Establishing Markov Equivalence in Cyclic Directed Graphs

Tom Claassen[1] **Joris M. Mooij**[2]

[1]Institute for Computing and Information Sciences, Radboud University, Nijmegen, Netherlands
[2]Korteweg-deVries Institute, University of Amsterdam, Amsterdam, Netherlands

Abstract

We present a new, efficient procedure to establish Markov equivalence between directed graphs that may or may not contain cycles under the *d*-separation criterion. It is based on the Cyclic Equivalence Theorem (CET) in the seminal works on cyclic models by Thomas Richardson in the mid '90s, but now rephrased from an ancestral perspective. The resulting characterization leads to a procedure for establishing Markov equivalence between graphs that no longer requires tests for *d*-separation, leading to a significantly reduced algorithmic complexity. The conceptually simplified characterization may help to reinvigorate theoretical research towards sound and complete cyclic discovery in the presence of latent confounders.
This version includes a correction to rule (iv) in Theorem 1, and the subsequent adjustment in part 2 of Algorithm 2.

1 INTRODUCTION

Discovering causal relations from observational and experimental data is one of the key goals in many research areas. Developing principled, automated causal discovery methods has been an active area of research within the machine learning community, which has resulted in a wide variety of algorithms and techniques. Two of the main challenges here are handling the impact of unobserved confounders, and the possible presence of feedback mechanisms or cycles in the system under investigation. Both have a long history in the field: in this article we solely focus on the latter.

Building on earlier work by Spirtes (1994, 1995) on (linear) cyclic directed models that obey the global directed Markov property (see section 2.1, below), Richardson (1996b) introduced the Cyclic Causal Discovery (CCD) algorithm that was able to infer a sound cyclic causal model from independence constraints on data. It was based on the so-called Cyclic Equivalence Theorem (Richardson, 1997) that characterized Markov equivalence between cyclic directed graphs.

Strangely enough, after this promising start progress in cyclic directed models slowly ground to a halt, even though many challenges remained: the CCD output was certainly not complete, and could not account for latent confounders.

In the mean time theory and methods for acyclic causal discovery took flight, where, for example Zhang (2008) managed to extend FCI to a provably sound and complete algorithm under latent confounders and selection bias.

And even to this day fundamental progress continues to be made: recently several new and faster algorithms and characterizations for establishing Markov equivalence between maximal ancestral graphs (graphical independence models closed under marginalization and conditioning) have been developed (Hu and Evans, 2020; Wienöbst et al., 2022; Claassen and Bucur, 2022), ultimately bringing it down to linear complexity for sparse graphs. However, despite a widely acknowledged need to handle feedback cycles in learning algorithms for real world causal discovery, major steps towards that goal have been few and far between.

A promising attempt to extend CCD to the case of unobserved confounders was made by Strobl (2018), but though the resulting CCI algorithm was sound, it was by no means complete, foregoing on key FCI elements like discriminating paths and selection bias, and the output was not guaranteed to uniquely identify the Markov equivalence class.

Fundamentally different approaches to cyclic causal discovery have also been developed: for example, Lacerda et al. (2008) employs independent component analysis, Mooij et al. (2011); Mooij and Heskes (2013) proposed likelihood-based structure learning approaches for additive noise models, Hyttinen et al. (2012) exploits experiments to build a complete model, and Rothenhäusler et al. (2015) builds on information from unknown shift interventions to reconstruct the underlying cyclic causal graph.

On another front, Forré and Mooij (2018) showed that for *nonlinear* causal models with cycles and confounders, the usual d-separation criterion needs to be replaced with their σ-separation criterion (see also section 3 in the supplement). More recently, Mooij and Claassen (2020) showed that vanilla FCI was in fact already sound and complete for these nonlinear cyclic models. However, it does not account for the peculiarities encountered when handling *linear* cyclic models, as in Figure 1.

For linear or discrete cyclic causal models, σ-separation is too weak, as the stronger d-separation may apply. Perhaps surprisingly, this significantly complicates the causal structure analysis. But even in nonlinear systems we often consider linear approximations, which means in practice we may expect to encounter similar complications there as well. In section 3 in the supplement we summarize some results from the literature under which cyclic causal models are known to satisfy the stronger d-separation criterion. For the current paper it suffices to know that we focus on d-separation equivalence between cyclic directed graphs with no unobserved confounders, which, for the important class of systems where the global directed Markov condition in combination with its corresponding faithfulness assumption holds, also implies Markov equivalence.

Part of the reason for the slow progress on cyclic models that satisfy the d-separation criterion may be that the associated theoretical machinery developed to characterize Markov equivalence is quite imposing, which may make extensions towards confounders seem an overly daunting task.

In this article we find things may not be quite as bad as perhaps once feared. We show, for example, that establishing Markov equivalence between directed graphs becomes more intuitive when viewed from an *ancestral* perspective, leading to a simplified characterization and an efficient algorithm that greatly speeds up identification. Although this is of course but a small step, we hope that it may inspire renewed investigation into full-fledged cyclic causal discovery in the presence of latent confounders and selection bias.

In the rest of the article, section 2 introduces the necessary tools to handle cyclic directed graphs, section 3 describes an alternative, ancestral formulation of the CET, section 4 shows how to infer a graphical characterization of the Markov equivalence class without the need for d-separation tests, and section 5 demonstrates the remarkable efficiency of the resulting procedure compared to current state of the art. Detailed proofs as well as some additional experimental results are provided in the supplement.

2 CYCLIC DIRECTED GRAPHS

In this section we start with a few standard graphical model definitions, and then continue with some perhaps less familiar terminology and results specific to cyclic graphs.
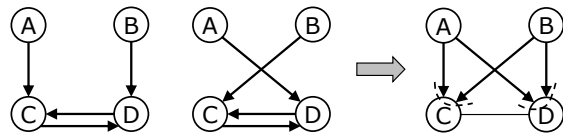

Figure 1: Two different cyclic graphs (left) that together form the only two members of the Markov equivalence class on the right, where the dashed lines signal two *virtual v-structures* (see §3.1). For linear/discrete models conditioning on C would make A and B dependent, but conditioning on $\{C, D\}$ would not.

2.1 GRAPH NOTATIONS AND TERMINOLOGY

Throughout this article we use capital letters for vertices/variables, boldface capitals to indicate sets, and calligraphic letters to indicate graphs or distributions.

A *directed graph* (DG) $\mathcal{G}$ is an ordered pair $\langle \mathbf{V}, \mathbf{E} \rangle$, where $\mathbf{V}$ is a set of vertices (nodes), and $\mathbf{E}$ is a set of directed edges (arcs) between vertices. Two nodes in $\mathcal{G}$ are *adjacent* if they are connected by an edge, two edges are *adjacent* if they share a node. A *path* in the graph $\mathcal{G}$ is a sequence of adjacent edges where each consecutive pair along the path is adjacent in $\mathcal{G}$ and each node occurs at most once, or just a single node (a *trivial* path). A *directed path* $X_0 \rightarrow X_1 \rightarrow .. \rightarrow X_k$ is a path where each pair of consecutive nodes is connected by an arc $X_i \rightarrow X_{i+1}$ in $\mathcal{G}$. A *cycle* is a directed path $X_0 \rightarrow .. \rightarrow X_k$ together with an edge $X_k \rightarrow X_0$. A directed graph with no cycles is called a *directed acyclic graph* (DAG). If $X \rightarrow Y$ in $\mathcal{G}$ then X is called a *parent* of Y, and Y a *child* of X. Similarly, if there is a directed path from X to Y in $\mathcal{G}$ then X is an *ancestor* of Y, and Y a *descendant* of X. We use $pa_{\mathcal{G}}(X)$ to denote the set of parents of X in graph $\mathcal{G}$. Idem $ch_{\mathcal{G}}(X)$, $an_{\mathcal{G}}(X)$ and $de_{\mathcal{G}}(X)$ for the sets of children, ancestors, and descendants of X in $\mathcal{G}$, with natural extensions to sets, e.g. $pa_{\mathcal{G}}(\mathbf{X}) : \{V : \exists X \in \mathbf{X}, V \in pa_{\mathcal{G}}(X)\}$. A node Z is a *collider* on a path $\langle .., X, Z, Y, .. \rangle$ if the subpath is of the form $X \rightarrow Z \leftarrow Y$, otherwise it is a *noncollider*. A triple of nodes $\langle .., X, Z, Y, .. \rangle$ on a path is said to be *unshielded* if X and Y are not adjacent in $\mathcal{G}$. An unshielded collider $X \rightarrow Z \leftarrow Y$ is known as a *v-structure*.

A *DG model* is an ordered pair $\langle \mathcal{G}, \mathcal{P} \rangle$ where $\mathcal{G}$ is a (cyclic or acyclic) directed graph and $\mathcal{P}$ is a probability distribution over the vertices (variables) in $\mathcal{G}$. The *global directed Markov property* links the structure of the graph $\mathcal{G}$ to probabilistic independences in $\mathcal{P}$ via the d-separation criterion: for sets of vertices $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ in a graph $\mathcal{G}$, $\mathbf{X}$ is *d-connected* to $\mathbf{Y}$ given $\mathbf{Z}$ iff there is an $X \in \mathbf{X}$ and $Y \in \mathbf{Y}$ such that there is a path π between X and Y on which every noncollider is not in $\mathbf{Z}$, and every collider on π is an ancestor of $\mathbf{Z}$; otherwise $\mathbf{X}$ and $\mathbf{Y}$ are said to be *d-separated* given $\mathbf{Z}$. Two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are said to be *d-separation (Markov) equivalent* iff every d-separation in $\mathcal{G}_1$ also holds in $\mathcal{G}_2$ and v.v. For more details on graphical causal models, see (Koller and

Friedman, 2009; Spirtes et al., 2000; Pearl, 2009; Bongers et al., 2021). In section 3 in the supplement, we provide more details on Markov properties in structural causal models, and describe some concrete classes of models for which the *d*-separation criterion applies.

2.2 FEATURES OF CYCLIC GRAPHS

Next we will state a few properties and definitions that are specific to directed graphs with cycles.

Definition 1 *In a directed graph $\mathcal{G}$ over set of vertices $\mathbf{V}$, a subset $\mathbf{S} \subseteq \mathbf{V}$ is a **strongly connected component (SCC)** of $\mathcal{G}$ iff $\mathbf{S}$ is a maximal set of vertices where every vertex is reachable via a directed path in $\mathcal{G}$ from every other vertex in $\mathbf{S}$.*

In cyclic graphs the presence of arcs into directed cycles can create dependencies that behave like additional induced edges:

Definition 2 *In a graph $\mathcal{G}$, two nodes A and B are said to be **virtually adjacent** iff there is no edge between A and B in $\mathcal{G}$, but A and B have a common child C which is an ancestor of A or B.*

Two nodes connected by a virtual edge cannot be *d*-separated by any set of nodes, and therefore appear like they are connected by an edge. In (Richardson, 1997) virtual edges were also called *p(seudo)-adjacent*.

These induced virtual edges can also be part of paths we have to consider, giving rise to the generalized concept of an itinerary:

Definition 3 *In a graph $\mathcal{G}$, a sequence of vertices $\langle X_0, ..., X_{n+1} \rangle$ where all neighbouring nodes in the sequence are (virtually) adjacent in the graph is said to be an **itinerary**. If none of the nodes on the itinerary are (virtually) adjacent to each other except for the ones that occur consecutively on it then the itinerary is said to be **uncovered**, otherwise it is said to be **covered**.*

Virtual edges can also appear in regular (non)collider triples, leading to the generalized notion of (non)conductors:

Definition 4 *In a graph $\mathcal{G}$, a triple $\langle A, B, C \rangle$ forms a **conductor** if $\langle A, B, C \rangle$ is an itinerary, and B is an ancestor of A and/or C. If $\langle A, B, C \rangle$ is an itinerary, but B is NOT an ancestor of A or C, then $\langle A, B, C \rangle$ is a **nonconductor**. A (non)conductor $\langle A, B, C \rangle$ is said to be **unshielded** if A and C are not (virtually) adjacent, otherwise it is **shielded**.*

In some case we can actually detect the presence of some induced edge, although we can never be sure which one:

Definition 5 *In a graph $\mathcal{G}$ a nonconductor triple $\langle A, B, C \rangle$ is a **perfect nonconductor** if B is also a descendant of a common child of A and C. If not, then $\langle A, B, C \rangle$ is an **imperfect nonconductor**.*

Key notion here is that for unshielded perfect nonconductors conditioning on a set that includes B *always* creates a dependence between A and C, whereas unshielded imperfect nonconductors do create a dependence when conditioning on B, but not for *every* set containing B. This is impossible in acyclic graphs and is therefore a hallmark for the presence of cycles. See the two virtual *v*-structures in Figure 1 for an example.

Finally, as pièce de résistance, we have some patterns that introduce a nonlocality aspect:

Definition 6 *If $\langle X_0, ..., X_{n+1} \rangle$ is a sequence of vertices such that each consecutive triple along the (uncovered) itinerary is a conductor, and all nodes $\{X_1, .., X_n\}$ are ancestors of each other, but not ancestors of either X_0 or X_{n+1}, then the triples $\langle X_0, X_1, X_2 \rangle$ and $\langle X_{n-1}, X_n, X_{n+1} \rangle$ are **mutually exclusive (m.e.) conductors w.r.t. an (uncovered) itinerary**.*

An example is depicted in Figure 2. As a result, graphs that have identical *d*-separation relations locally everywhere in the graph can still differ regarding a *d*-separation between nodes that are arbitrarily far apart in the graph (something that is impossible in the acyclic case).

