168). PMLR.

Pearl, J. (2010). Causal Inference. In *Proceedings of Workshop on Causality: Objectives and Assessment at NIPS 2008* (pp. 39–58). PMLR.

Richardson, T. (1996a). *Discovering cyclic causal structure*. Carnegie Mellon [Department of Philosophy].

Richardson, T. (1996b). A discovery algorithm for directed cyclic graphs. In *Proceedings of the twelfth international conference on uncertainty in artificial intelligence* (p. 454–461). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Richardson, T. (2003). Markov Properties for Acyclic Directed Mixed Graphs. *Scandinavian Journal of Statistics*, *30*(1), 145–157. https://doi.org/10.1111/1467-9469.00323

Richardson, T., & Spirtes, P. (2002). Ancestral graph Markov models. *The Annals of Statistics*, *30*(4), 962 – 1030. https://doi.org/10.1214/aos/1031689015

Robinaugh, D. J., Hoekstra, R. H. A., Toner, E. R., & Borsboom, D. (2020). The network approach to psychopathology: a review of the literature 2008–2018 and an agenda for future research. *Psychological Medicine*, *50*(3), 353–366. https://doi.org/10.1017/S0033291719003404

Rothenhäusler, D., Heinze, C., Peters, J., & Meinshausen, N. (2015, November). *backShift: Learning causal cyclic graphs from unknown shift interventions.* arXiv. (arXiv:1506.02494 [stat]) https://doi.org/10.48550/arXiv.1506.02494

Ryan, O., Bringmann, L. F., & Schuurman, N. K. (2022). The challenge of generating causal hypotheses using network models. *Structural Equation Modeling: A Multidisciplinary Journal*, *0*(0), 1-18. https://doi.org/10.1080/10705511.2022.2056039

Spirtes, P. (1993). Directed cyclic graphs, conditional independence, and non-recursive linear

structural equation models..

Spirtes, P. (1994). Conditional independence in directed cyclic graphical models for feedback..

Spirtes, P., & Glymour, C. (1991). An Algorithm for Fast Recovery of Sparse Causal Graphs. *Social Science Computer Review*, *9*(1), 62–72. (Publisher: SAGE Publications Inc) https://doi.org/10.1177/089443939100900106

Spirtes, P., Glymour, C. N., Scheines, R., & Heckerman, D. (2000). *Causation, prediction, and search*. MIT press.

Spirtes, P., Meek, C., & Richardson, T. (1999). An algorithm for causal inference in the presence of latent variables and selection bias. *Computation, causation, and discovery*, *21*, 211–252.

Strobl, E. V. (2019). A constraint-based algorithm for causal discovery with cycles, latent variables and selection bias. *International Journal of Data Science and Analytics*, *8*(1), 33–56. https://doi.org/10.1007/s41060-018-0158-2

Wittenborn, A. K., Rahmandad, H., Rick, J., & Hosseinichimeh, N. (2016). Depression as a systemic syndrome: mapping the feedback loops of major depressive disorder. *Psychological Medicine*, *46*(3), 551–562. https://doi.org/10.1017/S0033291715002044

Zhang, J. (2008). On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, *172*(16), 1873–1896. https://doi.org/10.1016/j.artint.2008.08.001

Zhang, J., & Spirtes, P. (2005). *A Characterization of Markov Equivalence Classes for Ancestral Graphical Models.*

Appendix A CCD Details

Algorithm 1 Cyclic Causal Discovery (CCD)

Input: A conditional independent oracle for a distribution $\mathcal{P}$, satisfying global directed Markov property and faithfulness conditions with respect to a directed graph $\mathcal{G}$ with vertex set $\mathcal{V}$.

Output: A PAG Ψ for the Markov equivalence class Equiv($\mathcal{G}$).

