TECHNICAL REPORT
R-128
June, 2025

Causal Discovery over Clusters of Variables in Markovian Systems

Tara V Anand
Causal Artificial Intelligence Laboratory
Columbia University
tara.v.anand@columbia.edu

Adèle H Ribeiro
Institute of Medical Informatics
University of Münster
adele.ribeiro@uni-muenster.de

Jin Tian
Mohamed bin Zayed University
of Artificial Intelligence
jin.tian@mbzuai.ac.ae

George Hripcsak
Department of Biomedical Informatics
Columbia University
gh13@cumc.columbia.edu

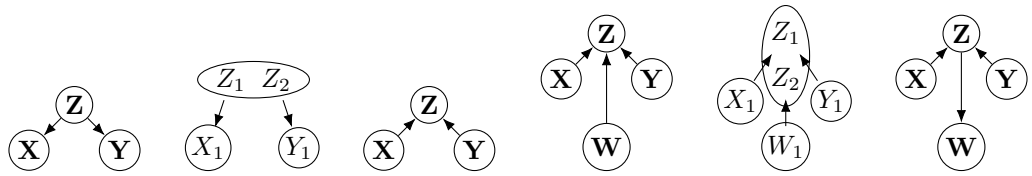
Elias Bareinboim
Causal Artificial Intelligence Laboratory
Columbia University
eb@cs.columbia.edu

Abstract

Causal discovery approaches are limited by scalability and are primarily for learning relationships among variables. Learning causal relationships among sets or clusters of variables is of interest because for some applications, relationships among variables grouped in semantically meaningful ways is the goal, and in others, clusters improve causal discovery in high-dimensions by reducing dimensionality. Here, we introduce an approach for learning over clusters in Markov causal systems. We develop a new graphical model to encode knowledge of relationships between user-defined clusters while fully representing independencies and dependencies over clusters, faithful to a given distribution. Then we define and characterize a graphical equivalence class of these models that share cluster-level independence information. Lastly, we introduce an algorithm for causal discovery, leveraging these new representations, to soundly represent learnable causal relationships between clusters of variables.

1 Introduction

Causal discovery, where observational data are used to uncover causal relationships between variables, is a task of interest in many domains [13, 16]. However, existing algorithms are often computationally prohibitive with many variables and prone to errors in practice [7]. One approach to improve scalability in high-dimensional settings is to group variables into clusters and infer relationships between these clusters. In the context of diagrams constructed from knowledge used for identification of causal effects, Cluster Directed Acyclic Graphs (C-DAGs) [1] are introduced as causal diagrams defined over clusters, allowing the visual representation of a high-dimensional system to be simplified and the requisite knowledge for graph specification lessened. In a C-DAG, nodes are clusters of variables, and an edge exists if a variable in one cluster causally influences a variable in another. C-DAGs are assumed to be constructed based on partial knowledge of causal and confounding relationships between variables across clusters, oblivious to variable-level relationships within clusters.



(a) $G_{\mathbf{C}_1}$ (b) G_1 (c) $\mathcal{P}_{\mathbf{C}_1}$ (d) $G_{\mathbf{C}_2}$ (e) G_2 (f) $\mathcal{P}_{\mathbf{C}_2}$

Figure 1: (a) and (d) are C-DAGs. (b) and (e) are DAGs in the class of $G_{\mathbf{C}}$ and $G_{\mathbf{C}_2}$, respectively. (c): an attempted graphical equivalence class for (a) after applying a collider search test given a distribution from G_1. (f): an attempted graphical equivalence class for $G_{\mathbf{C}_2}$ after applying a modified collider search test, where $\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y}|\mathbf{Z}$ is required, and applying an orientation rule, given a distribution from G_2.

In this work, we address causal discovery over clusters of variables. We assume that the underlying causal model is a DAG over individual variables $\mathbf{V} = \{V_1, ...V_n\}$ with no latent variables, known as a Markovian system. Given a predefined partition of $\mathbf{V}$ into clusters $\mathbf{C} = \{\mathbf{C}_1, \ldots, \mathbf{C}_k\}$, we aim to learn causal relationships between these clusters based on observed conditional (in)dependencies between clusters encoded in the distribution $P(\mathbf{C}) = P(\mathbf{C}_1, \ldots, \mathbf{C}_k)$ without access to variable-level relationships.

One might attempt to simply treat each cluster as a multivariate random variable and apply existing causal discovery algorithms like PC [15]. However, consider the DAG G_1 and its corresponding C-DAG $G_{\mathbf{C}_1}$ in Figure 1(b) and 1(a), respectively. Assuming a probability distribution faithful to G_1, PC will correctly construct the skeleton $\mathbf{X} - \mathbf{Z} - \mathbf{Y}$, but observing the independence $\mathbf{X} \perp\!\!\!\perp \mathbf{Y}$ will lead to the collider structure $\mathcal{P}_{\mathbf{C}_1}$ in Figure 1(c), clearly misrepresenting the true causal directions. In fact, we have both $\mathbf{X} \perp\!\!\!\perp \mathbf{Y}$ and $\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z}$ according to G_1. No DAG structures over clusters $\mathbf{X} - \mathbf{Z} - \mathbf{Y}$ can simultaneously capture both independencies. This implies the need for a new graphical object to represent (in)dependence information between clusters. Suppose we revise our collider test to only assigna collider to a triplet $\langle \mathbf{A}, \mathbf{B}, \mathbf{C} \rangle$ when $\mathbf{A} \perp\!\!\!\perp \mathbf{C}$ and $\mathbf{A} \not\perp\!\!\!\perp \mathbf{C}|\mathbf{B}$. Consider G_2 and its C-DAG $G_{\mathbf{C}_2}$ in Figure 1(e), and 1(d), respectively. In this context, our modified collider test allows one to correctly deduce a collider structure $\mathbf{X} \to \mathbf{Z} \leftarrow \mathbf{Y}$, and this is the only collider learned. Applying the standard orientation rule that for the triplet $\mathbf{X} \to \mathbf{Z} - \mathbf{W}$, $\mathbf{Z} - \mathbf{W}$ should be oriented as $\mathbf{Z} \to \mathbf{W}$ to reflect that since $\langle \mathbf{X}, \mathbf{Z}, \mathbf{W} \rangle$ is not a collider, it must be a non-collider, and again this results in a misdirected edge.

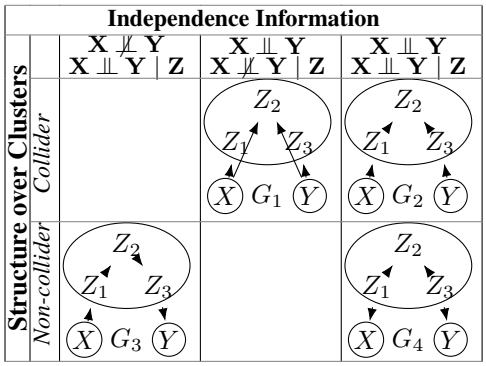
These somewhat surprising results illustrate the complexities of representing causal and independence relationships over clusters and show that naively applying existing algorithms like PC over clusters can lead to incorrect orientations. PC over individual variables learns a Markov equivalence class of causal diagrams with the same conditional (in)dependencies [16, 17, 9, 19], represented as a completed partially directed acyclic graph (CPDAG) [6, 9, 2]. Analogously, for clusters, the goal is to recover a Markov equivalence class reflecting the same (in)dependencies between clusters.

Summary of Contributions Our contributions are as follows:

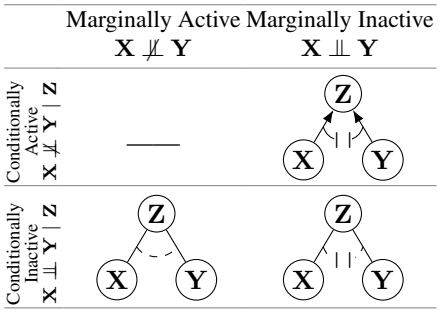
1. In section 2, we define a new graphical object, αC-DAG (Definition 7), that, in addition to causal relations, explicitly represents all (in)dependence information over clusters. We define a new criterion for d-separation in αC-DAGs (Definition 8) which we show is sound and complete for extracting conditional independencies over cluster variables (Theorem 1).
2. In section 3, we define *Cluster Completed Partially Directed Acyclic Graphs*, or αC-CPDAGs, to represent a Markov equivalence class of αC-DAGs (Definition 10). We introduce a learning algorithm for sound and complete causal discovery over clusters to learn an αC-CPDAG by testing conditional independencies over clusters (Algorithm 1).

1.1 Related work and Preliminaries

In the literature, clusters are mainly used as an intermediate step in learning a graphical equivalence class over variables. Typically, clusters of nodes sharing some properties are learned, then structures within or between these clusters are learned, and ultimately integrated into a graph over variables representing a class of DAGs [18, 12, 4, 5, 22]. Prior approaches that learn structures over clusters either group variables heuristically based on structural similarity [10], assume clusters with strict internal structural constraints [3, 14], including where structures such as those in Figure 1 are



(a) Example DAGs representing non-colliders and colliders with possible independence information for clusters.



(b) Independence-arcs for marginal/ conditional independence/dependence combinations.

Figure 2

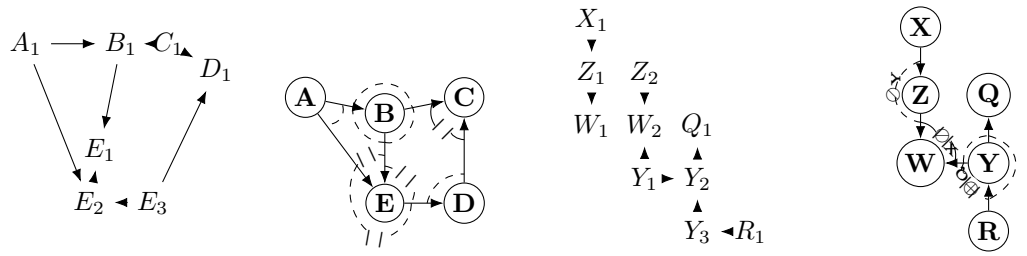
disallowed [11, 21], or consider only two clusters [20]. In contrast, we consider a user-defined partition of variables and learn a structure representing a cluster-level equivalence class.

Notation. A boldfaced uppercase letter $\mathbf{X}$ denotes a set (or a cluster) of variables. We use kinship relations, defined via edges in the graph. We denote by $Pa(\mathbf{X})_G$, $An(\mathbf{X})_G$, and $De(\mathbf{X})_G$, the sets of parents, ancestors, and descendants in graph G, respectively. A vertex V is said to be *active* on a path relative to $\mathbf{Z}$ if 1) V is a collider and V or any of its descendants are in $\mathbf{Z}$ or 2) V is a non-collider and is not in $\mathbf{Z}$. A path p is said to be *active* given (or conditioned on) $\mathbf{Z}$ if every vertex on p is active relative to $\mathbf{Z}$. Otherwise, p is said to be *inactive*. Given a graph G, $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ if every path between $\mathbf{X}$ and $\mathbf{Y}$ is inactive given $\mathbf{Z}$. We denote this d-separation by $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})_G$.
Learned Equivalence Classes. A completed partially directed acyclic graph (CPDAG) $\mathcal{G}$ can have either directed ($\rightarrow$) or undirected ($-$) edges. Directed edges are common for all members of the Markov equivalence class represented by the CPDAG whereas undirected edges are variant. A triplet of vertices $\langle X, Y, Z\rangle$ is unshielded if X and Z are not adjacent to each other. If X and Z are adjacent to one another, the triplet is said to be shielded. In a consecutive triplet $\langle X, Z, Y\rangle$, Z is a definite collider if edges from X and Y are into it ($X \rightarrow Z \leftarrow Y$). Z is a definite non-collider if at least one edge is out of it ($X \leftarrow Y - Z$, $X - Y \rightarrow Z$) or both edges are undirected and the triplet is unshieleded ($X - Y - Z$). Otherwise, Y has a non-definite status. **Definition. Cluster DAG or C-DAG (Markov)**[1] Given an DAG $G(\mathbf{V}, \mathbf{E})$ and a partition $\mathbf{C} = \{\mathbf{C}_1, \ldots, \mathbf{C}_k\}$ of $\mathbf{V}$, construct a graph $G_{\mathbf{C}}(\mathbf{C}, \mathbf{E}_{\mathbf{C}})$ over $\mathbf{C}$ with a set of edges $\mathbf{E}_{\mathbf{C}}$ defined as follows: An edge $\mathbf{C}_i \rightarrow \mathbf{C}_j$ is in $\mathbf{E}_{\mathbf{C}}$ if exists some $V_i \in \mathbf{C}_i$ and $V_j \in \mathbf{C}_j$ such that $V_i \in Pa(V_j)$ in G. If $G_{\mathbf{C}}(\mathbf{C}, \mathbf{E}_{\mathbf{C}})$ contains no cycles, then we say that $\mathbf{C}$ is an *admissible partition* of $\mathbf{V}$. We then call $G_{\mathbf{C}}$ a *cluster DAG*, or *C-DAG*, compatible with G.