2.3 THE CYCLIC EQUIVALENCE THEOREM

With the features introduced in the previous section Richardson (1997) established the following characterization:

Cyclic Equivalence Theorem (CET): Two directed graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ over vertices $\mathbf{V}$ are Markov (*d*-separation) equivalent iff

(i) they have the same (virtual) adjacencies,

(ii).a they have the same unshielded conductors,

(ii).b they have the same unshielded perfect nonconductors,

(iii) two triples $\langle A, B, C \rangle$ and $\langle X, Y, Z \rangle$ are mutually exclusive conductors on some uncovered itinerary $P = \langle A, B, C, .., X, Y, Z \rangle$ in $\mathcal{G}_1$ iff they are also m.e. conductors on some uncovered itinerary in $\mathcal{G}_2$,

(iv) if $\langle A, X, B \rangle$ and $\langle A, Y, B \rangle$ are unshielded imperfect nonconductors in $\mathcal{G}_1$ and $\mathcal{G}_2$, then X is an ancestor of Y in $\mathcal{G}_1$ iff X is an ancestor of Y in $\mathcal{G}_2$,

(v) if $\langle A, B, C \rangle$ and $\langle X, Y, Z \rangle$ are m.e. conductors on an uncovered itinerary $P = \langle A, B, C, .., X, Y, Z \rangle$, and $\langle A, M, Z \rangle$ is an unshielded imperfect nonconductor (in $\mathcal{G}_1$ and $\mathcal{G}_2$), then M is a descendant of B in $\mathcal{G}_1$ iff M is a descendant of B in $\mathcal{G}_2$.
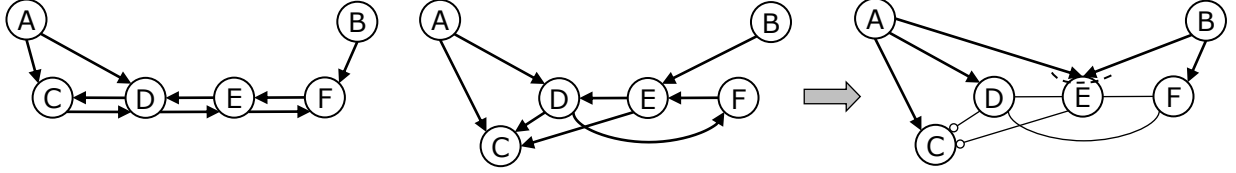
Figure 2: Two Markov equivalent graphs (left) with $\langle A, D, F \rangle$ and $\langle D, F, B \rangle$ a pair of m.e. conductors on uncovered itinerary $\langle A, D, F, B \rangle$; (right) corresponding (maximally informative) CPAG.

2.4 CYCLIC PAGS

To characterize the (*d*-separation) Markov equivalence class of a cyclic directed graph $\mathcal{G}$, denoted $MEC(\mathcal{G})$, Richardson (1996c) described an algorithm that created a set of exhaustive lists of instances in the graph matching one of the individual rules in the CET, above. Establishing Markov equivalence then boils down to comparing the lists constructed for each.

Later on, Richardson (1996b) introduced a more intuitive graphical representation in the form of a (cyclic) partial ancestral graph that also captured enough elements to uniquely identify the equivalence class of a directed graph:

Definition 7 *A graph $\mathcal{P}$ is a **partial ancestral graph (PAG)** for directed (a)cyclic graph $\mathcal{G}$ with vertex set $\mathbf{V}$, iff*

(i) *there is an edge between vertices A and B iff A and B are d-connected given any subset $\mathbf{W} \subseteq \mathbf{V} \smallsetminus \{A, B\}$,*

(ii) *If $A \longrightarrow\!\!* B$ is in $\mathcal{P}$, then in every graph in $MEC(\mathcal{G})$, A is ancestor of B,*

(iii) *If $A \leftrightarrow B$ is in $\mathcal{P}$, then in every graph in $MEC(\mathcal{G})$, B is NOT an ancestor of A,*

(iv) *if $A *\!\!\longrightarrow\!\!\underline{* B *}\!\!\longrightarrow\!\!* C$ in $\mathcal{P}$, then B is ancestor of A and/or C in every $\mathcal{G}' \in MEC(\mathcal{G})$,*

(v) *if $A \longrightarrow \underline{B} \longleftarrow C$ in $\mathcal{P}$, then B is NOT a descendant of a common child of A and C in every $\mathcal{G}' \in MEC(\mathcal{G})$,*

(vi) *any remaining edge mark not oriented in the above ways obtains a circle mark $\circ\!\!\longrightarrow\!\!*$ in $\mathcal{P}$.*

*We use the term **cyclic PAG (CPAG)** of a graph $\mathcal{G}$ to denote a PAG $\mathcal{P}$ that captures invariant ancestral relations shared by all and only the graphs in the Markov equivalence class of $\mathcal{G}$.*

In these rules the asterisk $*\!\!-$ mark on an edge is used as a meta symbol that represents any of the other marks $\{-, >, \circ\}$. The solid underlining in rule (iv), indicating that the middle node is *not* a collider between the other two, is superfluous and therefore often omitted from the graph $\mathcal{P}$. The dashed underlining in rule (v), however, *is* essential, and unique to cyclic graphs, and appears in the virtual *v*-structures introduced in §3.1. See Figure 2 for an example CPAG.

The CPAG has the same purpose and interpretation as the familiar PAG output by the well-known FCI algorithm (Spirtes et al. (2000); Zhang (2008)), including circle marks $X \circ\!\!\longrightarrow\!\!* Y$ from rule (vi) to explicitly denote 'not determined'. This can be either because the implied ancestral relation is not invariant between all members in the Markov equivalence class of $\mathcal{G}$, i.e. there are some graphs where X is an ancestor of Y and some where it is not ('can't know'), or because the relation *is* invariant but we have not determined what it is yet ('don't know'). As a result, a graph $\mathcal{G}$ can correspond to different CPAGs $\mathcal{P}$ that differ in level of completeness. In this paper we are not concerned with obtaining the (unique) *maximally informative* CPAG, but instead settle for any *Markov complete* PAG that represents a unique (*d*-separation) Markov equivalence class.

2.5 CPAG-FROM-GRAPH ALGORITHM

Using the CPAG definition above we now describe an algorithm by Richardson (1996a) that takes as input a (possibly cyclic) directed graph $\mathcal{G}$ and outputs a CPAG $\mathcal{P}$ such that two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are Markov equivalent iff the algorithm outputs the same CPAG for both. In other words, the algorithm is *d-separation complete*.

(a) form the complete undirected graph $\mathcal{P}$ with all circle edges $\circ\!\!-\!\!\circ$, and then for every edge $A \circ\!\!-\!\!\circ B$ in $\mathcal{P}$, if A is *d*-separated from B given $\mathbf{C} = An(\{A, B\}) \smallsetminus \{A, B\}$ then remove edge $A \circ\!\!-\!\!\circ B$ from $\mathcal{P}$ and record $\mathbf{C}$ in $Sepset(A, B)$ and $Sepset(B, A)$,

(b) for each unshielded triple $A *\!\!\longrightarrow\!\!* B *\!\!\longrightarrow\!\!* C$ in $\mathcal{P}$, orient $A \rightarrow B \leftarrow C$ if $B \notin Sepset(A, C)$,

(c) for each triple $\langle A, X, Y \rangle$ such that $X *\!\!\longrightarrow\!\!* Y$ in $\mathcal{P}$, A is not adjacent to X or Y in $\mathcal{P}$, $X \notin Sepset(A, Y)$, orient $X \leftarrow Y$ if A and X are d-connected given $Sepset(A, Y)$,

(d) for each unshielded triple $A \rightarrow B \leftarrow C$ in $\mathcal{P}$, if A and C are *d*-separated given a specific set $\mathbf{R}$,[1] then orient $A \longrightarrow \underline{B} \longleftarrow C$ in $\mathcal{P}$ and record $\mathbf{R}$ in $SupSepset\langle A, B, C \rangle$ (and $SupSepset\langle C, B, A \rangle$),

(e) for each quadruple $\langle A, B, C, D \rangle$, if $A \longrightarrow \underline{B} \longleftarrow C$ in $\mathcal{P}$, $A \rightarrow D \leftarrow C$ or $A \longrightarrow \underline{D} \longleftarrow C$ in $\mathcal{P}$, B and D are

---

[1]We omit the definition of the set $\mathbf{R}$ here for brevity.

adjacent in $\mathcal{P}$, then if $D \in SupSepset\langle A, B, C\rangle$ then orient $B \ast\!\!-\!\!- D$, otherwise orient $B \to D$ in $\mathcal{P}$,

(f) for each quadruple $\langle A, B, C, D\rangle$, such that $A \longrightarrow \underline{B} \longleftarrow C$ in $\mathcal{P}$, and D is not adjacent to both A and C in $\mathcal{P}$, if A and C are d-connected given $SupSepset\langle A, B, C\rangle \cup \{D\}$, then orient $B \ast\!\!-\!\!\circ D$ as $B \to D$.

The algorithm has complexity $O(N^7)$, and is d-separation complete:

Theorem 2 in (Richardson, 1996a): For two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, the CPAG-from-Graph algorithm outputs corresponding CPAGs $\mathcal{P}_1$ and $\mathcal{P}_2$ that are identical iff $\mathcal{G}_1$ and $\mathcal{G}_2$ are d-separation equivalent.

Actually, the theorem was formulated for the CCD algorithm (Richardson, 1996b) for obtaining a CPAG from (oracle) independence information, but the two are so similar that the proof automatically carries over to the CPAG-from-Graph algorithm. The algorithm is an improvement by a factor $O(N^2)$ on the earlier list-based Cyclic Classification algorithm in (Richardson, 1996c, §5.4).

3 AN ANCESTRAL PERSPECTIVE ON THE CET

On reflection of the characterization of Markov equivalence between cyclic graphs obtained, one may note that the rather daunting definitions and terminology in the CET seem to contrast quite sharply with the apparent simplicity of the actual invariant features contained in the CPAG. At the same time complicated again by the fact that some of these 'invariant features' like edges in the CPAG are not actually invariant in the underlying graph at all.

Furthermore, there is no clear match from some rules in the CET to specific invariant features in the CPAG. In particular the 'mutually exclusive conductors on an uncovered itinerary'[2] in rule CET-(iii) are never explicitly recorded, even though they can of course be inferred from the CPAG afterwards.

A natural question, inspired by the familiar DAG-MAG-PAG triad for acyclic graphs, would be whether it might make sense to also consider an intermediate ancestral stage for cyclic graphs.

In this section we answer that question with an emphatic: yes! We first introduce the CMAG as the cyclic analogue to the (acyclic) maximal ancestral graph (Richardson and Spirtes, 2002), and rephrase the CET in terms of ancestral graphs. This results in a simplified set of rules that each are in direct correspondence with invariant features in the

---

[2]Actually this term is a bit of a misnomer, as the two conductors need not be mutually exclusive when there is an induced virtual edge along the uncovered itinerary connecting the two.

CPAG. In the next section we will show that this approach also leads to an efficient procedure to establish Markov equivalence that no longer needs to rely on d-separation tests.

3.1 INTRODUCING THE CMAG

In keeping with the spirit of regular (acyclic) maximal ancestral graphs, we will define a cyclic MAG as:

Definition 8 *The **cyclic maximal ancestral graph (CMAG)** $\mathcal{M}$ corresponding to (cyclic) directed graph $\mathcal{G}$ over set of vertices $\mathbf{V}$ is a graph where:*

(i) *there is an edge between every distinct pair of vertices $\{X, Y\}$ iff they cannot be d-separated by any subset of $\mathbf{V} \smallsetminus \{X, Y\}$ in $\mathcal{G}$,*

(ii) *there is a tail mark $X \longrightarrow\!\ast Y$ at vertex X on the edge to Y iff there exists a directed path from X to Y in $\mathcal{G}$, otherwise there is an arrowhead mark $X \leftrightarrow\!\ast Y$,*

(iii) *every unshielded collider triple $X \to Z \leftarrow Y$ in $\mathcal{M}$ where Z is not a descendant of a common child of X and Y in $\mathcal{G}$ obtains a dashed underline $X \longrightarrow \underline{Z} \longleftarrow Y$.*

*Unshielded collider triples without underlining are called **v-structures**. The 'dashed-underlined' collider triples in a CMAG are referred to as **virtual v-structures**.*

With this definition, a CPAG becomes a straightforward collection of invariant edges and edge marks (rather than 'ancestral relations') shared by all and only the CMAGs corresponding to graphs in the same Markov equivalence class.

The 'virtual' in the dashed-underlined *v*-structures from rule (iii) emphasises that they resemble regular *v*-structures in the CMAG, but look and behave differently in the underlying directed graph $\mathcal{G}$. They are a direct consequence of rule (v) in Def. 7, and correspond to unshielded imperfect nonconductors in $\mathcal{G}$, that are unique to cyclic graphs. In a CMAG $\mathcal{M}$, node A is an *ancestor* of node B (and B a *descendant* of A) iff there exists an ancestral path $A \longrightarrow\!\ast .. \longrightarrow\!\ast B$ in $\mathcal{M}$.

An SCC in directed graph $\mathcal{G}$ corresponds to a *maximal set of nodes in a connected, undirected subgraph* in $\mathcal{M}$, as each node in an SCC is ancestor of all other nodes in the same SCC. Given this one-to-one correspondence we will also use $SCC(Z)$ in the context of a CMAG $\mathcal{M}$ to denote the nodes in the strongly connected component of Z in $\mathcal{G}$.

3.2 VIRTUAL COLLIDER TRIPLES

Having brought out the CMAG we can make a straightforward mapping from elements in the CET to their ancestral counterpart: (virtual) adjacencies become edges, itineraries

become paths, unshielded conductors become unshielded noncolliders, unshielded (perfect) nonconductors become v-structures, and unshielded imperfect nonconductors become virtual v-strucutures.

That only leaves the 'mutually exclusive conductors w.r.t. an uncovered itinerary'. For that we note that these only appear in the CPAG as the invariant arcs into a cycle, oriented in step (c) of the CPAG-from-Graph algorithm. In other words, from an ancestral perspective it is not about the conductor triples at the beginning and end of the uncovered itinerary, but only about the first and last edge along the corresponding path in the CMAG.