Step 1. Form a complete graph (Ψ) with the edge $\circ\!\!-\!\!\circ$ between every pair of vertices in $\mathcal{V}$.
$n = 0$
repeat
  repeat
    Select an ordered pair of variables X and Y that are adjacent in Ψ such that the number of vertices in **Adjacent**$(\Psi, X)\backslash\{Y\} \geq n$, and select a subset $\mathcal{S}$ of **Adjacent**$(\Psi, X)\backslash\{Y\}$ with n vertices.
    If $X \perp\!\!\!\perp Y \mid \mathcal{S}$, then delete the edge $X\circ\!\!-\!\!\circ Y$ and record $\mathcal{S}$ in **Sepset**$\langle X, Y\rangle$ and **Sepset**$\langle X, Y\rangle$.
  until all pairs of adjacent variables X and Y such that the number of vertices in **Adjacent**$(\Psi, X)\backslash\{Y\} \geq n$ and all sets $\mathcal{S}$ such that the number of vertices in $\mathcal{S} = n$ have been tested.
  $n = n + 1$;
until for all ordered pairs of adjacent vertices X and Y, **Adjacent**$(\Psi, X)\backslash\{Y\} < n$.
Step 2. For each triple of vertices A, B, C such that each of the pair of A, B and the pair B, C are adjacent in Ψ but the pair A, C are not adjacent in Ψ, then:
(i) orient $A{*\!\!-\!\!*}B{*\!\!-\!\!*}C$ as $A \rightarrow B \leftarrow C$ *iff* $B \notin$ **Sepset**$\langle A, B\rangle$.
(ii) orient $A{*\!\!-\!\!*}B{*\!\!-\!\!*}C$ as $A{*\!\!-\!\!*}\underline{B}{*\!\!-\!\!*}C$ *iff* $B \in$ **Sepset**$\langle A, B\rangle$.
Step 3. For each triple of vertices A, X, Y in Ψ such that (a) A is not adjacent to X or Y, (b) X and Y are adjacent, (c) $X \notin$ **Sepset**$\langle A, Y\rangle$, then orient $X{*\!\!-\!\!*}Y$ as $X \leftarrow Y$ if $A \not\perp\!\!\!\perp X \mid$ **Sepset**$\langle A, Y\rangle$.
Step 4. For each vertex V in Ψ form the following set: $X \in$ **Local**(Ψ, V) or there is a vertex Y such that $X \rightarrow Y \leftarrow V$ in Ψ.
$m = 0$
repeat
  repeat
    Select an ordered triple $\langle A, B, C\rangle$ such that $A \rightarrow B \leftarrow C$, A and C are not adjacent, and **Local**$(\Psi, A)\backslash\{B, C\}$ has $\geq m$ vertices.
    Select a set $T \subseteq$ **Local**$(\Psi, A)\backslash\{B, C\}$ with m vertices. If $A \perp\!\!\!\perp C \mid T \cup \{B\}$, then orient $A \rightarrow B \leftarrow C$ as $A \rightarrow \underline{B} \leftarrow C$ and record $T \cup \{B\}$ in **Supset**$\langle A, B, V\rangle$.

until for all triples such that $A \rightarrow B \leftarrow C$ (not $A \rightarrow \underline{\mathrm{B}} \leftarrow C$), A and C are not adjacent, **Local**$(\Psi, A)\backslash\{B\}$ has $\geq m$ vertices, every subset T with m vertices has been considered.

$m = m + 1$;

until all ordered triples $\langle A, B, C\rangle$ such that $A \rightarrow B \leftarrow C$, A and C are not adjacent, are such that **Local**$(\Psi, A)\backslash\{B\}$ have $< m$ vertices.