2 αC-DAGs: a new graphical object for encoding causal relationships and independences over clusters

2.1 Representing Independence Information Over Clusters

In DAGs, marginal and conditional independencies align consistently with structural edges and arrowhead orientations between variables. As d-separation rules familiarly show, for an unshielded triplet X, Z, Y, a collider structure exists if and only if $X \perp\!\!\!\perp Y, X \not\perp\!\!\!\perp Y|Z$ and a non-collider structure exists if and only if $X \not\perp\!\!\!\perp Y, X \perp\!\!\!\perp Y|Z$. It is only possible for $X \not\perp\!\!\!\perp Y, X \not\perp\!\!\!\perp Y|Z$ if the triplet is shielded. The last combination of independence information, $X \perp\!\!\!\perp Y, X \perp\!\!\!\perp Y|Z$ such that X and Y are adjacent as well as Z and Y, never occurs. With C-DAGs, ambiguity is introduced and the correspondence between graphical structure and independence information changes. Consider G_1 and G_2 in Figure 2(a) which are both colliders over the clusters $\langle \mathbf{X}, \mathbf{Z}, \mathbf{Y}\rangle$, but are each associated with distinct independence information. G_3 and G_4 illustrate analogous behavior for non-colliders, whether a chain or fork. Therefore, neither collider nor non-collider structures over clusters can be singularly associated with specific independencies or dependencies, unlike with variables. Fortunately, the converse is true: certain independence tests can singularly inform structure, and we can leverage this property for learning over clusters in some cases. However, a new representation is needed to ensure complete representation of independence information for structural inference.



(a) G_1 (b) $G_{\mathbf{C}_1}$ (c) G_2 (d) $G_{\mathbf{C}_2}$

Figure 3: G_1 is an ADMG in the class of C-DAGs (with Independence Arcs) $G_{\mathbf{C}_1}$. Independence arcs encode (in)dependencies between clusters, for example that $\mathbf{A} \perp\!\!\!\perp \mathbf{D}$ and $\mathbf{A} \not\perp\!\!\!\perp \mathbf{D}|\mathbf{C}$. $G_{\mathbf{C}_2}$ is a C-DAG (with Independence Arcs and Separation Marks, or αC-DAG) and G_2 is a compatible DAG.

2.2 A novel representation of independence information

We introduce a new semantic representation called "independence arcs" to graphically encode known independence information. These arcs explicitly conveys independence information between variables, decoupled from ancestral relationships. We note that while the terms of "edges" and "arcs" are often used interchangeably to refer to the connections between nodes in a graph, we use the term "independence arc" to refer to a novel symbolic representation of an arc *drawn between two edges* of a cluster graph. The form and representation of the arc conveys information about the conditional and marginal (in)dependencies of the triplet of which these two edges are a part. This is in contrast to what we consistently refer to as *edges*, meaning the connections between nodes in a graph.

Figure 2(b) shows the three new independence arc markings and their meanings, defined formally in Definition 2. A break in the independence arc indicates a marginally inactive triplet, while an arc without any break represents a marginally active triplet. A dashed arc indicates a conditionally inactive triplet, while a solid line indicates a conditionally active triplet. Under this new representation, edges preserve their semantics with regards to conveying parent-child relationships between nodes, and independence information of a triplet is determined exclusively through the independence arc.

Independence arcs annotate both unshielded triplets, $\langle \mathbf{C}_i, \mathbf{C}_k, \mathbf{C}_j \rangle$, where $\mathbf{C}_k$ is adjacent to both $\mathbf{C}_i$ and $\mathbf{C}_j$, and $\mathbf{C}_i$ and $\mathbf{C}_j$ are not adjacent, and shielded triplets, $\langle \mathbf{C}'_i, \mathbf{C}'_k, \mathbf{C}'_j \rangle$, where $\mathbf{C}'_k$ is adjacent to both $\mathbf{C}'_i$ and $\mathbf{C}'_j$, and $\mathbf{C}'_i$ and $\mathbf{C}'_j$ are adjacent. To determine the arc for a shielded triplet, we introduce the concept of a *manipulated shielded triplet* where one edge of the triplet is removed so that the triplet can become unshielded, and the arc describes the behavior of this induced unshielded triplet.

Definition 1 (Manipulation of a shielded triplet)**.** *Given a shielded triplet over clusters $\langle \mathbf{C}_i, \mathbf{C}_m, \mathbf{C}_j \rangle$, its manipulation involves removing the edge between $\mathbf{C}_i$ and $\mathbf{C}_j$, corresponding to removal of any edges between variables in these clusters. After manipulation, the shielded triplet becomes unshielded and this manipulated unshielded triplet is referenced as $\langle \mathbf{C}_i, \mathbf{C}_m, \mathbf{C}_j \rangle^{-\mathbf{C}_i\mathbf{C}_j}$.*

Example 1: Consider Figure 3. Triplet $\langle \mathbf{A}, \mathbf{B}, \mathbf{E} \rangle$ in $G_{\mathbf{C}_1}$ is shielded. To manipulate the triplet, the edge $\mathbf{A} \to \mathbf{E}$ is removed, corresponding to removing the edge $A_1 \to E_2$ in G_1. This manipulated unshielded triplet in $G_{\mathbf{C}_1}$ is referred to as $\langle \mathbf{A}, \mathbf{B}, \mathbf{E} \rangle^{-\mathbf{AE}}$. The complete process for adding independence arcs to a graph is described below in Definition 2.

Definition 2 (**Independence Arcs**)**.** *Consider a graph $\mathbf{G}_C$ over clusters $\mathbf{C} = \langle \mathbf{C}_0, ..., \mathbf{C}_n \rangle$. For any unshielded triplet $\langle \mathbf{C}_i, \mathbf{C}_k, \mathbf{C}_j \rangle$ (or manipulated unshielded triplet $\langle \mathbf{C}_i, \mathbf{C}_k, \mathbf{C}_j \rangle^{-\mathbf{C}_i\mathbf{C}_j}$), let $\mathbf{S}$ equal a (possibly empty) set of clusters $\mathbf{S} \subset (\mathbf{C} \setminus \langle \mathbf{C}_i, \mathbf{C}_j, \mathbf{C}_k \rangle)$ such that $\mathbf{C}_i \perp\!\!\!\perp \mathbf{C}_j|\mathbf{S}$, if such a set exists. For some triplet $\langle \mathbf{C}_i, \mathbf{C}_k, \mathbf{C}_j \rangle$, an independence arc, $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_k,\mathbf{C}_j} \in \boldsymbol{\mathcal{A}}$, can be drawn from some point on the edge between $\mathbf{C}_i$ and $\mathbf{C}_k$ to some point on the edge between $\mathbf{C}_j$ and $\mathbf{C}_k$ in the following way:*

1. *A marginally-connecting independence arc of - - - - is drawn if and only if $\mathbf{C}_i \not\perp\!\!\!\perp \mathbf{C}_j|\mathbf{S} \setminus \mathbf{C}_k$ and $\mathbf{C}_i \perp\!\!\!\perp \mathbf{C}_j|\mathbf{S}$, where $\mathbf{C}_k \in \mathbf{S}$.*
2. *A conditionally-connecting independence arc of —||— is drawn if and only if $\mathbf{C}_i \perp\!\!\!\perp \mathbf{C}_j|\mathbf{S} \setminus \mathbf{C}_k$ and $\mathbf{C}_i \not\perp\!\!\!\perp \mathbf{C}_j|\mathbf{S}$, where $\mathbf{C}_k \in \mathbf{S}$*
3. *A never-connecting independence arc of $--||--$ is drawn if and only if $\mathbf{C}_i \perp\!\!\!\perp \mathbf{C}_j|\mathbf{S} \setminus \mathbf{C}_k$ and $\mathbf{C}_i \perp\!\!\!\perp \mathbf{C}_j|\mathbf{S}$, where $\mathbf{C}_k \in \mathbf{S}$*

Shielded triplets are annotated according to the behavior of their respective manipulated triplets.

Example 2: Consider DAG G_1 in Figure 3. Unshielded triplets $\langle \mathbf{A}, \mathbf{B}, \mathbf{C} \rangle$, $\langle \mathbf{E}, \mathbf{B}, \mathbf{C} \rangle$, and $\langle \mathbf{C}, \mathbf{D}, \mathbf{E} \rangle$, are marked with a marginally-connecting arc, as are manipulated unshielded triplets $\langle \mathbf{E}, \mathbf{A}, \mathbf{B} \rangle^{-\mathbf{EB}}$ and $\langle \mathbf{A}, \mathbf{B}, \mathbf{E} \rangle^{-\mathbf{AE}}$. A conditionally-connecting arc is drawn for $\langle \mathbf{B}, \mathbf{C}, \mathbf{D} \rangle$ Never-connecting arcs are added to triplets $\langle \mathbf{A}, \mathbf{E}, \mathbf{D} \rangle$ and $\langle \mathbf{B}, \mathbf{E}, \mathbf{D} \rangle$, and manipulated unshielded triplet $\langle \mathbf{A}, \mathbf{E}, \mathbf{B} \rangle^{-\mathbf{AB}}$.

Remark 1. *In a Markov C-DAG with independence arcs, a conditionally-connecting independence arc always implies a collider structure.*

While a collider structure $\mathbf{X} \to \mathbf{Z} \leftarrow \mathbf{Y}$ in a C-DAG does not necessarily imply that $\mathbf{X} \perp\!\!\!\perp \mathbf{Y}$; $\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y} | \mathbf{Z}$, Remark 1 notes that the converse is true. Independence arcs allow for d-separations to be read in a new way, unrelated to edge connections. For an isolated triplet with clusters $\langle \mathbf{C}_i, \mathbf{C}_k, \mathbf{C}_j \rangle$, the triplet is active (d-connecting) relative to the (possibly empty) set of cluster vertices $\mathbf{Z}$ if a) $\langle \mathbf{C}_i, \mathbf{C}_k, \mathbf{C}_j \rangle$ is marked with a marginally-connecting independence arc and $\mathbf{C}_k \notin \mathbf{Z}$ or b) $\langle \mathbf{C}_i, \mathbf{C}_k, \mathbf{C}_j \rangle$ is marked with a conditionally-connecting independence arc and $\mathbf{C}_k \in \mathbf{Z}$. Otherwise, $\langle \mathbf{C}_i, \mathbf{C}_k, \mathbf{C}_j \rangle$ is d-separated relative to $\mathbf{Z}$. In a larger graph, we introduce the notion of *arc trajectories*, or the sequence of independence arcs corresponding to a path between two variables. Arc trajectories can be analyzed to determine if two variables are connected or not.

Definition 3 (Arc Trajectory). *Given a graph $\mathbf{G}_C$, for some path over clusters $\langle \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, ..., \mathbf{C}_n \rangle$, the* arc trajectory *refers to the sequence of independence arcs for each triplet along the path, $\mathbf{a} = \langle \mathcal{A}_{\mathbf{C}_1,\mathbf{C}_2,\mathbf{C}_3}, ..., \mathcal{A}_{\mathbf{C}_{n-2},\mathbf{C}_{n-1},\mathbf{C}_n} \rangle$.*

Example 3: Consider the example in Figure 3. To determine if $\mathbf{A}$ and $\mathbf{D}$ are d-separated ($\mathbf{A} \perp\!\!\!\perp \mathbf{D}$) in $G_{\mathbf{C}_1}$, we first identify all simple paths between $\mathbf{A}$ and $\mathbf{D}$, of which there are three: $\mathbf{A} \to \mathbf{B} \to \mathbf{C} \leftarrow \mathbf{D}$, $\mathbf{A} \to \mathbf{B} \to \mathbf{E} \to \mathbf{D}$, and $\mathbf{A} \to \mathbf{E} \to \mathbf{D}$. The arc trajectory corresponding to the first path is $\langle \mathcal{A}_{\mathbf{A},\mathbf{B},\mathbf{C}}, \mathcal{A}_{\mathbf{B},\mathbf{C},\mathbf{D}} \rangle$, consisting of a marginally-connecting arc and a conditionally-connecting arc. Because there is no conditioning set in the query, only $\mathcal{A}_{\mathbf{A},\mathbf{B},\mathbf{C}}$ indicates an active triplet but not $\mathcal{A}_{\mathbf{B},\mathbf{C},\mathbf{D}}$, and therefore $\mathbf{A}$ and $\mathbf{D}$ are not connected along this path. For the second path, the arc trajectory is $\langle \mathcal{A}_{\mathbf{A},\mathbf{B},\mathbf{E}}, \mathcal{A}_{\mathbf{B},\mathbf{E},\mathbf{D}} \rangle$. $A_{\mathbf{A},\mathbf{B},\mathbf{E}}$ is an always-connecting arc, but $\mathcal{A}_{\mathbf{B},\mathbf{E},\mathbf{D}}$ is a never-connecting arc, so $\mathbf{A}$ and $\mathbf{D}$ are not connected by this path either. The last path has the arc trajectory $\langle \mathcal{A}_{\mathbf{A},\mathbf{E},\mathbf{D}} \rangle$, and its only independence arc is never-connecting. Therefore, we can conclude that $\mathbf{A} \perp\!\!\!\perp \mathbf{D}$. By a similar analysis, we can conclude that $\mathbf{A} \not\perp\!\!\!\perp \mathbf{D} | \mathbf{C}$.

With some simple examples, we illustrate that determining d-separations by independence arcs can sometimes be more complex. Consider Figure 3. From G_2, the following independence information is clear: $\mathbf{X} \not\perp\!\!\!\perp \mathbf{W}$ and $\mathcal{A}_{\mathbf{X},\mathbf{Z},\mathbf{W}}$ is a marginally-connecting arc, $\mathbf{Z} \not\perp\!\!\!\perp \mathbf{Y} | \mathbf{W}$, and $\mathcal{A}_{\mathbf{Z},\mathbf{W},\mathbf{Y}}$ is a conditionally-connecting arc. Then the arc trajectory in $G_{\mathbf{C}_2}$ from $\mathbf{X}$ to $\mathbf{Y}$ might lead us to believe that $\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y} | \mathbf{W}$, but this is not true. Independence arcs indicate information with regards to a triplet of clusters, but alone, may misrepresent d-separation for paths over clusters. We enrich independence arcs with a new semantic representation to denote unexpected independencies. We introduce a new symbol, $\oslash_{\mathbf{C}}$, which we call a "separation mark." This mark annotates an independence arc of a triplet to indicate a cluster (specified by the subscript of the separation mark) further along on a path that, by independence arcs, would appear to have a d-connection to the variables in the triplet, but is actually separated. This notion is formalized in definition 5. First, we define a supporting concept below.