This brings us to the following definition:

Definition 9 *In a CMAG $\mathcal{M}$, a quadruple of distinct nodes $\langle X, Z, Z', Y \rangle$ is a **u-structure** if there is an uncovered path $X \to Z — .. — Z' \leftarrow Y$ in $\mathcal{M}$, where all intermediate nodes are also in $SCC(Z)$.*

The term *u*-structure reflects the fact that it is similar to a *v*-structure, but with the central collider node replaced by an uncovered path through a strongly connected component.

There is a straightforward connection between *u*-structures and the 'm.e. conductors w.r.t. an uncovered itinerary' from Definition 6:

Lemma 1 *For a directed graph $\mathcal{G}$ and corresponding CMAG $\mathcal{M}$, there is a u-structure $\langle X, Z, Z', Y \rangle$ in $\mathcal{M}$ iff there is an uncovered itinerary $\pi = \langle X, Z, U, .., U', Z', Y \rangle$ in $\mathcal{G}$, possibly with $Z = U'$ or $U = U'$, where $\langle X, Z, U \rangle$ and $\langle U', Z', Y \rangle$ are a pair of m.e. conductors w.r.t. the uncovered itinerary π in $\mathcal{G}$.*

(For proof details for this and other results in the rest of this article, see supplement.)

Crucially, in the CMAG or CPAG we do not actually record the *u*-structure explicitly. In fact, the only elements of a *u*-structure that need to be oriented in the CPAG are the first and last edge *into* the strongly connected component (cf. step (c) of the CPAG-from-Graph algorithm, §2.5).

As a result, we do not have to identify the full quadruple $\langle X, Z, Z', Y \rangle$ of each *u*-structure, but only if an edge $X - Z$ is part of *some* u-structure pattern. For that, we can rely on the following result:

Lemma 2 *In a CMAG $\mathcal{M}$, a pair of nodes $\langle X, Z \rangle$ is part of a u-structure $\langle X, Z, Z', Y \rangle$ with a node $Y \in \mathbf{Y} \subseteq pa(SCC(Z)) \setminus adj(\{X, Z\})$, iff $X \in pa(Z)$, and X and Y are connected in the subgraph over $((SCC(Z) \setminus adj(X)) \cup \{X, Z\} \cup \mathbf{Y}$.*

This significantly reduces the complexity of establishing Markov equivalence later on, as it means we only need to

search over triples rather than quadruples in the CMAG. More importantly, it motivates the introduction of the following invariant element, which in turn will significantly simplify the CET.

Definition 10 *In a CMAG $\mathcal{M}$, a triple of distinct nodes $\langle X, Z, Y \rangle$ is a **virtual collider triple** iff $\langle X, Z, Y \rangle$ is a virtual v-structure, or there is some $Z' \in SCC(Z)$, such that either $\langle X, Z, Z', Y \rangle$ or $\langle X, Z', Z, Y \rangle$ is a u-structure.*

Intuitively, a virtual collider triple $\langle X, Z, Y \rangle$ implies that X and Y are connected by an uncovered itinerary via nodes in $SCC(Z)$ that *identifiably* contains one or more virtual edges. The strongly connected component of Z fulfils the role of collider in $X \to SCC(Z) \leftarrow Y$, and the *virtual* emphasises there is no 'real' collider triple $X \to Z \leftarrow Y$ in the underlying directed graph.
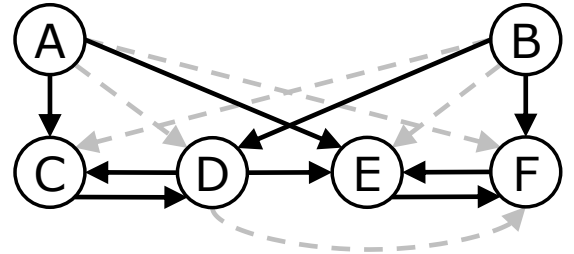

Figure 3: Example CET orientation rule (iv) on virtual collider triples $\langle A, D, B \rangle$ and $\langle A, E, B \rangle$ for invariant edge $D \to E$, with virtual edges as dashed grey arcs.

3.3 A NEW CET

We are now ready to restate the Cyclic Equivalence Theorem in terms of CMAGs:

Theorem 1 *Two directed graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, corresponding to CMAGs $\mathcal{M}_1$ and $\mathcal{M}_2$, are Markov (d-separation) equivalent iff*

(i) *$\mathcal{M}_1$ and $\mathcal{M}_2$ have the same skeleton,*

(ii) *$\mathcal{M}_1$ and $\mathcal{M}_2$ have the same v-structures,*

(iii) *$\mathcal{M}_1$ and $\mathcal{M}_2$ have the same virtual collider triples,*

(iv) *if $\langle A, B, C \rangle$ is a virtual collider triple, and $\langle A, D, C \rangle$ a virtual v-structure, then B is an ancestor of D in $\mathcal{M}_1$ iff B is an ancestor of D in $\mathcal{M}_2$.*[3]

In this case, we call $\mathcal{M}_1$ and $\mathcal{M}_2$ 'Markov equivalent'.

---

[3]In the original published version, $\langle A, D, C \rangle$ was erroneously included as a virtual collider triple, but the distinction is needed to restrict the pairs of virtual collider triples to check.

Each rule in this ancestral CET can be linked directly to specific invariant elements in the CPAG: rule (i) to the edges in the CPAG, (ii) to all *v*-structures, (iii) to remaining invariant arcs into strongly connected components (incl. under-dashed marks for virtual *v*-structures), and (iv) to invariant edges within or between (identifiable) cycles.

Comparing to the original CET in §2.3, we can see that the ancestral formulation greatly simplifies the Markov equivalence characterization, leading to two fewer rules and only requiring (collider) triples.

An interesting observation is that in the acyclic case going from DAGs to MAGs (to allow for unobserved confounders) implied going from '(unshielded) collider triples' (*v*-structures) to 'collider triples with order' in the characterization of Markov equivalence between graphs (Ali et al., 2009; Claassen and Bucur, 2022). Given that analogy we conjecture that for the cyclic case allowing for latent confounders can similarly be accomplished by extending to '(virtual) collider triples with order'.

4 ESTABLISHING MARKOV EQUIVALENCE FOR CYCLIC GRAPHS

We now show that with the intermediate CMAG representation we can derive a consistent CPAG that uniquely defines the equivalence class of a (cyclic) directed graph without the need for any *d*-separation tests. The resulting algorithm is extremely fast, and allows to determine Markov equivalence between graphs by directly comparing the output CPAGs.

4.1 OBTAINING THE CMAG

To capture the first rule of the new CET, we need to obtain the skeleton of the CPAG. To avoid the *d*-separation tests in step (a) of the CPAG-from-Graph algorithm in §2.5, we can use the following result:

Lemma 3 *In a CMAG $\mathcal{M}$ corresponding to directed graph $\mathcal{G}$, two variables X and Y are adjacent, iff X and Y are (virtually) adjacent in $\mathcal{G}$.*

It implies we can read off the CMAG skeleton directly from the graph $\mathcal{G}$, by starting from the skeleton of $\mathcal{G}$, and adding an edge between X and Y for every *v*-structure $X \to Z \leftarrow Y$ in $\mathcal{G}$ with $Y \in SCC_{\mathcal{G}}(Z)$.

It does mean that we first need to partition the vertices in the graph into the set of strongly connected components. This can be achieved in time linear in the number of vertices and edges $O(Nd)$ using e.g. Tarjan's algorithm (Tarjan, 1972).[4]

---

[4] Actually, we use a modified version of Tarjan's algorithm that also tracks ancestral relations in one go. For details on this and all

Subsequent orientations of edges in $\mathcal{M}$ follow orientations in $\mathcal{G}$, where edges between nodes in the same SCC become undirected edges, signifying they are all ancestor of each other. Induced edges between nodes in the same cycle also become undirected, and induced edges by a triple $X \to Z \leftarrow Y$ in $\mathcal{G}$ with $X \notin SCC_{\mathcal{G}}(Z)$ become $X \to Y$.

Alternatively, we can process each node X in $\mathcal{G}$ in turn, and draw undirected edges between all of its parents in the same cycle as X (incl. X) in $\mathcal{M}$, and add arcs from all remaining parents into the first set of parents (again incl. X), which is what we do in Algorithm 1, below.

Algorithm 1 Graph-to-CMAG

Input: directed cyclic graph $\mathcal{G}$ over nodes $\mathbf{V}$
Output: CMAG $\mathcal{M}$, SCCs,
$SCC \leftarrow Get\_StronglyConnComps(\mathcal{G})$
part 1: CMAG rules (i) + (ii)
for all $X \in \mathbf{V}$ **do**
   $\mathbf{Z} \leftarrow pa_{\mathcal{G}}(X)$
   $\mathbf{Z}_{cyc} \leftarrow \mathbf{Z} \cap SCC(X)$
   $\mathbf{Z}_{acy} \leftarrow \mathbf{Z} \setminus \mathbf{Z}_{cyc}$
   add all arcs $\mathbf{Z}_{acy} \to \mathbf{Z}_{cyc} \cup \{X\}$ to $\mathcal{M}$
   add all undirected edges $\mathbf{Z}_{cyc} — \mathbf{Z}_{cyc} \cup \{X\}$ to $\mathcal{M}$
end for
part 2: CMAG rule (iii)
for all $X \in \mathbf{V} : |SCC(X)| \geq 2$ **do**
   $\mathbf{Z} \leftarrow pa_{\mathcal{M}}(X)$
   for all non-adjacent pairs $\{Z_i, Z_j\} \subseteq \mathbf{Z}$ **do**
     if $\{Z_i, Z_j\} \nsubseteq adj_{\mathcal{G}}(X)$ **then**
      if $X \notin de_{\mathcal{G}}(ch_{\mathcal{G}}(Z_i) \cap ch_{\mathcal{G}}(Z_j))$ **then**
       mark virtual v-structure $\langle Z_i, X, Z_j \rangle$ in $\mathcal{M}$
   end for
end for

The second part of Algorithm 1 simply involves checking all *v*-structures in $\mathcal{M}$ with central collider node in a non-trivial SCC, and with at least one virtual edge in $\mathcal{G}$. Here we use the matrix of ancestral relations, constructed when identifying the SCCs at the start of the algorithm, to reduce the 'descendant of' check in the second 'if'-clause to constant time per node.

4.2 CONSTRUCTING THE CPAG

Before we can go on to construct a CPAG from the CMAG $\mathcal{M}$ obtained above, we still need to recognise the virtual collider triples corresponding to so-called *u*-structures. These are not marked explicitly in the CMAG (contrary to virtual *v*-structures), but they *are* needed to orient certain invariant edges in the CPAG corresponding to rules (iii) and (iv) in Theorem 1. Fortunately, for that we can rely on Lemma 2, where the fact that we only need to consider straightforward

---

other algorithms used in the paper, see source code available at
`https://github.com/tomc-ghub/CET_uai2023`

'connected subgraphs' means the complexity of this step scales linearly with the number of edges in the subgraph.

It also means that, in the construction of the CPAG, to cover invariant arcs from *u*-structures, we only need to consider edges $X \to Z$ in $\mathcal{M}$ that are not yet oriented in $\mathcal{P}$, and where Z is part of a nontrivial SCC (size $|SCC(Z)| \geq 2$), and the **Y** in Lemma 2 are all other parents of $SCC(Z)$ that are not adjacent to X and/or Z in $\mathcal{M}$. Note that the arcs oriented thusly were previously captured by the exhaustive search in step (c) of the CPAG-from-Graph algorithm in section 2.5.

We can now bring these steps together in Algorithm 2.[5]

Algorithm 2 Graph-to-CPAG

Input: directed cyclic graph $\mathcal{G}$ over nodes **V**,
Output: CPAG $\mathcal{P}$,
$(\mathcal{M}, SCC) \leftarrow$ Graph-to-CMAG($\mathcal{G}$)
part 1: new-CET rules (i)-(iii)
$\mathcal{P} \leftarrow$ skeleton of $\mathcal{M}$ with all $\circ\!\!-\!\!\circ$ edges
$\mathcal{P} \leftarrow$ copy all (virtual) *v*-structures from $\mathcal{M}$
for all $X \circ\!\!-\!\!\circ Z$ in $\mathcal{P}$, $X \to Z$ in $\mathcal{M}$, $|SCC(Z)| \geq 2$ **do**
 if $\exists \langle X, Z, Z', Y \rangle$ as *u*-structure in $\mathcal{M}$ **then**
  orient $X \to Z$ in $\mathcal{P}$ {*Lemma 2*}
end for
part 2: new-CET rule (iv)
for all $Z \circ\!\!-\!\!\circ W$ at virtual *v*-structures $\langle X, Z, Y \rangle$ **do**
 if $\langle X, W, Y \rangle$ is virtual collider triple **or**
   $\exists \langle X, B, Y \rangle$ as virt. coll. triple, with $B \notin an(Z)$,
   uncovered $B \,\rightarrow\!\!\ast \,.. \,\rightarrow\!\!\ast \,W \leftarrow Z$ in $\mathcal{M}$, and
   $\nexists U : \{v\text{-structure } X \to U \leftarrow Y \text{ in } \mathcal{M}, W \in de(U)\}$
 then copy edge $Z \,\ast\!\!-\!\!\ast\, W$ from $\mathcal{M}$ to $\mathcal{P}$
end for

In practice we already copy invariant features to the CPAG while constructing the CMAG to improve efficiency. Note that the final output CPAG is *d*-separation complete, but *not* guaranteed to be identical to the CPAG from the original CPAG-from-Graph algorithm. This is because steps (c) and (f) there contain an exhaustive search that also orients certain arcs that are sound but not needed for the CET, but could also be obtained from subsequent implied orientation rules, similar to the PC/FCI algorithm. Therefore the CPAGs from the two algorithms cannot be compared directly against each other to establish Markov equivalence. However the main result remains the same:

Theorem 2 *For two different directed graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, let $\mathcal{P}_1$ and $\mathcal{P}_2$ be the corresponding CPAGs output by algorithm 2. Then $\mathcal{G}_1$ is Markov (d-separation) equivalent to $\mathcal{G}_2$ iff $\mathcal{P}_1 = \mathcal{P}_2$.*

---

[5]The second clause in the 'if' statement in part 2 was added as a result of the correction to CET rule (iv).