Step 5. If there is a quadruple A, B, C, D in Ψ of distinct vertices such that:
(i) $A \rightarrow \underline{\mathrm{B}} \leftarrow C$,
(ii) $A \rightarrow D \leftarrow C$ or $A \rightarrow \underline{\mathrm{D}} \leftarrow C$,
(iii) B and D are adjacent,
then orient $B{*\!\!-\!\!*}D$ ad $B \rightarrow D$ in Ψ if $D \notin$ **Subset**$\langle A, B, C\rangle$. Else orient $B{*\!\!-\!\!*}D$ as $B{*\!\!-}D$ in Ψ.
Step 6. For each quadruple A, B, C, D in Ψ of distinct vertices such that:
(i) D is not adjacent to both A and C,
(ii) $A \rightarrow \underline{\mathrm{B}} \leftarrow C$,
if $A \not\perp\!\!\!\perp D \mid$ **Supset**$\langle A, B, C\rangle \cup \{D\}$, then orient $B{*\!\!-\!\!*}D$ as $B \rightarrow D$ in Ψ.

Appendix B FCI Details

Algorithm 2 Fast Causal Inference (FCI)

Input: A conditional independent oracle for a distribution $\mathcal{P}$, satisfying global directed Markov property and faithfulness conditions with respect to a directed graph $\mathcal{G}$ with vertex set $\mathcal{V}$.

Output: A PAG $\hat{\mathcal{G}}'$ for the Markov equivalence class of DMGs (Equiv($\mathcal{G}$)).

Step 1. Form the complete undirected graph Q on the vertex set $\mathcal{V}$.
$n = 0$
repeat
  repeat
    Select an ordered pair of variables X and Y that are adjacent in Q such that the number of vertices in **Adjacent**$(Q, X)\backslash\{Y\} \geq n$, and select a subset $\mathcal{S}$ of **Adjacent**$(Q, X)\backslash\{Y\}$ with n vertices.
    If $X \perp\!\!\!\perp Y \mid \mathcal{S}$, then delete the edge $X{\circ\!\!-\!\!\circ}Y$ and record $\mathcal{S}$ in **Sepset**$\langle X, Y\rangle$ and **Sepset**$\langle X, Y\rangle$.
  until all pairs of adjacent variables X and Y such that the number of vertices in **Adjacent**$(Q, X)\backslash\{Y\} \geq n$ and all sets $\mathcal{S}$ such that the number of vertices in $\mathcal{S} = n$ have been tested.
  $n = n + 1$;

7: **until** for all ordered pairs of adjacent vertices X and Y, $\mathbf{Adjacent}(Q, X)\backslash\{Y\} < n$.

8: ***Step 2.*** Let Q' be the undirected graph resulting from step 1. For each triplet $\langle A, B, C\rangle$ such that each of the pair of A, B and the pair B, C are adjacent in Q' but the pair A, C are not adjacent in Q', then orient $A{*\!\!-\!\!*}B{*\!\!-\!\!*}C$ as $A{*\!\!\rightarrow}B{\leftarrow\!\!*}C$ *iff* $B \notin \mathbf{Sepset}\langle A, B\rangle$.

9: ***Step 3.*** For each pair of variables A and B adjacent in Q', if A and B are d-separated given any subset S of $\mathbf{Possible\text{-}d\text{-}sepset}\langle A, B\rangle)\backslash\{A, B\}$ or any subset S of $\mathbf{Possible\text{-}d\text{-}sepset}\langle B, A\rangle)\backslash\{A, B\}$ in Q', then remove the edge between A and B, and record S in $\mathbf{Sepset}\langle A, B\rangle$ and $\mathbf{Sepset}\langle B, A\rangle$.

10: ***Step 4.*** Execute the following orientation rules iteratively until none applies:

(i) If $A {*\!\!\rightarrow} B {\circ\!\!-\!\!*} C$, and A and C are not adjacent, then orient the triple as $A{*\!\!\rightarrow}B{\rightarrow}C$.

(ii) If $A{\rightarrow}B{*\!\!\rightarrow}C$ or $A{*\!\!\rightarrow}B{\rightarrow}C$, and $A {*\!\!-\!\!\circ} C$, then orient $A {*\!\!-\!\!\circ} C$ as $A {*\!\!\rightarrow} C$.

