[Technical Appendix Updated] Relaxing partition admissibility in Cluster-DAGs: a causal calculus with arbitrary variable clustering

Anonymous Author(s)
Affiliation
Address
email

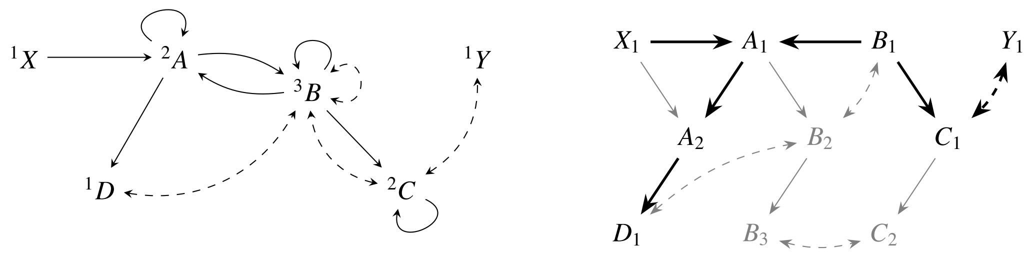
Abstract

Cluster DAGs (C-DAGs) provide an abstraction of causal graphs in which nodes represent clusters of variables, and edges encode both cluster-level causal relationships and dependencies arisen from unobserved confounding. C-DAGs define an equivalence class of acyclic causal graphs that agree on cluster-level relationships, enabling causal reasoning at a higher level of abstraction. However, when the chosen clustering induces cycles in the resulting C-DAG, the partition is deemed inadmissible under conventional C-DAG semantics. In this work, we extend the C-DAG framework to support arbitrary variable clusterings by relaxing the partition admissibility constraint, thereby allowing cyclic C-DAG representations. We extend the notions of d-separation and causal calculus to this setting, significantly broadening the scope of causal reasoning across clusters and enabling the application of C-DAGs in previously intractable scenarios. Our calculus is both sound and atomically complete with respect to the do-calculus: all valid interventional queries at the cluster level can be derived using our rules, each corresponding to a primitive do-calculus step.

Knowing the effect of a treatment X on an outcome Y, encoded by an intervention $do(X)$ in the interventional distribution $P(Y|do(X))$, is crucial in many applications. However, performing interventions is often impractical due to ethical concerns, potential harm or prohibitive costs. In such cases, one can instead aim to identify do-free formulas that estimate the effects of interventions using only observational (non-experimental) data and a causal graph (Pearl, 2009). Solving the identifiability problem typically involves establishing graphical criteria under which the total effect is identifiable, and providing a do-free formula for estimating it from observational data. However, specifying a causal diagram requires prior knowledge of the causal relationships between all observed variables, a requirement that is often unmet in real-world applications. This challenge is particularly acute in complex, high-dimensional settings, limiting the practical applicability of causal inference methods.

One way to circumvent this difficulty is to rely on abstract representations which group several variables into macro-variables, a mapping usually referred to as causal representation learning (Schölkopf et al., 2021), which are connected through causal relationships and dependencies arisen from unobserved confounding. Several studies have been devoted to causal discovery of and causal inference in specific abstract representations, both for static and dynamic (time series) variables, as Assaad et al. (2022); Ferreira and Assaad (2024); Anand et al. (2023); Wahl et al. (2023). Recent studies have also tackled the related problem of defining mappings from clusters to variables while preserving specific causal properties (Chalupka et al., 2015, 2016; Rubenstein et al., 2017; Beckers and Halpern, 2019), and have provided a thorough theoretical analysis of the relationship between micro- and macro-level causal models with a view on causal discovery assumptions (Wahl et al., 2024).

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.



(a) $\mathcal{G}^C$

(b) $\mathcal{G}^m \in C(\mathcal{G}^C)$

Figure 1: Left: a C-DAG $\mathcal{G}^C = (\mathcal{V}^C, \mathcal{E}^C)$, Right: a graph $\mathcal{G}^m = (\mathcal{V}^m, \mathcal{E}^m)$ that is compatible with $\mathcal{G}^C$. For example, $^2A \in \mathcal{V}^C$ corresponds to $\{A_1, A_2\} \subseteq \mathcal{V}^m$. In $\mathcal{G}^m$, a *structure of interest* (Definition 3) is highlighted in **bold black**, with all other nodes and edges shown in gray.

The starting point of our study is the framework recently proposed in Anand et al. (2023), which relaxes the strict requirement of a fully specified causal diagram and provides a foundation for valid inference over clusters of variables. However, it focuses on abstract graphs, called Cluster-DAGs, which do not contain cycles between clusters, a restriction known as *partition admissibility*. We extend this framework in this paper by removing this restriction and consider arbitrary clusterings of variables, potentially resulting in abstract graphs with self-loops and cycles between clusters.

To tackle identification in causal abstractions, recent work has introduced separation criteria that generalize d-separation to specific abstraction types (Jaber et al., 2022; Perković et al., 2018; Ferreira and Assaad, 2025). While effective, this has led to a proliferation of abstraction-specific rules. Yet, all these methods operate over the same underlying object: the class of graphs compatible with the abstraction. An alternative line of work could seek to construct a transformed graph on which standard d-separation can be directly applied. This is feasible, for example, when the union of all compatible graphs is itself compatible — but such cases are rare, especially when the abstraction introduces cycles. Our approach adopts a hybrid strategy. Rather than relying solely on path-based d-separation, we define a structure-based criterion that captures all necessary information for assessing separation. These structures avoid pitfalls such as reattaching colliders into conditioned paths and are simple to construct — typically by tracing backward along directed edges from root nodes. Crucially, our criterion remains tightly aligned with standard d-separation: every d-connecting path can define such a structure, and every connecting structure contains a d-connecting path. To support identification under abstraction, we introduce a two-step method grounded in a tractable search space. First, we derive an extended graph from the abstraction that, while not necessarily compatible, conservatively includes all potentially connecting structures. This enables efficient exploration using standard graph-traversal techniques. Second, a lightweight compatibility test is applied to filter out invalid structures, *i.e.*, those which do not correspond to any graph in the compatible class.

Specifically, we make the following contributions:

1. We extend the framework of Anand et al. (2023) by removing the assumption of partition admissibility, thereby broadening its applicability to a wider range of causal abstractions.

2. We reformulate the d-separation criterion in an ADMG using a structure-based separation criterion that remains faithful to classical d-separation.

3. We introduce a calculus which is sound and atomically complete.

4. We further show that any cluster can be reduced to a cluster of limited size, leading to efficient calculus rules.

The remainder of the paper is structured as follows: Section 1 introduces the main notions while Section 2 presents our main result regarding sound and atomically complete calculus; Section 3 presents an efficient way to look at causal abstractions based on clusters; lastly, Section 4 discusses some extensions of our work while Section 5 concludes the paper. All proofs are provided in the Technical Appendices.

1 Preliminaries

We follow the notations of Pearl (2009). A single variable is denoted by an uppercase letter X and its realized value by a small letter x. A calligraphic uppercase letter $\mathcal{X}$ denotes a set.

Graphs. We denote by $\mathrm{Anc}(\mathcal{X}, \mathcal{G})$ and $\mathrm{Desc}(\mathcal{X}, \mathcal{G})$ the sets of ancestors and descendants of $\mathcal{X}$ in the graph $\mathcal{G}$, respectively. We denote by $\mathrm{Root}(\mathcal{G})$ the set of roots of $\mathcal{G}$, i.e., the vertices that have no child in $\mathcal{G}$. A vertex V is said to be *active* on a path relative to $\mathcal{Z}$ if 1) V is a collider and V or any of its descendants are in $\mathcal{Z}$ or 2) V is a non-collider and is not in $\mathcal{Z}$. A path π is said to be *active* given (or conditioned on) $\mathcal{Z}$ if every vertex on π is active relative to $\mathcal{Z}$. Otherwise, π is said to be *inactive* given $\mathcal{Z}$. Given a graph $\mathcal{G}$, the sets $\mathcal{X}$ and $\mathcal{Y}$ are said to be d-separated by $\mathcal{Z}$ if every path between $\mathcal{X}$ and $\mathcal{Y}$ is inactive given $\mathcal{Z}$. We denote this by $\mathcal{X} \perp\!\!\!\perp_{\mathcal{G}} \mathcal{Y} \mid \mathcal{Z}$. Otherwise, $\mathcal{X}$ and $\mathcal{Y}$ are d-connected given $\mathcal{Z}$, which we denote by $\mathcal{X} \not\perp\!\!\!\perp_{\mathcal{G}} \mathcal{Y} \mid \mathcal{Z}$. The mutilated graph $\mathcal{G}_{\overline{\mathcal{X}}\underline{\mathcal{Z}}}$ is the result of removing from a graph $\mathcal{G}$ edges with an arrowhead into $\mathcal{X}$ (e.g., $A \to \mathcal{X}$, $A \leftrightarrow \mathcal{X}$), and edges with a tail from $\mathcal{Z}$ (e.g., $A \leftarrow \mathcal{Z}$). Let π be a path in a graph $\mathcal{G}$ and let A and B be two nodes of π. We denote by $\pi_{[A,B]}$, the subpath of π between A and B. For two graphs $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1)$ and $\mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$, the union is $\mathcal{G}_1 \cup \mathcal{G}_2 \coloneqq (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{E}_1 \cup \mathcal{E}_2)$.

Structural Causal Models. Formally, a Structural Causal Model (SCM) $\mathcal{M}$ is a 4-tuple $\langle \mathcal{U}, \mathcal{V}, \mathcal{F}, P(\mathcal{U}) \rangle$, where $\mathcal{U}$ is a set of exogenous (latent) mutually independent variables and $\mathcal{V}$ is a set of endogenous (measured) variables. $\mathcal{F}$ is a collection of functions $\{f_i\}_{i=1}^{|\mathcal{V}|}$ such that each endogenous variable $V_i \in \mathcal{V}$ is a function $f_i \in \mathcal{F}$ of $\mathcal{U}_i \cup \mathrm{Pa}(V_i)$, where $\mathcal{U}_i \subseteq \mathcal{U}$ and $\mathrm{Pa}(V_i) \subseteq \mathcal{V} \setminus V_i$. The uncertainty is encoded through a probability distribution over the exogenous variables, $P(\mathcal{U})$. Each SCM $\mathcal{M}$ induces a directed acyclic graph (DAG) with bidirected edges – or an acyclic directed mixed graph (ADMG) – $G(\mathcal{V}, \mathcal{E} = (\mathcal{E}_D, \mathcal{E}_B))$, known as a *causal diagram*, that encodes the structural relations among $\mathcal{V} \cup \mathcal{U}$, where every $V_i \in \mathcal{V}$ is a vertex. We potentially distinguish edges $\mathcal{E}$ into directed edges $\mathcal{E}_D$, which connect each variable $V_i \in \mathcal{V}$ to its parents $V_j \in \mathrm{Pa}(V_i)$ as $(V_j \to V_i)$, and bidirected edges, which appear as dashed edges $(V_j \leftrightarrow V_i)$ between variables $V_i, V_j \in \mathcal{V}$ that share a common exogenous parent, i.e., such that $\mathcal{U}_i \cap \mathcal{U}_j \neq \emptyset$. Performing an intervention $\mathcal{X}{=}x$ is represented through the do-operator, $do(\mathcal{X}{=}x)$, which represents the operation of fixing a set $\mathcal{X}$ to a constant x, and induces a submodel $\mathcal{M}_{\mathcal{X}}$, which is $\mathcal{M}$ with f_X replaced to x for every $X \in \mathcal{X}$. The post-interventional distribution induced by $\mathcal{M}_{\mathcal{X}}$ is denoted by $P(\mathcal{V} \setminus \mathcal{X} | do(\mathcal{X}))$.

Cluster-DAGs In this study, we further develop the Cluster-DAG framework introduced in Anand et al. (2023). The individual variables, called micro-variables and denoted by $\mathcal{V}^m$, are grouped into clusters forming a partition $\mathcal{V}^C$, where each cluster contains one or more micro-variables.

Definition 1 (Cluster-DAG)**.** *Let $\mathcal{G}^m = (\mathcal{V}^m, \mathcal{E}^m)$ be an ADMG and let $\mathcal{V}^C$ be a partition of $\mathcal{V}^m$. We construct the mixed graph $\mathcal{G}^C = (\mathcal{V}^C, \mathcal{E}^C)$, possibly with self-loops and cycles, by defining $\mathcal{E}^C$ as follows. For all clusters $V, W \in \mathcal{V}^C$:*

- *$V \to W$ is in $\mathcal{E}^C$* if and only if *there exists V_v in V and W_w in W such that $V_v \to W_w$ in $\mathcal{G}^m$.*
- *$V \leftrightarrow W$ is in $\mathcal{E}^C$* if and only if *there exists V_v in V and W_w in W such that $V_v \leftrightarrow W_w$ in $\mathcal{G}^m$.*

We say that $\mathcal{G}^m$ and $\mathcal{G}^C$ are compatible.

A key novelty of our approach is the allowance of cycles at the cluster level, in contrast to Anand et al. (2023), which restricts the cluster graph to be acyclic. We nonetheless retain the term "Cluster-DAG" (C-DAG) to emphasize that the underlying graph on micro-variables remain acyclic. We denote a node in $\mathcal{G}^C$ by V^C, and its corresponding set of micro-variables in a compatible graph $\mathcal{G}^m$ by $V^m = \{V_1, \cdots, V_{\#V}\}$ where the indices follow the topological ordering induced by $\mathcal{G}^m$. We will use the same notations for any intersection or union of clusters. The cardinality of each cluster is displayed in the upper left corner of its corresponding node, as represented in Figure 1a.

A C-DAG $\mathcal{G}^C$ is, by definition, derived from an ADMG over the micro-variable set $\mathcal{V}^m$; however, in practical applications the true causal diagram on $\mathcal{V}^m$ is typically unknown. Then, we are interested in all the ADMGs compatible with $\mathcal{G}^C$.

Definition 2 (Class of Compatible Graphs)**.** *Let $\mathcal{V}^C$ be a partition of $\mathcal{V}^m$, and $\mathcal{G}^C$ be a mixed graph on $\mathcal{V}^C$. We denote $C(\mathcal{G}^C) := \{\mathcal{G}^m \mid \mathcal{G}^m$ is compatible with $\mathcal{G}^C\}$ the equivalence class[1] of graphs compatible with $\mathcal{G}^C$.*

A C-DAG is a valid causal abstraction of any underlying causal diagram on micro-variables if and only if it contains no directed cycle composed entirely of singleton clusters, which ensure the existence of at least one ADMG compatible with the mixed graph (see Proposition 4 in Appendix).

2 A Causal Calculus for Cyclic C-DAGs

We now introduce an atomically complete calculus for reasoning about cluster queries in C-DAGs. For each rule of Pearl's calculus, Theorem 2 gives a graphical criterion that is sound and complete (Theorem 3): if the separation criterion for a given rule applies, then the rule is valid in all compatible graphs; if not, then there is at least one compatible graph in which the rule fails. Our work is constructive: if such a graph exists, then our criterion enables its construction. To establish Theorems 2 and 3, we first introduce the concept of *structure of interest*, then describe the associated graphs needed to define efficiently our calculus, and finally define the corresponding calculus in the third subsection.

2.1 Structure of interest

Demonstrating that a rule of Pearl's calculus fails on a compatible graph (at least one) requires exhibiting a graph on micro-variables in which the corresponding d-separation fails. Concretely, this involves the three following steps on the micro-variables, for a given C-DAG: (i) find a path connecting variables; (ii) for each collider on that path, provide a directed path to a conditioning variable; and (iii) ensure all these paths coexist in a single compatible graph. While (i)–(ii) could be decided via a graphical test on the C-DAG, step (iii) is nontrivial since paths may conflict and form cycles. To avoid this, we directly look at *structures of interest*, which correspond to paths to which we add for each collider a directed path to a conditioning variable. Thus, we only need to test that this structure of interest connects two sets of variables. Definitions 3 and 4 refine and formalize this intuition.

Definition 3. *A* structure of interest *σ is an ADMG, with a single connected component, in which each node V satisfies the following property:*

- *V has at most one outgoing arrow, or,*
- *V has two outgoing arrows but no incoming arrow.*

In an ADMG, executing a breadth-first search from the root set against the arrow orientation produces a subgraph that, by construction, satisfies the first condition of Definition 3. Moreover, by enforcing the second condition at each exploration step, one obtains an efficient procedure for constructing the desired structures of interest within the ADMG.