4.3 COMPUTATIONAL COMPLEXITY

The scaling behaviour of Algorithm 2 depends primarily on the number of vertices N and average node degree d corresponding to $N * d$ edges in the graph.

The first part of algorithm 1 requires order $O(N + N * d)$ steps to find the strongly connected components, followed by a loop over N vertices comparing d^2 parents, so overall $O(N * d^2)$. Similarly, the second part of algorithm 1 considers d^2 parents for N nodes, checking it is not a descendant of d possible common children for $O(N * d^3)$ (provided the $Get\_StronglyConnComps$ step also tracks the ancestral matrix for constant-time descendant checks).

Next, the first two steps in part 1 of algorithm 2, initializing the skeleton and (virtual) v-structures, are also $O(N * d^2)$. Next, for the *u*-structures we may need to loop over $O(N * d)$ edges and establish connectedness in a subgraph over at most N nodes, which can be done in order $O(N * d)$ steps (similar to the SCC procedure) leading to overall $O(N^2 * d^2)$. Finally, in part 2 of algorithm 2 we need to loop over $N * d$ edges to virtual *v*-structures, considering d^2 other virtual collider triples (previously identified in part 1), each with links to d candidate nodes B, followed by a (single) test for connectedness per edge, order $O(N * d)$, giving a total of $O(N * d * (d^3 + N * d))$.

So overall worst case complexity scales with $O(N^5)$ for arbitrary density, which is a significant improvement over the $O(N^7)$ achieved by the current state-of-the-art CPAG-from-Graph algorithm.

In practice, even for large graphs there is typically only a relatively small number of cases to consider in the final steps, and so for both procedures the actual scaling behaviour is usually much better than this worst-case bound suggests, as evidenced by the next section.

5 EXPERIMENTAL EVALUATION

In order to evaluate the performance of the CPAG-from-graph procedure as a function of size and density of the graph we generate collections of random directed cyclic graphs and track both average and worst-case performance in terms of number of elementary operations and time.

Note that in generating the random cyclic graphs we introduced a few parameters to be able to tweak the number and type of cycles included, as for increasing size and density truly random cyclic graphs quickly tend to collapse into the 'one big cycle' type, avoiding most of the intricacies from CET rules (iv) and (v) that relate to invariate edges between cycles; see section 1.1 in the supplement for details.
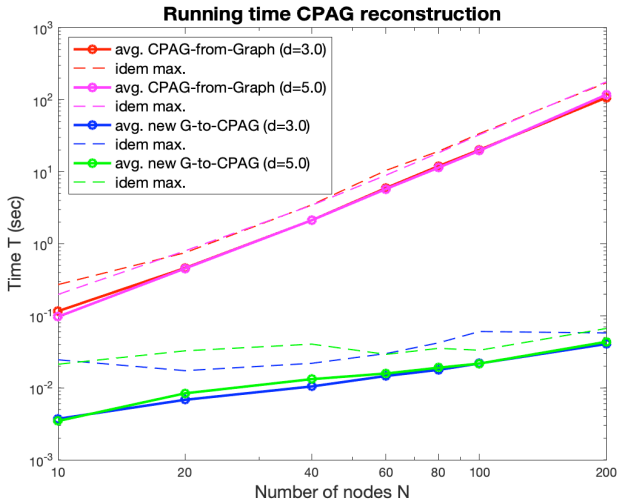
Figure 4: Log-log plot depicting scaling behaviour of original (red/magenta) and new CPAG algorithms (blue/green), as a function of size of the graph N, for two different densities $d \in \{3.0, 5.0\}$. Solid lines indicate average performance over 100 instances, dashed lines the worst case encountered.

5.1 SCALING BEHAVIOUR

Figure 4 shows the results for the two CPAG-from-graph procedures. As expected, the scaling behaviour of the new procedure in Algorithm 2 is much more benign. As a result, for graphs of $N = 200$ nodes with density $d = 3.0$, the latter requires only about 0.05 sec. on average to construct the CPAG, whereas the original version takes about 78 sec.: a speed-up by 3 orders of magnitude.

In the supplement we see that the original CPAG-from-Graph procedure spends the vast majority of its time in the expensive d-separation searches in stage (a) and (c), whereas for sparse graphs the new Graph-to-CPAG version spends roughly equal amounts in each phase. For denser graphs, the final stage in the latter starts to dominate, as expected from the complexity analysis in section 4.3.

Finally, note that for both algorithms in this experiment there is not much difference between average and worst-case scaling behaviour in the collection of randomly sampled graphs (around $1.5 - 2.0$ times more expensive for both versions), and both stay well below their theoretical worst-case limits. The reason is that, in order to reach the dreaded 'worst case' scenario, the graphs require very specific configurations that are extremely unlikely to occur in truly random graphs. As a result, a reassuring message of Figure 4 is that in practice the challenge of handling even (very) large cyclic directed graphs is likely to remain feasible in practice, despite the quite imposing theoretical worst-case limit.

6 DISCUSSION

We presented a new, ancestral perspective on the Cyclic Equivalence Theorem for directed graphs that resulted in a fast and efficient procedure to obtain the CPAG from an arbitrary directed graph.

The resulting CPAGs can be compared directly to establish Markov equivalence between cyclic directed graphs, but so far we made no attempt to derive *all* invariant features shared by all (and only) the CMAGs in the same equivalence class. In other words, we did not yet aim for the *maximally informative* CPAG. As a result, not all identifiable cycles are guaranteed to appear in an easily recognisable form. Squeezing out all available information would likely entail a set of additional orientation propagation rules, similar to augmented FCI in (Zhang, 2008).

The obtained efficiency of the Graph-to-CPAG procedure in algorithm 2 also means it is fast enough to be a viable route for extending score-based greedy equivalence search algorithms like GES (Chickering, 2002) towards cyclic graphs, similar to recent extensions for acyclic graphs in the presence of confounders (Claassen and Bucur, 2022).

However, we consider the most promising aspect of our results the significantly reduced conceptual complexity provided by the ancestral perspective. The new ancestral CET is notably simpler than the original version, and suggests a natural extension to cyclic models with confounders, analogous to that for MAGs.

Finally, the CMAG under d-separation treats strongly connected components more similar to the nonlinear case under σ-separation (Mooij and Claassen, 2020), which suggests they may be merged to handle arbitrary cyclic relationships in the near future. We hope this may encourage researchers to renew work towards extending available constraint-based algorithms towards sound and complete causal discovery in the presence of confounders, cycles, and selection bias.

References

Ali, R. A., Richardson, T. S., and Spirtes, P. (2009). Markov equivalence for ancestral graphs. The Annals of Statistics, 37(5B):2808–2837.

Bongers, S., Forré, P., Peters, J., and Mooij, J. M. (2021). Foundations of structural causal models with cycles and latent variables. Annals of Statistics, 49(5):2885–2915.

Chickering, D. M. (2002). Optimal structure identification with greedy search. Journal of machine learning research, 3(Nov):507–554.

Claassen, T. and Bucur, I. G. (2022). Greedy equivalence search in the presence of latent confounders. In

Uncertainty in Artificial Intelligence, pages 443–452. PMLR.

Forré, P. and Mooij, J. M. (2017). Markov properties for graphical models with cycles and latent variables. arXiv.org preprint, arXiv:1710.08775 [math.ST].

Forré, P. and Mooij, J. M. (2018). Constraint-based causal discovery for non-linear structural causal models with cycles and latent confounders. In *Proceedings of the 34th Annual Conference on Uncertainty in Artificial Intelligence (UAI-18)*.

Hu, Z. and Evans, R. (2020). Faster algorithms for markov equivalence. In *Conference on Uncertainty in Artificial Intelligence*, pages 739–748. PMLR.

Hyttinen, A., Eberhardt, F., and Hoyer, P. (2012). Learning linear cyclic causal models with latent variables. *Journal of Machine Learning Research*, 13:3387–3439.

Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Koster, J. (1996). Markov properties of nonrecursive causal models. *The Annals of Statistics*, 24(5):2148–2177.

Lacerda, G., Spirtes, P., Ramsey, J., and Hoyer, P. O. (2008). Discovering cyclic causal models by independent components analysis. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI-08)*.

Mooij, J. M. and Claassen, T. (2020). Constraint-based causal discovery using partial ancestral graphs in the presence of cycles. In *Conference on Uncertainty in Artificial Intelligence*, pages 1159–1168. PMLR.

Mooij, J. M. and Heskes, T. (2013). Cyclic causal discovery from continuous equilibrium data. In Nicholson, A. and Smyth, P., editors, *Proceedings of the 29th Annual Conference on Uncertainty in Artificial Intelligence (UAI-13)*, pages 431–439. AUAI Press.

Mooij, J. M., Janzing, D., Heskes, T., and Schölkopf, B. (2011). On causal discovery with cyclic additive noise models. In Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 24 (NIPS*2011)*, pages 639–647.

Neal, R. (2000). On deducing conditional independence from d-separation in causal graphs with feedback. *Journal of Artificial Intelligence Research*, 12:87–91.

Pearl, J. (2009). *Causality: Models, Reasoning and Inference*. Cambridge University Press.

Pearl, J. and Dechter, R. (1996). Identifying independence in causal graphs with feedback. In *Proceedings of the 12th Annual Conference on Uncertainty in Artificial Intelligence (UAI-96)*, pages 420–426.

Richardson, T. (1996a). Discovering cyclic causal structure. Technical Report CMU-PHIL-68, Carnegie Mellon University.

Richardson, T. (1996b). A discovery algorithm for directed cyclic graphs. In *Proceedings of the Twelfth international conference on Uncertainty in Artificial Intelligence (UAI-96)*, pages 454–461.

Richardson, T. (1996c). A polynomial-time algorithm for deciding Markov equivalence of directed cyclic graphical models. In *Proceedings of the Twelfth international conference on Uncertainty in artificial intelligence*, pages 462–469.

Richardson, T. (1997). A characterization of Markov equivalence for directed cyclic graphs. *International Journal of Approximate Reasoning*, 17(2-3):107–162.

Richardson, T. S. and Spirtes, P. (2002). Ancestral graph Markov models. *The Annals of Statistics*, 30(4):962–1030.

Rothenhäusler, D., Heinze, C., Peters, J., and Meinshausen, N. (2015). BACKSHIFT: Learning causal cyclic graphs from unknown shift interventions. In *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, pages 1513–1521.

Spirtes, P. (1994). Conditional independence in directed cyclic graphical models for feedback. Technical Report CMU-PHIL-54, Carnegie Mellon University.

Spirtes, P. (1995). Directed cyclic graphical representations of feedback models. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI-95)*, pages 499–506.

Spirtes, P., Glymour, C., and Scheines, R. (2000). *Causation, Prediction, and Search*. MIT press, 2nd edition.

Strobl, E. V. (2018). A constraint-based algorithm for causal discovery with cycles, latent variables and selection bias. *International Journal of Data Science and Analytics*, 8:33–56.

Tarjan, R. (1972). Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160.

Wienöbst, M., Bannach, M., and Liśkiewicz, M. (2022). A new constructive criterion for markov equivalence of mags. In *Uncertainty in Artificial Intelligence*, pages 2107–2116. PMLR.

Wright, S. (1921). Correlation and causation. *Journal of Agricultural Research*, 20:557–585.

Zhang, J. (2008). On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16-17):1873–1896.

Supplement - Establishing Markov Equivalence in Cyclic Directed Graphs

Tom Claassen[1]   **Joris M. Mooij**[2]

[1]Institute for Computing and Information Sciences, Radboud University, Nijmegen, Netherlands
[2]Korteweg-deVries Institute, University of Amsterdam, Amsterdam, Netherlands

Abstract

This part contains the revised version of the supplement to the original UAI2023 publication 'Establishing Markov Equivalence in Cyclic Directed Graphs'. It includes the correction to rule (iv) in Theorem 1 and the subsequent adjustment in Algorithm 2, as well as a number of extensions that should make the proofs essentially self-contained. Numbering and notations follow the main article.

7 ADDITIONAL EXPERIMENTAL RESULTS

This section elaborates on the random cyclic graph generating process, and a result that offers some added insight into the inner workings of the two CPAG algorithms.

7.1 GENERATING RANDOM CYCLIC GRAPHS

In contrast to the familiar acyclic graphs, in cyclic graphs there can be *two* edges between each pair of nodes, corresponding to a total of $N(N-1)$ possible directed edges for graphs over N nodes. However, in both the Erdos-Renyi model (all graphs with n edges equally likely) and the Gilbert model (all edges appear with equal probability p), as density or size of the graph increases, the resulting graph is overwhelmingly likely to contain just one, big strongly connected component, with only a few other nodes on its periphery. As a key part of the CET is about invariant edges *between* components in rule (iv) (see e.g. Figure 3 in the main article), just evaluating on arbitrary random graphs would likely lead to an incomplete or biased perspective. In addition, a number of challenges in finding the correct CPAG are related to sequences of connected two-cycles (see main, Figure 2), which in larger fully random graphs are also exceedingly unlikely to appear.

Therefore we tweak the random graph generating process to allow some control over the number and size of the strongly connected components. We introduce a 3-stage process parameterized by size N and density d, as well as parameters p_{two} for the proportion of two-cycles, and p_{acy} and p_{cyc} for the proportion of recursive resp. nonrecursive edges that remain:

1. randomly sample the required number of two-cycles,

2. add random arcs from lower to higher numbered nodes,

3. add completely random arcs for the remaining edges.

Afterwards a random permutation of the nodes is applied to ensure there is no implicit bias in the ordering.