Definition 4 (Analogous Paths). *Given a C-DAG $G_{\mathbf{C}}$ and a compatible ADMG G, we define a simple path in G over variables, $p = \langle V_1, V_2, V_3, ..., V_m \rangle$ to be considered analogous to a path in $G_{\mathbf{C}}$ over clusters $p_{\mathbf{C}} = \langle \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, ..., \mathbf{C}_n \rangle$ (and $p_{\mathbf{C}}$ analogous to p) if and only if the following hold: 1) for every variable V_i on p, V_i is in some cluster $\mathbf{C}_i$ on $p_{\mathbf{C}}$, 2) for every cluster C_j on $p_{\mathbf{C}}$, there exists some variable $V_j \in \mathbf{C}_j$ where V_j is on p, and 3) for any variable $V_n \in \mathbf{C}_n$, there does not exist any variable that appears after V_n on p that is in a cluster before $\mathbf{C}_n$ on $p_{\mathbf{C}}$.*

In Fig. 3, the path over variables $p_v = \langle A_1, B_1, C_1, D_1 \rangle$ in G_1 is an analogous path for the path over clusters $p_c = \langle \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D} \rangle$ in $G_{\mathbf{C}_1}$, but the path over variables $p'_v = \langle A_1, B_1, E_1, E_2, E_3, D_1 \rangle$ is not analogous to p_c, since $\mathbf{E}$ is not on p_c but $\exists V_e \in \mathbf{E}$ on p'_v and $\nexists V_c \in \mathbf{C}$ on p'_v, but $\mathbf{C}$ is on p_c.

Definition 5 (Separation Marks). *Let G be an ADMG, and let $G_{\mathbf{C}}$ denote a possible C-DAG for G. Consider a path $p_{\mathbf{C}}$ in $G_{\mathbf{C}}$ over clusters $\langle \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, ..., \mathbf{C}_n \rangle$ and its corresponding arc trajectory $\mathbf{a} = \langle \mathcal{A}_{\mathbf{C}_1,\mathbf{C}_2,\mathbf{C}_3}, ... \mathcal{A}_{\mathbf{C}_{n-2},\mathbf{C}_{n-1},\mathbf{C}_n} \rangle$ such that:*

1. *there is no arc $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_{i+1},\mathbf{C}_{i+2}} \in \mathbf{a}$ that is a never-connecting arc,*
2. *there is no d-connecting path p in G over variables relative to clusters $\mathbf{Z}$, analogous to $p_{\mathbf{C}}$,*
3. *there exists a d-connecting path p' in G over variables relative to some set of clusters $\mathbf{Z}'$ that is analogous to the path in $G_{\mathbf{C}}$, $p'_{\mathbf{C}} = \langle \mathbf{C}_1, ..., \mathbf{C}_{n-1} \rangle$, and*

4. *there exists a d-connecting path p'' in G over variables relative to some set $\mathbf{Z}''$ of clusters that is analogous to the path in $G_{\mathbf{C}}$, $p''_{\mathbf{C}} = \langle \mathbf{C}_2, ..., \mathbf{C}_n \rangle$.*

Then, a separation mark, $\oslash_{\mathbf{C}_1}$ is placed on the arc $\mathcal{A}_{\mathbf{C}_{n-2},\mathbf{C}_{n-1},\mathbf{C}_n}$, and a separation mark, $\oslash_{\mathbf{C}_n}$ is placed on the arc $\mathcal{A}_{\mathbf{C}_1,\mathbf{C}_2,\mathbf{C}_3}$.

Example 4: In Figure 3, we identify where a separation mark is needed by traversing paths of length greater than 3 in $G_{\mathbf{C}_2}$ and compare to the paths over variables in G_2. For example, traversing the path $\langle \mathbf{X}, \mathbf{Z}, \mathbf{W}, \mathbf{Y} \rangle$ in $G_{\mathbf{C}_2}$ and comparing to G_2, we see that there is no path between any variable in $\mathbf{X}$ and a variable in $\mathbf{Y}$. We place a separation mark with the subscript $\mathbf{Y}$, as in $\oslash_{\mathbf{Y}}$, on the independence arc of $\mathcal{A}_{\mathbf{X},\mathbf{Z},\mathbf{W}}$. This indicates that when traversing a path starting at $\mathbf{X}$ where $\mathcal{A}_{\mathbf{X},\mathbf{Z},\mathbf{W}}$ is in the arc trajectory associated with the path, $\mathbf{Y}$ is separated from $\mathbf{X}$ (in addition to any nodes past $\mathbf{Y}$ on the path). We place a mirroring separation mark, $\oslash_{\mathbf{X}}$, along arc trajectory $\mathcal{A}_{\mathbf{Z},\mathbf{W},\mathbf{Y}}$ to reflect the reverse. $G_{\mathbf{C}_2}$ in Figure 3 shows the C-DAG with independence arcs and separation marks. Further discussion on separation marks can be found in the appendix.

Separation marks indicate separations on paths masked by the clusters and independence arcs. Connections may also be masked if conditioning on a descendant of a collider within a cluster, where the descendant is in a different cluster from the collider. We introduce a new *connection mark*, which, like separation marks, annotates independence arcs. Specifically, a connection mark, $\oplus_{\mathbf{C}_x}$ in an independence arc $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_j,\mathbf{C}_k}$ denotes that the triplet $\langle \mathbf{C}_i, \mathbf{C}_j, \mathbf{C}_k \rangle$ is activated by conditioning on $\mathbf{C}_x$ due to some variable $V_x \in \mathbf{C}_x$ being a descendant of some collider variable $V_j \in \mathbf{C}_j$. Definition 6 formalizes this.

Definition 6 (Connection Marks). *Let G be an ADMG and let $G_{\mathbf{C}}$ denote a possible C-DAG for G with independence arcs. Consider a triplet over clusters in $G_{\mathbf{C}}$, $\langle \mathbf{C}_i, \mathbf{C}_j, \mathbf{C}_k \rangle$, and its corresponding independence arc, $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_j,\mathbf{C}_k}$. If $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_j,\mathbf{C}_k}$ is a never-connecting or conditionally-connecting independence arc, and there exists a path p in G over variables through the triplet $\langle V_i, ..., V_j, ...V_k \rangle$ such that $V_i \in \mathbf{C}_i$, $V_j \in \mathbf{C}_j$, and $V_k \in \mathbf{C}_k$ then $\forall V'_j \in \mathbf{C}_j$ and on p, where V'_j is a collider, let $\mathbf{D}$ be the set of clusters that are children of $\mathbf{C}_j$ and which include descendants of all colliders along the path, ($\mathbf{D} = \bigcup \{\mathbf{C}_d : V_d \in \mathbf{C}_d\}$ where $V_d \notin \{\mathbf{C}_i, \mathbf{C}_j, \mathbf{C}_k\}$ and $V_d \in Ch(V_j)$). Then the connection mark $\oplus_{\mathbf{D}}$ is added to $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_j,\mathbf{C}_k}$.*

Example 5: Consider again Figure 3. Collider Y_2 in the triplet $\langle Y_1, Y_2, Y_3 \rangle$ in G_2 is not discernible in triplet $\mathbf{W}, \mathbf{Y}, \mathbf{R}$ in $G_{\mathbf{C}_2}$, which is marked by a never-connecting independence arc. However, conditioning on $\mathbf{Q}$ renders $\mathbf{R}$ and $\mathbf{W}$ dependent. The connection mark $\oplus_{\mathbf{Q}}$ is placed along arc $\mathcal{A}_{\mathbf{W},\mathbf{Y},\mathbf{R}}$, as shown. Further discussion on connection marks can be found in the appendix.

2.3 αC-DAG Definition and Properties

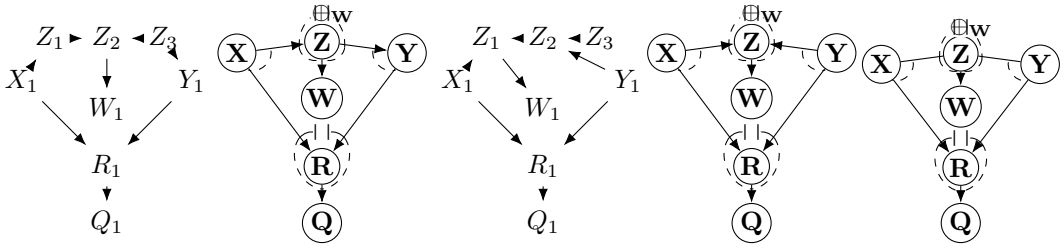
With the introduction of the new symbolic representations of independence arcs, separation marks, and connection marks we can fully define a new graphical model for C-DAGs with independence arcs, which we call αC-DAGs, for short. The "α" prefix will be used to indicate graphical representations making use of the new semantics of independence arcs, separation marks and connection marks.

Definition 7 (αC-DAG (C-DAG with Independence Arcs)). *Given a DAG $G(\mathbf{V}, \mathbf{E})$ and a partition $\mathbf{C} = \{\mathbf{C}_1, \ldots, \mathbf{C}_n\}$ of $\mathbf{V}$, construct a graph $G_{\mathbf{C}}(\mathbf{C}, \mathbf{E}_{\mathbf{C}}, \boldsymbol{\mathcal{A}})$ over $\mathbf{C}$.*

- *An edge $\mathbf{C}_i \to \mathbf{C}_j$ is in $\mathbf{E}_{\mathbf{C}}$ if exists some $V_i \in \mathbf{C}_i$ and $V_j \in \mathbf{C}_j$ such that $V_i \in Pa(V_j)$ in G;*
- *The set of independence arcs $\boldsymbol{\mathcal{A}}$ is defined over all triplets $\langle \mathbf{C}_i, \mathbf{C}_k, \mathbf{C}_j \rangle$, by to Definition 2;*
- *For each arc trajectory in $G_{\mathbf{C}}$, separation marks are added according to Definition 5;*
- *For each path in $G_{\mathbf{C}}$, connection marks are added according to Definition 6.*

If for all pairs of clusters $\mathbf{C}_i, \mathbf{C}_j$ where there exists an edge $\mathbf{C}_i \to \mathbf{C}_j$, there is no directed path $\mathbf{C}_j \to ... \to \mathbf{C}_i$, then we say that $\mathbf{C}$ is an admissible partition *of $\mathbf{V}$. We then call $G_{\mathbf{C}}$ a* cluster DAG with independence arcs*, or an α*C-DAG*, compatible with G.*

As with the definition of C-DAGs, αC-DAGs include an assumption about acyclicity over the clusters. Specifically, we disallow what we define as **apparent directed cycles** (or just apparent cycles), where edges over clusters give the appearance of a cycle such that for some pair of clusters $\mathbf{C}_i, \mathbf{C}_j$ there exists an edge $\mathbf{C}_i \to \mathbf{C}_j$ and a directed path $\mathbf{C}_j \to ... \to \mathbf{C}_i$. While Definition 7 takes as input a DAG, we also note that construction of an αC-DAG could alternatively take as input a C-DAG and a probability distribution $P(\mathbf{C})$ where $P(\mathbf{C})$ is faithful to the true data-generating process. This



(a) G_1 (b) $G_{\mathbf{C}_1}$ (c) G_2 (d) $G_{\mathbf{C}_2}$ (e) $\mathcal{P}_{\mathbf{C}}$

Figure 4: G_1 and G_2 are DAGs in the class of the αC-DAGs $G_{\mathbf{C}_1}$ and $G_{\mathbf{C}_2}$, respectively. $G_{\mathbf{C}_1}$ and $G_{\mathbf{C}_2}$ are in the Markov equivalence class of αC-CPDAG, $\mathcal{P}_{\mathbf{C}}$. In $\mathcal{P}_{\mathbf{C}}$, $\mathscr{R}_0$ is applied to $\langle \mathbf{X}, \mathbf{R}, \mathbf{Y} \rangle$, and then $\mathscr{R}_1$ is applied to $\langle \mathbf{X}, \mathbf{R}, \mathbf{Q} \rangle$. Lastly, $\mathscr{R}_5$ is applied to $\langle \mathbf{X}, \mathbf{Z}, \mathbf{Y} \rangle$ with descendant $\mathbf{W}$.

approach would simply rely on $P(\mathbf{C})$ to inform independence arcs, separation marks and connections marks such that the αC-DAG is still an object constructed from knowledge, rather than one that is learned.

Remark 2. *An αC-DAG can be constructed by modification to Definition 7 where a C-DAG $G_{\mathbf{C}}$ and faithful probability distribution $P(\mathbf{C})$ are taken as input. $G_{\mathbf{C}}$ informs the relationships over clusters and $P(\mathbf{C})$ is leveraged to set independence arcs, separation marks and connection marks.*

D-separation over αC-DAGs can be determined according to the criteria below. In the theorem that follows, we show these d-separation rules are sound and complete in αC-DAGs.