(iii) If $A{*\!\!\rightarrow}B{\leftarrow\!\!*}C$, $A{*\!\!-\!\!\circ}D{\circ\!\!-\!\!*}C$, A and C are not adjacent, and $D{*\!\!-\!\!\circ}B$, then orient $D{*\!\!-\!\!\circ}B$ ad $D{*\!\!\rightarrow}B$.

(iv) If $u = \langle D, ..., A, B, C\rangle$ is a discriminating path between D and C for B, and $B{\circ\!\!-\!\!*}C$; then if $B \in \mathbf{Sepset}\langle D, C\rangle$, orient $B{\circ\!\!-\!\!*}C$ as $B{\rightarrow}C$; otherwise orient the triple $\langle A, B, C\rangle$ as $A{\leftrightarrow}B{\leftrightarrow}C$.

(v) For every (remaining) $A{\circ\!\!-\!\!\circ}B$, if there is an uncovered circle path $p = \langle A, C, ..., D, B\rangle$ between A and B such that A, D are not adjacent and B, C are not adjacent, then orient $A{\circ\!\!-\!\!\circ}B$ and every edge on p as undirected edges $(-)$.

(vi) If A — $B{\circ\!\!-\!\!*}C$ (A and C may or may not be adjacent), then orient $B{\circ\!\!-\!\!*}C$ as $B{-\!\!*}C$.

(vii) If $A{-\!\!\circ}B{\circ\!\!-\!\!*}C$ and A, C are not adjacent, then orient $B{\circ\!\!-\!\!*}C$ ad $B{-\!\!*}C$.

(viii) If $A{\rightarrow}B{\rightarrow}C$ or $A{-\!\!\circ}B{\rightarrow}C$, and $A{\circ\!\!\rightarrow}C$, orient $A{\circ\!\!\rightarrow}C$ as $A{\rightarrow}C$.

(ix) If $A{\circ\!\!\rightarrow}C$, and $p = \langle A, B, D, ..., C\rangle$ is an uncovered potentially directed path from A to C such that C and B are not adjacent, then orient $A{\circ\!\!\rightarrow}C$ as $A{\rightarrow}C$.

(x) Suppose $A{\circ\!\!\rightarrow}C$, $B{\rightarrow}C{\leftarrow}D$, p_1 is an uncovered potentially directed path from A to B, and p_2 is an uncovered potentially directed path from A to D. Let m be the vertex adjacent to A on p_1 (m could be B), and w be the vertex adjacent to A on p_2 (w could be D). If m and w are distinct, and are not adjacent, then orient $A{\circ\!\!\rightarrow}D$ as $A{\rightarrow}D$.

Appendix C CCI Details

Algorithm 3 Cyclic Causal Inference (CCI)

Input: A conditional independent oracle for a distribution $\mathcal{P}$, satisfying global directed Markov property and faithfulness conditions w.r.t. a directed graph $\mathcal{G}$ with vertex set $\mathcal{V}$. ($\mathcal{V} = \mathbf{O} \cup \mathbf{L} \cup \mathbf{S}$), where **O**, **L**, and **S** refer to the sets of observable, latent, and selection variables, respectively.

Output: A PAG ($\hat{\mathcal{G}}'$) for the Markov equivalence class of DMGs (Equiv($\mathcal{G}$)).

Step 1. Run FCI's skeleton discovery procedure.

Step 2. Run FCI's collider structure (v-structure) orientation procedure.

Step 3. For any triplet $\langle O_i, O_k, O_j \rangle$, such that we have $O_k \circ\!\!-\!\!* O_i$, if $O_i \perp\!\!\!\perp_{\mathcal{G}} O_j \mid \mathbf{Sepset}\langle O_i, O_j \rangle \cup \mathbf{S}$, where $\mathbf{Sepset}\langle O_i, O_j \rangle$ is a separating set discovered in step 1, $O_k \notin \mathbf{Sepset}\langle O_i, O_j \rangle$, $O_i \not\perp\!\!\!\perp_{\mathcal{G}} O_k \mid \mathbf{Sepset}\langle O_i, O_j \rangle \cup \mathbf{S}$ and $O_j \not\perp\!\!\!\perp_{\mathcal{G}} O_k \mid \mathbf{Sepset}\langle O_i, O_j \rangle \cup \mathbf{S}$, then orient $O_k \circ\!\!-\!\!* O_i$ as $O_k \leftarrow\!\!* O_i$.