With this procedure, setting $[p_{two}, p_{acy}, p_{cyc}] = [0, 1, 0]$ would lead to a random acyclic graph, whereas setting $[0.1, 0.9, 0]$ would lead to a random acyclic graph with some edges turned into two-cycles. Setting $[0, 0, 1]$ would lead to a fully random cyclic graph in the Erdos-Renyi model. In practice setting e.g. $[p_{two}, p_{acy}, p_{cyc}] = [0.1, 0.82, 0.08]$ leads to a varied number and size of the strongly connected components for graphs of up to $N = 200$ nodes with density $d = 3.0$. For $N = 200$ this leads on average to about 11 nontrivial strongly connected components with average largest component size of about 17 vertices.

For larger/higher density graphs the p_{cyc} proportion should be reduced to avoid collapsing into the 'one big cycle' trap. In our experiments for $d = 5.0$ we used $[p_{two}, p_{acy}, p_{cyc}] = [0.05, 0.93, 0.02]$, which, for $N = 200$ resulted on average in about 5 nontrivial strongly connected components, with an average largest size of about 70 vertices.

Additional implementation details will be published with the accompanying source code.

7.2 RELATIVE TIME SPENT PER STAGE

To take a closer look at the relative contribution of each stage in the two different CPAG procedures to the overall time

complexity we also timed each stage separately. Average results are depicted below.
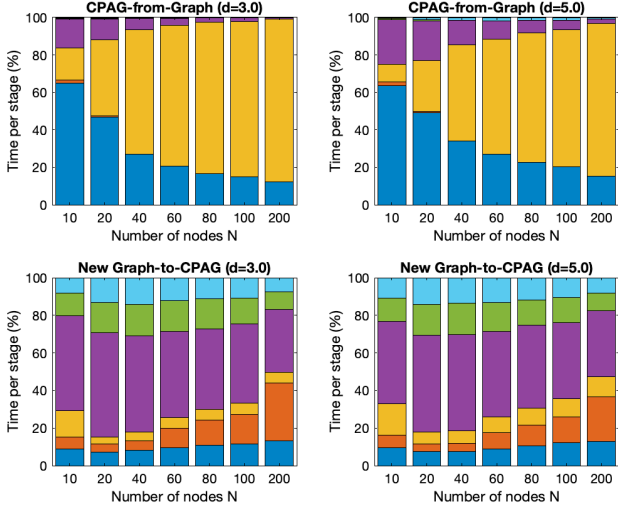

Figure 5: Plots depicting the relative proportion each algorithm spends on average in the different stages, as a function of the size of the graph N, for two different densities $d \in \{3.0, 5.0\}$. Stages are ordered bottom up, i.e. first stage on the x-axis, second stage on top of that etc.

We see that the original CPAG-from-Graph procedure spends the vast majority of its time in the expensive d-separation searches in stage (a) (blue) and (c) (yellow), whereas the new Graph-to-CPAG version spends roughly constant amounts in each phase. For denser graphs, the final stage (cyan) in the latter is theoretically the most expensive worst case, but remains at nearly constant proportion in practice, as it is extremely rare to encounter such instances in arbitrary random graphs.

The correction to rule (iv) in Theorem 1 resulted in an additional clause in the 'if' statement in part 2 of the Graph-to-CPAG algorithm, but also allowed for more efficient filtering of candidate edges to check. As a result, the implementation now spends a bit more time preparing in the 4th stage (purple), but significantly less in the final stage (cyan), making the new implementation as a whole scale slightly better than the original version (about twice as fast for $N = 200$).

8 PROOF DETAILS

First a few results on properties of *ancestral paths* in CMAGs, i.e. paths of the form $X_1 \longrightarrow * \; .. \; \longrightarrow * \; X_n$, so that every vertex X_i is ancestor of all $X_{>i}$, and in particular X_1 is ancestor of X_n, and X_n is a descendant of X_1.

Lemma 3 *In a CMAG $\mathcal{M}$ corresponding to directed graph $\mathcal{G}$, two variables X and Y are adjacent, iff X and Y are (virtually) adjacent in $\mathcal{G}$.*

Proof This is Lemma 1 in (Richardson, 1997). ∎

Lemma 4 *In a CMAG $\mathcal{M}$ corresponding to directed graph $\mathcal{G}$, A is ancestor of B in $\mathcal{M}$ iff A is ancestor of B in $\mathcal{G}$.*

Proof If A is ancestor of B in $\mathcal{M}$, then there exists a path $\pi = \langle A = X_0 \longrightarrow * (X_1 \longrightarrow * .. \longrightarrow * X_{n-1}) \longrightarrow * X_n = B \rangle$ in $\mathcal{M}$. By Definition 8-(ii), each edge $X_i \longrightarrow * X_{i+1}$ along π in $\mathcal{M}$ implies a directed path in $\mathcal{G}$ from X_i to X_{i+1}. Concatenating them provides a directed path from A to B in $\mathcal{G}$ which implies A is ancestor of B in $\mathcal{G}$. Conversely, if A is ancestor of B in $\mathcal{G}$, then this implies the existence of a directed path $\pi = \langle A = X_0 \rightarrow (X_1 \rightarrow .. \rightarrow X_{n-1}) \rightarrow X_n = B \rangle$ in $\mathcal{G}$. By Definition 8-(i), each edge $X_i \ast\!\!\longrightarrow\ast X_{i+1}$ along π in $\mathcal{G}$ is also present in $\mathcal{M}$, and again by 8-(ii) is of the form $X_i \longrightarrow * X_{i+1}$. Concatenating them creates the required path in $\mathcal{M}$ which proves the Lemma. ∎

Lemma 5 *In a CMAG $\mathcal{M}$, if $\pi = X_1 \ast\!\!\longrightarrow\ast .. \ast\!\!\longrightarrow\ast X_n$ is a path in $\mathcal{M}$, then there is a subsequence of the X_i's that forms an uncovered path between X_1 and X_n in $\mathcal{M}$. Similarly, if π is an ancestral path $X_1 \longrightarrow * .. \longrightarrow * X_n$ from X_1 to X_n, then there is a subsequence of the X_i's that forms an uncovered ancestral path from X_1 to X_n in $\mathcal{M}$.*

Proof Follows directly from Lemma 4 in combination with Lemma 13 in (Richardson, 1996a). ∎

Now the proofs for some results in the main article.

Lemma 1 *For a directed graph $\mathcal{G}$ and corresponding CMAG $\mathcal{M}$, there is a u-structure $\langle X, Z, Z', Y \rangle$ in $\mathcal{M}$ iff there is an uncovered itinerary $\pi = \langle X, Z, U, .., U', Z', Y \rangle$ in $\mathcal{G}$, possibly with $Z = U'$ or $U = U'$, where $\langle X, Z, U \rangle$ and $\langle U', Z', Y \rangle$ are a pair of m.e. conductors w.r.t. the uncovered itinerary π in $\mathcal{G}$.*

Proof By Definition 9, a u-structure $\langle X, Z, Z', Y \rangle$ implies the existence of an uncovered path $\pi = \langle X, Z, U_1, .., U_k, Z', Y \rangle$ (possibly with $U_1 = U_k$ or $U_1 = Z', U_k = Z$) between nonadjacent X and Y in $\mathcal{M}$, corresponding to an uncovered itinerary in $\mathcal{G}$ where all nodes $\{Z, Z', U_1, .., U_k\}$ are ancestors of each other, but not of X or Y, which implies $\langle X, Z, U_1 \rangle$ and $\langle U_k, Z', Y \rangle$ are a pair of m.e. conductors w.r.t. the uncovered itinerary π in $\mathcal{G}$.

Conversely, if $\langle X, Z, U \rangle$ and $\langle U', Z', Y \rangle$ are a pair of m.e. conductors w.r.t. an uncovered itinerary $\pi = \langle X, Z, U, .., U', Z', Y \rangle$ in $\mathcal{G}$, then π is a also an uncovered path $\langle X, Z, .., Z', Y \rangle$ in $\mathcal{M}$, where all intermediate nodes are ancestor of each other, as $\{Z, U, .., U', Z'\} \subset SCC(Z)$, but not ancestors of X or Y,

and so $X \to Z$ and $Z' \leftarrow Y$ in $\mathcal{M}$, which by Definition 9 implies $\langle X, Z, Z', Y \rangle$ is a u-structure. ∎

Lemma 2 *In a CMAG $\mathcal{M}$, a pair of nodes $\langle X, Z \rangle$ is part of a u-structure $\langle X, Z, Z', Y \rangle$ with a node $Y \in \mathbf{Y} \subseteq pa(SCC(Z)) \smallsetminus adj(\{X, Z\})$, iff $X \in pa(Z)$, and X and Y are connected in the subgraph over $((SCC(Z) \smallsetminus adj(X)) \cup \{X, Z\} \cup \mathbf{Y}$.*

Proof The given implies the existence of some path from X, via adjacent nodes in the undirected part of the subgraph, to some node from $\mathbf{Y}$. Let Y be the first node from $\mathbf{Y}$ encountered along this path, then $\langle X, Z_1, .., Z_k, Y \rangle$ is a path over distinct nodes where all $Z_i \in SCC(Z)$ are ancestors of each other, but not of X or Y. If the path $\langle X, Z_1, .., Z_k, Y \rangle$ is not uncovered, then by Lemma 5 some subsequence $\langle X, U_1, .., U_m, Y \rangle$ with $\{U_1, .., U_m\} \subset \{Z_1, .., Z_k\}$ can be chosen so that $\langle X, U_1, .., U_m, Y \rangle$ is an uncovered path consisting solely of nodes in the subgraph. Furthermore, as all nodes adjacent to X in $\mathcal{M}$ are excluded from this subgraph with the exception of Z, it means that $Z = U_1 = Z_1$. We also know that $m \geq 2$, as all $Y \in \mathbf{Y}$ were taken not to be adjacent to Z, so $Z' = U_m \neq Z$.

Finally, as all $U_i \in SCC(Z)$ are ancestors of each other, but not of X or Y, in accordance with Definition 9, $\langle X, Z, Z', Y \rangle$ is a u-structure. ∎

8.1 PROOF OF THEOREM 1

In the proof of Theorem 1 we use the following (straightforward) implication:

Lemma 6 *In a CMAG $\mathcal{M}$, a virtual collider triple $\langle A, B, C \rangle$ uniquely corresponds to either:*

1. *a virtual v-structure $\langle A, B, C \rangle$, or*

2. *a u-structure $\langle A, B, B', C \rangle$, or*

3. *a u-structure $\langle A, B', B, C \rangle$,*

where for the latter two the complementary triple $\langle A, B', C \rangle$ is also a virtual collider triple.

Proof If virtual collider triple $\langle A, B, C \rangle$ corresponds to a virtual v-structure, then it cannot be part of a u-structure $\langle A, B, B', C \rangle$ or $\langle A, B', B, C \rangle$, as that would imply the path from A to C via B is not uncovered, contrary Definition 9. Similarly, if virtual collider triple $\langle A, B, C \rangle$ corresponds to a u-structure $\langle A, B, B', C \rangle$, then it cannot also correspond to a u-structure $\langle A, B', B, C \rangle$, as the combination would imply the presence of edges $A \to B$ and $B \leftarrow C$ in $\mathcal{M}$, which again would contradict the fact that the path $\langle A, B, .., B', C \rangle$ in $\mathcal{M}$ is uncovered. By Definition

10, in both cases the u-structure would imply that the complementary $\langle A, B', C \rangle$ also satisfies the definition of a virtual collider triple. ∎

We are now ready to prove the new ancestral CET:

Theorem 1 *Two directed graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, corresponding to CMAGs $\mathcal{M}_1$ and $\mathcal{M}_2$, are Markov (d-separation) equivalent iff*

(i) *$\mathcal{M}_1$ and $\mathcal{M}_2$ have the same skeleton,*

(ii) *$\mathcal{M}_1$ and $\mathcal{M}_2$ have the same v-structures,*

(iii) *$\mathcal{M}_1$ and $\mathcal{M}_2$ have the same virtual collider triples,*

(iv) *if $\langle A, B, C \rangle$ is a virtual collider triple, and $\langle A, D, C \rangle$ a virtual v-structure, then B is an ancestor of D in $\mathcal{M}_1$ iff B is an ancestor of D in $\mathcal{M}_2$.*

Proof We show that in terms of the CPAG the first 3 rules are equivalent to the first 4 rules in the original CET, and that the last rule is sound and implies the last two rules in the original CET, which means the combined set of rules is sound and sufficient to ensure Markov equivalence.

(i) By Lemma 3, two nodes in a CMAG $\mathcal{M}$ are adjacent if and only if they are (virtually) adjacent in the underlying graph $\mathcal{G}$, and so rule (i) is equivalent between the two CETs.

(ii)+(iii) By definitions 4 and 5 and rule (i), an unshielded triple $\langle A, B, C \rangle$ in a CPAG is either an unshielded conductor, an unshielded perfect nonconductor, or an unshielded imperfect nonconductor in $\mathcal{G}$. Therefore (ii).a+(ii).b in the original CET are equivalent to 'have the same unshielded perfect and imperfect nonconductors' (as the remaining unshielded triples then all must correspond to unshielded conductors). An unshielded perfect nonconductor in $\mathcal{G}$ is a v-structure in the CMAG $\mathcal{M}$, and by Definition 8 the subset of unshielded *imperfect* nonconductors is equivalent to the set of virtual v-structures. By Lemma 6, a virtual collider triple $\langle A, B, C \rangle$ is either a virtual v-structure, or part of a u-structure $\langle A, B, B', C \rangle$ or $\langle A, B', B, C \rangle$, for which, by Definition 10, the complementary $\langle A, B', C \rangle$ is also a virtual collider triple. By Lemma 1, that means that, depending on the skeleton from rule (i), either $\langle A, B, U \rangle$ and $\langle U', B', C \rangle$ are a pair of m.e. conductors w.r.t. uncovered itinerary $\langle A, B, U, .., U', B', C \rangle$, or $\langle A, B', U' \rangle$ and $\langle U, B, C \rangle$ are a pair of m.e. conductors w.r.t. uncovered itinerary $\langle A, B', U', .., U, B, C \rangle$. The latter all follow from rule (iii) in the original CET, and therefore rules (ii) + (iii) combined are equivalent to rules (ii).a + (ii).b + (iii) in the original CET.