Figure 5 in Appendix shows how arrows look like around a vertex in a structure of interest. We draw in Figure 1b in bold an example of a structure of interest in a graph on micro-variables.

Definition 4. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a mixed graph. Let $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ be pairwise disjoint subsets of $\mathcal{V}$. We say that a structure of interest $\sigma \subseteq \mathcal{G}$* connects *$\mathcal{X}$ and $\mathcal{Y}$ under $\mathcal{Z}$ and we write $\mathcal{X} \not\perp\!\!\!\perp_\sigma \mathcal{Y} \mid \mathcal{Z}$ if the following conditions hold:*

- *$\mathcal{X} \cap \sigma \neq \emptyset$ and $\mathcal{Y} \cap \sigma \neq \emptyset$* *($\sigma$ connects $\mathcal{X}$ and $\mathcal{Y}$)*
- $\mathrm{Root}(\sigma) \subseteq \mathcal{Z} \cup \mathcal{X} \cup \mathcal{Y}$ *and* *(all vertices of σ are ancestors of $\mathcal{Z} \cup \mathcal{X} \cup \mathcal{Y}$)*
- $(\sigma \setminus \mathrm{Root}(\sigma)) \cap \mathcal{Z} = \emptyset$ *(neither chains nor forks of σ are in $\mathcal{Z}$)*

Example 1. *Let us consider the graph $\mathcal{G}^m$ and the structure of interest σ^m depicted in Figure 1b. The roots of σ^m are $\mathrm{Root}(\sigma^m) = \{D_1, C_1, Y_1\}$. According to Definition 4, σ^m connects X_1 and Y_1 under $C^m \cup D^m = \{C_1, C_2, D_1\}$. Indeed, σ^m contains the path $\pi^m = \langle X_1, A_1, B_1, C_1, Y_1 \rangle$ which d-connects X_1 and Y_1 under $C^m \cup D^m$ in $\mathcal{G}^m$.*

[1] If we denote $\phi(\mathcal{G}^m; \mathcal{V}^C)$ the cluster-DAG obtained from $\mathcal{G}^m$ via Definition 1, the equivalence relation is given by $\mathcal{G}_1^m \sim_{\mathcal{V}^C} \mathcal{G}_2^m \Leftrightarrow \phi(\mathcal{G}_1^m; \mathcal{V}^C) = \phi(\mathcal{G}_2^m; \mathcal{V}^C)$.

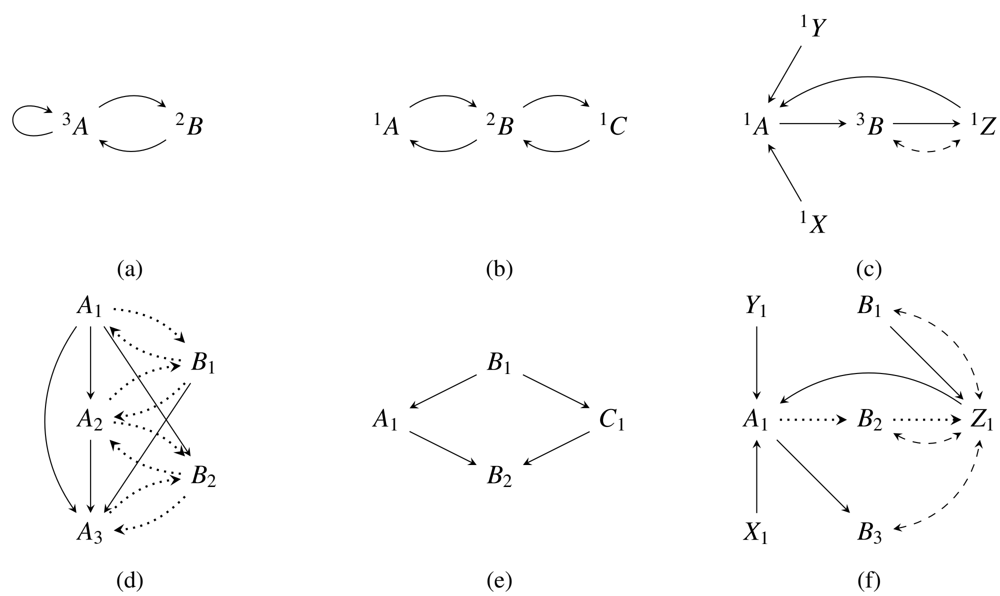


Figure 2: On the first row (Figures 2a, 2b and 2c), three examples of C-DAG are given. On the second row (respectively, Figures 2d, 2e and 2f), we represent the corresponding unfolded and canonical compatible graphs. The plain and dashed arrows corresponds to $\mathcal{G}^m_{\text{can}}$, whereas the dotted arrows represent the "eligible" arrows. Lemma 2 and Figure 2e show that there is no graph compatible with the C-DAG depicted in Figure 2b such that A_1 and C_1 are connected by a directed path. Similarly, Proposition 3 and Figure 2f show that there is no graph $\mathcal{G}^m$ compatible with the C-DAG depicted in Figure 2c such that $A_1 \in \text{Anc}(Z_1, \mathcal{G}^m)$.

We remark that a path is always a structure of interest, but it may not be a connecting structure of interest. Theorem 1 shows that there exists a d-connecting path if and only if there exists a connecting structure of interest.

Theorem 1 (D-connection with structures of interests)**.** *Let $\mathcal{G}$ be an ADMG. Let $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ be pairwise disjoint subsets of nodes of $\mathcal{G}$. The following properties are equivalent:*

1. $\mathcal{X} \not\perp\!\!\!\perp_{\mathcal{G}} \mathcal{Y} \mid \mathcal{Z}$.
2. *$\mathcal{G}$ contains a structure of interest σ such that $\mathcal{X} \not\perp\!\!\!\perp_{\sigma} \mathcal{Y} \mid \mathcal{Z}$.*

Theorem 1 reduces the problem of determining whether a rule from Pearl's calculus fails for a given C-DAG to the search for a compatible graph which contains a *structure of interest* that violates the corresponding d-separation.

2.2 Associated graphs

Enumerating all compatible graphs is generally infeasible due to their potentially large number. To address this challenge, we define two mixed graphs. The canonical compatible graph (Definition 5) allows efficient verification of whether given structures of interest exist in some compatible graph. The unfolded graph (Definition 6) aggregates all structures of interest present in at least one compatible graph. To look for a structure of interest, we first check in the unfolded graph. However, since it may include spurious structures that do not correspond to any actual compatible graph, we afterward filter out these spurious structures from the canonical compatible graph.

Definition 5 (Canonical Compatible Graph)**.** *Let $\mathcal{G}^C$ be a C-DAG. Its corresponding* canonical compatible graph *is the ADMG $\mathcal{G}^m_{can} = (\mathcal{V}^m_{can}, \mathcal{E}^m_{can})$, where the set of nodes is $\mathcal{V}^m_{can} := \mathcal{V}^m$, and the set of edges is constructed by the following procedure:*

1. *For all dashed-bidirected-arrows $V^C \dashleftarrow\!\!\dashrightarrow W^C$ in $\mathcal{G}^C$, add the dashed-bidirected-arrows $V_v \dashleftarrow\!\!\dashrightarrow W_w$ for all $v, w \in \{1, \cdots, \#V^C\} \times \{1, \cdots, \#W^C\}$ such that $V_v \neq W_w$.*

2. *For all self-loop* $\circlearrowright V^C$, *add the arrow* $V_i \to V_j$ *for all* $i, j \in \{1, \cdots, \#V^C\}^2$ *such that* $i < j$.

3. *For all arrows* $V^C \to W^C$, *with* $V^C \neq W^C$, *add the arrow* $V_1 \to W_{\#W^C}$.

Examples of canonical compatible graphs are given in Figure 2. As stated in Proposition 1, the canonical compatible graph is itself compatible and canonical in the sense that it can be added to any compatible graph without violating compatibility.

Proposition 1. *Let* $\mathcal{G}^C$ *be a C-DAG and* $\mathcal{G}^m_{can}$ *be its corresponding canonical compatible graph. Then, the following properties hold:*

1. $\mathcal{G}^m_{can} \in C(\mathcal{G}^C)$.

2. *For all* $\mathcal{G}^m \in C(\mathcal{G}^C)$, $\mathcal{G}^m_{can} \cup \mathcal{G}^m \in C(\mathcal{G}^C)$.

Consequently, if a structure of interest exists in some compatible graph, it must also coexist with the canonical compatible graph, meaning that its addition to the canonical compatible graph does not create cycles[2].

We now introduce the unfolded graph.

Definition 6 (Unfolded graph)**.** *Let* $\mathcal{G}^C$ *be a C-DAG and* $\mathcal{G}^m_{can} = (\mathcal{V}^m_{can}, \mathcal{E}^m_{can})$ *be its corresponding canonical compatible graph. Its corresponding* unfolded graph *is* $\mathcal{G}^m_u = (\mathcal{V}_u, \mathcal{E}_u)$, *the mixed graph defined by the following procedure:*

- $\mathcal{V}_u := \mathcal{V}^m$.
- *Let us consider the following set:*

$$\mathcal{E}_{eligible} := \left\{ V_v \to W_w \;\middle|\; \begin{cases} V^C \to W^C \subseteq \mathcal{G}^C, \text{ and,} \\ \mathcal{G}^m_{can} \cup \{V_v \to W_w\} \text{ is acyclic.} \end{cases} \right\}$$

Then $\mathcal{E}_u := \mathcal{E}^m_{can} \cup \mathcal{E}_{eligible}$.

Examples of unfolded graphs are given in Figure 2. As shown in Proposition 2, the unfolded graph is a supergraph of any compatible graph. Therefore, if there exists a compatible graph containing a structure of interest σ^m, it follows that σ^m must also appear in the unfolded graph. This implies that it is no longer necessary to enumerate all compatible graphs in the search for a structure of interest; instead, it suffices to search within the unfolded graph alone. Figure 2 illustrates how the unfolded graph can be used to demonstrate the non-existence of certain structures within the set of compatible graphs.

Proposition 2. *Let* $\mathcal{G}^C$ *be a C-DAG and* $\mathcal{G}^m_u$ *be its corresponding unfolded graph. Then, any compatible graph* $\mathcal{G}^m$ *is a subgraph of* $\mathcal{G}^m_u$ *up to a permutation of indices in each cluster.*

The unfolded graph defines the search space for the structures of interest, while the canonical compatible graph, as stated in Proposition 3, ensures that these structures can indeed be realized within a compatible graph.

Proposition 3. *Let* $\mathcal{G}^C$ *be a C-DAG and* $\mathcal{G}^m_u$ *be its corresponding unfolded graph. Let* σ^m *be a structure of interest in* $\mathcal{G}^m_u$. *If* $\mathcal{G}^m_{can} \cup \sigma^m$ *is acyclic, then* $\mathcal{G}^m_{can} \cup \sigma^m$ *is compatible with* $\mathcal{G}^C$.

By keeping these two notions distinct, we are able to apply mutilations directly on the unfolded graph without restricting the overall class of compatible graphs.

2.3 Calculus

It is important to note that mutilating all graphs compatible with a C-DAG generally yields a strictly smaller set of graphs than the set of graphs compatible with the mutilated C-DAG, as illustrated in Example 2.[3] This means that one cannot do the do-calculus on the class of graphs defined by the mutilated C-DAG.

[2]Let $\sigma^m \subseteq \mathcal{G}^m$. Since $\mathcal{G}^m \cup \mathcal{G}^m_{\text{can}}$ is acyclic, $\sigma^m \cup \mathcal{G}^m_{\text{can}}$ is also acyclic.

[3]A similar phenomenon was observed by Zhang (2008) in the context of ancestral graphs.

Example 2. *Let us consider the C-DAG $\mathcal{G}^C := {}^2A \rightleftarrows {}^1B$. We have the following identities:*

- $\left\{\mathcal{G}^m{}_{\underline{B_1}} \mid \mathcal{G}^m \in C(\mathcal{G}^C)\right\} = \left\{ \begin{array}{ccc} A_1 & \searrow & \\ & & B_1 \\ A_2 & & \end{array}, \quad \begin{array}{ccc} A_1 & & \\ & & B_1 \\ A_2 & \nearrow & \end{array} \right\}$

- $C\left(\mathcal{G}^C{}_{\underline{B^C}}\right) = \left\{ \begin{array}{ccc} A_1 & \searrow & \\ & & B_1 \\ A_2 & & \end{array}, \quad \begin{array}{ccc} A_1 & & \\ & & B_1 \\ A_2 & \nearrow & \end{array}, \quad \begin{array}{ccc} A_1 & \searrow & \\ & & B_1 \\ A_2 & \nearrow & \end{array} \right\}$

And then, $\left\{\mathcal{G}^m{}_{\underline{B_1}} \mid \mathcal{G}^m \in C(\mathcal{G}^C)\right\} \subsetneq C\left(\mathcal{G}^C{}_{\underline{B^C}}\right)$.

However, thanks to the unfolded graph and the canonical compatible graph, the rules of do-calculus for cluster queries can be encoded in a sound and complete manner. This principle is formally established in Theorems 2 and 3. An illustrative application of Theorem 2 is provided in Example 3 (a complementary example is provided in Figure 8 in Appendix).

Theorem 2 (Calculus)**.** *Let $\mathcal{G}^C$ be a C-DAG and let $\mathcal{G}^m_u$ be its corresponding unfolded graph. Let $\mathcal{X}^C, \mathcal{Y}^C, \mathcal{Z}^C, \mathcal{W}^C$ be pairwise distinct subsets of nodes. Then for any density P compatible with any compatible graph, the following rules apply:*

R1. $P(\mathbf{y^m} \mid do(\mathbf{w^m}), \mathbf{x^m}, \mathbf{z^m}) = P(\mathbf{y^m} \mid do(\mathbf{w^m}), \mathbf{z^m})$ *if $\mathcal{G}^m_{u\,\overline{\mathcal{W}^m}}$ does not contain a structure of interest σ^m such that $\mathcal{X}^m \not\perp\!\!\!\perp_{\sigma^m} \mathcal{Y}^m \mid \mathcal{W}^m, \mathcal{Z}^m$ and $\mathcal{G}^m_{can} \cup \sigma^m$ is acyclic.*

R2. $P(\mathbf{y^m} \mid do(\mathbf{w^m}), do(\mathbf{x^m}), \mathbf{z^m}) = P(\mathbf{y^m} \mid do(\mathbf{w^m}), \mathbf{x^m}, \mathbf{z^m})$ *if $\mathcal{G}^m_{u\,\overline{\mathcal{W}^m},\underline{\mathcal{X}^m}}$ does not contain a structure of interest σ^m such that $\mathcal{X}^m \not\perp\!\!\!\perp_{\sigma^m} \mathcal{Y}^m \mid \mathcal{W}^m, \mathcal{Z}^m$ and $\mathcal{G}^m_{can} \cup \sigma^m$ is acyclic.*

R3. $P(\mathbf{y^m} \mid do(\mathbf{w^m}), do(\mathbf{x^m}), \mathbf{z^m}) = P(\mathbf{y^m} \mid do(\mathbf{w^m}), \mathbf{z^m})$ *if $\mathcal{G}^m_{u\,\overline{\mathcal{W}^m}}$ does not contain a structure of interest σ^m such that $\mathcal{X}^m \not\perp\!\!\!\perp_{\sigma^m} \mathcal{Y}^m \mid \mathcal{W}^m, \mathcal{Z}^m$, $\mathcal{G}^m_{can} \cup \sigma^m$ is acyclic and* $\mathrm{Root}(\sigma^m) \subseteq (\mathcal{W}^m \cup \mathcal{Z}^m) \cup \mathcal{Y}^m$.