Step 4. Find additional non-minimal d-separating sets.

$m = 0$

repeat

  repeat

    Select an ordered triplet $\langle O_i, O_j, O_k \rangle$ with the collider structure $O_i *\!\!\rightarrow O_j \leftarrow\!\!* O_k$ such that $|\mathbf{PD\text{-}Sep}(O_i)| \geq m$

    repeat

      Select a subset
      $\mathbf{W} \subseteq \mathbf{PD\text{-}Sep}(O_i) \backslash \{\mathbf{Sepset}\langle O_i, O_k \rangle \cup \{O_j, O_k\}\}$ with m vertices
      $\mathbf{T} = \mathbf{W} \cup \mathbf{Sepset}\langle O_i, O_k \rangle \cup O_j$
      if O_i and O_k are d-separated given $\mathbf{T} \cup \mathbf{S}$, then record the set **T** in $\mathbf{Supset}\langle O_i, O_j, O_k \rangle$

    until all subsets
      $\mathbf{W} \subseteq \mathbf{PD\text{-}Sep}(O_i) \backslash \{\mathbf{Sepset}\langle O_i, O_k \rangle \cup \{O_j, O_k\}\}$ have been considered or a d-separating set of O_i and O_k has been recorded in $\mathbf{Supset}\langle O_i, O_j, O_k \rangle$;

  until all triplets $\langle O_i, O_j, O_k \rangle$ with the collider structure $O_i *\!\!\rightarrow O_j \leftarrow\!\!* O_k$ and $|\mathbf{PD\text{-}Sep}(O_i)| \geq m$ have been selected;

until all ordered triplets $\langle O_i, O_j, O_k \rangle$ with the collider structure $O_i *\!\!\rightarrow O_j \leftarrow\!\!* O_k$ have $|\mathbf{PD\text{-}Sep}(O_i)| < m$;

Step 5. Find all quadruples of vertices $\langle O_i, O_j, O_k, O_l \rangle$ such that O_i and O_k are non-adjacent, $O_i *\!\!\rightarrow O_l \leftarrow\!\!* O_k$, and $O_i \perp\!\!\!\perp_{\mathcal{G}} O_k \mid \mathbf{W} \cup \mathbf{S}$ with $O_j \in \mathbf{W}$ and $\mathbf{W} \subseteq \mathbf{O} \backslash \{O_i, O_k\}$. If $O_l \notin \mathbf{W} = \mathbf{Sepset}\langle O_i, O_k \rangle$ as discovered in step 2, then orient $O_j *\!\!-\!\!\circ O_l$ as $O_j *\!\!\rightarrow O_l$. If we also have $O_i *\!\!\rightarrow O_j \leftarrow\!\!* O_k$ and $O_l \in \mathbf{W} = \mathbf{Supset}\langle O_i, O_j, O_k \rangle$ as discovered in step 4, then orient $O_j *\!\!-\!\!\circ O_l$ as $O_j *\!\!-\!\!* O_l$.

15: ***Step 6.*** For any two vertices O_i and O_k, if we have $O_i \perp\!\!\!\perp_{\mathcal{G}} O_k \mid \mathbf{W} \cup \mathbf{S}$ for some $\mathbf{W} \subseteq \mathbf{O} \backslash \{O_i, O_k\}$ discovered in step 1 or step 4 with $O_j \in \mathbf{W}$ but we have $O_i \not\perp\!\!\!\perp_{\mathcal{G}} O_k \mid O_l \cup \mathbf{W} \cup \mathbf{S}$, then orient $O_l \circ\!\!-\!\!* O_j$ as $O_l \leftarrow\!\!* O_j$.