(iv) If virtual collider triple $\langle A, B, C \rangle$ is a virtual v-structure, then, by Definition 8, rule (iv) is equivalent to the original CET rule (iv), and therefore sound. If $\langle A, B, C \rangle$ is part of a u-structure, then, by Lemma 1, rule (iv) is equivalent to the original CET rule (v), and therefore also sound. By Lemma

13

6, these are the only two possibilities for virtual collider triple $\langle A, B, C \rangle$, and so rule (iv) is sound.

In conclusion, all rules in Theorem 1 are sound, and imply the rules in the original CET. Therefore, Theorem 1 suffices to establish d-separation equivalence, which in turn, under the assumed global directed Markov property, ensures Markov equivalence between two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$. ∎

8.2 PROOF OF THEOREM 2

First a result on invariant ancestral paths from virtual collider triples in CMAGs.

Lemma 7 *Let CMAGs $\mathcal{M}_1$ and $\mathcal{M}_2$ agree on CET(i)-(iii) in Theorem 1. Let $\langle A, B, C \rangle$ be a virtual collider triple in $\mathcal{M}_1$, and let there be an uncovered ancestral path $\pi = B \longrightarrow\!\!* X_1 \longrightarrow\!\!* .. \longrightarrow\!\!* X_k$ in $\mathcal{M}_1$, so that B is an ancestor of X_k in $\mathcal{M}_1$. Assume there are no (virtual) v-structures $\langle A, X_i, C \rangle$ along the path π in $\mathcal{M}_1$. Then $\langle A, B, C \rangle$ is also a virtual collider triple in $\mathcal{M}_2$, $B \longrightarrow\!\!* X_1 \longrightarrow\!\!* .. \longrightarrow\!\!* X_k$ is an uncovered ancestral path in $\mathcal{M}_2$, and in particular, B is ancestor of X_k in $\mathcal{M}_2$ as well.*

Proof If $\mathcal{M}_1$ and $\mathcal{M}_2$ agree on CET(i)-(iii), then both have the same skeleton, v-structures, and virtual collider triples. In particular, it implies that $\langle A, B, C \rangle$ is also a virtual collider triple in $\mathcal{M}_2$, and that the uncovered path π in $\mathcal{M}_1$ corresponds to an uncovered path $\pi_2 = B \,*\!\!-\!\!* X_1 \,*\!\!-\!\!* .. \,*\!\!-\!\!* X_k$ in $\mathcal{M}_2$ as well. Remains to show this path is also ancestral in $\mathcal{M}_2$.

By contradiction: assume the path π_2 is *not* ancestral in $\mathcal{M}_2$, and let X_j be the first node along π_2 (starting from $X_0 = B$) that has $X_j \leftarrow X_{j+1}$ in $\mathcal{M}_2$. No (virtual) v-structures $\langle A, X_{i \geq 1}, C \rangle$ implies that every node X_i along π_2 has at most one edge to A or C (i.e. not to both), so assume no edge between A and X_{j+1}.

Now virtual collider triple $\langle A, B, C \rangle$ implies there is an uncovered path $A = X_{-(m+1)} \to X_{-m} - .. - X_0 = B$ (possibly $X_{-m} = B$) in $\mathcal{M}_2$. We concatenate this path and π_2 to obtain $\pi^+ = X_{-(m+1)} \to X_{-m} - .. - X_0 \longrightarrow\!\!* X_1 \longrightarrow\!\!* .. \longrightarrow\!\!* X_j \leftarrow X_{j+1}$ in $\mathcal{M}_2$. Let $X_{i \leq j}$ be the node along π^+, closest to X_j, that has an arc $X_{i-1} \to X_i$ in $\mathcal{M}_2$. Note there is no edge between X_{i-1} and X_{j+1}: for $0 < i \leq j$ because π_2 was uncovered, and for $i = -m$ because we assumed no edge between A and X_{j+1}, and for the remaining $-m < i \leq 0$ the edges are all undirected.

Then there is a subpath $X_{i-1} \to X_i - .. - X_j \leftarrow X_{j+1}$ (possibly $X_i = X_j$) in $\mathcal{M}_2$ where all intermediate nodes are ancestors of each other, but not of X_{i-1} or X_{j+1}, and so by Lemma 5 there is also an uncovered path $X_{i-1} \to X_p - .. - X_q \leftarrow X_{j+1}$ in $\mathcal{M}_2$. This path would either correspond

to a (virtual) v-structure (if $p = q$), or a u-structure (if $p < q$) in $\mathcal{M}_2$, and so, if $\mathcal{M}_1$ and $\mathcal{M}_2$ agree on CET(i)-(iii), would also appear as $X_q \leftarrow X_{j+1}$ in $\mathcal{M}_1$. But that would imply X_q is not an ancestor of X_{j+1}, in contradiction with the ancestral path from A via X_q to X_{j+1} in $\mathcal{M}_1$.

Therefore, there can be no edge $X_j \leftarrow X_{j+1}$ along π_2 in $\mathcal{M}_2$, and so $\pi_2 = B \longrightarrow\!\!* X_1 \longrightarrow\!\!* .. \longrightarrow\!\!* X_k$ is also an uncovered ancestral path in $\mathcal{M}_2$, and in particular, $B \in an(X_k)$ as well. ∎

We can apply the same approach to invariant descendants of v-structures.

Lemma 8 *Let CMAGs $\mathcal{M}_1$ and $\mathcal{M}_2$ agree on CET(i)-(iii) in Theorem 1. Let X_k be a descendant of a node X_0 in a v-structure $\langle A, X_0, C \rangle$ in $\mathcal{M}_1$. Then X_k is also a descendant of some node X_j (possibly $X_j = X_0$) in a v-structure $\langle A, X_j, C \rangle$ in $\mathcal{M}_2$.*

Proof If $\mathcal{M}_1$ and $\mathcal{M}_2$ agree on CET(i)-(iii), then both have the same skeleton, v-structures, and virtual collider triples. Therefore, if X_k is part of a v-structure $\langle A, X_k, C \rangle$ in $\mathcal{M}_1$, then also in $\mathcal{M}_2$, and so the lemma is trivially true.

If not, then $X_k \in de(X_0)$ in $\mathcal{M}_1$ implies there is an ancestral path from X_0 to X_k in $\mathcal{M}_1$, and so by Lemma 5 also an uncovered ancestral path $\pi = X_0 \longrightarrow\!\!* X_1 \longrightarrow\!\!* .. \longrightarrow\!\!* X_k$ in $\mathcal{M}_1$. By CET(i) the path π is also an uncovered path in $\mathcal{M}_2$. Furthermore, by Definition 8-(iii), none of nodes X_i along the path can be part of a virtual v-structure $\langle A, X_i, C \rangle$ in $\mathcal{M}_1$, and so also not in $\mathcal{M}_2$.

Let X_j be the node closest to X_k along the path π that is part of a v-structure $\langle A, X_j, C \rangle$ (possibly $X_j = X_0$), so there are no other (virtual) v-structures $\langle A, X_{i>j}, C \rangle$ along the path in $\mathcal{M}_1$.

We can now apply the same argument as in the proof of Lemma 7 to conclude that $X_j \longrightarrow\!\!* .. \longrightarrow\!\!* X_k$ is also an uncovered ancestral path in $\mathcal{M}_2$, and therefore indeed X_k is also a descendant of a v-structure $\langle A, X_j, C \rangle$ in $\mathcal{M}_2$. ∎

Lemma 5 has the following implication, analogous to the Claim in Case 5 in Theorem 2 in (Richardson, 1996a, p.38), showing that CET (iv) is only needed to distinguish edges at virtual v-structures.

Lemma 9 *Let CMAGs $\mathcal{M}_1$ and $\mathcal{M}_2$ agree on CET(i)-(iii) in Theorem 1. Let $\langle A, B, C \rangle$ be a virtual collider triple and $\langle A, D, C \rangle$ be a virtual v-structure in $\mathcal{M}_1$, with B an ancestor of D in $\mathcal{M}_1$, but not in $\mathcal{M}_2$, i.e. they do not agree on CET (iv) and so $\mathcal{M}_1$ and $\mathcal{M}_2$ are not Markov equivalent. Then they also differ on CET (iv) for some virtual collider triple $\langle A, B', C \rangle$ (possibly $B' = B$), and virtual v-structure $\langle A, D', C \rangle$ (possibly $D' = D$), such that $B' \in an(D')$ in*

$\mathcal{M}_1$, but $B' \notin an(D')$ in $\mathcal{M}_2$, where there is an uncovered ancestral path $B' \rightarrow\!\!* X_1 \rightarrow\!\!* .. \rightarrow\!\!* X_k \rightarrow\!\!* D'$ in $\mathcal{M}_1$ (possibly $B' = X_k$), corresponding to an uncovered path $B' \rightarrow\!\!* X_1 \rightarrow\!\!* .. \rightarrow\!\!* X_k \leftarrow D'$ in $\mathcal{M}_2$.

Proof In words: if CET (iv) is needed to distinguish between two CMAGs that are not Markov equivalent, but agree on CET (i)-(iii), then they differ on an edge $X_k \ast\!\!-\!\!\ast D'$ at a virtual v-structure $\langle A, D', C \rangle$ along an uncovered ancestral path (except for edge $X_k \ast\!\!-\!\!\ast D$) from another virtual collider triple $\langle A, B', C \rangle$ that is an ancestor of X_k in both $\mathcal{M}_1$ and $\mathcal{M}_2$.

As B is ancestor of D in $\mathcal{M}_1$ there is an ancestral path from B to D in $\mathcal{M}_1$, and so by Lemma 5 also an uncovered ancestral path from B to D.

Starting from B, let D' be the first node in a virtual v-structure $\langle A, D', C \rangle$ along this uncovered ancestral path in $\mathcal{M}_1$, such that $B \notin an(D')$ in $\mathcal{M}_2$ (possibly $D' = D$). Let B' be the node in a virtual v-structure $\langle A, B', C \rangle$ closest to D' on the path from B, or $B' = B$ if no such triple exists.

Then there is an uncovered ancestral path $\pi = B' \rightarrow\!\!* X_1 \rightarrow\!\!* .. \rightarrow\!\!* X_k \rightarrow\!\!* D'$ in $\mathcal{M}_1$, and so $B' \in an(D')$ in $\mathcal{M}_1$. However $B' \notin an(D)$ in $\mathcal{M}_2$: by construction B is an ancestor of B' in $\mathcal{M}_2$ (otherwise B' would satisfy the criterion for D' but be closer to B, contrary the assumed), so if $B' \in an(D')$ in $\mathcal{M}_2$, then B would also be ancestor of D' in $\mathcal{M}_2$, again contrary the assumed.

At least one node along the path π is not an ancestor of its successor along the path in $\mathcal{M}_2$, otherwise B' would be ancestor of D' in $\mathcal{M}_2$ as well. Let X_j be the first such node along the path starting from B', such that $B' \rightarrow\!\!* X_1 \rightarrow\!\!* .. \rightarrow\!\!* X_j \leftarrow X_{j+1} \ast\!\!-\!\!\ast .. \ast\!\!-\!\!\ast X_k \ast\!\!-\!\!\ast D'$ in $\mathcal{M}_2$. By definition, no node X_i along this path appears in a v-structure $\langle A, X_i, C \rangle$ in $\mathcal{M}_1$, otherwise $\langle A, D', C \rangle$ would not be a virtual v-structure. By construction, there are also no virtual v-structures $\langle A, X_i, C \rangle$ on this path between B' and D' in $\mathcal{M}_1$.

We now show this implies $X_{j+1} = D'$. By contradiction, suppose $X_{j+1} \neq D'$. Then there exists an uncovered ancestral path $B' \rightarrow\!\!* .. \rightarrow\!\!* X_j \rightarrow\!\!* X_{j+1}$ in $\mathcal{M}_1$, where no node X_i along this path (except perhaps B') is part of a virtual v-structure $\langle A, X_i, C \rangle$. Therefore by Lemma 7, then $B' \rightarrow\!\!* .. \rightarrow\!\!* X_j \rightarrow\!\!* X_{j+1}$ in $\mathcal{M}_2$, in contradiction with the assumed $X_j \leftarrow X_{j+1}$. Therefore $X_{j+1} = D'$, and $X_j = X_k$, and so $B' \rightarrow\!\!* .. \rightarrow\!\!* X_k \leftarrow D'$ is an uncovered path in $\mathcal{M}_2$. It also implies that $B' \in an(X_k)$ in both $\mathcal{M}_1$ and $\mathcal{M}_2$. ∎

It means that if two CMAGs $\mathcal{M}_1$ and $\mathcal{M}_2$ are only different on CET rule (iv), then they (also) differ on rule (iv) between two nodes connected by a very specific path configuration, which we will use in the subsequent *Graph-to-CPAG* al-gorithm.

Next we show that for the case $\mathcal{M}_2$ with $B \notin an(D)$ above, the final edge $X_k \leftarrow D'$ is an invariant edge in any CMAG that is Markov equivalent to $\mathcal{M}_2$. We also show that in order to identify this invariant edge, we do not need to find all possible pairs of nodes B and D that satisfy the conditions in Lemma 9 above, but only the existence of *some* pair that do.