Definition 8 (d-separation over αC-DAGs.). *A path $p_{\mathbf{C}}$ in an α-C-DAG, $G_{\mathbf{C}}$, is said to be d-separated (or blocked) by a set of clusters $\mathbf{Z} \subset \mathbf{C}$ if and only if its corresponding arc trajectory $\mathbf{a}$ contains an independence arc $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_j,\mathbf{C}_k}$ that is:*

1. *a marginally-connecting independence arc (a) $\mathbf{C}_j$ is in $\mathbf{Z}$ or (b) there exists a separation mark $\oslash_{\mathbf{C}_x}$ on $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_j,\mathbf{C}_k}$ where $\mathbf{C}_x$ is on $p_{\mathbf{C}}$,*
2. *a conditionally-connecting independence arc and (a) $\mathbf{C}_j$ is not in $\mathbf{Z}$ nor is any true descendant of $\mathbf{C}_j$ in $\mathbf{Z}$, (b) there exists a separation mark on $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_j,\mathbf{C}_k}$ $\oslash_{\mathbf{C}_x}$ where $\mathbf{C}_x$ is on $p_{\mathbf{C}}$, or, (c) for any connection mark $\oplus_{\mathbf{C}_x}$ on $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_j,\mathbf{C}_k}$, $\mathbf{C}_x$ is not in $\mathbf{Z}$*
3. *a never-connecting independence arc and for connection mark $\oplus_{\mathbf{C}_x}$ on $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_j,\mathbf{C}_k}$, $\mathbf{C}_x \notin \mathbf{Z}$*

Theorem 1. ***[Soundness and completeness of d-separation in αC-DAGs.]*** *In an αC-DAG $G_{\mathbf{C}}$, let $\mathbf{X}, \mathbf{Z}, \mathbf{Y} \subset \mathbf{C}$. $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in $G_{\mathbf{C}}$, if and only if for any DAG, G compatible with $G_{\mathbf{C}}$, $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in G. $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})_{G_{\mathbf{C}}} \iff (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})_{G}$.*

With this d-separation definition, we have a tool for reading independence information over clusters in an αC-DAG. This new graph can be used to represent knowledge not only of connections between clusters but also of independence information over clusters. Semantics of αC-DAGs are discussed further in the appendix. In the next section, we build on the semantics introduced in the context of α-CDAGs to define new graphical objects to serve as the foundation for learning over clusters.

3 αC-CPDAGs and learning

3.1 Equivalence classes of αC-DAGs

Now we define of a new graphical object that represents the equivalence class of cluster graphs sharing the same independence structure induced by that distribution. This graphical object will be analogous to a completed partially directed acyclic graph (CPDAG) which uniquely represents a Markov equivalence class of variables, and will represent an equivalence class of αC-DAGs. We introduce this new graph, a cluster CPDAG, or αC-CPDAG, define how an αC-DAG can be mapped to an αC-CPDAG, and describe how this new object can be learned from an observational distribution.

Two DAGs, G_1 and G_2 with the same vertices are Markov equivalent if for any three disjoint sets of vertices $\mathbf{X}, \mathbf{Z}, \mathbf{Y}$, $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in G_1 if and only if $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in G_2. We extend a similar notion for clusters and αC-DAGs in Definition 9. From the definition of

d-separation for αC-DAGs, we know that such separations are discernible from independence arcs, separation marks and connection marks alone, which leads to the theorem following the definition.

Definition 9 (**Cluster Markov Equivalence**). *Two αC-DAGs, $G_{\mathbf{C}_1}$ and $G_{\mathbf{C}_2}$ (with the same partition $\mathbf{C}$ over the same variables $\mathbf{V}$) are cluster Markov equivalent if for any three disjoint sets of clusters $\mathbf{X}, \mathbf{Z}, \mathbf{Y}$, $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in $G_{\mathbf{C}_1}$ iff $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in $G_{\mathbf{C}_2}$.*

Theorem 2. *Two αC-DAGs, $G_{\mathbf{C}_1}$ and $G_{\mathbf{C}_2}$ (with the same partition $\mathbf{C}$ over the same set of variables $\mathbf{V}$) are cluster Markov equivalent if and only if they share the same: 1) adjacencies, 2) independence arcs, 3) separation marks and 4) connection marks.*

Figure 4 illustrates example DAGs and αC-DAGs in the same cluster Markov equivalence class. Markov equivalent αC-DAGs share some unshielded colliders, namely those marked by a conditionally-connecting arc. This characterization of equivalent αC-DAGs leads to the definition of the cluster CPDAGs (αC-CPDAGs). As with a partially directed acyclic graph, an αC-CPDAG may contain both directed and undirected edges and does not contain any directed cycles. As with αC-DAGs, an αC-CPDAG is defined over a user-defined partition of clusters $\mathbf{C}$ over the variables $\mathbf{V}$.

Definition 10 (α**Cluster CPDAG**). *Let $[G_{\mathbf{C}}]$ be the Markov equivalence class of an arbitrary αC-DAG, $G_{\mathbf{C}}$. The αC-CPDAG for $[G_{\mathbf{C}}]$, denoted $\mathcal{P}$, is a cluster completed partially directed acyclic graph (αC-CPDAG) such that:*

1. *$\mathcal{P}$ has the same adjacencies as $G_{\mathbf{C}}$ (and therefore any member of $[G_{\mathbf{C}}]$) does.*
2. *A directed edge is in $\mathcal{P}$ iff shared by all DAGs in $[G_{\mathbf{C}}]$ and otherwise the edge is undirected*
3. *$\mathcal{P}$ has the same independence arcs, separation marks, and connection marks as $G_{\mathbf{C}}$ (and therefore any member of $[G_{\mathbf{C}}]$) does.*

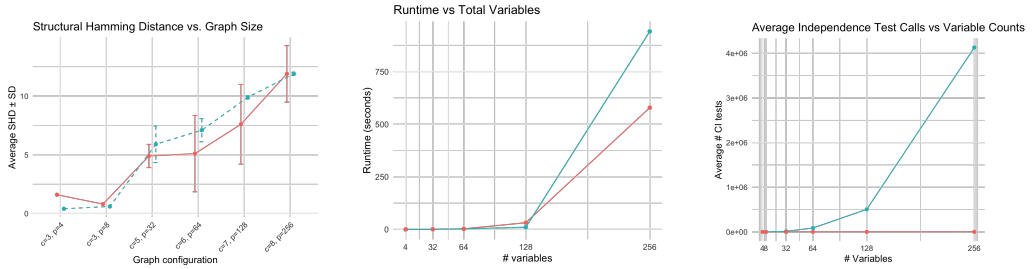
3.2 A Constraint-Based Learning Algorithm for αC-CPDAG

Algorithm 1: CLOC: Algorithm for Learning an αC-CPDAG

Input: Admissible partition $\mathbf{C} = \{\mathbf{C}_1, ..., \mathbf{C}_n\}$, $P(\mathbf{C})$
Output: αC-CPDAG, $\mathcal{P}$

Form complete graph $\mathcal{P}$ over $\mathbf{C}$ with undirected edges.
for $\mathbf{X}, \mathbf{Y} \in \mathbf{C}$ **do**
  for $\mathbf{S} \subseteq \mathbf{C} \setminus \{\mathbf{X}, \mathbf{Y}\}$ **do**
    if $P(\mathbf{y}|\mathbf{s}, \mathbf{x}) = P(\mathbf{y}|\mathbf{s})$ **then**
      SepSet $\leftarrow \mathbf{S}$, SepFlag $\leftarrow True$, **break**
  if *SepFlag* $= True$ **then**
    Remove the edge between $\mathbf{X}, \mathbf{Y}$ in $\mathcal{P}$
for *every unshielded triplet* $\langle \mathbf{X}, \mathbf{Z}, \mathbf{Y} \rangle$ *in* $\mathcal{P}$ **do**
  if $\mathbf{Z} \notin SepSet(\mathbf{X}, \mathbf{Y})$ *and* $\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y} | \mathbf{Z} \cup SepSet(\mathbf{X}, \mathbf{Y})$ **then**
    Mark $\langle \mathbf{X}, \mathbf{Z}, \mathbf{Y} \rangle$ in $\mathcal{P}$ with a conditionally-connecting arc, and orient as $\mathbf{X} \rightarrow \mathbf{Z} \leftarrow \mathbf{Y}$
  else if $\mathbf{Z} \in SepSet(\mathbf{X}, \mathbf{Y})$ **then**
    Mark $\langle \mathbf{X}, \mathbf{Z}, \mathbf{Y} \rangle$ in $\mathcal{P}$ with a marginally-connecting arc
  else
    Mark $\langle \mathbf{X}, \mathbf{Z}, \mathbf{Y} \rangle$ in $\mathcal{P}$ with a never-connecting arc
for *each path* $p = \langle \mathbf{C}_0, ..., \mathbf{C}_n \rangle \in \mathcal{P}$ **do**
  if *length*$(p) \geq 4$ *and arc trajectory* $\mathbf{a}$ *for p is only marginal/conditionally-connecting arcs (with no marks $\oslash_{\mathbf{C}_a}$ where $\mathbf{C}_a \in p$)* **then**
    Let $\mathbf{K} \leftarrow \bigcup \{C_z \mid \mathcal{A}_{C_x, C_z, C_y} \in \mathbf{a}$ is conditionally-connecting$\}$
    if $\mathbf{C}_0 \perp\!\!\!\perp \mathbf{C}_n | \mathbf{K} \cup (SepSet(\mathbf{C}_0, \mathbf{C}_n) \setminus p)$ **then**
      For shortest subpath $p' = \langle \mathbf{C}_i, ..., \mathbf{C}_j \rangle \subseteq p$ s.t. length$(p') \geq 4$ and $\mathbf{C}_i \perp\!\!\!\perp \mathbf{C}_j | \mathbf{K} \cup (SepSet(\mathbf{C}_0, \mathbf{C}_n) \setminus p)$
      Add $\oslash_{\mathbf{C}_j}$ to $\mathcal{A}_{\mathbf{C}_i, \mathbf{C}_{i+1}, \mathbf{C}_{i+2}}$
      Add $\oslash_{\mathbf{C}_i}$ to $\mathcal{A}_{\mathbf{C}_{j-2}, \mathbf{C}_{j-1}, \mathbf{C}_j}$
for *each triplet* $\langle \mathbf{X}, \mathbf{Z}, \mathbf{Y} \rangle$ *in* $\mathcal{P}$ **do**
  if $\mathcal{A}_{\mathbf{X}, \mathbf{Z}, \mathbf{Y}}$ *is a conditionally or never connecting arc and* $\exists$ *some* $\mathbf{W}$ *such that* $\mathbf{Z} - \mathbf{W}$ **then**
    Let $\boldsymbol{\mathcal{W}}_{\mathbf{Z}} \leftarrow \{\mathbf{W} \mid \mathbf{Z} - \mathbf{W}$ exists in $\mathcal{P}\}$
    Let $\mathcal{S}$ be the power set of $\boldsymbol{\mathcal{W}}_{\mathbf{Z}} \setminus \emptyset$
    for *each subset* $\boldsymbol{\mathcal{D}} \in \mathcal{S}$ **do**
      if $\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y} | \boldsymbol{\mathcal{D}} \cup SepSet(\mathbf{X}, \mathbf{Y})$ **then**
        Add $\oplus_{\boldsymbol{\mathcal{D}}}$ to $\mathcal{A}_{\mathbf{X}, \mathbf{Z}, \mathbf{Y}}$
Apply the five orientation rules until none apply

Given how to define an αC-DAG from a DAG, and an αC-CPDAG from an αC-DAG, we can understand the reverse process of constructing an αC-CPDAG from independence information in



(a) SHD plot (b) CI test (c) Run time

Figure 5: Plots comparing the (a) structural Hamming distance for each combination of cluster count (c) and variable count (p), (b) algorithm run time, and (c) number of conditional-independence tests calculated for CLOC(red) compared to the PC-then-cluster approach (blue).

an observational dataset. This procedure is shown in Algorithm 1, which we call Causal Learning Over Clusters (CLOC) and is based on the assumption that an available distribution $P(\mathbf{C})$ (or data representing it) is *faithful* to the true underlying αC-DAGs (in addition to the aforementioned assumptions of causal sufficiency and an admissible partition $\mathbf{C}$). Figure 4 illustrates an αC-CPDAG learned by the algorithm.

Definition 11 (**Faithfulness for αC-DAGs**). *Given an αC-DAG $G_{\mathbf{C}}$ and probability distribution over the clusters $P(\mathbf{C})$ that is generated by an SCM consistent with any causal diagram compatible with $G_{\mathbf{C}}$, we say that $P(\mathbf{C})$ is* faithful *to $G_{\mathbf{C}}$ if* $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z})_{P(\mathbf{c})} \Rightarrow (\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z})_{P(G_{\mathbf{C}})}$.

CLOC has three phases to it. In the first, edges between two nodes $\mathbf{X}$ and $\mathbf{Y}$ are removed from a complete graph with undirected edges if there exists some separating set of clusters $\mathbf{S}$ such that $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{S})$. In the second phase, independence arcs, separation marks, and connection marks are added. Unshielded colliders are also determined from conditionally-connecting arcs (Remark 1). ($\mathscr{R}_0$: If $\mathbf{X} - \mathbf{Z} - \mathbf{Y}$, $\mathbf{X}$ and $\mathbf{Y}$ are not adjacent, and $\mathcal{A}_{\mathbf{X},\mathbf{Z},\mathbf{Y}}$ is conditionally-connecting, then orient the triplet as $\mathbf{X} \to \mathbf{Z} \leftarrow \mathbf{Y}$ (i.e. $\mathbf{Z} \notin SepSet(\mathbf{X}, \mathbf{Y})$ and $\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y}|\mathbf{Z} \cup SepSet(\mathbf{X}, \mathbf{Y})$). In the final phase, five orientation rules are applied until none apply. Rules 1, 3 and 4 extend from PC, leveraging independence arcs to determine where the logic is sound. Rule 2 extends precisely, and Rule 5 is our contribution. This algorithm gives us an αC-CPDAG, which represents the Cluster Markov equivalence class of αC-DAGs compatible with the distribution $P(\mathbf{C})$. We review the rules below and proofs are in the appendix. We demonstrate after that the orientation rules as well as the learning algorithm overall are sound for learning causal relations between clusters. Note that in the orientation rules, asterisks indicate either an arrowhead or tail is possible.

$\mathscr{R}_1$: If $\mathbf{X} \to \mathbf{Z} - \mathbf{Y}$, $\mathbf{X}$ and $\mathbf{Y}$ are not adjacent, and $\mathcal{A}_{\mathbf{X},\mathbf{Z},\mathbf{Y}}$ is marginally-connecting, then orient the triplet as $\mathbf{X} \to \mathbf{Z} \to \mathbf{Y}$.