The first two rules of Theorem 2 are very similar to the first two rules of Pearl's do-calculus. In contrast, the third rule of Pearl's do-calculus requires verifying the d-separation condition $\mathcal{Y}^m \perp\!\!\!\perp_{\mathcal{G}^m{}_{\overline{\mathcal{W}^m},\overline{\mathcal{X}^m(\mathcal{Z}^m)}}} X^m \mid \mathcal{W}^m, \mathcal{Z}^m$, in all compatible graph $\mathcal{G}^m$, where $\mathcal{X}^m(\mathcal{Z}^m) = X^m \setminus \mathrm{Anc}(\mathcal{Z}^m, \mathcal{G}^m{}_{\overline{\mathcal{W}^m}})$. Since $\mathcal{X}^m(\mathcal{Z}^m)$ is not, in general, a union of clusters, the associated mutilation depends on the particular graph $\mathcal{G}^m$. As a result, it is not possible to directly apply this mutilation to the unfolded graph to derive an atomically complete criterion for Rule 3.

Nonetheless, if Rule 3 does not hold in some compatible graph $\mathcal{G}^m$, then there exists a structure of interest between $\mathcal{Y}^m$ and X^m in $\mathcal{G}^m{}_{\overline{\mathcal{W}^m},\overline{\mathcal{X}^m(\mathcal{Z}^m)}}$. If this structure includes a root $X_x \in \mathcal{X}^m$, then X_x must be an ancestor of some $Z_z \in \mathcal{Z}^m$ in the mutilated graph. In such a case, we can augment the structure of interest by explicitly adding the directed path from X_x to $\mathcal{Z}^m$, resulting in a new structure whose roots lie outside $\mathcal{X}^m$. This constructive process of eliminating roots from $\mathcal{X}^m$ is behind the third rule in Theorem 2.

For all the rules, structures of interest are sought in the unfolded graph. The canonical compatible graph is then used to ensure that the identified structure of interest actually exists in a compatible graph.

All the calculus rules given in Theorem 2 are atomatically complete, as stated by Theorem 3.

Theorem 3 (Atomic completeness)**.** *The calculus in Theorem 2 is atomically complete i.e. if the rule does not hold given a C-DAG, then there exists a compatible graph in which the corresponding rule in Pearl's calculus fails.*

Example 3. *Let us consider $\mathcal{G}^C$, the C-DAG depicted in Figure 2c. Figure 2f displays the corresponding unfolded graph and canonical compatible graphs. The plain and dashed arrows represent $\mathcal{G}^m_{can}$, while the dotted arrows denote the "eligible" edges. According to the second rule of Theorem 2, we have $P(\mathbf{y^m} \mid do(\mathbf{z^m})) = P(\mathbf{y^m} \mid \mathbf{z^m})$. Indeed any structure of interest σ^m which connects Y^m and Z^m under $\emptyset$ contains the arrows $A_1 \to B_2 \to Z_1$. Since $\mathcal{G}^m_{can}$ contains $Z_1 \to A_1$, we know that $\mathcal{G}^m_{can} \cup \sigma^m$ contains a cycle. Therefore, $\mathcal{G}^m_{u\,\overline{\emptyset},\underline{\mathcal{Z}^m}}$ does not contain a structure of interest σ^m that connects Y^m and Z^m under $\emptyset$ such that $\mathcal{G}^m_{can} \cup \sigma^m$ is acyclic.*

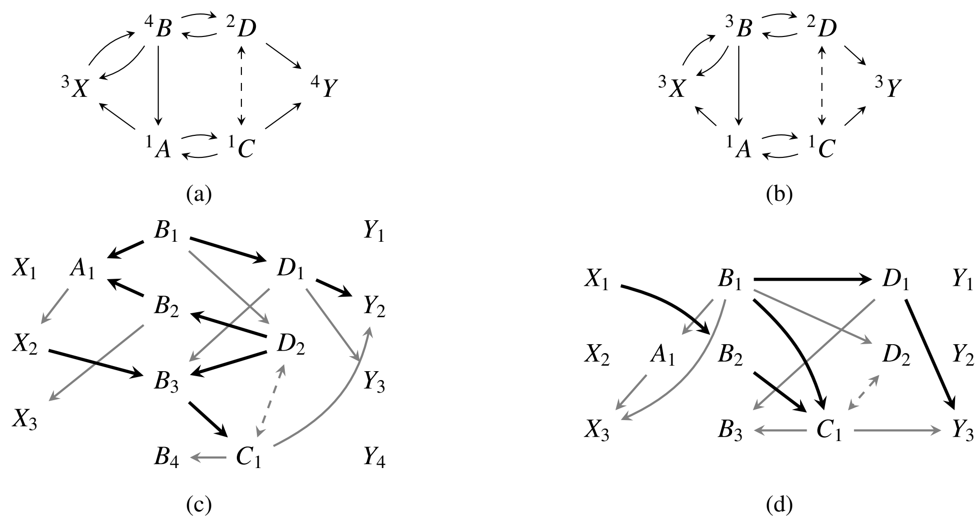


Figure 3: Top: $\mathcal{G}^C$ (left) and $\mathcal{G}^C_{\leq 3}$ (right). Bottom: a graph compatible with $\mathcal{G}^C$ (left) and a graph compatible with $\mathcal{G}^C_{\leq 3}$ (right). In Figure 3c, the arrows in **bold black** represent a structure of interest that connects X^m and Y^m under $C^m \cup A^m$. In Figure 3d, the arrows in **bold black** represent the structure of interest that connects $X^m_{\leq 3}$ and $Y^m_{\leq 3}$ under $C^m_{\leq 3} \cup A^m_{\leq 3}$, which is obtained by applying the strategy used in the proof of Theorem 4.

3 Computational efficiency: reducing clusters of large size to 3 nodes

While Theorem 2 provides a sound and atomically complete calculus, its direct application may be impractical for large clusters, as computing $\mathcal{G}^m_{\mathrm{u}}$ becomes intractable due to a combinatorial explosion in the number of edges. To address this, we associate with any C-DAG $\mathcal{G}^C$ a simplified C-DAG $\mathcal{G}^C_{\leq 3}$ on the same set of nodes (but with different cardinals), where each cluster of size greater than 3 is reduced to size 3. The set of edges of $\mathcal{G}^C_{\leq 3}$ is the set of edges of $\mathcal{G}^C$. The key difference is that $C(\mathcal{G}^C_{\leq 3}) \neq C(\mathcal{G}^C)$, because the graphs compatible with $\mathcal{G}^C_{\leq 3}$ contain fewer nodes and different edges than the graphs compatible with $\mathcal{G}^C$. We illustrate this in Figure 3. Notably, Theorem 4 shows that applying the calculus on $\mathcal{G}^C$ or on $\mathcal{G}^C_{\leq 3}$ leads to the same results.

Theorem 4 (Infinity is at most three)**.** *Let $\mathcal{G}^C$ be a C-DAG and $\mathcal{G}^C_{\leq 3}$ be the corresponding C-DAG where all clusters of size greater than 3 are reduced to size 3. Let $\mathcal{W}^C$, $\mathcal{X}^C$, $\mathcal{Y}^C$ and $\mathcal{Z}^C$ be pairwise disjoint subsets of nodes. For $i \in \{1, 2, 3\}$, let $R_i(\mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{Z})$ be the i^{th} rule of Pearl's Calculus applied to $(\mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{Z})$, and say it "does not holds in $\mathcal{G}$" whenever its associated d-separation condition in the associated mutilated graph is not satisfied. The following propositions are equivalent:*

1. *There exists $\mathcal{G}^m \in C(\mathcal{G}^C)$ in which $R_i(\mathcal{W}^m, \mathcal{X}^m, \mathcal{Y}^m, \mathcal{Z}^m)$ does not hold.*

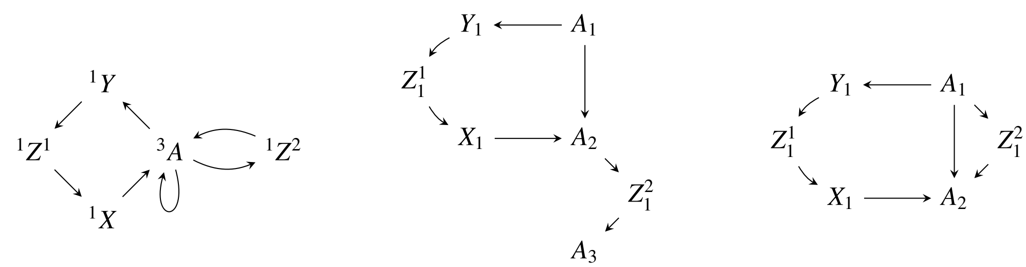
2. *There exists $\mathcal{G}^m_{\leq 3} \in C(\mathcal{G}^C_{\leq 3})$ in which $R_i(\mathcal{W}^m_{\leq 3}, \mathcal{X}^m_{\leq 3}, \mathcal{Y}^m_{\leq 3}, \mathcal{Z}^m_{\leq 3})$ does not hold.*

Where $\mathcal{W}^m_{\leq 3}$, $\mathcal{X}^m_{\leq 3}$, $\mathcal{Y}^m_{\leq 3}$ and $\mathcal{Z}^m_{\leq 3}$ are the sets of nodes corresponding to $\mathcal{W}^C$, $\mathcal{X}^C$, $\mathcal{Y}^C$ and $\mathcal{Z}^C$ in $\mathcal{G}^C_{\leq 3}$

Figure 3 illustrates Theorem 4 by showing how a graph $\mathcal{G}^m \in C(\mathcal{G}^C)$, in which a given d-separation does not hold, is transformed into $\mathcal{G}^m_{\leq 3} \in C(\mathcal{G}^C_{\leq 3})$, in which the corresponding d-separation does not hold as well. The figure highlights how this transformation impacts the structure of interest that violates the d-separation by omitting all irrelevant dependencies (see Figure 3d).

Theorem 4 shows that reducing cluster size to at most three preserves all relevant dependencies. This bound is tight: in some C-DAGs, any further reduction of the size of the clusters (by removing more nodes) would necessarily lose causal information. Example 4 illustrates such a case.

Example 4 (Infinity is at least three)**.** *Let $\mathcal{G}^C$ be the C-DAG defined by Figure 4a. Let $\mathcal{X}^C = \{X^C\}$, $\mathcal{Y}^C = \{Y^C\}$ and $\mathcal{Z}^C = \{Z^{1^C}, Z^{2^C}\}$. There exists a compatible graph $\mathcal{G}^m$ in which $\mathcal{X}^m \not\perp\!\!\!\perp_{\mathcal{G}^m} \mathcal{Y}^m \mid \mathcal{Z}^m$: we displayed it in Figure 4b.*



(a) $\mathcal{G}^C$ (b) $\mathcal{G}^m \in C(\mathcal{G}^C)$ (c) $C(\mathcal{G}^C_{\leq 2}) = \left\{\mathcal{G}^m_{\leq 2}\right\}$

Figure 4: Figure 4a depicts a cluster $\mathcal{G}^C$. Figure 4b illustrates a graph compatible with $\mathcal{G}^C$. Figure 4c illustrates the unique graph compatible with $\mathcal{G}^C_{\leq 2}$.

There is only one graph compatible with $\mathcal{G}^C_{\leq 2}$: $C(\mathcal{G}^C_{\leq 2}) = \{\mathcal{G}^m_{\leq 2}\}$ (the one displayed in Figure 4c). Moreover, in $\mathcal{G}^m_{\leq 2}$, the corresponding dependence does not hold.

4 Discussion

On Cluster d-Separation Theorem 2 provides a sound and atomically complete calculus for causal identification. In addition, our results offer a sound and atomically complete solution to the problem of cluster d-separation. Specifically, the criterion for cluster d-separation corresponds to the first rule of Theorem 2 when taking $\mathcal{W}^c = \emptyset$. Furthermore, Theorem 6 in Appendix establishes cluster d-separation under cluster-level mutilations. Our results thus encompass both the first (association) and second (intervention) rungs of Pearl's ladder of causation.

Recovering the results on standard (acyclic) C-DAGs Theorem 2 recovers the results of Anand et al. (2023). When a C-DAG $\mathcal{G}^C$ is acyclic, its corresponding unfolded graph $\mathcal{G}^m_{\mathrm{u}}$ is also acyclic. As a result, $\mathcal{G}^m_{\mathrm{u}}$ is a compatible graph and standard d-separation is both sound and complete in $\mathcal{G}^m_{\mathrm{u}}$.

A 2-step strategy In our study, the unfolded graph defines the search space for the structures of interest, while the canonical compatible graph ensures that these structures can indeed be realized within a compatible graph. By keeping these two notions distinct, we are able to apply mutilations directly on the unfolded graph without restricting the overall class of compatible graphs. This approach resolves the non-commutativity between mutilation and enumeration of compatible graphs, since performing mutilation before enumeration generally produces a strictly larger set of graphs than enumerating first before mutilating. A similar phenomenon was observed in Zhang (2008) in the context of ancestral graphs.

5 Conclusion

We have addressed in this study the problem of identification in causal abstractions based on arbitrary clusterings of variables in ADMGs, extending the framework considered in Anand et al. (2023) to abstract graphs which potentially contain self-loops and cycles between clusters. This extension is important in practice as the structure induced on clusters of variables in a given ADMG is likely to contain cycles between clusters. In this framework, we have first reformulated the notion of d-separation in an ADMG using structures of interest, a reformulation which remains faithful to the original formulation as finding a structure of interest is sufficient to d-connect two sets, and then provided a causal calculus which is both sound and atomically complete. We further showed that any cluster can be reduced to a cluster of limited size, leading to efficient calculus rules.

In the future, we aim to establish the global completeness of the calculus, as it is currently only atomically complete. We also plan to extend this work by considering micro-level interventions, *i.e.*, interventions on individual variables rather than on clusters of variables, when only the C-DAG is known.

References

Anand, T. V., Ribeiro, A. H., Tian, J., and Bareinboim, E. (2023). Causal Effect Identification in Cluster DAGs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(10):12172–12179.

Assaad, C. K., Devijver, E., and Gaussier, E. (2022). Discovery of extended summary graphs in time series. In Cussens, J. and Zhang, K., editors, *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 96–106. PMLR.

Beckers, S. and Halpern, J. Y. (2019). Abstracting causal models. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI Press.

Chalupka, K., Eberhardt, F., and Perona, P. (2016). Multi-level cause-effect systems. In Gretton, A. and Robert, C. C., editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 361–369, Cadiz, Spain. PMLR.

Chalupka, K., Perona, P., and Eberhardt, F. (2015). Visual causal feature learning. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, page 181–190, Arlington, Virginia, USA. AUAI Press.

Ferreira, S. and Assaad, C. K. (2024). Identifiability of direct effects from summary causal graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(18):20387–20394.

Ferreira, S. and Assaad, C. K. (2025). Identifying macro conditional independencies and macro total effects in summary causal graphs with latent confounding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(25):26787–26795.

Jaber, A., Ribeiro, A., Zhang, J., and Bareinboim, E. (2022). Causal identification under markov equivalence: Calculus, algorithm, and completeness. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 3679–3690. Curran Associates, Inc.

Pearl, J. (2009). *Causality*. Cambridge University Press, Cambridge, 2 edition.

Perković, E., Textor, J., Kalisch, M., and Maathuis, M. H. (2018). Complete Graphical Characterization and Construction of Adjustment Sets in Markov Equivalence Classes of Ancestral Graphs. *Journal of Machine Learning Research*, 18:1–62.

Rubenstein, P. K., Weichwald, S., Bongers, S., Mooij, J. M., Janzing, D., Grosse-Wentrup, M., and Schölkopf, B. (2017). Causal consistency of structural equation models. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence (UAI)*, page ID 11.

Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., and Bengio, Y. (2021). Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634.

Wahl, J., Ninad, U., and Runge, J. (2023). Vector causal inference between two groups of variables. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(10):12305–12312.

Wahl, J., Ninad, U., and Runge, J. (2024). Foundations of causal discovery on groups of variables. *Journal of Causal Inference*, 12(1).

Zhang, J. (2008). Causal Reasoning with Ancestral Graphs. *Journal of Machine Learning Research*, 9:1437–1474.