16: ***Step 7.*** Execute orientation rules until no more endpoints can be oriented. The orientation rules are as follows:

(i) If we have $O_i *\!\!\rightarrow O_j \circ\!\!-\!\!* O_k$ with O_i and O_k non-adjacent, then orient $O_j \circ\!\!-\!\!* O_k$ as $O_j \!-\!\!* O_k$. Furthermore, if $O_i *\!\!\rightarrow O_j$ is not potentially 2-triangulated w.r.t. O_k, then orient $O_j \!-\!\circ O_k$ as $O_j \rightarrow O_k$.

(ii) If we have $O_i \!-\!\!* O_j \circ\!\!-\!\!* O_k$ with O_i and O_k non-adjacent, and $O_j \circ\!\!- O_k$ is not potentially 2-triangulated w.r.t. O_i, then orient $O_j \circ\!\!-\!\!* O_k$ as $O_j \!-\!\!* O_k$.

(iii) Suppose we have $O_i *\!\!\rightarrow O_j \!-\! O_k$ with O_i and O_k non-adjacent, and $O_i *\!\!\rightarrow O_j$ is potentially 2-triangulated w.r.t. O_k. If $O_i *\!\!\rightarrow O_j$ can be potentially 2-triangulated w.r.t. O_k using only one vertex O_l in the triangle involve $\{O_i, O_j, O_l\}$, then orient $O_i *\!\!-\!\circ O_l$ as $O_i *\!\!\rightarrow O_l$, $O_j \circ\!\!-\!\!* O_l$ as $O_j *\!\!\rightarrow O_l$ and/or $O_j *\!\!-\!\circ O_l$ as $O_j *\!\!- O_l$. Next, if there exists only one potentially undirected path $\prod_{O_l, O_k}$ between O_l and O_k, then substitute all circle endpoints on $\prod_{O_l, O_k}$ with tail endpoints (−).

(iv) If $O_i *\!\!\rightarrow O_j \!-\!\!* O_k$, there exists a path $\prod = \langle O_k, ..., O_i \rangle$ with at least $n \geq 3$ vertices such that we have $O_h \!-\!\!* O_{h+1}$ for all $1 \leq i \leq n-1$, and we have $O_1 \circ\!\!-\!\!* O_n$, then orient $O_1 \circ\!\!-\!\!* O_n$ as $O_1 \!-\!\!* O_n$.

(v) If we have the sequence of vertices $\langle O_1, ..., O_n \rangle$ such that $O_i \!-\!\!* O_{i+1}$ with $1 \leq i \leq n-1$, and we have $O_1 \circ\!\!-\!\!* O_n$, then orient $O_1 \circ\!\!-\!\!* O_n$ as $O_1 \!-\!\!* O_n$.

(vi) If we have $O_k *\!\!-\!\circ O_i$, there exists a non-potentially 2-triangulated path $\prod = \langle O_i, O_j, O_l, ..., O_k \rangle$ such that $O_k *\!\!-\!\circ O_i$ is not potentially 2-triangulated w.r.t. O_j, and $O_j *\!\!-\!\!* O_i *\!\!-\!\!* O_k$ is an unshielded non-v-structure, then orient $O_k *\!\!-\!\circ O_i$ as $O_k *\!\!- O_i$.

(vii) Suppose we have $O_i \circ\!\!- O_k$, $O_i \!-\!\!* O_k *\!\!- O_l$, a non-potentially 2-triangulated path $\prod_1$ from O_i to O_j, and a non-potentially 2-triangulated path $\prod_2$ from O_i to O_l. Let O_m be a vertex adjacent to O_i on $\prod_1$, and let O_n be the vertex adjacent to O_i on $\prod_2$. If further $O_m *\!\!-\!\!* O_i *\!\!-\!\!* O_n$ is an unshielded non-v-structure and $O_i \circ\!\!-\!\!* O_k$ is not potentially 2-triangulated w.r.t. both O_n and O_m, then orient $O_i \circ\!\!-\!\!* O_k$ as $O_i \!-\!\!* O_k$.