$\mathscr{R}_2$: If $\mathbf{X} \to \mathbf{Z} \to \mathbf{Y}$ and $\mathbf{X} - \mathbf{Y}$, then orient $\mathbf{X} - \mathbf{Y}$ as $\mathbf{X} \to \mathbf{Y}$.

$\mathscr{R}_3$: If $\mathbf{X} \to \mathbf{Z} \leftarrow \mathbf{Y}$, $\mathbf{X} - \mathbf{W} - \mathbf{Y}$, $\mathbf{X}$ and $\mathbf{Y}$ are not adjacent, $\mathbf{W} - \mathbf{Z}$, and $\mathcal{A}_{\mathbf{X},\mathbf{W},\mathbf{Y}}$ is marginally-connecting, then orient $\mathbf{W} - \mathbf{Z}$ as $\mathbf{W} \to \mathbf{Z}$.

$\mathscr{R}_4$: If $\mathbf{X} \to \mathbf{Z} \to \mathbf{Y}$, $\mathbf{X} - \mathbf{W} - \mathbf{Y}$, $\mathbf{X}$ and $\mathbf{Y}$ are not adjacent, $\mathbf{W} *\!\!-\!\!* \mathbf{Z}$, and $\mathcal{A}_{\mathbf{X},\mathbf{W},\mathbf{Y}}$ is marginally-connecting, then orient $\mathbf{W} - \mathbf{Y}$ as $\mathbf{W} \to \mathbf{Y}$.

$\mathscr{R}_5$: If $\mathbf{X} *\!\!-\!\!* \mathbf{Z} *\!\!-\!\!* \mathbf{Y}$, $\mathbf{Z} - \mathbf{W}$, $\mathbf{X}$ and $\mathbf{W}$ are not adjacent, $\mathbf{Y}$ and $\mathbf{W}$ are not adjacent, and $\mathcal{A}_{\mathbf{X},\mathbf{Z},\mathbf{Y}}$ is never-connecting or conditionally-connecting with connection mark $\oplus_{\boldsymbol{\mathcal{D}}}$ such that $\mathbf{W} \in \boldsymbol{\mathcal{D}}$, then orient $\mathbf{Z} - \mathbf{W}$ as $\mathbf{Z} \to \mathbf{W}$.

Theorem 3. ***[Soundness and Completeness of Orientation Rules and CLOC]*** *The five orientation rules and the procedure of CLOC are sound and complete.*

4 Experiments

We show performance of our algorithm in comparison to applying PC over the entire graph of variables and then imposing clusters on the graph. We generate random C-DAGs (c = 3, 5, 6, 7, 8 clusters), and random DAGs (p = 4, 8, 32, 64, 128, 256 variables) compatible with the C-DAGs. A Gaussian distribution (samples of n= 500, 1000, 3000) faithful to the DAG is drawn over which PC and CLOC are run. Runtime, conditional independence test counts called, and the structural hamming distances between the graph returned from each method and the true C-DAG are shown in Figure 5.

PC requires exponentially more independence tests relative to CLOC. Runtime also is improved for CLOC. As more efficient multivariate independence tests are developed, the runtime for CLOC can be expected to show even greater improvements. We find that the distances of the graphs generated by PC and CLOC are similar in value. Additional details and results are in the appendix.

5 Conclusions

In this work, we address the need for causal discovery over Markov causal systems by proposing a new graphical representation, αC-DAGs, that captures both causal directions and independence information over the clusters. We also introduce and characterize a novel graphical object for an equivalence class of αC-DAGs. We then propose a sound algorithm, CLOC, to learn this new graphical representation of an equivalence class from observational data. We illustrate that our proposed algorithm learns a graphical equivalence class over clusters that is just as (if not more accurate than) what can be learned by applying PC over variables and then applying clustering, and that our algorithm achieves this with fewer independence tests and faster runtime. Limitations of the approach include assumptions of causal sufficiency and faithfulness which may not apply for a given practical question. Users are required to have knowledge of a partition of variables into clusters that does not induce a cycle, which is non-negligible, while feasible for many applications. While in practice, CLOC may be limited by slow multi-variate conditional independence tests for certain data distributions or types, the foundational work introduced here sets the stage for improved scalability.

Acknowledgements

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

References

[1] Tara V. Anand, Adèle H. Ribeiro, Jin Tian, and Elias Bareinboim. Causal effect identification in cluster dags. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12172–12179. AAAI Press, 2023. doi: 10.1609/aaai.v37i10.26435.

[2] S A Andersson, D Madigan, and M D Perlman. A characterization of {M}arkov equivalence classes for acyclic digraphs. *Annals of Statistics*, 24:505–541, 1997.

[3] Bryan Andrews, Peter Spirtes, and Gregory F. Cooper. On the completeness of causal discovery in the presence of latent confounding with tiered background knowledge. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4002–4011. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/andrews20a.html.

[4] Shaofan Chen, Yuzhong Peng, Guoyuan He, Hao Zhang, Li Cai, and Chengdong Wei. Cdsc: Causal decomposition based on spectral clustering. *Information Sciences*, 657: 119985, 2024. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2023.119985. URL https://www.sciencedirect.com/science/article/pii/S0020025523015700.

[5] Wei Chen, Yunjin Wu, Ruichu Cai, Yueguo Chen, and Zhifeng Hao. Ccsl: A causal structure learning method from multiple unknown environments. *ArXiv*, abs/2111.09666, 2021. URL https://api.semanticscholar.org/CorpusID:244346148.

[6] David Maxwell Chickering. Learning equivalence classes of bayesian-network structures. *Journal of Machine Learning Research*, 2:445–498, 2002. doi: 10.1162/153244302760200696.

[7] Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, 10, 2019.

[8] Markus Kalisch, Martin Mächler, Diego Colombo, Marloes H. Maathuis, and Peter Bühlmann. Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software*, 47(11):1–26, 2012. doi: 10.18637/jss.v047.i11.

[9] C Meek. Causal inference and causal explanation with background knowledge. In P. Besnard and S. Hanks, editors, *Uncertainty in Artificial Intelligence 11*, pages 403–410. Morgan Kaufmann, San Francisco, 1995.

[10] Xueyan Niu, Xiaoyun Li, and Ping Li. Learning cluster causal diagrams: An information-theoretic approach. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4871–4877. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/675. URL https://doi.org/10.24963/ijcai.2022/675. Main Track.

[11] Pekka Parviainen and Samuel Kaski. Learning structures of bayesian networks for variable groups. *International Journal of Approximate Reasoning*, 88:110–127, 2017. ISSN 0888-613X. doi: https://doi.org/10.1016/j.ijar.2017.05.006. URL https://www.sciencedirect.com/science/article/pii/S0888613X17303134.

[12] Sepideh Pashami, Anders Holst, Juhee Bae, and Sławomir Nowaczyk. Causal discovery using clusters from observational data. In *Proceedings of the FAIM'18 Workshop on CausalML*, Stockholm, Sweden, July 2018. FAIM. URL https://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-39216. Refereed Conference Paper.

[13] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, NY, USA, 2nd edition, 2000.

[14] Eran Segal, Dana Pe'er, Aviv Regev, Daphne Koller, and Nir Friedman. Learning module networks. *Journal of Machine Learning Research*, 6(19):557–588, 2005. URL http://jmlr.org/papers/v6/segal05a.html.

[15] P Spirtes, C N Glymour, and R Scheines. *Causation, Prediction, and Search*. MIT Press, Cambridge, MA, 2nd edition, 2000.

[16] Peter Spirtes, Clark N Glymour, and R Scheines. *Causation, Prediction, and Search*. Springer-Verlag, New York, 1993.

[17] Peter Spirtes, Christopher Meek, and Thomas Richardson. Causal Inference in the Presence of Latent Variables and Selection Bias. *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 499–506, 1995. ISSN 0717-6163. doi: 10.1007/s13398-014-0173-7.2.

[18] Chandler Squires, Annie Yun, Eshaan Nichani, Raj Agrawal, and Caroline Uhler. Causal structure discovery between clusters of nodes induced by latent factors. In Bernhard Schölkopf, Caroline Uhler, and Kun Zhang, editors, *Proceedings of the First Conference on Causal Learning and Reasoning*, volume 177 of *Proceedings of Machine Learning Research*, pages 669–687. PMLR, 11–13 Apr 2022. URL https://proceedings.mlr.press/v177/squires22a.html.

[19] Thomas Sadanand Verma and Judea Pearl. An algorithm for deciding if a set of observed independencies has a causal explanation. In D. Dubois, M.P. Wellman, B. D'Ambrosio, and P. Smets, editors, *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence*, pages 323–330. Morgan Kaufmann, Stanford, CA, 1992.

[20] Jonas Wahl, Urmi Ninad, and Jakob Runge. Vector causal inference between two groups of variables. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i10.26450. URL https://doi.org/10.1609/aaai.v37i10.26450.

[21] Jonas Wahl, Urmi Ninad, and Jakob Runge. Foundations of causal discovery on groups of variables. *Journal of Causal Inference*, 12(1):20230041, 2024. doi: doi:10.1515/jci-2023-0041. URL https://doi.org/10.1515/jci-2023-0041.

[22] Raanan Yehezkel and Boaz Lerner. Bayesian network structure learning by recursive autonomy identification. *Journal of Machine Learning Research*, 10(53):1527–1570, 2009. URL http://jmlr.org/papers/v10/yehezkel09a.html.

List of Appendices

A Proofs

Remark 1. *In a Markov C-DAG with independence arcs, a conditionally-connecting independence arc always implies a collider structure.*

Proof. Consider an unshielded triplet $\langle \mathbf{C}_i, \mathbf{C}_k, \mathbf{C}_j \rangle$ such that $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_k,\mathbf{C}_j}$ is a conditionally-connecting independence arc. This implies that $\mathbf{C}_i \perp\!\!\!\perp \mathbf{C}_j | \mathbf{S} \setminus \mathbf{C}_k; \mathbf{C}_i \not\perp\!\!\!\perp \mathbf{C}_j | \mathbf{C}_k \cup \mathbf{S}$ where $\mathbf{S}$ is a separating set for $\mathbf{C}_i$ and $\mathbf{C}_j$. Then there must exist some path, $p = V_i, ..., V_k, ...V_j$ where $V_i \in \mathbf{C}_i$, $V_k \in \mathbf{C}_k$, and $V_j \in \mathbf{C}_j$, such that every non-endpoint node is a collider. In Markovian cases, this can only occur if there is only one non-endpoint. Therefore, V_k must be the only non-endpoint node on p such that V_k is a collider. Moreover, due to the admissibility of the partition, it follows that no additional variable in $\mathbf{C}_k$ can act as a cause of any variable in $\mathbf{C}_i$ or $\mathbf{C}_j$. Therefore, $\mathbf{C}_i, \mathbf{C}_k, \mathbf{C}_j$ must follow a collider structure. □

Theorem 1. ***[Soundness and completeness of d-separation in αC-DAGs.]*** *In an αC-DAG $G_{\mathbf{C}}$, let $\mathbf{X}, \mathbf{Z}, \mathbf{Y} \subset \mathbf{C}$. $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in $G_{\mathbf{C}}$, if and only if for any DAG, G compatible with $G_{\mathbf{C}}$, $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in G. $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})_{G_{\mathbf{C}}} \iff (\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z})_{G}$.*

Proof. First we prove the soundness of d-separation by showing that if $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in $G_{\mathbf{C}}$, then, in any ADMG, G, compatible with $G_{\mathbf{C}}$, $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in G. We show by contradiction. Assume $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in $G_{\mathbf{C}}$ but in some compatible ADMG, G, there exists a path p between a variable $X \in \mathbf{X}$ and $Y \in \mathbf{Y}$ that is active when the set of variables contained in cluster $\mathbf{Z}$ are conditioned on. By the preservation of paths and adjacencies, no connection is destroyed through clustering, so p in G is contained in a path $p_{\mathbf{C}}$ of $G_{\mathbf{C}}$ between clusters $\mathbf{X}$ and $\mathbf{Y}$. Since $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in $G_{\mathbf{C}}$, $p_{\mathbf{C}}$ is blocked, and $\mathbf{X}$ and $\mathbf{Y}$ are not adjacent. Therefore, by definition 8, there is at least one triplet of clusters in $p_{\mathbf{C}}$ that indicates a block on the path. Let this triplet be $\langle \mathbf{C}_i, \mathbf{C}_m, \mathbf{C}_j \rangle$, and let its associated independence arc be $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_m,\mathbf{C}_j}$ where $\mathbf{C}_m$ is distinct from $\mathbf{X}$ and $\mathbf{Y}$. Consider the subpath p_{ij} of p contained in the triplet $\langle \mathbf{C}_i, \mathbf{C}_m, \mathbf{C}_j \rangle$ in $p_{\mathbf{C}}$. Since p is active by assumption, every subpath of p is active, including p_{ij}. The triplet $\langle \mathbf{C}_i, \mathbf{C}_m, \mathbf{C}_j \rangle$ indicates a block on the path either if 1) $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_m,\mathbf{C}_j}$ is a never connecting arc with no connection marks $\oplus_{\mathbf{C}_d}$ such that $\mathbf{C}_d \in \mathbf{Z}$, 2) if $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_m,\mathbf{C}_j}$ is a marginally-connecting arc where $\mathbf{C}_m \in \mathbf{Z}$, 3) if $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_m,\mathbf{C}_j}$ is a conditionally-connecting arc such that $\mathbf{C}_m \notin \mathbf{Z}$ and with no connection mark $\oplus_{\mathbf{C}_d}$ such that $\mathbf{C}_d \notin \mathbf{Z}$ or 4) if there is a separation mark $\oslash_{\mathbf{C}_x}$ on $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_m,\mathbf{C}_j}$ such that $\mathbf{C}_x$ is on $p_{\mathbf{C}}$. In case 1, p_{ij} cannot be a connecting path or a collider path so p_{ij} would be inactive. In case 2, p_{ij} cannot be a collider path, and since $\mathbf{C}_m \in \mathbf{Z}$, p_{ij} cannot be active. In case 3, p_{ij} cannot be a connecting path and since $\mathbf{C}_m \notin \mathbf{Z}$ and for any connection mark $\oplus_{\mathbf{C}_d}$, $\mathbf{C}_d \notin \mathbf{Z}$, p_{ij} cannot be active. In case 4, definition 5 states that if $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_m,\mathbf{C}_j}$ is an always-connecting path, if $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_m,\mathbf{C}_j}$ is a marginally-connecting arc such that $\mathbf{C}_m \notin \mathbf{Z}$, or if $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_m,\mathbf{C}_j}$ is a conditionally-connecting arc such that $\mathbf{C}_m \in \mathbf{Z}$, then p_{ij} may be active, but since $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_m,\mathbf{C}_j}$ is marked with a separation mark $\oslash_{\mathbf{C}_x}$, there must exist some sub-path p_{ix} of p from some $V_i \in \mathbf{C}_i$ to some $V_x \in \mathbf{C}_x$ such that $\mathbf{C}_x$ is on $p_{\mathbf{C}}$ that is inactive. Therefore, p must be inactive, there is a contradiction, and we conclude that if

$\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in $G_{\mathbf{C}}$, then, in any ADMG, G, compatible with $G_{\mathbf{C}}$, $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in G.