Contents

A Technical Appendices and Supplementary Material

Notation 1. *Let $\mathcal{G}^C$ be a C-DAG, and let V be a cluster in $\mathcal{G}^C$. When V is seen as a node of $\mathcal{G}^C$, V will be written as V^C; but when V is seen as a set of variables of a graph $\mathcal{G}^m$ on micro-variables, V will be written as $V^m = \{V_1, \cdots, V_{\#V}\}$ where the indices follow the topological ordering induced by $\mathcal{G}^m$. We will use the same notations for any intersection or union of cluster.*

Terminolgy on paths. Let π be a path and $\mathcal{X}$ a subset of variables. We say that π intersects or encounters $\mathcal{X}$ if they share at least one common vertex. We treat paths as ordered lists of variables, which allows us to define the first and last encounter of π with $\mathcal{X}$. The first encounter is the first vertex in the order of π that also belongs to $\mathcal{X}$. Similarly, the last encounter is the last such vertex in the order of π. Let π be a path from $\mathcal{X}$ to $\mathcal{Y}$. Let A be a vertex on π. We denote by $\pi_{[\mathcal{X},A]}$ the subpath of π from its first vertex to A and $\pi_{[A,\mathcal{Y}]}$ from A to its last vertex.

A.1 Basic Properties of C-DAGs

In this section, following the notations of Perković et al. (2018), an arrow ($\leftarrow\!\!\bullet$) represents either a directed arrow ($\leftarrow$) or a dashed-bidirected arrows ($\leftrightarrow$).

Proposition 4. *Let $\mathcal{V}^C$ be a partition of $\mathcal{V}^m$. Let $\mathcal{G}^C$ be a mixed graph over $\mathcal{V}^C$ and let $C\left(\mathcal{G}^C\right)$ denote the class of graphs compatible with $\mathcal{G}^C$. Then the following propositions are equivalent:*

- $C\left(\mathcal{G}^C\right) \neq \emptyset$
- *$\mathcal{G}^C$ does not contain any cycle on clusters of size* 1.

Proof. Let us prove the two implications:

- $\Rightarrow$: If $\mathcal{G}^C$ contains the cycle ${}^1A \rightarrow \cdots \rightarrow {}^1A$, then any compatible graph would contain the cycle $A_1 \rightarrow \cdots \rightarrow A_1$, which is not allowed because compatible graphs have to be acyclic.
- $\Leftarrow$: If $\mathcal{G}^C$ does not contain any cycle on cluster of size 1. Let us construct a compatible graph. For all cluster V of size 1, we put all incoming and outgoing edges at V_1. This does not create a cycle because, otherwise, $\mathcal{G}^C$ would contain a cycle on cluster of size 1. For all other clusters V, as they are at least of size 2, we can deal with V_1 and V_2. We put all outgoing edges at V_1 and all incoming edges at V_2. This construction cannot introduce a directed cycle, since no vertex in V^m ever has both an incoming and an outgoing edge.

□

Proposition 5. *Let $\mathcal{G}^C$ be a C-DAG, $\mathcal{G}^m$ be a compatible graph with $\mathcal{G}^C$ and V^C and W^C be nodes of $\mathcal{G}^C$. If $\mathcal{G}^m$ contains two similar (same type) arrows between V^m and W^m, then removing one of these arrows create another compatible graph.*

Proof. By definition, only one arrow between V^m and W^m is necessary. □

Proposition 6. *Let $\mathcal{G}^C$ be a C-DAG, $\mathcal{G}^m$ be a compatible graph with $\mathcal{G}^C$ and V^C be a node in $\mathcal{G}^C$, i.e. a cluster. If there exists (i, j) with $i > j$ such that $\mathcal{G}^m$ contains the arrow $V_i \rightarrow W_w$, then there exists a compatible graph $\mathcal{G}^{m\prime}$ that contains the arrow $V_j \rightarrow W_w$ and not $V_i \rightarrow W_w$ if desired.*

Proof. If $i > j$, thus by the convention of Notation 1, we know that V_j is before V_i in the topological order of $\mathcal{G}^m$. Thus, V_j is not a descendant of V_i in $\mathcal{G}^m$. Thus, V_j is not a descendant of W_w in $\mathcal{G}^m$. Therefore, adding the arrow $V_j \rightarrow W_w$ into $\mathcal{G}^m$ does not create a cycle, because otherwise V_j would be a descendant of W_w in $\mathcal{G}^m$.

By Proposition 5, $V_i \rightarrow W_w$ can be thereafter removed if desired. □

Corollary 1. *Let $\mathcal{G}^C$ be a C-DAG, $\mathcal{G}^m$ be a compatible graph and V^C be a cluster. If $\mathcal{G}^m$ contains a path which contains a fork on V_i with $i > j$, then there exists a compatible graph $\mathcal{G}^{m\prime}$ which contains the same path except that the fork is on V_j.*

Proof. Apply Proposition 6 twice. □

Proposition 7. *Let $\mathcal{G}^C$ be a C-DAG, $\mathcal{G}^m$ be a compatible graph and V^C be a cluster. If there exists $i < j$ such that $\mathcal{G}^m$ contains the arrow $V_i \leftarrow\!\!\bullet W_w$, where $\leftarrow\!\!\bullet$, then there exists a compatible graph $\mathcal{G}^{m\prime}$ that contains the arrow $V_j \leftarrow\!\!\bullet W_w$ and not $V_i \leftarrow\!\!\bullet W_w$ if desired.*

Proof. $i < j$, thus by the convention of Notation 1, we know that W_w is not a descendant of V_j. Therefore, adding the arrow $V_j \leftarrow\!\!\bullet W_w$ does not create a cycle.

By Proposition 5, $V_i \leftarrow\!\!\bullet W_w$ can be removed if desired. □

A.2 Proof of Theorem 1

We introduce Figure 5 which helps the reader understanding structures of interest.

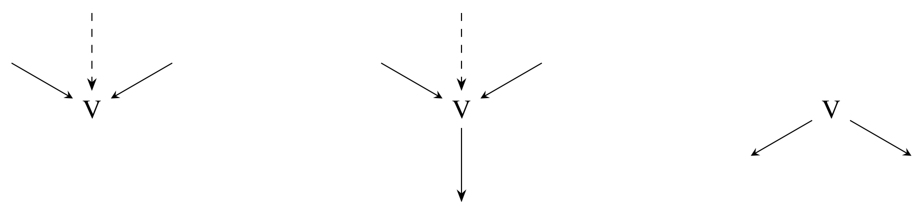


Figure 5: We represent the three forms that arrows can take around a vertex with multiple arrow in a structure of interest. Some vertices have incoming arrows (left), some have incoming arrows and a single outgoing arrow (middle), and some have exactly two outgoing arrows and no incoming arrows (right).

During the proofs, we are often led to construct structures of interest in which a root lies outside the target set. However, the graph containing such a structure also includes a directed path from this problematic root (outside the target set) to a vertex within the target set. Lemma 1 shows how to exploit this directed path to construct a new structure of interest where the problematic root has been removed. Figure 6 depicts this idea.

(a) (b)

Figure 6: Figure 6a depicts a graph containing a structure of interest shown in bold black. Let us assume that the target set of roots is $\{R_1, R_2\}$. In this graph, B is a root outside the target set. However, the graph contains a directed path $\langle B, C, R_1 \rangle$ from B to a vertex in the target set. Figure 6b illustrates how this path can be used to construct a structure of interest in which B is no longer a root, without introducing a new root outside the target set.

Lemma 1 (Add a Path to Remove a Problematic Root)**.** *Let $\mathcal{G}$ be an ADMG. Let $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ be pairwise distinct subsets of nodes of $\mathcal{G}$. Let σ be a structure of interest such that:*

- $\sigma \subseteq \mathcal{G}$
- $\mathcal{X} \cap \sigma \neq \emptyset$ *and* $\mathcal{Y} \cap \sigma \neq \emptyset$.
- $(\sigma \setminus \mathrm{Root}(\sigma)) \cap \mathcal{Z} = \emptyset$.

If there exists $R \in (\mathrm{Root}(\sigma) \setminus \mathcal{Z}) \cap Anc(\mathcal{Z}, \mathcal{G})$ then $\mathcal{G}$ contains a structure of interest σ' such that:

(a) $\sigma' \subseteq \mathcal{G}$

(b) $\mathcal{X} \cap \sigma' \neq \emptyset$ *and* $\mathcal{Y} \cap \sigma' \neq \emptyset$*.*

(c) $(\sigma' \setminus \mathrm{Root}(\sigma')) \cap \mathcal{Z} = \emptyset$*.*

(d) $\mathrm{Root}(\sigma') \setminus \mathcal{Z} \subseteq (\mathrm{Root}(\sigma) \setminus \mathcal{Z}) \setminus \{R\}$.

Sketch of proof. $R \in \mathrm{Anc}(\mathcal{Z}, \mathcal{G})$, thus $\mathcal{G}$ contains a directed path π from R to $\mathcal{Z}$. We add this path to σ to remove R from the root set. The end of the path is in $\mathcal{Z}$, thus we do not add a problematic root. □

Proof. $R \in \mathrm{Anc}(\mathcal{Z}, \mathcal{G})$, thus $\mathcal{G}$ contains a directed path π from R to $\mathcal{Z}$. Without loss of generality, we assume that π meets $\mathcal{Z}$ only at its last vertex. We construct σ' with the following procedure:

- If $\sigma \cap \pi \setminus \{R\} = \emptyset$, then $\sigma \cup \pi$ is a structure of interest. In this case, we set $\sigma' \leftarrow \sigma \cup \pi$. We have the following properties:

  (a) $\sigma' \subseteq \mathcal{G}$ because $\sigma \subseteq \mathcal{G}$ and $\pi \subseteq \mathcal{G}$.
  (b) $\mathcal{X} \cap \sigma \subseteq \mathcal{X} \cap \sigma'$ and $\mathcal{Y} \cap \sigma \subseteq \mathcal{Y} \cap \sigma'$ thus $\mathcal{X} \cap \sigma' \neq \emptyset$ and $\mathcal{Y} \cap \sigma' \neq \emptyset$.
  (c) By construction, $\mathrm{Root}(\sigma') = \mathrm{Root}(\sigma) \cup \mathrm{Root}(\pi) \setminus \{R\}$. Thus, $\sigma' \setminus \mathrm{Root}(\sigma') = \{R\} \cup (\sigma \cup \pi) \setminus (\mathrm{Root}(\sigma) \cup \mathrm{Root}(\pi)) \subseteq \{R\} \cup (\sigma \setminus \mathrm{Root}(\sigma)) \cup (\pi \setminus \mathrm{Root}(\pi))$. Since $R \notin \mathcal{Z}$, $(\sigma \setminus \mathrm{Root}(\sigma)) \cap \mathcal{Z} = \emptyset$, and $(\pi \setminus \mathrm{Root}(\pi)) \cap \mathcal{Z} = \emptyset$ (because π meets $\mathcal{Z}$ only at its last vertex), we can conclude that $(\sigma' \setminus \mathrm{Root}(\sigma')) \cap \mathcal{Z} = \emptyset$.
  (d) $\mathrm{Root}(\sigma') = \mathrm{Root}(\sigma) \cup \mathrm{Root}(\pi) \setminus \{R\}$. Since $\mathrm{Root}(\pi) \subseteq \mathcal{Z}$, we conclude that $\mathrm{Root}(\sigma') \setminus \mathcal{Z} \subseteq (\mathrm{Root}(\sigma) \setminus \mathcal{Z}) \setminus \{R\}$.

- Otherwise, $\sigma \cap \pi \setminus \{R\} \neq \emptyset$. Let W be the first encounter of π and $\sigma \setminus \{R\}$ and let π' be the subpath of π from R to W.[4] In this case, we set $\sigma' \leftarrow \sigma \cup \pi'$. We have (a), (b), but also

  (c) By construction, $\mathrm{Root}(\sigma') = \mathrm{Root}(\sigma) \setminus \{R\}$. Thus, $\sigma' \setminus \mathrm{Root}(\sigma') = \{R\} \cup (\sigma \setminus \mathrm{Root}(\sigma)) \cup (\pi' \setminus \mathrm{Root}(\sigma))$. Since π meets $\mathcal{Z}$ only at its last vertex, we know that $\pi' \cap \mathcal{Z} \subseteq \{W\}$.[5] Thus, $(\pi' \setminus \mathrm{Root}(\sigma)) \cap \mathcal{Z} = (\pi' \cap \mathcal{Z}) \setminus \mathrm{Root}(\sigma) \subseteq \{W\} \setminus \mathrm{Root}(\sigma) \subseteq \sigma \setminus \mathrm{Root}(\sigma)$. Since $R \notin \mathcal{Z}$ and $(\sigma \setminus \mathrm{Root}(\sigma)) \cap \mathcal{Z} = \emptyset$, we can conclude that $(\sigma' \setminus \mathrm{Root}(\sigma')) \cap \mathcal{Z} = \emptyset$.
  (d) $\mathrm{Root}(\sigma') = \mathrm{Root}(\sigma) \setminus \{R\}$. Therefore, $\mathrm{Root}(\sigma') \setminus \mathcal{Z} \subseteq (\mathrm{Root}(\sigma) \setminus \mathcal{Z}) \setminus \{R\}$.

  However, σ' is not necessarily a structure of interest. By construction, W is the only node of $\sigma \cup \pi'$ which does not necessarily satisfy the conditions of Definition 3. Since σ is a structure of interest, the arrows around W in σ are necessarily in one of the three cases described by Figure 5. The right hand case is the only case where adding an incoming arrow prevents σ' from being a structure of interest. Thus, if σ' is not a structure of interest, it means that W has one incoming arrow and two outgoing arrows in σ'. Let A and B be the two children of W in σ'. Moreover, since $\sigma \cap \pi \neq \emptyset$, we know that σ' has a single connected component. Let $\pi^\star$ be a path between $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ in σ'. Let us show that we can assume that $\pi^\star$ does not use $A \leftarrow W \rightarrow B$:

  - If $\pi^\star$ uses $A \leftarrow W \rightarrow B$. Without loss of generality, we assume that A is before B in $\pi^\star$. Let π^{XR} be a path from X to R in σ. We distinguish two cases:
    * If π^{XR} does not encounter W (cf Figure 7a). We consider $\theta \coloneqq \pi^{XR} \cup \pi' \cup \pi^\star_{[W,Y]}$. θ is a subgraph of σ' in which $\mathcal{X}$ and $\mathcal{Y}$ are connected.[6] By construction, θ does not contain $A \leftarrow W$. Thus $\theta \subseteq \sigma'$ contains a path between $\mathcal{X}$ and $\mathcal{Y}$ that does not use $A \leftarrow W \rightarrow B$.
    * Otherwise, π^{XR} encounters W. Since $\pi^{XR} \subseteq \sigma$, π^{XR} uses $A \leftarrow W \rightarrow B$. We distinguish two cases:
      · If A is before B in π^{XR} (cf Figure 7b), we consider $\theta \coloneqq \pi^\star_{[XW]} \cup \pi' \cup \pi^{XR}_{[R,B]} \cup \pi^\star_{[B,Y]}$. θ is a subgraph of σ' in which $\mathcal{X}$ and $\mathcal{Y}$ are connected. By construction, θ does

[4] π and π' may be equal.

[5] $\pi' \cap \mathcal{Z}$ is empty if $\pi' \neq \pi$, otherwise, it is equal to {W}.

[6] θ is not necessarily a path.

not contain $W \to B$. Thus $\theta \subseteq \sigma'$ contains a path between $\mathcal{X}$ and $\mathcal{Y}$ that does not use $A \leftarrow W \to B$.

- Otherwise, A is after B in π^{XR} (cf Figure 7c), we consider $\theta := \pi^{XR}_{[X,B]} \cup \pi^{\star}_{[B,Y]}$. θ is a subgraph of σ' in which $\mathcal{X}$ and $\mathcal{Y}$ are connected. By construction, θ does not contain $A \leftarrow W \to B$. Thus $\theta \subseteq \sigma'$ contains a path between $\mathcal{X}$ and $\mathcal{Y}$ that does not use $A \leftarrow W \to B$.

Thus, without loss of generality, we assume that $\pi^{\star}$ does not use $A \leftarrow W \to B$. We remove from σ' one outgoing arrow from W that is not used by $\pi^{\star}$. By doing so, W satisfies the conditions of Definition 3. Thus, all the vertices in σ' now satisfy the conditions of Definition 3, and satisfies (a) and (b), and

(c) $(\sigma' \setminus \mathrm{Root}(\sigma')) \cap \mathcal{Z} = \emptyset$ because removing the arrow does not change the vertices nor the roots of σ'.