Lemma 10 *In CMAG $\mathcal{M}_1$, let $\langle A, D, C \rangle$ be a virtual v-structure, with $X_k \leftarrow D$ in $\mathcal{M}_1$, with X_k is not a descendant of some node X' in a v-structure $\langle A, X', C \rangle$. Assume there exists a virtual collider triple $\langle A, B, C \rangle$ in $\mathcal{M}_1$ such that B is not an ancestor of D, B is an ancestor of X_k, and there exists an uncovered path $\pi = B \rightarrow\!\!* X_1 \rightarrow\!\!* .. \rightarrow\!\!* X_k \leftarrow D$ in $\mathcal{M}_1$. Then there exists a node B' in a virtual collider triple $\langle A, B', C \rangle$ in $\mathcal{M}_1$ (possibly $B' = B$), such that for any CMAG $\mathcal{M}_2$ that is Markov equivalent to $\mathcal{M}_1$, in both $\mathcal{M}_1$ and $\mathcal{M}_2$ it holds that: 1) B' is not an ancestor of D, 2) B' is an ancestor of X_k, 3) X_k is not a descendant of some node X' in a v-structure $\langle A, X', C \rangle$, and 4) there exists an uncovered path $\pi = B' \rightarrow\!\!* X_j \rightarrow\!\!* .. \rightarrow\!\!* X_k \leftarrow D$, and in particular, then $X_k \leftarrow D$ in $\mathcal{M}_2$ as well.*

Proof The given implies that $\mathcal{M}_1$ and $\mathcal{M}_2$ agree on CET rules (i)-(iv). By rules (i)-(iii), both have the same skeleton, uncovered paths, v-structures, and virtual collider triples. Then if X_k is not a descendant of some node X' in a v-structure $\langle A, X', C \rangle$ in $\mathcal{M}_1$, then by Lemma 8 neither in $\mathcal{M}_2$, so (3) holds.

Let B' be the node closest to D along the path π in $\mathcal{M}_1$ that is part of a virtual v-structure $\langle A, B', C \rangle$ (possibly $B' = B$ or $B' = X_k$), or $B' = B$ if no such virtual v-structure exists. Then the subpath $\pi' = B' \rightarrow\!\!* X_j \rightarrow\!\!* .. \rightarrow\!\!* X_k \leftarrow D$ is an uncovered path in $\mathcal{M}_1$, and by construction there are no other v-structures or virtual v-structures $\langle A, X_i, C \rangle$ along the path π' between B' and D. Then by Lemma 7 the subpath $B' \rightarrow\!\!* X_j \rightarrow\!\!* .. \rightarrow\!\!* X_k$ is also an uncovered ancestral path in $\mathcal{M}_2$, and by CET(i), π' itself is also an uncovered path in $\mathcal{M}_2$, and therefore $B' \rightarrow\!\!* X_j \rightarrow\!\!* .. \rightarrow\!\!* X_k \ast\!\!-\!\!\ast D$ is an uncovered path in $\mathcal{M}_2$, which implies (2) holds as well.

Node B' is not an ancestor of D in $\mathcal{M}_1$ (otherwise B as ancestor of B' would be as well, contrary the given). For Markov equivalent graph $\mathcal{M}_2$, CET rule (iv) on virtual collider triple $\langle A, B', C \rangle$ and virtual v-structure $\langle A, D, C \rangle$ then implies B' is not an ancestor of D in $\mathcal{M}_2$ either, which ensures (1).

But then by contradiction, if $X_k \rightarrow D$ or $X_k - D$ in $\mathcal{M}_2$, then π' would be an ancestral path from B' to D, which would imply B' is an ancestor of D in $\mathcal{M}_2$, contrary the given. Therefore $X_k \leftarrow D$ in $\mathcal{M}_2$ as well, which proves (4). ∎

The Lemma above implies that if the triggering conditions in the second clause of the 'if' statement in part 2 of Algorithm 2 apply to a CMAG $\mathcal{M}_1$, then the same conclusion $X_k \leftarrow D$ will (also) trigger on a pattern that is *identical* between all CMAGs Markov equivalent to $\mathcal{M}_1$.

Note that for $B \in an(D)$ in $\mathcal{M}_1$, the final edge $X_k \longrightarrow\!\!* D'$ is *not* necessarily an invariant edge in every CMAG that is Markov equivalent to $\mathcal{M}_1$, as an edge $X_k \leftarrow D'$ does not preclude the existence of another path π' along which B' is an ancestor of D'. Therefore, the invariance in Lemma 10 only applies to the explicit case '$B \notin an(D)$'.

We can also show that when Lemma 10 is needed to orient an invariant $X_k \leftarrow D$, then X_k is part of a cycle. **Corollary 1** In a CMAG $\mathcal{M}$, let $\langle A, B, C \rangle$ be a virtual collider triple, and $\langle A, D, C \rangle$ be a virtual v-structure, with $B \notin an(D)$. Let $\pi = B \longrightarrow\!\!* X_1 \longrightarrow\!\!* .. \longrightarrow\!\!* X_k \leftarrow D$ be an uncovered path in $\mathcal{M}$ (possibly $B = X_k$), with X_k not a descendant of a node X' in a v-structure $\langle A, X', C \rangle$. Then, if $X_k \leftarrow D$ is not already implied by CET rules (i)-(iii), then X_k is part of a cycle in $\mathcal{M}$.

Proof If X_k is part of a virtual collider triple $\langle A, X_k, C \rangle$, then by definition X_k is part of a cycle, and so then the claim holds.

If X_k is *not* part of a virtual collider triple $\langle A, X_k, C \rangle$, then if $X_{k-1} \rightarrow X_k$ in $\mathcal{M}$, then $X_{k-1} \rightarrow X_k \leftarrow D$ would be a (virtual) v-structure, and so already be implied by CET (i)-(iii). Therefore, if CET (iv) is needed to orient $X_k \leftarrow D$, then $X_{k-1} \longrightarrow\!\!* X_k \leftarrow D$ must be an unshielded noncollider, and so $X_{k-1} \text{---} X_k$, which implies X_{k-1} and X_k are part of a cycle. ∎

We use this in the implementation to quickly filter out candidate nodes W in part 2 of Algorithm 2 that are not part of a cycle ($|SCC(W)| = 1$), in order to avoid unnecessary path searches.

Finally, we show that all edges between two virtual collider nodes in a CMAG are invariant.

Lemma 11 *If two CMAGs $\mathcal{M}_1$ and $\mathcal{M}_2$ are Markov equivalent, then if $\langle A, B, C \rangle$ and $\langle A, D, C \rangle$ are virtual collider triples with B and D adjacent, then B is ancestor of D in $\mathcal{M}_1$ iff B is ancestor of D in $\mathcal{M}_2$.*

Proof We consider three cases: 1) both virtual collider triples correspond to virtual v-structures, 2) one virtual collider triple corresponds to a virtual v-structure, and the other is part of a u-structure, or 3) both are part of a u-structure. Below we will tackle each of these cases in turn:

Case 1: this is equivalent to rule (iv) of the original CET.

Case 2: let $\langle A, B, C \rangle$ be the virtual v-structure, and $\langle A, D, C \rangle$ be part of a u-structure $\langle A, D, D', C \rangle$. Note this implies there is no edge between C and D. Then if $B \rightarrow D$

in $\mathcal{M}_1$, i.e. B is NOT a descendant of D, then this satisfies rule (v) of the original CET, meaning B is also not a descendant of D in $\mathcal{M}_2$, and so $B \rightarrow D$ in $\mathcal{M}_2$ as well. If $B \leftarrow D$ in $\mathcal{M}_1$, then $C \rightarrow B \leftarrow D$ would be a (virtual) v-structure, and be invariant by CET rule (ii)/(iii). The only remaining possibility is $D \text{---} B$ in $\mathcal{M}_1$, which by symmetry then must also apply to $\mathcal{M}_2$.

Case 3: now both $\langle A, B, C \rangle$ and $\langle A, D, C \rangle$ are part of a u-structure, but neither are virtual v-structures.

Consider $B \rightarrow D$ in $\mathcal{M}_1$. As A and C cannot both have an edge to B (for then $\langle A, B, C \rangle$ would be a virtual v-structure), assume there is no edge between B and C. Then if virtual collider triple $\langle A, D, C \rangle$ corresponds to a u-structure $\langle A, D', D, C \rangle$, then $B \rightarrow D \leftarrow C$ corresponds to a (virtual) v-structure, and would be invariant by CET rule (ii)/(iii). If virtual collider triple $\langle A, D, C \rangle$ corresponds to a u-structure $\langle A, D, D', C \rangle$, then $B \rightarrow D \text{---} .. \text{---} D' \leftarrow C$ would be a path in $\mathcal{M}_1$. Let D^* be the node closest to C along the path with an edge to B. If $D^* = D'$, then $B \rightarrow D^* \leftarrow C$ would be a invariant (virtual) v-structure. If $D^* \neq D'$ then $B \rightarrow D^* \text{---} .. \text{---} D' \leftarrow C$ would be an invariant u-structure in $\mathcal{M}_1$, and so then $B \rightarrow D^*$ in $\mathcal{M}_2$. Given that D and D^* are both ancestors/descendants of each other, it follows that then $B \rightarrow D$ in $\mathcal{M}_2$ as well. In all cases this implies edge $B \rightarrow D$ would be invariant by CET rule (ii)/(iii), and so appear as $B \rightarrow D$ in $\mathcal{M}_2$.

For $B \leftarrow D$ in $\mathcal{M}_1$ we can repeat the previous argument with the roles of B and D reversed, which implies $B \leftarrow D$ in $\mathcal{M}_2$ as well. That leaves $B \text{---} D$ in $\mathcal{M}_1$ as the only remaining option, and so necessarily must appear as $B \text{---} D$ in $\mathcal{M}_2$ as well. ∎

We can now prove that the Graph-to-CPAG algorithm in the main article is sound and d-separation complete, meaning that the output CPAG can be used to establish Markov equivalence between cyclic directed graphs.

Theorem 2 *For two different directed graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, let $\mathcal{P}_1$ and $\mathcal{P}_2$ be the corresponding CPAGs output by the Graph-to-CPAG algorithm. Then $\mathcal{G}_1$ is Markov (d-separation) equivalent to $\mathcal{G}_2$ iff $\mathcal{P}_1 = \mathcal{P}_2$.*

Proof We cover three aspects of the claim: 1) soundness of the output PAG, 2) d-separation completeness of the output PAG (which implies it is a CPAG), and 3) equality between CPAGs if and only if they are Markov equivalent.

1) Soundness of the algorithm follows from Theorem 1, in combination with the fact that: a) each orientation in part 1 of the algorithm has a direct match to an invariant feature implied by the CET rules (i)-(iii), b) all orientations between adjacent triples from the first 'if' clause in part 2 of the algorithm are sound by Lemma 11, and c) the remaining orientations between nonadjacent triples implied

by CET rule (iv) from the second 'if' clause in part 2 are sound by Lemma 10. Therefore all orientations correspond to invariant features in the Markov equivalence class of the input graph $\mathcal{G}$, which guarantees the output is a valid PAG.

2) d-separation completeness follows from the fact that if two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are not Markov equivalent, then the algorithm will make at least one different orientation in the corresponding output PAGs $\mathcal{P}_1$ and $\mathcal{P}_2$ (which therefore qualify as CPAGs).

Part 1 of the algorithm captures the entire skeleton, all v-structures, all virtual v-structures, and all invariant edges from u-structures into a cycle. Therefore, if CMAGs $\mathcal{M}_1$ and $\mathcal{M}_2$ corresponding to graphs $\mathcal{G}_1$ resp. $\mathcal{G}_2$ differ in *any* feature corresponding to CET rules (i)-(iii), then this will lead to at least one different edge/orientation between the output PAGs $\mathcal{P}_1$ and $\mathcal{P}_2$.

If $\mathcal{M}_1$ and $\mathcal{M}_2$ agree on CET (i)-(iii), but are not Markov equivalent (i.e. they disagree only on CET (iv)), then by Lemma 9 they differ on (at least) one $X_k \ast\!\!-\!\!\ast D$ along an uncovered path $B \rightarrow\!\!\ast X_1 \rightarrow\!\!\ast .. \rightarrow\!\!\ast X_k \ast\!\!-\!\!\ast D$ between some virtual collider triple $\langle A, B, C \rangle$ and virtual v-structure $\langle A, D, C \rangle$, for which $B \in an(D)$ in $\mathcal{M}_1$, but $B \notin an(D)$ in $\mathcal{M}_2$.

In part 2 of the algorithm, if $B \notin an(D)$ in the CMAG $\mathcal{M}$ corresponding to input graph $\mathcal{G}$, then *all* such uncovered paths (or edge) between B and D will obtain an orientation $X_k \leftarrow D$ (possibly $X_k = B$) in the output PAG $\mathcal{P}$. If $B \in an(D)$ in the CMAG $\mathcal{M}$, then *at least one* of these paths must have $X_k \rightarrow\!\!\ast D$ in $\mathcal{M}$ (again possibly $X_k = B$), otherwise there would be no ancestral path from B to D in $\mathcal{M}$, contrary the assumption that $B \in an(D)$. By the soundness of the algorithm, this edge will obtain either $X_k \rightarrow D$, $X_k - D$, or $X_k \circ\!\!-\!\!\circ D$ in the output PAG $\mathcal{P}$ (again possibly with $X_k = B$).

Therefore if two CMAGs $\mathcal{M}_1$ and $\mathcal{M}_2$ are not Markov equivalent, then there is at least one edge or orientation different between the corresponding output PAGs $\mathcal{P}_1$ and $\mathcal{P}_2$. Therefore the output PAG $\mathcal{P}$ uniquely identifies the Markov equivalence class of input graph $\mathcal{G}$, which also implies the output PAG $\mathcal{P}$ is indeed a CPAG.

3) equality: the 'only if' part follows from the d-separation completeness of the algorithm, above. Remains to show that when two graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ are Markov equivalent then the corresponding output CPAGs $\mathcal{P}_1$ and $\mathcal{P}_2$ are also *identical*. For an input graph $\mathcal{G}_1$, part 1 of the algorithm exhaustively searches for all and only the edges (skeleton), v-structures, and virtual collider triples implied by CET (i)-(iii). Any graph $\mathcal{G}_2$ that is Markov equivalent to $\mathcal{G}_1$ must have the same skeleton, v-structures, and virtual collider triples, and therefore the two (intermediate) PAGs $\mathcal{P}_1$ and $\mathcal{P}_2$ must be identical after part 1.