Then, we prove the completeness of d-separation by showing that if $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in a ADMG G, then $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in a compatible αC-DAG $G_{\mathbf{C}}$. We prove by contradiction. Assume all paths from some $X \in \mathbf{X}$ to some $Y \in \mathbf{Y}$ are blocked by $\mathbf{Z}$ in some ADMG G, but $\mathbf{X}$ and $\mathbf{Y}$ are not d-separated by $\mathbf{Z}$ in $G_{\mathbf{C}}$, i.e. $(\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y}|\mathbf{Z})_{G_{\mathbf{C}}}$. If all paths from any $X \in \mathbf{X}$ to any $Y \in \mathbf{Y}$ are inactive by $\mathbf{Z}$, then by preservation of paths and adjacencies, $\mathbf{X}$ and $\mathbf{Y}$ are not adjacent in $G_{\mathbf{C}}$. No connections are destroyed through clustering so any p in G is contained in a path $p_{\mathbf{C}}$ of $G_{\mathbf{C}}$ between clusters $\mathbf{X}$ and $\mathbf{Y}$. Because $\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y}|\mathbf{Z}$ in $G_{\mathbf{C}}$, by Definition 8, there must exist some path $p_{\mathbf{C}}$ such that 1) for any triplet $\langle \mathbf{C}_i, \mathbf{C}_m, \mathbf{C}_j \rangle$ on $p_{\mathbf{C}}$, the independence arc $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_m,\mathbf{C}_j}$ marking it must not be marked by a separation mark $\oslash_{\mathbf{C}_k}$ where $\mathbf{C}_k$ is on $p_{\mathbf{C}}$, 2) for all marginally-connecting arcs $\mathbf{C}_m \notin \mathbf{Z}$, 3) for all conditionally connecting arcs $\mathbf{C}_m \in \mathbf{Z}$, or $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_m,\mathbf{C}_j}$ is marked with a connection mark $\oplus_{\mathbf{C}_d}$ and $\mathbf{C}_d$ or a true descendant is in $\mathbf{Z}$, 4) for all never-connecting arcs, $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_m,\mathbf{C}_j}$ is marked by a connection mark $\oplus_{\mathbf{C}_d}$ and $\mathbf{C}_d$ or a true descendant is in $\mathbf{Z}$.

For all paths p from some $X \in \mathbf{X}$ to some $Y \in \mathbf{Y}$ in G to be blocked, there must exist at least one triplet, $\langle V_i, V_m, V_j \rangle$, contained either within 1 cluster (i.e. $\langle V_i, V_m, V_j \rangle \in \mathbf{C}_m$) or between 2 (i.e. $\langle V_i, V_m \rangle \in \mathbf{C}_m, V_j \in \mathbf{C}_j$ or $V_i \in \mathbf{C}_i, \langle V_m, V_j \rangle \in \mathbf{C}_m$) or 3 clusters (i.e. $V_i \in \mathbf{C}_i, V_m \in \mathbf{C}_m, V_j \in \mathbf{C}_j$) on $p_{\mathbf{C}}$, that is blocked.

1. If the blocked triplet is a non-collider, $V_i \leftarrow V_m \rightarrow V_j$ or $V_i \rightarrow V_m \rightarrow V_j$, then V_m must be in $\mathbf{Z}$, which implies that $\mathbf{C}_m \in \mathbf{Z}$. As there could be multiple paths through a cluster, the triplet over clusters, $\langle \mathbf{C}_i, \mathbf{C}_m, \mathbf{C}_j \rangle$ could still be marked by any independence arc.
   (a) If $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_m,\mathbf{C}_j}$ is a marginally-connecting arc or never-connecting arc, since $\mathbf{C}_m \in \mathbf{Z}$, there is a contradiction with the implications of $(\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y}|\mathbf{Z})_{G_{\mathbf{C}}}$.
   (b) If $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_m,\mathbf{C}_j}$ is a conditionally-connecting arc, then then there must exist a different path, p', over variables through the triplet from some some $V_i' \in \mathbf{C}_i$ to $V_j' \in \mathbf{C}_j$ through $\mathbf{C}_m$ that is a collider path. Because $\mathbf{C}_m \in \mathbf{Z}$, either there is no $X \in \mathbf{X}$ or $Y \in \mathbf{Y}$ on p' or there must be another triplet, V_q, V_r, V_w, on p' that is blocked.
2. If the triplet is a collider, $V_i \rightarrow V_m \leftarrow V_j$, then V_m nor any of its descendants, V_d can be in $\mathbf{Z}$, implying that $C_m \notin \mathbf{Z}$ and $C_d \notin \mathbf{Z}$ where $V_d \in \mathbf{C}_d$ and $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_m,\mathbf{C}_j}$ is marked with the connection mark $\oplus_{\mathbf{C}_d}$.
   (a) If $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_m,\mathbf{C}_j}$ is a marginally-connecting arc, then there must exist a different path, p', over variables through the triplet from some some $V_i' \in \mathbf{C}_i$ to $V_j' \in \mathbf{C}_j$ through $\mathbf{C}_m$ that is a connecting path. Because $\mathbf{C}_m \notin \mathbf{Z}$, either there is no $X \in \mathbf{X}$ or $Y \in \mathbf{Y}$ on p' or there must be another triplet, V_q, V_r, V_w, on p' that is blocked.
   (b) If $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_m,\mathbf{C}_j}$ is a conditionally-connecting arc or a never-connecting arc, because $\mathbf{C}_m \notin \mathbf{Z}$, and there is a connection mark $\oplus_{\mathbf{C}_d}$, $\mathbf{C}_d \notin \mathbf{Z}$, there is a contradiction with the implications of $(\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y}|\mathbf{Z})_{G_{\mathbf{C}}}$.

For any path p' with a blocked triplet $\langle V_q, V_r, V_w \rangle$, either one of the conditions above leading to a contradiction (case 1a or 2b) applies, or there is a contradiction because a separation mark must exist along the path $p_{\mathbf{C}}$. By definition 5, the separation mark would be required because by assumption, all paths between any $X \in \mathbf{X}$ and $Y \in \mathbf{Y}$ are blocked by $\mathbf{Z}$ in G, so it is not possible for there to be a d-connecting path relative to $\mathbf{Z}$ in G analogous to $p_{\mathbf{C}}$ in $\mathbf{G}_C$. However, p is a d-connecting path relative to $\mathbf{Z}$ analogous to $p'_{\mathbf{C}} = \langle \mathbf{C}_i, ..., \mathbf{C}_r \rangle$ and p' is a d-connecting path relative to $\mathbf{Z}$ analogous to $p''_{\mathbf{C}} = \langle \mathbf{C}_m, ..., \mathbf{C}_w \rangle$, so by definition 5, the criteria is met and a separation must be placed.

If $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in G, it is also possible that there is no path from any $X \in \mathbf{X}$ to any $Y \in \mathbf{Y}$, and $\mathbf{Z}$ would equal the empty set. In this case, by preservation of adjacencies, for any triplet $\langle \mathbf{C}_i, \mathbf{C}_m, \mathbf{C}_j \rangle$ along $p_{\mathbf{C}}$, there must be some $V_i \in \mathbf{C}_i$ adjacent to some $V_m \in \mathbf{C}_m$, and some $V_m' \in \mathbf{C}_m$ adjacent to some $V_j \in \mathbf{C}_j$. Then, there must exist some such triplet where V_m is not adjacent to V_m'. If for all V_m and V_m' in C_m, V_m and V_m' are not adjacent, then $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_m,\mathbf{C}_j}$ must be marked with a never-connecting arc in $G_{\mathbf{C}}$ with no connection mark, and there would be a contradiction with the implications of $(\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y}|\mathbf{Z})_{G_{\mathbf{C}}}$. Otherwise, because $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in G, there must exist some connecting subpaths of $p_{\mathbf{C}}$, $\mathbf{C}_i, ..., \mathbf{C}_n$ and $\mathbf{C}_{i+1}, ..., \mathbf{C}_n + 1$ such that $\mathbf{C}_i \not\perp\!\!\!\perp \mathbf{C}_{n+1}$, which, by definition 5, necessitates a separation mark and then there would be a contradiction with the implications of $(\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y}|\mathbf{Z})_{G_{\mathbf{C}}}$. □

Theorem 2. *Two αC-DAGs, $G_{\mathbf{C}_1}$ and $G_{\mathbf{C}_2}$ (with the same partition $\mathbf{C}$ over the same set of variables $\mathbf{V}$) are cluster Markov equivalent if and only if they share the same: 1) adjacencies, 2) independence arcs, 3) separation marks and 4) connection marks.*

Proof. The proof follows directly from the definitions of cluster Markov equivalence, and d-separation for αC-DAGs. Because d-separation is determined solely by the independence arcs, separation marks, and connection marks in a graph for a series of adjacent clusters, two αC-DAGs with the same adjacencies, independence arcs, separation marks, and connection marks will necessarily lead to the same d-separations between clusters and will therefore be cluster Markov equivalent. □

Theorem 3. ***[Soundness and Completeness of Orientation Rules and CLOC]*** *The five orientation rules and the procedure of CLOC are sound and complete.*

First we prove the soundness of the collider search and each of the five orientation rules. We then establish orientation completeness by showing that, whenever no more rules can be applied, there exist two Markov-equivalent αC-DAGs that differ in orientation of any undirected edge. The proof for the soundness and completeness of CLOC follows. First, we introduce two remarks complementing remark 1, and an associated lemma.

Remark 3. *In a Markov C-DAG with independence arcs, a marginally-connecting independence arc always implies a non-collider structure.*

Proof. We prove by contradiction. Consider an unshielded triplet $\langle \mathbf{C}_i, \mathbf{C}_k, \mathbf{C}_j \rangle$ such that $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_k,\mathbf{C}_j}$ is a marginally-connecting independence arc. We show that orienting the triple as $\mathbf{C}_i \to \mathbf{C}_k \leftarrow \mathbf{C}_j$ necessarily leads to a contradiction. By definition of a marginally-connecting independence arc, we have $\mathbf{C}_i \not\perp\!\!\!\perp \mathbf{C}_j | \mathbf{S} \setminus \mathbf{C}_k; \mathbf{C}_i \perp\!\!\!\perp \mathbf{C}_j | \mathbf{S} \cup \mathbf{C}_k$, where $\mathbf{S}$ is a separating set for $\mathbf{C}_i$ and $\mathbf{C}_j$. Assume that the structure over clusters forms a collider, $\mathbf{C}_i \to \mathbf{C}_k \leftarrow \mathbf{C}_j$. There are two possible cases: either there is no path at all between $\mathbf{C}_i$ and $\mathbf{C}_j$ through $\mathbf{C}_k$, or such a path exists. If no such path exists, then the dependence implied by the marginally-connecting independence arc $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_k,\mathbf{C}_j}$ cannot hold, leading to a contradiction. If there exists a path p between $\mathbf{C}_i$ and $\mathbf{C}_j$ through $\mathbf{C}_k$, then, since $\mathbf{C}_i$ is assumed to point to $\mathbf{C}_k$, there must be a pair of nodes $V_i \in \mathbf{C}_i$ and $V_k \in \mathbf{C}_k$ on p such that $V_i \to V_k$. By the admissibility of the partition, an edge of the form $V_i \leftarrow V_k$ is not allowed. To preserve the marginal dependence implied by the marginally-connecting independence arc $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_k,\mathbf{C}_j}$, every subsequent edge between $V_k, V_{k+1} \in \mathbf{C}_k$ along the path p must be of the form $V_k \to V_{k+1}$. Otherwise, a collider would be introduced, rendering the path inactive and violating the assumed marginal dependence, leading to a contradiction. Now, because $\mathbf{C}_k \leftarrow \mathbf{C}_j$, there must also exist some $V_j \in \mathbf{C}_j$ and some $V'_k \in \mathbf{C}_k$ such that $V'_k \leftarrow V_j$ where V'_k is on p. Because of the assumption of the admissibility of the partition, there can be no edge $V'k \to V_j$. Then there must exist a collider and there is a contradiction. Therefore, the triplet $\langle \mathbf{C}_i, \mathbf{C}_k, \mathbf{C}_j \rangle$ must be a non-collider. □