(d) $\mathrm{Root}(\sigma') \setminus \mathcal{Z} \subseteq (\mathrm{Root}(\sigma) \setminus \mathcal{Z}) \setminus \{R\}$ because removing the arrow does not change the roots of σ'.

However, σ' does not contain necessarily a single connected component. Thus, we only keep the connected component of X, which contains Y via $\pi^{\star}$. By doing so, σ' is now a structure of interest and we have:

(c) $(\sigma' \setminus \mathrm{Root}(\sigma')) \cap \mathcal{Z} = \emptyset$ because we consider a subgraph.

(d) $\mathrm{Root}(\sigma') \setminus \mathcal{Z} \subseteq (\mathrm{Root}(\sigma) \setminus \mathcal{Z}) \setminus \{R\}$ because we consider a subgraph.

□

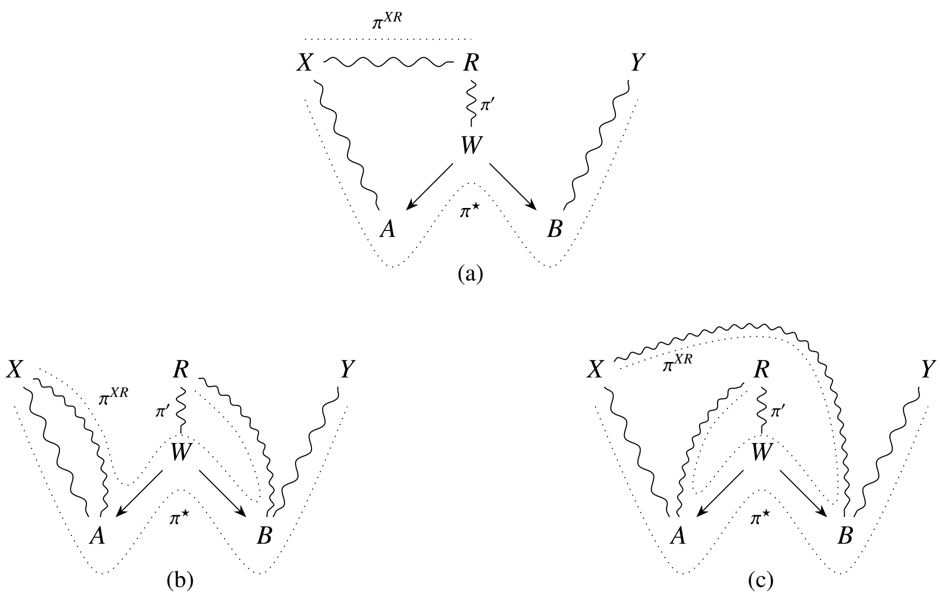


Figure 7: Helping figure for the Proof of Lemma 1. $\pi^{\star}$ and π^{XR} are represented by the dotted paths following the arrows. A squiggly arrow represents an arbitrary path.

Corollary 2. *Let $\mathcal{G}$ be an ADMG. Let $\mathcal{X}$, $\mathcal{Y}$, $\mathcal{Z}$ and $\mathcal{R}$ be pairwise distinct subsets of nodes of $\mathcal{G}$. Let $\mathcal{G}$ contains a structure of interest σ such that:*

- $\mathcal{X} \cap \sigma \neq \emptyset$ *and* $\mathcal{Y} \cap \sigma \neq \emptyset$
- $\mathrm{Root}(\sigma) \subseteq \mathcal{Z} \cup \mathcal{X} \cup \mathcal{Y} \cup \mathcal{R}$
- $(\sigma \setminus \mathrm{Root}(\sigma)) \cap \mathcal{Z} = \emptyset$

If for all $R \in \mathcal{R}$, $R \in Anc(\mathcal{Z}, \mathcal{G})$, then $\mathcal{G}$ contains a structure of interest that connects $\mathcal{X}$ and $\mathcal{Y}$ under $\mathcal{Z}$.

Proof. We apply Lemma 1 iteratively for each $R \in \mathcal{R}$, we get a structure of interest σ' such that:

- $\sigma' \subseteq \mathcal{G}$
- $\mathcal{X} \cap \sigma' \neq \emptyset$ and $\mathcal{Y} \cap \sigma' \neq \emptyset$
- $\text{Root}(\sigma') \setminus \mathcal{Z} \subseteq (\text{Root}(\sigma) \setminus \mathcal{Z}) \setminus \mathcal{R}$. Thus, $\text{Root}(\sigma) \subseteq \mathcal{Z} \cup \mathcal{X} \cup \mathcal{Y}$.
- $(\sigma' \setminus \text{Root}(\sigma')) \cap \mathcal{Z} = \emptyset$

Therefore, σ' a structure of interest that connects $\mathcal{X}$ and $\mathcal{Y}$ under $\mathcal{Z}$. □

Theorem 1 (D-connection with structures of interests)**.** *Let $\mathcal{G}$ be an ADMG. Let $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ be pairwise disjoint subsets of nodes of $\mathcal{G}$. The following properties are equivalent:*

1. *$\mathcal{X} \not\perp\!\!\!\perp_{\mathcal{G}} \mathcal{Y} \mid \mathcal{Z}$.*
2. *$\mathcal{G}$ contains a structure of interest σ such that $\mathcal{X} \not\perp\!\!\!\perp_{\sigma} \mathcal{Y} \mid \mathcal{Z}$.*

Proof. Let us prove the two implications:

- $1 \Rightarrow 2$: If $\mathcal{X} \not\perp\!\!\!\perp_{\mathcal{G}} \mathcal{Y} \mid \mathcal{Z}$, then, by definition, the ADMG $\mathcal{G}$ contains a path π d-connecting $\mathcal{X}$ and $\mathcal{Y}$. Without loss of generality, we assume that π encounters $\mathcal{X}$ only at its first vertex and $\mathcal{Y}$ only at its last vertex. Since π is a d-connecting path, it is a structure of interest which satisfies the following properties:

  - $\pi \subseteq \mathcal{G}$.
  - $\mathcal{X} \cap \pi \neq \emptyset$ and $\mathcal{Y} \cap \pi \neq \emptyset$.
  - $(\pi \setminus \text{Root}(\pi)) \cap \mathcal{Z} = \emptyset$.

  However, some roots of π may not be in $\mathcal{Z} \cup \mathcal{X} \cup \mathcal{Y}$, preventing π from being a connecting structure of interest. Necessarily, theses roots are colliders. Define $\mathcal{R} := \text{Root}(\pi) \setminus (\mathcal{Z} \cup \mathcal{X} \cup \mathcal{Y})$ to be the set of these colliders. Since π is d-connecting, for all $R \in \mathcal{R}$, $R \in \text{Anc}(\mathcal{Z}, \mathcal{G})$. By Corollary 2, $\mathcal{G}$ contains a structure of interest which connects $\mathcal{X}$ and $\mathcal{Y}$ under $\mathcal{Z}$.

- $2 \Rightarrow 1$: By definition, σ has a single connected component. Thus σ contains a path π from $\mathcal{X}$ to $\mathcal{Y}$. Without loss of generality, we assume that π encounters $\mathcal{X}$ only at its first vertex and $\mathcal{Y}$ only at its last vertex. Let us first prove that without loss of generality, we can assume that all colliders on π are ancestors of $\mathcal{Z}$. If it's not the case, since $\text{Root}(\sigma) \subseteq \mathcal{Z} \cup \mathcal{X} \cup \mathcal{Y}$, all colliders that are not ancestors of $\mathcal{Z}$ are ancestors of $\mathcal{X} \cup \mathcal{Y}$. Without loss of generality, assume that a collider C on π is an ancestor of $\mathcal{X}$. Thus, σ contains a directed path π^1 from C to $\mathcal{X}$. Let π^2 be the subpath of π between C and $\mathcal{Y}$. Since C belongs to both π^1 and π^2, these two paths intersect at C, and possibly at other vertices. Let T be the last vertex of π^1 that is in $\pi^1 \cap \pi^2$. Let $\pi' := \pi^1_{[T,\mathcal{X}]} \cup \pi^2_{[T,\mathcal{Y}]}$. π' is a path. Moreover, between X and T, π' is a directed path, and, after T to $\mathcal{Y}$, π' is a subpath of π_2. Therefore, π' contains at least one fewer collider than π that is not an ancestor of $\mathcal{Z}$. Repeating this procedure iteratively allows us to construct a path in σ from $\mathcal{X}$ to $\mathcal{Y}$ in which all colliders are ancestors of $\mathcal{Z}$.

  Finally, note that all forks and chains on π are not roots by definition, and hence do not belong to $\mathcal{Z}$. Therefore, the resulting path π is *d*-connecting between $\mathcal{X}$ and $\mathcal{Y}$ given $\mathcal{Z}$.

□

A.3 Proofs of the Properties of the canonical compatible Graph and the Unfolded Graph

A.3.1 Canonical Compatible Graph

Proposition 1. *Let $\mathcal{G}^C$ be a C-DAG and $\mathcal{G}^m_{can}$ be its corresponding canonical compatible graph. Then, the following properties hold:*

1. *$\mathcal{G}^m_{can} \in C(\mathcal{G}^C)$.*

2. *For all* $\mathcal{G}^m \in C(\mathcal{G}^C)$, $\mathcal{G}^m_{can} \cup \mathcal{G}^m \in C(\mathcal{G}^C)$.

We break the proof of Proposition 1 in Lemma 2 and Lemma 3.

Lemma 2. *Let* $\mathcal{G}^C$ *be a C-DAG. Then,* $\mathcal{G}^m_{can}$*, its canonical compatible graph, is compatible with* $\mathcal{G}^C$.

Proof. By construction, all arrows in $\mathcal{G}^C$ are represented in $\mathcal{G}^m_{\text{can}}$. Moreover, all arrows that are added correspond to an arrow in $\mathcal{G}^C$. Therefore, we only need to check for acyclicity to prove that $\mathcal{G}^m_{\text{can}}$ is a compatible graph with $\mathcal{G}^C$. By contradiction, let us assume that $\mathcal{G}^m_{\text{can}}$ contains a cycle π. First, we know that π is not within a single cluster. Indeed, in each cluster V, $\mathcal{G}^m_{\text{can}|V}$ is a $V_{\#V}$-rooted tree. Thus, π encounters at least two clusters and has an arrow between two clusters. Let $A_1 \to B_{\#B}$ be such an arrow. We prove that necessarily, $\#A^C = 1$. Indeed, if $\#A^C \geq 2$, then no arrow in $\mathcal{G}^m_{\text{can}}$ is pointing on A_1. Similarly, we can show that $\#B^C = 1$. Thus, the next arrow cannot be pointing into B^C, thus the next arrow is also an arrow between two clusters. By induction, we show that all the clusters encountered by π have a cardinal of 1. This contradicts Proposition 4. Therefore, $\mathcal{G}^m_{\text{can}}$ is acyclic.

Therefore, $\mathcal{G}^m_{\text{can}}$ is a compatible graph with $\mathcal{G}^C$. □

Lemma 3. *Let* $\mathcal{G}^C$ *be a C-DAG,* $\mathcal{G}^m_{can}$ *be its corresponding canonical compatible graph, and* $\mathcal{G}^m$ *be a compatible graph. Then,* $\mathcal{G}^m_{can} \cup \mathcal{G}^m$ *is a compatible graph.*

Proof. Since $\mathcal{G}^m$ and $\mathcal{G}^m_{\text{can}}$ are compatible, then we only need to check for acyclicity. Let label indices according to Notation 1. Let a be an arrow in $\mathcal{G}^m_{\text{can}}$ that is not in $\mathcal{G}^m$. We distinguish three cases:

- If a is a dashed-bidirected arrow, then adding a does not create a cycle.
- If a is a directed arrow inside a cluster V^C, since $\mathcal{G}^m$ follows Notation 1, then the indices in V^m follow the topological ordering induced by $\mathcal{G}^m$. Therefore, a can be added without creating a cycle.
- Otherwise, a corresponds to an arrow between two clusters V^m and U^m. $\mathcal{G}^m$ is compatible thus it also contains an arrow from U^m to V^m. By applying Propositions 6 and 7, we see that we can add a without creating a cycle.

Therefore, $\mathcal{G}^m_{\text{can}} \cup \mathcal{G}^m$ is a compatible graph. □

A.3.2 Unfolded Graph

Proposition 2. *Let* $\mathcal{G}^C$ *be a C-DAG and* $\mathcal{G}^m_u$ *be its corresponding unfolded graph. Then, any compatible graph* $\mathcal{G}^m$ *is a subgraph of* $\mathcal{G}^m_u$ *up to a permutation of indices in each cluster.*

Proof. Let $\mathcal{G}^m = (\mathcal{V}^m, \mathcal{E}^m)$ be a compatible graph. By definition, we already know that $\mathcal{V}^m = \mathcal{V}_{\text{u}}$. $\mathcal{G}^m$ is a DAG, thus, in each cluster, we can permute the indices of the vertices so that the topological order of $\mathcal{G}^m$ agrees with the order of the indices. Let a be an arrow of $\mathcal{G}^m$. We distinguish three cases:

- If a is a dashed-bidirected-arrow, then a is also in $\mathcal{G}^m_{\text{u}}$ because $\mathcal{G}^m_{\text{u}}$ is a super graph of $\mathcal{G}^m_{\text{can}}$ which contains all possible dashed-bidirected-arrows.
- If a corresponds to a self-loop $\circlearrowleft V^C$ in $\mathcal{G}^C$. Necessarily, in $\mathcal{G}^m$, $a = V_i \to V_j$ with $i < j$. Thus, a corresponds to an arrow added during step 2 of Definition 5. Therefore, a is also an arrow in $\mathcal{G}^m_{\text{u}}$.
- Otherwise, a corresponds to an arrow $U^C \to V^C$, with $U^C \neq V^C$. We distinguish two cases:
  - If a is added at step 3 of Definition 5, then a is also an arrow in $\mathcal{G}^m_{\text{u}}$.
  - Otherwise, by Lemma 3, we know that $\mathcal{G}^m_{\text{can}} \cup \mathcal{G}^m$ is compatible. Thus $\mathcal{G}^m_{\text{can}} \cup \mathcal{G}^m$ is acyclic. Since $\mathcal{G}^m_{\text{can}} \cup a$ is a subgraph of $\mathcal{G}^m_{\text{can}} \cup \mathcal{G}^m$, we can conclude that a does not create a cycle in $\mathcal{G}^m_{\text{can}}$. Thus, $a \in \mathcal{E}_{\text{eligible}}$. Therefore a is an arrow in $\mathcal{G}^m_{\text{u}}$

Hence, $\mathcal{E}^m \subseteq \mathcal{E}_{\text{u}}$. Therefore, $\mathcal{G}^m$ is a subgraph of G_{u}.

□

A.3.3 Proof of Proposition 3

Proposition 3. *Let $\mathcal{G}^C$ be a C-DAG and $\mathcal{G}^m_u$ be its corresponding unfolded graph. Let σ^m be a structure of interest in $\mathcal{G}^m_u$. If $\mathcal{G}^m_{can} \cup \sigma^m$ is acyclic, then $\mathcal{G}^m_{can} \cup \sigma^m$ is compatible with $\mathcal{G}^C$.*

Proof. By Proposition 1, $\mathcal{G}^m_{\text{can}}$ is compatible with $\mathcal{G}^C$. Therefore, $\mathcal{G}^m_{\text{can}} \cup \sigma^m$ contains all necessary arrows to be compatible with $\mathcal{G}^C$. Moreover, by definition of $\mathcal{G}^m_{\text{u}}$, all arrow $V_v \to W_w$ in $\sigma^m \subseteq \mathcal{G}^m_{\text{u}}$ correspond to an arrow $V \to W$ in $\mathcal{G}^C$. Therefore, $\mathcal{G}^m_{\text{can}} \cup \sigma^m$ does not contain any arrow preventing it from being compatible with $\mathcal{G}^m_{\text{u}}$. $\mathcal{G}^m_{\text{can}} \cup \sigma^m$ is acyclic, thus it is an ADMG. Therefore, $\mathcal{G}^m_{\text{can}} \cup \sigma^m$ is compatible with $\mathcal{G}^C$. □

A.4 Proofs of the Calculus

First of all let us recall the rules of Pearl's calculus for cluster queries.