For the remaining orientations in part 2 of the algorithm,

note that all required elements: virtual v-structures, virtual collider triples, and skeleton are implied by CET rules (i)-(iii), and therefore identical between the corresponding CMAGs $\mathcal{M}_1$ and $\mathcal{M}_2$. By Lemma 10, if an orientation in part 2 is triggered for some CMAG $\mathcal{M}_1$, then it would also trigger on a pattern in $\mathcal{M}_1$ that is *invariant* between all Markov equivalent CMAGs.

Therefore, for Markov equivalent $\mathcal{M}_1$ and $\mathcal{M}_2$, any orientation for $\mathcal{P}_1$ in part 2 of the algorithm will also be oriented identically in $\mathcal{P}_2$ and v.v. Combined with the fact that $\mathcal{P}_1$ and $\mathcal{P}_2$ are identical after part 1 of the algorithm, this proves the 'if' part of the Theorem.

As a result, for graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, the corresponding output $\mathcal{P}_1$ and $\mathcal{P}_2$ of the Graph-to-CPAG algorithm is identical iff $\mathcal{G}_1$ and $\mathcal{G}_2$ are Markov equivalent. ∎

9 MARKOV PROPERTIES FOR STRUCTURAL CAUSAL MODELS

We state here some of the key definitions and results in the theory of Structural Causal Models (SCMs). These models, also known as Structural Equation Models (SEMs), were introduced a century ago by Wright (1921) and popularized in AI by Pearl (2009). We follow here the treatment of Bongers et al. (2021), as it deals with cycles in a mathematically rigorous way.

Definition 11 (SCM) *A Structural Causal Model (SCM) is a tuple $M = \langle \mathbf{V}, \mathbf{W}, \mathcal{X}_\mathbf{V}, \mathcal{X}_\mathbf{W}, \mathbf{f}, P_M \rangle$ of:*

1. *finite disjoint index sets $\mathbf{V}, \mathbf{W}$ for the endogenous and exogenous variables in the model, respectively;*

2. *a product of standard measurable spaces $\mathcal{X}_\mathbf{V} = \prod_{v \in \mathbf{V}} \mathcal{X}_v$, which define the domains of the endogenous variables;*

3. *a product of standard measurable spaces $\mathcal{X}_\mathbf{W} = \prod_{w \in \mathbf{W}} \mathcal{X}_w$, which define the domains of the exogenous variables;*

4. *a measurable function $\mathbf{f} : \mathcal{X}_\mathbf{V} \times \mathcal{X}_\mathbf{W} \rightarrow \mathcal{X}_\mathbf{V}$, the <u>causal mechanism</u>;*

5. *a product probability measure $P_M = \prod_{w \in \mathbf{W}} P_w$ on $\mathcal{X}_\mathbf{W}$, with each P_w a probability measure on $\mathcal{X}_w$, specifying the <u>exogenous distribution</u>.*

The causal structure of the SCM is encoded by the dependences of the components of $\mathbf{f}$ on the variables in the model. This is formalized by:

Definition 12 (Parent) *Let M be an SCM. We call $i \in \mathbf{V} \cup \mathbf{W}$ a parent of $k \in \mathbf{V}$ if and only if there does not exist a*

measurable function $\tilde{f}_k : \mathcal{X}_{\mathbf{V} \setminus \{i\}} \times \mathcal{X}_{\mathbf{W} \setminus \{i\}} \to \mathcal{X}_k$ such that for P_M-almost every $\mathbf{w} \in \mathcal{X}_{\mathbf{W}}$, for all $\mathbf{v} \in \mathcal{X}_{\mathbf{V}}$,

$$v_k = \tilde{f}_k(\mathbf{v}_{\setminus i}, \mathbf{w}_{\setminus i}) \iff v_k = f_k(\mathbf{v}, \mathbf{w}).$$

Intuitively, this means that the k'th component of $\mathbf{f}$ does depend on the i'th variable. This definition allows us to define the directed mixed graph (DMG) associated to an SCM:

Definition 13 (Graph) *Let M be an SCM. The <u>graph</u> of M, denoted $\mathcal{G}(M)$, is defined as the directed mixed graph with nodes $\mathbf{V}$, directed edges $v_1 \to v_2$ iff v_1 is a parent of v_2 according to M, and bidirected edges $v_1 \leftrightarrow v_2$ iff there exists $w \in \mathbf{W}$ such that w is parent of both v_1 and v_2 according to M.*

If $\mathcal{G}(M)$ is acyclic, we call the SCM M <u>acyclic</u>, otherwise we call the SCM <u>cyclic</u>. If $\mathcal{G}(M)$ contains no bidirected edges, we call the endogenous variables in the SCM M <u>causally sufficient</u> (which is what we assumed in the present work for simplicity).

SCMs provide an implicit description of their solutions.

Definition 14 (Solutions) *A random variable $\mathfrak{X} = (\mathfrak{X}_{\mathbf{V}}, \mathfrak{X}_{\mathbf{W}})$ is called a <u>solution</u> of the SCM M if $\mathfrak{X}_{\mathbf{V}} = (\mathfrak{X}_v)_{v \in \mathbf{V}}$ with $\mathfrak{X}_v \in \mathcal{X}_v$ for all $v \in \mathbf{V}$, $\mathfrak{X}_{\mathbf{W}} = (\mathfrak{X}_w)_{w \in \mathbf{W}}$ with $\mathfrak{X}_w \in \mathcal{X}_w$ for all $w \in \mathbf{W}$, the distribution $\mathbb{P}(\mathfrak{X}_{\mathbf{W}})$ is equal to the exogenous distribution P_M, and the <u>structural equations</u>:*

$$\mathfrak{X}_v = f_v(\mathfrak{X}_{\mathbf{V}}, \mathfrak{X}_{\mathbf{W}}) \quad a.s.$$

hold for all $v \in \mathbf{V}$.

For acyclic SCMs, solutions exist and have a unique distribution that is determined by the SCM. This is not generally the case in cyclic SCMs, as these could have no solution at all, or could have multiple solutions with different distributions.

Definition 15 (Unique solvability) *An SCM M is said to be <u>uniquely solvable w.r.t.</u> $\mathbf{O} \subseteq \mathbf{V}$ if there exists a measurable mapping $\mathbf{g}_{\mathbf{O}} : \mathcal{X}_{\mathrm{pa}_{\mathcal{G}(M)}(\mathbf{O}) \setminus \mathbf{O}} \to \mathcal{X}_{\mathbf{O}}$ such that for P_M-almost every $\mathbf{w} \in \mathcal{X}_{\mathbf{W}}$, for all $\mathbf{v} \in \mathcal{X}_{\mathbf{V}}$:*

$$\mathbf{v}_{\mathbf{O}} = \mathbf{g}_{\mathbf{O}}(\mathbf{v}_{(\mathrm{pa}_{\mathcal{G}(M)}(\mathbf{O}) \setminus \mathbf{O}) \cap \mathbf{V}}, \mathbf{w}_{\mathrm{pa}_{\mathcal{G}(M)}(\mathbf{O}) \cap \mathbf{W}})$$
$$\iff \quad \mathbf{v}_{\mathbf{O}} = \mathbf{f}_{\mathbf{O}}(\mathbf{v}, \mathbf{w}).$$

Loosely speaking: the structural equations for $\mathbf{O}$ have an essentially unique solution for $\mathbf{v}_{\mathbf{O}}$ in terms of the other variables appearing in those equations. If M is uniquely solvable with respect to $\mathbf{V}$ (in particular, this holds if M is acyclic), then it induces a unique <u>observational distribution</u> $P_M(\mathfrak{X}_{\mathbf{V}})$, the push-forward of P_M through $\mathbf{g}_{\mathbf{V}}$.

One of the key aspects of SCMs—which we do not discuss here in detail because we do not make use of it in this work—is their causal semantics, which is defined in terms of interventions. Instead, we discuss only their probabilistic properties. In particular, under appropriate assumptions, the graph $\mathcal{G}(M)$ of an SCM M represents conditional independences that its solutions must satisfy. As shown already by Spirtes (1994, 1995), the directed global Markov property does *not* hold in general for cyclic SCMs.

Example 1 (d-separation fails) *Consider the SCM $M = \langle \{A, B, C, D\}, \{5, 6, 7, 8\}, \mathbb{R}^4, \mathbb{R}^4, \mathbf{f}, P_M \rangle$ where P_M is the standard-normal distribution on $\mathbb{R}^4$, and the causal mechanism is given by:*

$$\mathbf{f}(\mathbf{x}) = (x_5, x_6, x_A x_D + x_7, x_B x_C + x_8)$$

The graph $\mathcal{G}(M)$ is depicted in Figure 1 (left). This SCM is uniquely solvable with respect to its strongly connected components $\{A\}$, $\{B\}$, and $\{C, D\}$. One can check that for every solution $\mathfrak{X}$ of M, $\mathfrak{X}_A$ is not independent of $\mathfrak{X}_B$ given $\{\mathfrak{X}_C, \mathfrak{X}_D\}$. However, the nodes A and B are d-separated given $\{C, D\}$ in $\mathcal{G}(M)$. Hence the global directed Markov property does not hold for M.

For more concrete examples of cyclic SCMs, we refer the reader to (Bongers et al., 2021). Spirtes (1994) proved a weaker Markov property in terms of a 'collapsed graph', assuming causal sufficiency and densities. Forré and Mooij (2017) found the following formulation in terms of 'σ-separation' that is immediately applicable to the graph of the SCM itself.

Definition 16 (Blockable and unblockable noncolliders) *Let $\mathcal{G}$ be a directed mixed graph and π a path in $\mathcal{G}$. We call a noncollider on π <u>unblockable</u> if it is not an end-node and it only has outgoing edges on π to nodes in the same strongly connected component of $\mathcal{G}$; otherwise, it is called <u>blockable</u>.*

If $\mathcal{G}$ is acyclic then all noncolliders are blockable.

Definition 17 (σ-separation) *For a triple of node sets $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ in a graph $\mathcal{G}$, we say that $\mathbf{X}$ is σ-connected to $\mathbf{Y}$ given $\mathbf{Z}$ iff there is an $X \in \mathbf{X}$ and $Y \in \mathbf{Y}$ such that there is a path π between X and Y on which every blockable noncollider is not in $\mathbf{Z}$, and every collider on π is an ancestor of $\mathbf{Z}$; otherwise $\mathbf{X}$ and $\mathbf{Y}$ are said to be <u>σ-separated</u> given $\mathbf{Z}$.*

Note the small difference with the definition of d-connection: σ-connection only considers the *blockable* noncolliders. The following general result was shown by Forré and Mooij (2017).

Theorem 3 (σ-Separation Markov property) *Let M be an SCM that is uniquely solvable w.r.t. each strongly connected component of $\mathcal{G}(M)$. Then, the observational distribution of M exists and is unique. Furthermore, for a solution $\mathfrak{X}$ of M and for $\mathbf{A}, \mathbf{B}, \mathbf{C} \subseteq \mathbf{V}$: if $\mathbf{A}$ is σ-separated from $\mathbf{B}$ given $\mathbf{C}$ in $\mathcal{G}(M)$, then $\mathfrak{X}_{\mathbf{A}}$ is conditionally independent of $\mathfrak{X}_{\mathbf{B}}$ given $\mathfrak{X}_{\mathbf{C}}$.*

Proof See the proof of Theorem A.21 in Bongers et al. (2021). ∎

Under certain additional assumptions, one can show the stronger d-separation criterion (also known as the global directed Markov property).

Theorem 4 (d-Separation Markov property) *Let M be an SCM that satisfies one of the following three assumptions:*

1. *M is acyclic;*

2. • *all endogenous domains $\mathcal{X}_v$ for $v \in \mathbf{V}$ are discrete, and*
   • *M is uniquely solvable w.r.t. each ancestral subset $A \subseteq \mathbf{V}$ (that is, each subset $A \subseteq \mathbf{V}$ such that $\mathrm{an}_{\mathcal{G}(M)}(A) = A$);*

3. • *$\mathcal{X}_{\mathbf{V}} = \mathbb{R}^{\mathbf{V}}$ and $\mathcal{X}_{\mathbf{W}} = \mathbb{R}^{\mathbf{W}}$, and*
   • *$\mathbf{f}$ is a linear mapping, and*
   • *each $v \in \mathbf{V}$ has at least one parent in $\mathbf{W}$ according to M, and*
   • *P_M has a density w.r.t. the Lebesgue measure on $\mathbb{R}^{\mathbf{W}}$.*

Then, the observational distribution of M exists and is unique. Furthermore, for a solution $\mathfrak{X}$ of M and for $\mathbf{A}, \mathbf{B}, \mathbf{C} \subseteq \mathbf{V}$: if $\mathbf{A}$ is d-separated from $\mathbf{B}$ given $\mathbf{C}$ in $\mathcal{G}(M)$, then $\mathfrak{X}_{\mathbf{A}}$ is conditionally independent of $\mathfrak{X}_{\mathbf{B}}$ given $\mathfrak{X}_{\mathbf{C}}$.

Proof See the proof of Theorem A.7 in Bongers et al. (2021). The acyclic case is well known. The discrete case fixes the erroneous theorem by Pearl and Dechter (1996), for which a counterexample was found by Neal (2000), by adding the assumption of unique solvability with respect to each ancestral subset, and extends it to allow for bidirected edges in the graph. The linear case is an extension of existing results for the linear-Gaussian setting without bidirected edges Spirtes (1994, 1995); Koster (1996) to a linear (possibly non-Gaussian) setting with bidirected edges in the graph. ∎

For this paper, we assume that the global directed Markov property holds with respect to a graph that contains no bidirected edges. From the above theorem, it follows that this will hold if the data comes from the observational distribution of a causally sufficient SCM that falls into either the acyclic case (1), the discrete case (2), or the linear case (3). Note that these assumptions are sufficient, but not necessary.