Remark 4. *In a Markov C-DAG with independence arcs, a never-connecting independence arc could imply either a collider or a non-collider structure.*

Proof. Consider a triplet $\langle \mathbf{C}_i, \mathbf{C}_k, \mathbf{C}_j \rangle$ such that $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_k,\mathbf{C}_j}$ is a never-connecting independence arc. This implies that $\mathbf{C}_i \perp\!\!\!\perp \mathbf{C}_j | \mathbf{S} \setminus \mathbf{C}_k; \mathbf{C}_i \perp\!\!\!\perp \mathbf{C}_j | \mathbf{S} \cup \mathbf{C}_k$, where $\mathbf{S}$ is a separating set for $\mathbf{C}_i$ and $\mathbf{C}_j$. Then either there is no path from any $V_i \in \mathbf{C}_i$ to some $V_j \in \mathbf{C}_j$ through $\mathbf{C}_k$, or every such path p must include at least 4 nodes, $p = V_i, ..., V_{k_1}, V_{k_2}, ..., V_j$ where $V_i \in \mathbf{C}_i$, $V_{k_1}, V_{k_2}, \in \mathbf{C}_k$, and $V_j \in \mathbf{C}_j$, such that there is at least one collider triplet and at least one non-collider triplet on p. Consider the latter case. Let p be a path of exactly 4 nodes $\langle V_i, V_{k_1}, V_{k_2}, V_j \rangle$ such that $V_i \in \mathbf{C}_i$, $V_{k_1}, V_{k_2}, \in \mathbf{C}_k$ and $V_j \in \mathbf{C}_j$. Either V_{k_1} is a collider node and V_{k_2} is a non-collider node or V_{k_1} is a non-collider node and V_{k_2} is a collider node. In the first case, $V_i \to V_{k_1} \leftarrow V_{k_2} \to V_j$ or $V_i \to V_{k_1} \leftarrow V_{k_2} \leftarrow V_j$. In the second case, $V_i \to V_{k_1} \to V_{k_2} \leftarrow V_j$ or $V_i \leftarrow V_{k_1} \to V_{k_2} \leftarrow V_j$. Then $\langle \mathbf{C}_i, \mathbf{C}_k, \mathbf{C}_j \rangle$ may be either a collider or a non-collider. Adding any additional node, V_{k_i+1}, to p either creates an additional collider or an additional non-collider, but still allows for collider and non-collider structures over clusters. Now consider where there is no path from any $V_i \in \mathbf{C}_i$ to some $V_j \in \mathbf{C}_j$ through $\mathbf{C}_k$. Then the direction of any edge $V_i - V_k$ or $V'_k - V_j$ can be variant such that $\langle \mathbf{C}_i, \mathbf{C}_k, \mathbf{C}_j \rangle$ may be either a collider or a non-collider. □

Lemma 1. *For a distribution $P(\mathbf{C})$ over clusters $\mathbf{C} = \langle \mathbf{C}_1, ..., \mathbf{C}_n \rangle$ such that for every triplet $\langle \mathbf{C}_i, \mathbf{C}_k, \mathbf{C}_j \rangle$, $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_k,\mathbf{C}_j}$ is not a never-connecting independence arc, the orientation rules reduces to Meek's rules [9] and the PC algorithm [16].*

Proof. The proof follows from noting that modifications to Rules 1 and 3 require independence arcs aligning with the independence information typically associated with colliders and non-colliders over variables, and from Remarks 1, 3, and 4. The absence of never-connecting arcs ensure triplets exhibit expected behavior with regards to structure and observed independencies and dependencies. When there are no never-connecting arcs, Rule 5 reduces to Rule 1, as all triplets marked with conditionally-connecting arcs must be a collider, and any descendant of that collider is part of a non-collider triplet, so will be oriented by Rule 1. When there are no never-connecting arcs and there is no background knowledge, Rule 4 never applies, following from Meek, 1995 [9]. □

$\mathscr{R}_0$: If $\mathbf{X} - \mathbf{Z} - \mathbf{Y}$, $\mathbf{X}$ and $\mathbf{Y}$ are not adjacent, and $\mathcal{A}_{\mathbf{X},\mathbf{Z},\mathbf{Y}}$ is conditionally-connecting, then orient the triplet as $\mathbf{X} \to \mathbf{Z} \leftarrow \mathbf{Y}$

Proof. The proof of soundness follows directly from Remark 1.

□

$\mathscr{R}_1$: If $\mathbf{X} \to \mathbf{Z} - \mathbf{Y}$, $\mathbf{X}$ and $\mathbf{Y}$ are not adjacent, and $\mathcal{A}_{\mathbf{X},\mathbf{Z},\mathbf{Y}}$ is marginally-connecting, then orient the triplet as $\mathbf{X} \to \mathbf{Z} \to \mathbf{Y}$.

Proof. The proof for soundness follows directly from Remark 3.

□

$\mathscr{R}_2$: If $\mathbf{X} \to \mathbf{Z} \to \mathbf{Y}$ and $\mathbf{X} - \mathbf{Y}$, then orient $\mathbf{X} - \mathbf{Y}$ as $\mathbf{X} \to \mathbf{Y}$.

Proof. The soundness of the rule comes from observing that if $\mathbf{X} \leftarrow \mathbf{Y}$, a cycle would be induced, violating the admissible partition criteria of αC-DAGs. □

$\mathscr{R}_3$: If $\mathbf{X} \to \mathbf{Z} \leftarrow \mathbf{Y}$, $\mathbf{X} - \mathbf{W} - \mathbf{Y}$, $\mathbf{X}$ and $\mathbf{Y}$ are not adjacent, $\mathbf{W} - \mathbf{Z}$, and $\mathcal{A}_{\mathbf{X},\mathbf{W},\mathbf{Y}}$ is marginally-connecting, then orient $\mathbf{W} - \mathbf{Z}$ as $\mathbf{W} \to \mathbf{Z}$.

Proof. The soundness of the rule comes from observing that if $\mathbf{W} \leftarrow \mathbf{Z}$, then by two applications of rule 2, $\mathbf{Y} \to \mathbf{W}$, $\mathbf{X} \to \mathbf{W}$, and then there would be a collider at $\mathbf{W}$. Since $\mathcal{A}_{\mathbf{X},\mathbf{W},\mathbf{Y}}$ is marginally connecting, there is a contradiction by remark 3.

□

$\mathscr{R}_4$: If $\mathbf{X} \to \mathbf{Z} \to \mathbf{Y}$, $\mathbf{X} - \mathbf{W} - \mathbf{Y}$, $\mathbf{X}$ and $\mathbf{Y}$ are not adjacent, $\mathbf{W} \ast\!\!-\!\!\ast\, \mathbf{Z}$, and $\mathcal{A}_{\mathbf{X},\mathbf{W},\mathbf{Y}}$ is marginally-connecting, then orient $\mathbf{W} - \mathbf{Y}$ as $\mathbf{W} \to \mathbf{Y}$.

Proof. The soundness of the rule comes from observing that if $\mathbf{W} \leftarrow \mathbf{Y}$, then to avoid a cycle, it must be that $\mathbf{X} \to \mathbf{W}$. Then, however, there would be a collider at $\mathbf{W}$, but $\mathcal{A}_{\mathbf{X},\mathbf{W},\mathbf{Y}}$ is marginally connecting, so there is a contradiction. □

$\mathscr{R}_5$: If $\mathbf{X} \ast\!\!-\!\!\ast\, \mathbf{Z} \ast\!\!-\!\!\ast\, \mathbf{Y}$, $\mathbf{Z} - \mathbf{W}$, $\mathbf{X}$ and $\mathbf{W}$ are not adjacent, $\mathbf{Y}$ and $\mathbf{W}$ are not adjacent, and $\mathcal{A}_{\mathbf{X},\mathbf{Z},\mathbf{Y}}$ is never-connecting or conditionally-connecting with connection mark $\oplus_{\boldsymbol{\mathcal{D}}}$ such that $\mathbf{W} \in \boldsymbol{\mathcal{D}}$, then orient $\mathbf{Z} - \mathbf{W}$ as $\mathbf{Z} \to \mathbf{W}$.

Proof. The soundness of the rule comes from the definition of a connection mark, $\oplus_{\mathcal{D}}$, where any cluster $\mathbf{W} \in \mathcal{D}$ must be a descendant of a collider, such that $\mathbf{Z} \to \mathbf{W}$. □

Next we prove orientation completeness for Rules 1-5.

Lemma 2. *Rules 1-5 collectively are complete in the sense that all orientations determined from successive application are valid and result in all possible orientations.*

Proof. In the case that there are no never-connecting arcs, by lemma 1 the rules are complete following Meek 1995 [9]. If there is one or more never-connecting arc, the orientation rules of CLOC result in fewer orientations, as never-connecting arcs always imply ambiguous orientations by remark 4. For any edge between $\mathbf{C}_i$ and $\mathbf{C}_j$ left undirected by successive applications of Rules 1-5, either the edge is part of a triplet marked with a marginally connecting or conditionally connecting arcs, or it is part of a triplet marked with a never-connecting arc. In the former case, by lemma 1 the cluster Markov equivalence class includes at least one model with $\mathbf{C}_i \to \mathbf{C}_j$ and at least one with $\mathbf{C}_i \leftarrow \mathbf{C}_j$. In the latter case, by Remark 4, there exists at least one model in the cluster Markov equivalence class with $\mathbf{C}_i \to \mathbf{C}_j$ and at least one with $\mathbf{C}_i \leftarrow \mathbf{C}_j$.

Because CLOC and the orientation rules only make use of cluster level independence and dependence information, all marginal and conditional independencies for a given triplet are already evaluated. For a given triplet $\mathbf{C}_i, \mathbf{C}_k, \mathbf{C}_j$, by theorem 1, $\mathbf{C}_i$ and $\mathbf{C}_j$ can only be dependent if 1) they are adjacent, 2) they are not adjacent and $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_k,\mathbf{C}_j}$ is marginally connecting, 3) they are not adjacent, $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_k,\mathbf{C}_j}$ is conditionally connecting, and $\mathbf{C}_k$ is in the conditioning set, or 4) they are not adjacent, $\mathcal{A}_{\mathbf{C}_i,\mathbf{C}_k,\mathbf{C}_j}$ is never connecting, and there exists some descendant of a variable-level collider within $\mathbf{C}_k$ in cluster $\mathbf{C}_w$ where $\mathbf{C}_w$ is in the conditioning set. Cases 1, 2, and 3 are covered by the skeleton and collider search phases. Rule 5 captures conditional dependencies created by case 4, such that orientations for a non-oriented triplet can be made to reflect the dependence. As orientations of Rule 5 follow a non-standard pattern relative to Rules 1-3, we can consider information determined by Rule 5 to be a form of background knowledge introduced to the graph. Then, with Rule 4, and given the admissibility assumption of the partition, the proof for completeness extends directly from Meek 1995, where the PC algorithm with background knowledge is proved to be complete in that any subsequent orientations that can be determined following Rule 5 must be valid and complete. □

Finally, we prove Theorem 3 by showing that CLOC does return an αC-CPDAG.

Proof. An αC-CPDAG must reflect the cluster Markov equivalence class of αC-DAGs for a given partition. This means that all cluster level independencies and dependencies must be represented, all directed edges are non-variant and all undirected edges are variant. The proof for non-variant directed edges and variant undirected edges follows from lemma 2. To represent all independencies and dependencies, we must ensure that all adjacencies, independence arcs, separation marks, and connection marks are determined. The proof for valid adjacencies follows directly from the proof for skeleton construction of Spirtes et. al 1993 [16]. The procedure for determining independence arcs follows from definition 2, where for each triplet, searches for variables in or not in the separating set for any given pair of variables $\mathbf{X}$ and $\mathbf{Y}$ allows for determination of the appropriate arc. The procedure for determining separation marks follows from definition 5, where independence tests are performed to identify where the closest pair of clusters, appearing to be dependent, are in fact independent. Lastly, the procedure for determining connection marks follows from definition 6, where independence tests are performed to determine if any combination of possible descendants render two variables dependents such that the set of clusters are necessarily descendants. Therefore, by theorem 2, the αC-CPDAG completely represents a cluster Markov equivalence class. □

Remark 5. *CLOC is complete with background knowledge.*

Proof. The proof follows directly from the completeness of CLOC including the orientation rule (Rule 4) for background knowledge. □

B Further discussion on αC-DAG semantics

B.1 On separation marks, connection marks, and graph interpretation

In this section, we extend the discussion on the interpretation and semantics of αC-DAGs.

We first further explore separation marks and connection marks. We note that separation marks can be placed on any independence arc that signifies a connection: marginally-connecting arcs, or conditionally-connecting arcs. Separation marks can not be placed on never-connecting arcs, as there is no connection for the separation mark to dispute. When a separation mark is found on a marginally-connecting arc, a marginal connection is disputed. When a separation mark is on a

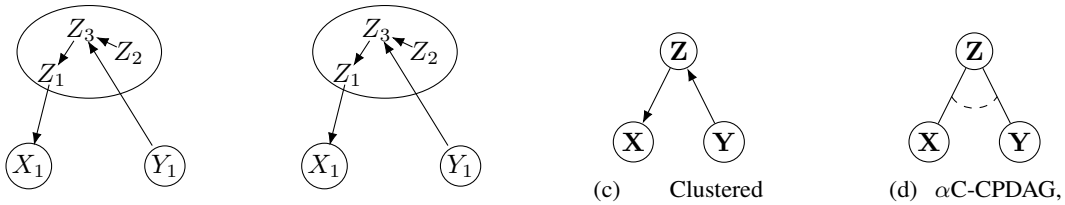
conditionally-connecting arc, the connection, conditional on the center node of the triplet marked by the independence arc, is disputed. Since paths can be traversed in two directions, and independence statements can be read in two ways ($\mathbf{X} \perp\!\!\!\perp \mathbf{Y}, \mathbf{Y} \perp\!\!\!\perp \mathbf{X}$), separation marks come in pairs.