Theorem 5 (Do-Calculus Rules for Cluster Queries (Pearl, 2009))**.** *Let $\mathcal{G}^C$ be a C-DAG and $\mathcal{X}^C, \mathcal{Y}^C, \mathcal{Z}^C, \mathcal{W}^C$ be pairwise distinct subsets of nodes. Let $\mathcal{G}^m$ be a compatible graph. The following rules hold:*

1. ***Insertion/deletion of observations:***

   $P(\mathbf{y^m} \mid do(\mathbf{w^m}), \mathbf{x^m}, \mathbf{z^m}) = P(\mathbf{y^m} \mid do(\mathbf{w^m}), \mathbf{z^m}) \quad \textit{if } \mathcal{Y}^m \perp\!\!\!\perp_{\mathcal{G}_{\overline{\mathcal{W}^m}}} \mathcal{X}^m \mid \mathcal{W}^m, \mathcal{Z}^m$

2. ***Action/observation exchange:***

   $P(\mathbf{y^m} \mid do(\mathbf{w^m}), do(\mathbf{x^m}), \mathbf{z^m}) = P(\mathbf{y^m} \mid do(\mathbf{w^m}), \mathbf{x^m}, \mathbf{z^m}) \quad \textit{if } \mathcal{Y}^m \perp\!\!\!\perp_{\mathcal{G}_{\overline{\mathcal{W}^m},\underline{\mathcal{X}^m}}} \mathcal{X}^m \mid \mathcal{W}^m, \mathcal{Z}^m$

3. ***Insertion/deletion of actions:***

   $P(\mathbf{y^m} \mid do(\mathbf{w^m}), do(\mathbf{x^m}), \mathbf{z^m}) = P(\mathbf{y^m} \mid do(\mathbf{w^m}), \mathbf{z^m}) \quad \textit{if } \mathcal{Y}^m \perp\!\!\!\perp_{\mathcal{G}_{\overline{\mathcal{W}^m},\overline{\mathcal{X}^m(\mathcal{Z}^m)}}} \mathcal{X}^m \mid \mathcal{W}^m, \mathcal{Z}^m$

   where $\mathcal{X}^m(\mathcal{Z}^m) \coloneqq \mathcal{X}^m \setminus Anc(\mathcal{Z}^m, \mathcal{G}_{\overline{\mathcal{W}^m}})$.

The first two rules hinge on graphical conditions expressed as d-separations under cluster-level mutilations. We begin by rigorously characterizing this form of dependency; Theorem 6 provides precisely this characterization.

A.4.1 Cluster D-separation with Cluster Mutilations

Theorem 6. *Let $\mathcal{G}^C = \left(\mathcal{V}^C, \mathcal{E}^C\right)$ be a C-DAG. Let $\mathcal{G}^m_{can}$ be its corresponding canonical compatible graph. Let $\mathcal{G}^m_u$ be the corresponding unfolded graph. Let $\mathcal{X}^C, \mathcal{Y}^C$ and $\mathcal{Z}^C$ be pairwise distinct subsets of nodes of $\mathcal{G}^C$. Let $\mathcal{A}$ and $\mathcal{B}$ be subsets of nodes of $\mathcal{G}^C$. Then the following properties are equivalent:*

1. *$\exists\, \mathcal{G}^m \in C\left(\mathcal{G}^C\right)$ $\mathcal{X}^m \not\perp\!\!\!\perp_{\mathcal{G}^m_{\overline{\mathcal{A}^m}\underline{\mathcal{B}^m}}} \mathcal{Y}^m \mid \mathcal{Z}^m$.*
2. *$\mathcal{G}^m_{u\,\overline{\mathcal{A}^m}\underline{\mathcal{B}^m}}$ contains a structure of interest σ^m such that $\mathcal{X}^m \not\perp\!\!\!\perp_{\sigma^m} \mathcal{Y}^m \mid \mathcal{Z}^m$ and $\mathcal{G}^m_{can} \cup \sigma^m$ is acyclic.*

Proof. Let us prove the two implications:

- $1 \Rightarrow 2$: Let $\mathcal{G}^m$ be a compatible graph such that $\mathcal{G}^m{}_{\overline{\mathcal{A}^m}\underline{\mathcal{B}^m}}$ contains a structure of interest σ^m which connects $\mathcal{X}^m$ and $\mathcal{Y}^m$ under $\mathcal{Z}^m$. By Lemma 2, $\sigma^m \subseteq \mathcal{G}^m{}_{\overline{\mathcal{A}^m}\underline{\mathcal{B}^m}} \subseteq \mathcal{G}^m \subseteq \mathcal{G}^m_{\text{u}}$. Moreover, since $\sigma^m \subseteq \mathcal{G}^m{}_{\overline{\mathcal{A}^m}\underline{\mathcal{B}^m}}$, we know that σ^m does not contain any incoming arrow in $\mathcal{A}^m$ and no outgoing arrow from $\mathcal{B}^m$. Therefore, $\sigma^m \subseteq \mathcal{G}^m_{\text{u}\,\overline{\mathcal{A}^m}\underline{\mathcal{B}^m}}$ and σ^m connects $\mathcal{X}^m$ and $\mathcal{Y}^m$ under $\mathcal{Z}^m$.

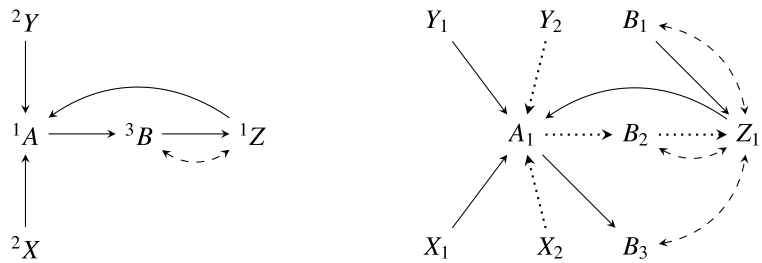
  By Lemma 3, $\mathcal{G}^m_{\text{can}} \cup \mathcal{G}^m$ is a compatible graph, thus acyclic. Moreover, $\mathcal{G}^m_{\text{can}} \cup \sigma^m \subseteq \mathcal{G}^m_{\text{can}} \cup \mathcal{G}^m$. Therefore, $\mathcal{G}^m_{\text{can}} \cup \sigma^m$ is acyclic.

- $2 \Rightarrow 1$: $\mathcal{G}^m_{\text{can}} \cup \sigma^m$ is acyclic. By Lemma 2, $\mathcal{G}^m_{\text{can}}$ is compatible. Moreover, all arrows from σ^m come from $\mathcal{G}^m_{\text{u}}$. Therefore, $\mathcal{G}^m_{\text{can}} \cup \sigma^m$ is a compatible graph.

Moreover, since $\sigma^m \subseteq \mathcal{G}^m_{\mathrm{u}\,\overline{\mathcal{A}^m}\underline{\mathcal{B}^m}}$, we know that σ^m does not contain any incoming arrow in $\mathcal{A}^m$ and no outgoing arrow from $\mathcal{B}^m$. Thus, $\sigma^m \subseteq (\mathcal{G}^m_{\mathrm{can}} \cup \sigma^m)_{\overline{\mathcal{A}^m}\underline{\mathcal{B}^m}}$. Since σ^m connects $\mathcal{X}^m$ and $\mathcal{Y}^m$ under $\mathcal{Z}^m$, we can conclude that $\mathcal{X}^m \not\perp\!\!\!\perp_{(\mathcal{G}^m_{\mathrm{can}} \cup \sigma^m)_{\overline{\mathcal{A}^m}\underline{\mathcal{B}^m}}} \mathcal{Y}^C \mid \mathcal{Z}^m$.

□

Figure 8 illustrates Theorem 6.



(a) $\mathcal{G}^C$ (b) $\mathcal{G}^m_{\mathrm{u}}$

Figure 8: (a) a C-DAG $\mathcal{G}^C$. (b) its corresponding unfolded graphs. The plain and dashed arrows represent $\mathcal{G}^m_{\mathrm{can}}$, while the dotted arrows denote the "eligible" edges. According to Theorem 6, we have $\forall\, \mathcal{G}^m \in C\left(\mathcal{G}^C\right)\ \mathcal{X}^m \perp\!\!\!\perp_{\mathcal{G}^m} \mathcal{Y}^m \mid \mathcal{Z}^m$, as all structures of interest connecting X^m and Y^m given Z^m in $\mathcal{G}^m_{\mathrm{u}}$ include a directed path from A_1 to Z_1. Since $\mathcal{G}^m_{\mathrm{can}}$ already contains the edge $Z_1 \to A_1$, such paths would necessarily form a cycle.

A.4.2 Proofs of the Three Rules of the Calculus

As soon as Theorem 6 has been established, Rules 1 and 2 of the calculus follow almost immediately. In contrast, the third rule of Pearl's do-calculus requires verifying the d-separation condition $\mathcal{Y}^m \perp\!\!\!\perp_{\mathcal{G}^m_{\overline{\mathcal{W}^m},\overline{\mathcal{X}^m(\mathcal{Z}^m)}}} X^m \mid \mathcal{W}^m, \mathcal{Z}^m$, in all compatible graph $\mathcal{G}^m$, where $\mathcal{X}^m(\mathcal{Z}^m) = X^m \setminus \mathrm{Anc}(\mathcal{Z}^m, \mathcal{G}^m_{\overline{\mathcal{W}^m}})$. Since $\mathcal{X}^m(\mathcal{Z}^m)$ is not, in general, a union of clusters, the associated mutilation depends on the particular graph $\mathcal{G}^m$. As a result, this rule does not fall under the scope of Theorem 6.

Nonetheless, if Rule 3 does not hold in some compatible graph $\mathcal{G}^m$, then there exists a structure of interest between $\mathcal{Y}^m$ and X^m in $\mathcal{G}^m_{\overline{\mathcal{W}^m},\overline{\mathcal{X}^m(\mathcal{Z}^m)}}$. If this structure includes a root $X_x \in \mathcal{X}^m$, then X_x must be an ancestor of some $Z_z \in \mathcal{Z}^m$ in the mutilated graph. In such a case, we can augment the structure of interest by explicitly adding the directed path from X_x to $\mathcal{Z}^m$, resulting in a new structure whose roots lie outside $\mathcal{X}^m$.

Theorem 2 (Calculus)**.** *Let $\mathcal{G}^C$ be a C-DAG and let $\mathcal{G}^m_u$ be its corresponding unfolded graph. Let $\mathcal{X}^C, \mathcal{Y}^C, \mathcal{Z}^C, \mathcal{W}^C$ be pairwise distinct subsets of nodes. Then for any density P compatible with any compatible graph, the following rules apply:*

R1. $P(\mathbf{y^m} \mid do(\mathbf{w^m}), \mathbf{x^m}, \mathbf{z^m}) = P(\mathbf{y^m} \mid do(\mathbf{w^m}), \mathbf{z^m})$ *if $\mathcal{G}^m_{u\,\overline{\mathcal{W}^m}}$ does not contain a structure of interest σ^m such that $\mathcal{X}^m \not\perp\!\!\!\perp_{\sigma^m} \mathcal{Y}^m \mid \mathcal{W}^m, \mathcal{Z}^m$ and $\mathcal{G}^m_{can} \cup \sigma^m$ is acyclic.*

R2. $P(\mathbf{y^m} \mid do(\mathbf{w^m}), do(\mathbf{x^m}), \mathbf{z^m}) = P(\mathbf{y^m} \mid do(\mathbf{w^m}), \mathbf{x^m}, \mathbf{z^m})$ *if $\mathcal{G}^m_{u\,\overline{\mathcal{W}^m},\underline{\mathcal{X}^m}}$ does not contain a structure of interest σ^m such that $\mathcal{X}^m \not\perp\!\!\!\perp_{\sigma^m} \mathcal{Y}^m \mid \mathcal{W}^m, \mathcal{Z}^m$ and $\mathcal{G}^m_{can} \cup \sigma^m$ is acyclic.*

R3. $P(\mathbf{y^m} \mid do(\mathbf{w^m}), do(\mathbf{x^m}), \mathbf{z^m}) = P(\mathbf{y^m} \mid do(\mathbf{w^m}), \mathbf{z^m})$ *if $\mathcal{G}^m_{u\,\overline{\mathcal{W}^m}}$ does not contain a structure of interest σ^m such that $\mathcal{X}^m \not\perp\!\!\!\perp_{\sigma^m} \mathcal{Y}^m \mid \mathcal{W}^m, \mathcal{Z}^m$, $\mathcal{G}^m_{can} \cup \sigma^m$ is acyclic and* $\mathrm{Root}(\sigma^m) \subseteq (\mathcal{W}^m \cup \mathcal{Z}^m) \cup \mathcal{Y}^m$.

Proof. The first two rules are proven by Theorem 6. The third one is proved by the following reasoning.

We will show that if the third rule applies, then in all compatible graph $\mathcal{G}^m$, the third rule of Pearl's calculus applies. More precisely, we prove the contrapositive. Let $\mathcal{G}^m$ be a compatible graph in which the third rule does not apply. Then, $\mathcal{G}^m_{\overline{\mathcal{W}^m},\overline{\mathcal{X}^m(\mathcal{Z}^m)}}$ contains a structure of interest σ^m that connects $\mathcal{Y}^m$

and $\mathcal{X}^m$ under the conditioning set $\mathcal{W}^m \cup \mathcal{Z}^m$, where $\mathcal{X}^m(\mathcal{Z}^m) = X^m \setminus \mathrm{Anc}(\mathcal{Z}^m, \mathcal{G}^m_{\overline{\mathcal{W}^m}})$. By definition, we already know that σ^m follows the following properties:

- $\sigma^m \subseteq \mathcal{G}^m_{\overline{\mathcal{W}^m},\overline{\mathcal{X}^m(\mathcal{Z}^m)}} \subseteq \mathcal{G}^m_{\overline{\mathcal{W}^m}}$.
- $\mathcal{X}^m \cap \sigma^m \neq \emptyset$ and $\mathcal{Y}^m \cap \sigma^m \neq \emptyset$.
- $\mathrm{Root}(\sigma^m) \subseteq (\mathcal{W}^m \cup \mathcal{Z}^m) \cup \mathcal{Y}^m \cup \mathcal{X}^m$
- $(\sigma^m \setminus \mathrm{Root}(\sigma^m)) \cap (\mathcal{W}^m \cup \mathcal{Z}^m) = \emptyset$

Let us remark that $\mathrm{Root}(\sigma^m) \cap \mathcal{X}^m \subseteq \mathcal{X}^m \setminus \mathcal{X}^m(\mathcal{Z}^m) = \mathrm{Anc}(\mathcal{Z}^m, \mathcal{G}^m_{\overline{\mathcal{W}^m}})$. Indeed, let X_x be an element of $\mathrm{Root}(\sigma^m) \cap \mathcal{X}^m$. Since X_x is a root and σ^m has a single connected component, X_x must have an incoming edge within σ^m. Thus, X_x must have an incoming edge within $\mathcal{G}^m_{\overline{\mathcal{W}^m},\overline{\mathcal{X}^m(\mathcal{Z}^m)}}$. Thus $X_x \notin \mathcal{X}^m(\mathcal{Z}^m)$. For element of $\mathrm{Root}(\sigma^m) \cap \mathcal{X}^m$, we iteratively update σ^m using Lemma 1. At the end of this process, we obtain a structure of interest $\sigma^{m\prime}$ which satisfies the following identities:

- $\sigma^{m\prime} \subseteq \mathcal{G}^m_{\overline{\mathcal{W}^m}}$.
- $\mathcal{X}^m \cap \sigma^{m\prime} \neq \emptyset$ and $\mathcal{Y}^m \cap \sigma^{m\prime} \neq \emptyset$.
- $(\sigma^{m\prime} \setminus \mathrm{Root}(\sigma^{m\prime}) \cap (\mathcal{W}^m \cup \mathcal{Z}^m) = \emptyset$
- $\mathrm{Root}(\sigma^{m\prime}) \setminus (\mathcal{W}^m \cup \mathcal{Z}^m) \subseteq (\mathrm{Root}(\sigma^m) \setminus (\mathcal{W}^m \cup \mathcal{Z}^m)) \setminus (\mathrm{Root}(\sigma^m) \cap \mathcal{X}^m) \subseteq \mathcal{Y}^m$

Thus, $\mathrm{Root}(\sigma^{m\prime}) \subseteq (\mathcal{W}^m \cup \mathcal{Z}^m) \cup \mathcal{Y}^m$ and $\sigma^{m\prime}$ is a structure of interest which connects $\mathcal{X}^m$ and $\mathcal{Y}^m$ under $\mathcal{W}^m \cup \mathcal{Z}^m$. Since $\mathcal{G}^m_{\overline{\mathcal{W}^m}} \subseteq \mathcal{G}^m_{\mathrm{u}\,\overline{\mathcal{W}^m}}$, it follows that $\mathcal{G}^m_{\mathrm{u}}$ contains $\sigma^{m\prime}$. Moreover, since $\mathcal{G}^m_{\mathrm{can}} \cup \sigma^{m\prime} \subseteq \mathcal{G}^m_{\mathrm{can}} \cup \mathcal{G}^m$, it follows that $\mathcal{G}^m_{\mathrm{can}} \cup \sigma^{m\prime}$ is acyclic.

Therefore, $\mathcal{G}^m_{\mathrm{u}\,\overline{\mathcal{W}^m}}$ contains a structure of interest σ^m which $\mathcal{X}^m$ and $\mathcal{Y}^m$ under $\mathcal{W}^m \cup \mathcal{Z}^m$, with $\mathrm{Root}(\sigma^m) \subseteq (\mathcal{W}^m \cup \mathcal{Z}^m) \cup \mathcal{Y}^m$, and such that $\mathcal{G}^m_{\mathrm{can}} \cup \sigma^m$ is acyclic. □

Theorem 3 (Atomic completeness)**.** *The calculus in Theorem 2 is atomically complete i.e. if the rule does not hold given a C-DAG, then there exists a compatible graph in which the corresponding rule in Pearl's calculus fails.*

Proof. The first two rules are proven by Theorem 6. The third one is proved by the following reasoning.

If the rule does not hold, then $\mathcal{G}^m_{\mathrm{u}\,\overline{\mathcal{W}^m}}$ contains a structure of interest σ^m that connects $\mathcal{X}^m$ and $\mathcal{Y}^m$ under $\mathcal{W}^m \cup \mathcal{Z}^m$ such that $\mathrm{Root}(\sigma^m) \subseteq (\mathcal{W}^m \cup \mathcal{Z}^m) \cup \mathcal{Y}^m$ and $\mathcal{G}^m_{\mathrm{can}} \cup \sigma^m$ is acyclic. $\mathcal{G}^m_{\mathrm{can}} \cup \sigma^m$ is acyclic, thus $\mathcal{G}^m := \mathcal{G}^m_{\mathrm{can}} \cup \sigma^m$ is a compatible graph. Moreover, since $\sigma^m \subseteq \mathcal{G}^m_{\mathrm{u}\,\overline{\mathcal{W}^m}}$, then $\sigma^m \subseteq \mathcal{G}^m_{\overline{\mathcal{W}^m}}$. We will show that we can assume that $\sigma^m \subseteq \mathcal{G}^m_{\overline{\mathcal{W}^m},\overline{\mathcal{X}^m(\mathcal{Z}^m)}}$, where $\mathcal{X}^m(\mathcal{Z}^m) = X^m \setminus \mathrm{Anc}(\mathcal{Z}^m, \mathcal{G}^m_{\overline{\mathcal{W}^m}})$:

By definition, $\mathcal{X}^m \cap \sigma^m \neq \emptyset$. Let X_x be an element of $\mathcal{X}^m \cap \sigma^m$. We distinguish the cases:

- If σ^m contains an outgoing arrow from X_x i.e. $X_x \to\subseteq \sigma^m$. Then this arrow is not deleted by the mutilation $\overline{\mathcal{X}^m(\mathcal{Z}^m)}$. It exists in $\mathcal{G}^m_{\overline{\mathcal{W}^m},\overline{\mathcal{X}^m(\mathcal{Z}^m)}}$.
- If σ^m contains an incoming arrow to X_x i.e. $\to X_x \subseteq \sigma^m$. Since $\mathrm{Root}(\sigma^m) \subseteq (\mathcal{W}^m \cup \mathcal{Z}^m) \cup \mathcal{Y}^m$, we know that X_x is not a root. Let R_r be a root corresponding to X_x i.e. an element of $\mathrm{Desc}(X_x, \sigma^m) \cap \mathrm{Root}(\sigma^m)$. Since $\sigma^m \subseteq \mathcal{G}^m_{\overline{\mathcal{W}^m}}$, we know that $R_r \notin \mathcal{W}^m$. Thus, $R_r \in \mathcal{Z}^m \cup \mathcal{Y}^m$. We distinguish two cases:
  - If $R_r \in \mathcal{Z}^m$. Then $X_x \notin \mathcal{X}^m(\mathcal{Z}^m)$. Therefore, $\to X_x$ is not deleted by the mutilation $\overline{\mathcal{X}^m(\mathcal{Z}^m)}$ and it exists in $\mathcal{G}^m_{\overline{\mathcal{W}^m},\overline{\mathcal{X}^m(\mathcal{Z}^m)}}$.
  - Otherwise, $R_r \in \mathcal{Y}^m$. Thus, σ^m contains a proper causal path from $\mathcal{X}^m$ to $\mathcal{Y}^m$. We update σ^m to be this path. Now σ^m have no incoming arrows on $\mathcal{X}^m$, thus it exists in $\mathcal{G}^m_{\overline{\mathcal{W}^m},\overline{\mathcal{X}^m(\mathcal{Z}^m)}}$.

Therefore, we can assume that $\sigma^m \subseteq \mathcal{G}^m_{\overline{\mathcal{W}^m},\overline{\mathcal{X}^m(\mathcal{Z}^m)}}$. Therefore, by Theorem 1, $\mathcal{Y}^m \not\perp\!\!\!\perp_{\mathcal{G}^m_{\overline{\mathcal{W}^m},\overline{\mathcal{X}^m(\mathcal{Z}^m)}}} X^m \mid \mathcal{W}^m, \mathcal{Z}^m$, i.e. the third rule of Pearl's do-calculus does not hold in $\mathcal{G}^m$. □

A.5 Proof of Theorem 4

Theorem 4 presents three equivalences, one for each rule of do-calculus. To streamline the proofs, we introduce Corollary 3, which restates Theorem 2 in a form better suited to treating all three rules in a uniform manner.

Corollary 3. *Let $\mathcal{G}^C$ be a C-DAG. Let $\mathcal{W}^C, \mathcal{X}^C, \mathcal{Y}^C$ and $\mathcal{Z}^C$ be pairwise disjoint subsets of nodes. Let $\mathcal{W}^C$, $\mathcal{X}^C$, $\mathcal{Y}^C$ and $\mathcal{Z}^C$ be pairwise disjoint subsets of nodes. For $i \in \{1, 2, 3\}$, let $R_i(\mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{Z})$ be the i^{th} rule of Pearl's Calculus applied to $(\mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{Z})$, and say it "does not holds in $\mathcal{G}$" whenever its associated d-separation condition in the associated mutilated graph is not satisfied. The following propositions are equivalent:*

- *There exists $\mathcal{G}^m \in C(\mathcal{G}^C)$ in which $R_i(\mathcal{W}^m, \mathcal{X}^m, \mathcal{Y}^m, \mathcal{Z}^m)$ does not hold.*
- *$\mathcal{G}^m_{u\,\overline{\mathcal{W}^m}, \underline{\mathcal{M}^m_i}}$ contains a structure of interest σ^m such that $\mathcal{X}^m \not\perp\!\!\!\perp_{\sigma^m} \mathcal{Y}^m \mid \mathcal{W}^m, \mathcal{Z}^m$ and $\mathcal{G}^m_{can} \cup \sigma^m$ is acyclic and $\mathrm{Root}(\sigma^m) \subseteq \mathcal{R}^m_i$.*

where $\mathcal{M}^m_i = \begin{cases} \mathcal{X}^m & \text{if } i = 2, \\ \emptyset & \text{otherwise.} \end{cases}$ *and* $\mathcal{R}^m_i = \begin{cases} (\mathcal{W}^m \cup \mathcal{Z}^m) \cup \mathcal{Y}^m & \text{if } i = 3, \\ (\mathcal{W}^m \cup \mathcal{Z}^m) \cup \mathcal{X}^m \cup \mathcal{Y}^m & \text{otherwise.} \end{cases}$.

Proof. Directly follows from Theorem 2. □

In order to prove Theorem 4, we need to prove Lemma 4 on structures of interest.

Lemma 4. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a mixed graph. Let $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ be pairwise disjoint subsets of $\mathcal{V}$. Let $\sigma \subseteq \mathcal{G}$ be a structure of interest such that $\mathcal{X} \not\perp\!\!\!\perp_\sigma \mathcal{Y} \mid \mathcal{Z}$. Then there exists a structure of interest $\sigma' \subseteq \sigma$ such that $\mathcal{X} \not\perp\!\!\!\perp_{\sigma'} \mathcal{Y} \mid \mathcal{Z}$ and such that $\#\sigma' \cap \mathcal{X} = 1$ and $\#\sigma' \cap \mathcal{Y} = 1$.*

Proof. $\mathcal{X} \not\perp\!\!\!\perp_\sigma \mathcal{Y} \mid \mathcal{Z}$, thus σ contains a d-connecting path π from $\mathcal{X}$ to $\mathcal{Y}$. For all collider C on π, σ contains a directed path π_C from C to $\mathcal{Z}$. We can assume, without loss of generality, that for all collider C, π_C does not encounter $\mathcal{X}$. Indeed, let $C^\star$ denote the last collider on π such that $\pi_{C^\star}$ encounters $\mathcal{X}$. Let X be the first encounter of $\mathcal{X}$ and $\pi_{C^\star}$. We just need to consider the path $\pi' := \pi_{C^\star[C^\star, X]} \cup \pi_{[C^\star, \mathcal{Y}]}$. Similarly, we can assume, without loss of generality that for all collider C, π_C does not encounter $\mathcal{Y}$. We apply the construction of Lemma 1. By doing so, we obtain a structure of interest $\sigma' \subseteq \pi \cup \bigcup_{C \text{ collider on } \pi} \pi_C \subseteq \sigma$ such that $\mathcal{X} \not\perp\!\!\!\perp_{\sigma'} \mathcal{Y} \mid \mathcal{Z}$. Therefore, $\#\sigma' \cap \mathcal{X} = 1$ and $\#\sigma' \cap \mathcal{Y} = 1$. □

Theorem 4 (Infinity is at most three)**.** *Let $\mathcal{G}^C$ be a C-DAG and $\mathcal{G}^C_{\leq 3}$ be the corresponding C-DAG where all clusters of size greater than 3 are reduced to size 3. Let $\mathcal{W}^C$, $\mathcal{X}^C$, $\mathcal{Y}^C$ and $\mathcal{Z}^C$ be pairwise disjoint subsets of nodes. For $i \in \{1, 2, 3\}$, let $R_i(\mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{Z})$ be the i^{th} rule of Pearl's Calculus applied to $(\mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{Z})$, and say it "does not holds in $\mathcal{G}$" whenever its associated d-separation condition in the associated mutilated graph is not satisfied. The following propositions are equivalent:*

1. *There exists $\mathcal{G}^m \in C(\mathcal{G}^C)$ in which $R_i(\mathcal{W}^m, \mathcal{X}^m, \mathcal{Y}^m, \mathcal{Z}^m)$ does not hold.*
2. *There exists $\mathcal{G}^m_{\leq 3} \in C(\mathcal{G}^C_{\leq 3})$ in which $R_i(\mathcal{W}^m_{\leq 3}, \mathcal{X}^m_{\leq 3}, \mathcal{Y}^m_{\leq 3}, \mathcal{Z}^m_{\leq 3})$ does not hold.*

Where $\mathcal{W}^m_{\leq 3}$, $\mathcal{X}^m_{\leq 3}$, $\mathcal{Y}^m_{\leq 3}$ and $\mathcal{Z}^m_{\leq 3}$ are the sets of nodes corresponding to $\mathcal{W}^C$, $\mathcal{X}^C$, $\mathcal{Y}^C$ and $\mathcal{Z}^C$ in $\mathcal{G}^C_{\leq 3}$

Proof. We prove the two implications.

Proof of $2 \Rightarrow 1$**:**

We add as many vertices without arrow as necessary to construct a graph compatible with $\mathcal{G}^C$ in which $R_i(\mathcal{W}^m, \mathcal{X}^m, \mathcal{Y}^m, \mathcal{Z}^m)$ does not hold.

Proof of $1 \Rightarrow 2$**:** (Figure 9 illustrates the key steps of this implication with a concrete example.)

By Corollary 3, $\mathcal{G}^m_{\mathrm{u}\,\overline{\mathcal{W}^m},\underline{\mathcal{M}_i^m}}$ contains a structure of interest σ^m such that $\mathcal{X}^m \not\perp\!\!\!\perp_{\sigma^m} \mathcal{Y}^m \mid \mathcal{W}^m, \mathcal{Z}^m$ and $\mathcal{G}^m_{\mathrm{can}} \cup \sigma^m$ is acyclic and $\mathrm{Root}(\sigma^m) \subseteq \mathcal{R}_i^m$ where $\mathcal{M}_i^m = \begin{cases} \mathcal{X}^m & \text{if } i = 2, \\ \emptyset & \text{otherwise.} \end{cases}$ and $\mathcal{R}_i^m = \begin{cases} (\mathcal{W}^m \cup \mathcal{Z}^m) \cup \mathcal{Y}^m & \text{if } i = 3, \\ (\mathcal{W}^m \cup \mathcal{Z}^m) \cup \mathcal{X}^m \cup \mathcal{Y}^m & \text{otherwise.} \end{cases}$.

Let σ^m be such a structure of interest. By Lemma 4, we can assume that $\#\sigma^m \cap \mathcal{X}^m = 1$ and $\#\sigma^m \cap \mathcal{Y}^m = 1$. Let V^C be a cluster. We split $\sigma^m \cap V^m$ in two subsets as follows:

$$\mathcal{F} := \{V_v \in \sigma^m \cap V^m \mid V_v \text{ has no incoming arrows in } \sigma^m\}$$

$$\mathcal{NF} := \sigma^m \cap V^m \setminus \mathcal{F}$$

We will show that we can assume that $\#\mathcal{NF} \leq 1$ without loss of generality. Consider the case where $\#\mathcal{NF} \geq 2$. Necessarily, since $\#\sigma^m \cap \mathcal{X}^m = 1$ and $\#\sigma^m \cap \mathcal{Y}^m = 1$, we know that V^C is different from any cluster in $\mathcal{X}^C \cup \mathcal{Y}^C$. Let $V_{\max \mathcal{NF}}$ denote the element of $\mathcal{NF}$ with maximal index. We distinguish two cases:

- If $\mathcal{NF}$ contains a root of σ^m. Since $\mathcal{X}^m \not\perp\!\!\!\perp_{\sigma^m} \mathcal{Y}^m \mid \mathcal{W}^m, \mathcal{Z}^m$, we know that $\mathcal{NF} \subseteq (\mathcal{W}^m \cup \mathcal{Z}^m) \cup \mathcal{X}^m \cup \mathcal{Y}^m$. Since V^C is different from any cluster in $\mathcal{X}^C \cup \mathcal{Y}^C$, we can conclude that $\mathcal{NF} \subseteq \mathcal{W}^m \cup \mathcal{Z}^m$. Since, $\mathcal{X}^m \not\perp\!\!\!\perp_{\sigma^m} \mathcal{Y}^m \mid \mathcal{W}^m, \mathcal{Z}^m$, we know that $(\sigma^m \setminus \mathrm{Root}(\sigma^m)) \cap (\mathcal{W}^m \cup \mathcal{Z}^m) = \emptyset$. Therefore, all elements in $\mathcal{NF}$ are roots in σ^m. By Proposition 7 and since mutilations are done at cluster level, we know that for all arrows $W_w \to V_v$ with $V_v \in \mathcal{NF}$, $\mathcal{G}^m_{\mathrm{u}\,\overline{\mathcal{W}^m},\underline{\mathcal{M}_i^m}}$ also contains the arrow $W_w \to V_{\max \mathcal{NF}}$. Therefore, $\mathcal{G}^m_{\mathrm{u}\,\overline{\mathcal{W}^m},\underline{\mathcal{M}_i^m}}$ also contains a structure of interest $\sigma^{m\prime}$ which is equal to σ^m except that all arrows $W_w \to V_v$ with $V_v \in \mathcal{NF}$ are now pointing toward $V_{\max \mathcal{NF}}$. Therefore, $V_{\max \mathcal{NF}}$ is the only element of $\sigma^{m\prime} \cap V^m$ that has incoming arrows. Therefore, in this case, we can assume that $\#\mathcal{NF} = 1$.
- Otherwise, every element in $\mathcal{NF}$ has an outgoing arrow in σ^m. Since σ^m is a structure of interest, we know that σ^m contains a path π^m from $\mathcal{X}^m$ to $\mathcal{Y}^m$. We will construct a graph $\sigma^{m\prime}$, satisfying the following conditions:
  - $\sigma^m \subseteq \sigma^{m\prime} \subseteq \mathcal{G}^m_{\mathrm{u}\,\overline{\mathcal{W}^m},\underline{\mathcal{M}_i^m}}$
  - $\mathrm{Root}(\sigma^{m\prime}) = \mathrm{Root}(\sigma^m)$
  - $\sigma^{m\prime}$ differs from σ^m only inside V^m.