Connection marks are read in a way distinct from separation marks. The subscript of a connection mark indicates the directly connected nodes or sets of nodes that, when conditioned on, create a connecting triplet where there otherwise is not one. Any true descendants of the nodes in the subscript of the connection mark are understood to also create the connection, where a true descendant is identified by a true connecting path over clusters (see d-separation criteria, Def. 8). Connection marks can only be placed along never-connecting independence arcs. This is because a marginally active triplet can not have a new connection created due to conditioning on a descendant of a collider because the triplet is already active. If the center node of the triplet marked by a marginally-connecting independence arc is conditioned on, any descendant of a collider that is conditioned on would still fail to create a new connection as the independence arc necessitates there are non-colliders along any path the collider may appear on, which would be conditioned on, so the path would be blocked. As conditionally-connecting arcs require a collider, any true descendant will create a connection, following expected behavior, so there is no need to explicitly denote a connection mark. Lastly, we note that the subscript of a connection mark can be a set of sets of clusters. Each set of cluster denotes one way that the triplet can be made active, and it is noted that a path through a cluster with multiple colliders on it would need multiple descendants (possibly in different clusters) to be conditioned on for the triplet over clusters to be active.

There are certain graph semantics and attributes that require new interpretation for αC-DAGs. In particular, we can create a more refined class of descendants and ancestors, informed by connections through the clusters. In C-DAGs, similarly as in DAGs and other graphs, a directed path from some node $\mathbf{C}_0$ to $\mathbf{C}_n$ is a sequence of distinct vertices $\langle \mathbf{C}_0, ..., \mathbf{C}_n \rangle$ such that for $0 \leq i \leq n-1$, C_i is a parent of C_{i+1} in $G_{\mathbf{C}}$. In αC-DAGs, applying this same definition yields what we define as an **apparent directed path**, since even with the described pattern of edges, it is possible to have independence arcs and separation marks that describe a break or block which contradicts the notion of a directed path. By contrast a **true directed path** in an αC-DAG from some node $\mathbf{C}_0$ to $\mathbf{C}_n$ is defined as a sequence of distinct vertices $\langle \mathbf{C}_0, ..., \mathbf{C}_n \rangle$ such that for $0 \leq i \leq n-1$, $\mathbf{C}_i$ is a parent of $\mathbf{C}_{i+1}$ in $G_{\mathbf{C}}$ and where every arc on the corresponding arc trajectory is a marginally-connecting arc with no separation marks. Then, $\mathbf{C}_A$ is called a **true ancestor** of $\mathbf{C}_B$ and $\mathbf{C}_B$ a **true descendant** of $\mathbf{C}_A$ if $\mathbf{C}_A = \mathbf{C}_B$ or there is a true directed path from $\mathbf{C}_A$ to $\mathbf{C}_B$. We contrast these terms with what we call **apparent ancestors** and **apparent descendants** where there may only be an apparent directed path from $\mathbf{C}_A$ to $\mathbf{C}_B$. In αC-DAGs, we use the notation $An_{G_{\mathbf{C}}}(\mathbf{C}_B)$ and $De_{G_{\mathbf{C}}}(\mathbf{C}_A)$ to refer to the sets of **true ancestors** of $\mathbf{C}_B$ and **true descendants** of $\mathbf{C}_A$ in $G_{\mathbf{C}}$, respectively.

B.2 On relaxing the assumption of acyclicity

In our definition of αC-DAGs (and by extension for αC-CPDAGs), we require that there is no **apparent cycle** over clusters, that is where for some pair of clusters $\mathbf{C}_i, \mathbf{C}_j$, where there exists an edge $\mathbf{C}_i \to \mathbf{C}_j$, there is no directed path $\mathbf{C}_j \to .... \to \mathbf{C}_i$. We believe this is a reasonable assumption in the context of clusters as the user intentionally defines the partition over variables, likely because these variables represent together some semantically meaningful entity or are otherwise similar in some ways, such that knowledge of a potential cycle is available. However, we also note that in some cases, such an assumption may not be feasible, and it is easy to construct an example where the underlying graph over variables is acyclic, but a certain partition over the variables creates an apparent cycle. In such a case, αC-DAGs have the representational capacity to differentiate between a true cycle and an apparent cycle, as is clear by the discussion above differentiating between true and apparent ancestors and descendants. Specifically, if the assumption of acyclicity over clusters is relaxed (assuming an acyclic distribution over variables), then where there is an edge $\mathbf{C}_i \to \mathbf{C}_j$ and some directed path $\mathbf{C}_j \to .... \to \mathbf{C}_i$, there will necessarily exist some independence arc or separation mark along the path $\mathbf{C}_j \to .... \to \mathbf{C}_i$ that denotes that $\mathbf{C}_j$ is not a true ancestor of $\mathbf{C}_i$, and therefore there is no true cycle. In this context, properties such as d-separation extend soundly for αC-DAGs. However, the relaxation of the assumption of no apparent cycles over clusters does have implications in the context of structure learning. In particular, rules that leverage this assumption of acyclicity are no longer valid, such as Rule 2 and Rule 4. Rule 3 depends upon the validity of Rule 2 and therefore also becomes invalid. An area of future work is to determine sound extension of or different rules that allow for sound and complete learning over clusters when the acyclicity assumption is relaxed.



(a) DAG G_1 (b) CPDAG, $\mathcal{P}$ (c) Clustered CPDAG, $\mathcal{P}_{\mathbf{C}}$ (d) αC-CPDAG, $\mathcal{P}'_{\mathbf{C}}$

Figure 6: (a) is a DAG and (b) is the CPDAG that comes from G_1. Following the procedure in definition 12, (c) is the clustered CPDAG that comes from $\mathcal{P}$. This object reflects orientations that are determined from tests on $P(\mathbf{V})$. By contrast, (d) is the αC-CPDAG that corresponds to G_1. All edges are undirected as $\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y}; \mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z}$ and the edges cannot be oriented as, by Remark 3, the cluster level dependencies and independencies align with the representations of $\mathbf{X} \to \mathbf{Z} \to \mathbf{Y}$, $\mathbf{X} \leftarrow \mathbf{Z} \to \mathbf{Y}$, and $\mathbf{X} \leftarrow \mathbf{Z} \leftarrow \mathbf{Y}$.

B.3 On the special case of clusters of size 1

We note that when all clusters include at most 1 variable, CLOC reduces to PC, following Lemma 1. Independence arcs, separation marks, and connection marks all become redundant. When clusters have more than 1 variable, and there are no never-connecting arcs, the orientation rules also reduce to PC, however the graphical object still requires separation and connection marks to fully represent conditional independences and dependences. When clusters have at most 1 variable, this is no longer the case. For any triplet $\langle \mathbf{C}_i, \mathbf{C}_k, \mathbf{C}_j \rangle$ such that $\mathbf{C}_k$ is of size $n = 1$ (i.e. there is only one variable in the cluster), the alignment of the edge orientations and marginal and conditional independencies and dependencies will be aligned as the case is for variables. For a simplified representation in αC-DAGs and αC-CPDAG, independence arcs and connection marks could be removed for these triplets. The interpretation of this object is that wherever there is an omitted independence arc, the behavior for the triplet is as anticipated. If there exists another triplet in the graph $\langle \mathbf{C}_r, \mathbf{C}_q, \mathbf{C}_w \rangle$ such that $\mathbf{C}_q$ is not of size $n = 1$, it is possible a separation mark is required for $\langle \mathbf{C}_i, \mathbf{C}_k, \mathbf{C}_j \rangle$, in which case the independence arc, with the appropriate separation mark, would be required. If all clusters in an αC-DAG or αC-CPDAG include at most 1 variable, then the simplified representation holds for all triplets and the result would be a DAG or CPDAG respectively.

C Experimental details and additional results

C.1 Experimental Setup

All experiments were run on a machine with CPU: Intel i9 Chip, 32 GB of RAM, and macOS operating system. A single core was used for the experiments. Algorithms are implemented in R.

In our simulations, we compare two approaches to developing a clustered graphical equivalence class. The first approach consists of applying PC to the distribution over variables, $P(\mathbf{V})$, and then imposing clusters. The clustering procedure is shown below.

Definition 12 (Clustered CPDAG.). *Given a CPDAG, $\mathcal{P}$ over variables $\mathbf{V}$, and a partition $\mathbf{C} = \{\mathbf{C}_1, ..., \mathbf{C}_n\}$ of $\mathbf{V}$, construct a graph $\mathcal{P}_{\mathbf{C}}$ over $\mathbf{C}$ as follows.*

- *An edge $\mathbf{C}_i \to \mathbf{C}_j$ is in $\mathcal{P}_{\mathbf{C}}$ if there exists some $V_i \in \mathbf{C}_i$ and some $V_j \in \mathbf{C}_j$ such that $V_i \in Pa(V_j)$ in $\mathcal{P}$*
- *An edge $\mathbf{C}_i - \mathbf{C}_j$ is in $\mathcal{P}_{\mathbf{C}}$ if for all $V_i \in \mathbf{C}_i$ that are adjacent to some $V_j \in \mathbf{C}_j$, there is an undirected edge between V_i and V_j, i.e. $V_i - V_j$.*

We note that the graphical object created by the procedure above, which we refer to as a clustered CPDAG, determined by the PC-then-Cluster approach, is distinct from an αC-CPDAG. In particular, edges that may in fact be variant in a cluster Markov equivalence class may become oriented in the clustered CPDAG, due to some feature of the distribution over variables. For example, in Figure 6, the distribution over variables, $P(\mathbf{V})$ allows the collider over $\langle Z_2, Z_3, Y_1 \rangle$ to be learned, allowing for an orientation between $\mathbf{Y}$ and $\mathbf{Z}$ to be possible for the clustered CPDAG. Subsequent applications of Rule 1 of the PC algorithm allows for orientation of the edge $Z_1 \to X_1$, so that an orientation between $\mathbf{X}$ and $\mathbf{Z}$ is possible. By contrast, the αC-CPDAGis learned from the distribution $P(\mathbf{C})$ where cluster-level independence tests reveal $\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y}; \mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z}$. The cluster Markov equivalence

class for this information includes graphs with the orientations $\mathbf{X} \to \mathbf{Z} \to \mathbf{Y}$, $\mathbf{X} \leftarrow \mathbf{Z} \to \mathbf{Y}$, and $\mathbf{X} \leftarrow \mathbf{Z} \leftarrow \mathbf{Y}$, so no orientations in the αC-CPDAGcan be made.

For the experiments in the main body of the paper, we compare the methods of CLOC and the PC-cluster approach, as there is no other comparable method outputting an equivalence class over clusters. For the latter method, we use the built-in implementation of PC in the R package pcalg [8] using the gaussCItest built-in for the independence test over variables. The output is a CPDAG, which is then clustered by the procedure described in definition 12 using the defined partition over variables into clusters. In our implementation of CLOC the multi-variate conditional independence test used iterates over pair-wise tests of variable level independence tests with early stopping when a dependence is determined implying dependence over clusters.

C.2 Additional results

We show additional experimental results in Figure 7. In comparing oracle (ground truth) results by the PC-then-cluster approach with CLOC, we can note information that is lost by using only cluster-level information rather than variable-level information. As is illustrated in Figure 6, orientations beyond those representing the cluster Markov equivalence class are possible when the (variable-level) Markov equivalence class is learned by leveraging $P(\mathbf{V})$. The blue line on the plot shows how much of this sort of information, translating to orientations aligning with $P(\mathbf{V})$, is lost when only $P(\mathbf{C})$ is used. We expect this number to be non-zero. This tradeoff in orientation capacity can be weighed against improvements in required number of conditional independence tests and runtime, as demonstrated in the main body.

The green and red lines compare, for each method of CLOC and the PC-then-cluster approach, the structural hamming distance between a graph estimated from a data sample as compared to the ground truth equivalence class. We note that we see lower structural hamming distances for CLOC compared to the PC-then-Cluster approach, which reflects robustness of our proposed method to noise in data samples.

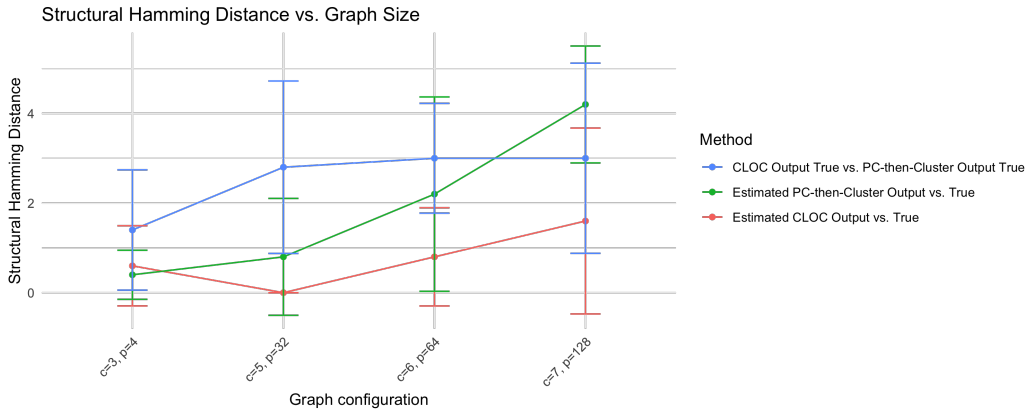


Figure 7: Red: comparison of CLOC output, estimated from a simulated Gaussian dataset, compared to the oracle for the corresponding data-generating process. Green: comparison of the PC-then-Cluster approach output, estimated from a simulated Guassian dataset, compared to the oracle for the corresponding data-generating process. Blue: Comparison of the oracle solutions by CLOC and the PC-then-Cluster approach.