  and such that $\sigma^{m\prime}$ contains a path $\pi^{m\prime}$ from $\mathcal{X}^m$ to $\mathcal{Y}^m$ which encounters $\mathcal{NF}$ at most once with a chain or a collider and that this intersection occurs a $V_{\max \mathcal{NF}}$. If σ^m and π^m does not satisfy these conditions, let us consider a_1 and a_2 be respectively the first and last arrows of π^m in $\mathcal{NF}$. We distinguish the cases:

  1. If both a_1 and a_2 are incoming arrows in $\mathcal{NF}$, i.e $\pi^m = \cdots W_w \bullet\!\!\rightarrow V^1_{v^1} \cdots V^2_{v^2} \leftarrow\!\!\bullet U_u \cdots$. By Proposition 7 and since mutilations are done at cluster level, we know that $\mathcal{G}^m_{\mathrm{u}\,\overline{\mathcal{W}^m},\underline{\mathcal{M}_i^m}}$ contains the arrows $W_w \bullet\!\!\rightarrow V_{\max \mathcal{NF}}$ and $V_{\max \mathcal{NF}} \leftarrow\!\!\bullet U_u$. We consider $\sigma^{m\prime} := \sigma^m \cup \{W_w \bullet\!\!\rightarrow V_{\max \mathcal{NF}}\} \cup \{V_{\max \mathcal{NF}} \leftarrow\!\!\bullet U_u\}$ and $\pi^{m\prime} := \pi^m_{[\mathcal{X}^m, W_w]} \cup \{W_w \bullet\!\!\rightarrow V_{\max \mathcal{NF}} \leftarrow\!\!\bullet U_u\} \cup \pi^m_{[U_u, \mathcal{Y}^m]}$. Note that $\pi^{m\prime}$ encounters $\mathcal{NF}$ only at $V_{\max \mathcal{NF}}$ and that $\mathrm{Root}(\sigma^{m\prime}) = \mathrm{Root}(\sigma^m)$.
  2. If both a_1 and a_2 are outgoing arrows in $\mathcal{NF}$, i.e $\pi^m = \cdots W_w \leftarrow\!\!\bullet V^1_{v^1} \cdots V^2_{v^2} \bullet\!\!\rightarrow U_u \cdots$. By Proposition 6 and since mutilations are done at cluster level, we know that $\mathcal{G}^m_{\mathrm{u}\,\overline{\mathcal{W}^m},\underline{\mathcal{M}_i^m}}$ contains the arrows $W_w \leftarrow\!\!\bullet V_1$ and $V_1 \bullet\!\!\rightarrow U_u$. We consider $\sigma^{m\prime} := \sigma^m \cup \{W_w \leftarrow\!\!\bullet V_1\} \cup \{V_1 \bullet\!\!\rightarrow U_u\}$ and $\pi^{m\prime} := \pi^m_{[\mathcal{X}^m, W_w]} \cup \{W_w \leftarrow\!\!\bullet V_1 \bullet\!\!\rightarrow U_u\} \cup \pi^m_{[U_u, \mathcal{Y}^m]}$. Note that $\pi^{m\prime}$ does not encounter $\mathcal{NF}$ and that $\mathrm{Root}(\sigma^{m\prime}) = \mathrm{Root}(\sigma^m)$.
  3. Otherwise, without loss of generality, we can assume that a_1 and a_2 are pointing towards $\mathcal{Y}^m$, i.e. $\pi^m = \cdots W_w \bullet\!\!\rightarrow V^1_{v^1} \cdots V^2_{v^2} \bullet\!\!\rightarrow U_u \cdots$. Indeed, otherwise, we just need to consider the path from $\mathcal{Y}^m$ to $\mathcal{X}^m$. Note that $V^1_{v^1}$ or $V^2_{v^2}$ could be equal to $V_{\max \mathcal{NF}}$ but not both. Since all elements in $\mathcal{NF}$ have an outgoing arrow, let us consider C_c, the child

of $V_{\max \mathcal{NF}}$ in σ^m. By Propositions 6, 7 and since mutilations are done at cluster level, we know that $\mathcal{G}^m_{\mathrm{u}\overline{\mathcal{W}^m},\underline{\mathcal{M}^m_i}}$ contains the arrows $W_w \bullet\!\!\rightarrow V_{\max \mathcal{NF}}$ and $C_c \leftarrow\!\!\bullet V^2_{v^2}$. We consider $\sigma^{m\prime} := \sigma^m \cup \{W_w \bullet\!\!\rightarrow V_{\max \mathcal{NF}}\} \cup \{C_c \leftarrow\!\!\bullet V^2_{v^2}\}$ and $\pi^{m\prime} := \pi^m_{[\mathcal{X}^m, W_w]} \cup \{W_w \bullet\!\!\rightarrow V_{\max \mathcal{NF}} \rightarrow C_c \leftarrow\!\!\bullet V^2_{v^2}\} \cup \pi^m_{[V^2_{v^2}, \mathcal{Y}^m]}$. Note that $\pi^{m\prime}$ encounters $\mathcal{NF}$ only once with a collider in $V_{\max \mathcal{NF}}$ and once with a fork in $V^2_{v^2}$ and that $\mathrm{Root}(\sigma^{m\prime}) = \mathrm{Root}(\sigma^m)$.

Note that in all cases, we have used Propositions 6 and 7, thus $\sigma^{m\prime} \cup \mathcal{G}^m_{\mathrm{can}}$ is acyclic. Moreover, in all cases, $\mathrm{Root}(\sigma^{m\prime}) = \mathrm{Root}(\sigma^m)$. Therefore, $\mathrm{Root}(\sigma^{m\prime}) \subseteq \mathcal{R}^m_i$. In addition, $\sigma^{m\prime}$ contains $\pi^{m\prime}$, a path from $\mathcal{X}^m$ to $\mathcal{Y}^m$ which encounters $\mathcal{NF}$ at most once with a chain or a collider and that this intersection occurs a $V_{\max \mathcal{NF}}$.

However, in Cases 2 and 3, we have added outgoing arrows to some vertices different from $V_{\max \mathcal{NF}}$. This could prevent $\sigma^{m\prime}$ from being a structure of interest. We apply the following transformation to construct a structure of interest from $\sigma^{m\prime}$:

1. **Move all incoming arrows to $\mathbf{V}_{\max \mathcal{NF}}$:** By Proposition 7 and since mutilations are done at cluster level, we know that for all arrows $W_w \rightarrow V_v$ with $V_v \in NF$, $\mathcal{G}^m_{\mathrm{u}\overline{\mathcal{W}^m},\underline{\mathcal{M}^m_i}}$ also contains the arrow $W_w \rightarrow V_{\max \mathcal{NF}}$. Therefore, $\mathcal{G}^m_{\mathrm{u}\overline{\mathcal{W}^m},\underline{\mathcal{M}^m_i}}$ also contains $\sigma^{m\prime\prime}$ which is equal to $\sigma^{m\prime}$ except that all arrows $W_w \rightarrow V_v$ with $V_v \in NF$ are now pointing toward $V_{\max \mathcal{NF}}$. Since we are using Proposition 7, we know that $\sigma^{m\prime\prime} \cup \mathcal{G}^m_{\mathrm{can}}$ is acyclic. Note that $\pi^{m\prime}$ still exists in $\sigma^{m\prime\prime}$ and that all vertices in $\sigma^{m\prime\prime} \cap V^m$ except $V_{\max \mathcal{NF}}$ have no incoming arrows in $\sigma^{m\prime\prime}$. Moreover, note that $\mathrm{Root}(\sigma^{m\prime\prime}) = \mathrm{Root}(\sigma^{m\prime})$.
2. **Remove problematic outgoing arrows:** To keep the notations simple, we update $\sigma^{m\prime} \leftarrow \sigma^{m\prime\prime}$. Some vertices, different from $V_{\max \mathcal{NF}}$, may have more than two outgoing arrows, preventing $\sigma^{m\prime}$ from being a structure of interest. We remove from $\sigma^{m\prime}$, all outgoing arrow from V^m that is not used by $\pi^{m\prime}$. Since, $\pi^{m\prime}$ uses at most two arrows around a vertex, we know that all vertices have now at most two outgoing arrows. Since we have just removed some arrows, we know that $\sigma^{m\prime} \cup \mathcal{G}^m_{\mathrm{can}}$ remains acyclic. Moreover, since $\pi^{m\prime}$ is preserved, we know that $\mathcal{X}^m$ and $\mathcal{Y}^m$ are still connected.
3. **Remove the problematic vertices:** At the end of the previous steps, some vertices are not connected to the others in $\sigma^{m\prime}$, preventing $\sigma^{m\prime}$ from being a structure of interest. More precisely, these vertices are $\mathcal{NF} \setminus \{V_{\max \mathcal{NF}}\}$ except V_1 in case 2 and except $V^2_{v^2}$ in case 3. Indeed, at step 1, they have lost their incoming arrows and at step 2, they have lost their outgoing arrows. We remove these vertices from $\sigma^{m\prime}$. By doing so, $\sigma^{m\prime} \cup \mathcal{G}^m_{\mathrm{can}}$ remains acyclic and $\sigma^{m\prime}$ is now a structure of interest, $\mathrm{Root}(\sigma^{m\prime}) = \mathrm{Root}(\sigma^m) \subseteq \mathcal{R}^m_i$.

To summarize, we have constructed a structure of interest $\sigma^{m\prime}$ in $\mathcal{G}^m_{\mathrm{u}\overline{\mathcal{W}^m},\underline{\mathcal{M}^m_i}}$, such that $\mathcal{X}^m \not\perp\!\!\!\perp_{\sigma^{m\prime}} \mathcal{Y}^m \mid \mathcal{W}^m, \mathcal{Z}^m$, $\mathcal{G}^m_{\mathrm{can}} \cup \sigma^{m\prime}$ is acyclic and $\mathrm{Root}(\sigma^{m\prime}) \subseteq \mathcal{R}^m_i$. Moreover, $V_{\max \mathcal{NF}}$ is the only element of $V^m \cap \sigma^{m\prime}$ with incoming arrows. Therefore, in this case, we can assume that $\#\mathcal{NF} = 1$.

Therefore, in all cases, we can assume that $\#\mathcal{NF} \leq 1$ without loss of generality.

We will now show that we can assume that $\mathcal{F} \subseteq \{V_1\}$ without loss of generality. If it is not the case we apply the following transformations:

1. **Move all arrows to V_1:** By Proposition 6 and since mutilations are done at cluster level, for all arrow $V_v \rightarrow W_w$ in $\mathcal{F} \cap \sigma^m$, we know that $\mathcal{G}^m_{\mathrm{u}\overline{\mathcal{W}^m},\underline{\mathcal{M}^m_i}}$ contains the arrow $V_1 \rightarrow W_w$. Thus, we consider $\sigma^{m\prime}$ the subgraph of $\mathcal{G}^m_{\mathrm{u}\overline{\mathcal{W}^m},\underline{\mathcal{M}^m_i}}$ obtained by moving all the arrows $V_v \rightarrow W_w$ with V_v in $\mathcal{F} \cap \sigma^m$ to $V_1 \rightarrow W_w$. Since, σ^m is a structure of interest, it contains a path π^m from $\mathcal{X}^m$ to $\mathcal{Y}^m$. Note that the above transformation yields a path $\pi^{m\prime}$ in $\sigma^{m\prime}$ which connects $\mathcal{X}^m$ and $\mathcal{Y}^m$.
2. **Remove problematic outgoing arrows:** In $\sigma^{m\prime}$, V_1 may have more than two outgoing arrows, preventing $\sigma^{m\prime}$ from being a structure of interest. Yet $\pi^{m\prime}$ uses at most two of these arrows. Thus, we only keep two of them without altering $\pi^{m\prime}$.

3. **Remove the problematic vertices:** At the end of the previous steps, some vertices are not connected to the others in $\sigma^{m\prime}$, preventing $\sigma^{m\prime}$ from being a structure of interest. More precisely, these vertices are $\mathcal{F} \setminus \{V_1\}$. Indeed, they have lost their outgoing arrow at step 1 and had no incoming arrows are they are in $\mathcal{F}$.

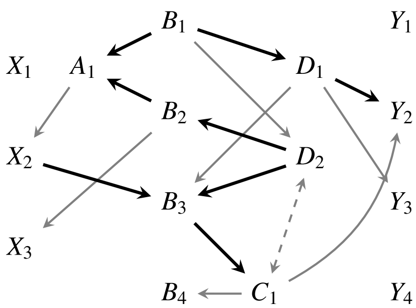
To summarize, we have constructed a structure of interest $\sigma^{m\prime}$ in $\mathcal{G}^m_{\mathrm{u}\,\overline{\mathcal{W}^m},\overline{\mathcal{M}^m_i}}$, such that $\mathcal{X}^m \not\perp\!\!\!\perp_{\sigma^{m\prime}} \mathcal{Y}^m \mid \mathcal{W}^m, \mathcal{Z}^m, \mathcal{G}^m_{\mathrm{can}} \cup \sigma^{m\prime}$ is acyclic and $\mathrm{Root}(\sigma^{m\prime}) \subseteq \mathcal{R}^m_i$. Moreover, V_1 is the only element of $V^m \cap \sigma^{m\prime}$ with no incoming arrows. Therefore, we can assume that $\mathcal{F} \subseteq \{V_1\}$.

End of the proof of $1 \Rightarrow 2$**:** Without loss of generality, we assume that $\#\mathcal{NF} \leq 1$ and $\mathcal{F} \subseteq \{V_1\}$. Let $V_{\mathcal{NF}}$ be the only element of $\mathcal{NF}$ (if exists). We consider the graph $\mathcal{G}^m := \mathcal{G}^m_{\mathrm{can}} \cup \sigma^m$. By Corollary 3, $R_i(\mathcal{W}^m, \mathcal{X}^m, \mathcal{Y}^m, \mathcal{Z}^m)$ does not hold in $\mathcal{G}^m$ and by construction $\mathcal{G}^m$ is compatible with $\mathcal{G}^C$. For every vertex in $V^m \setminus (\mathcal{F} \cup \mathcal{NF})$, we apply Proposition 6 to move all outgoing arrows to V_1 and Proposition 7 to move all incoming arrows to $V_{\#V^C}$. This yields a graph $\mathcal{G}^{m\prime} \in C(\mathcal{G}^C)$ in which $R_i(\mathcal{W}^m, \mathcal{X}^m, \mathcal{Y}^m, \mathcal{Z}^m)$ does not hold, and where no vertex outside $\mathcal{NF} \cup \{V_1\} \cup \{V_{\#V^C}\}$ is incident to an arrow.

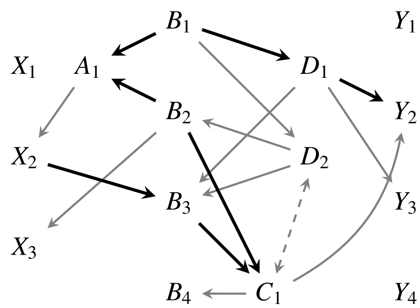
We repeat this construction for each cluster, ultimately obtaining $\mathcal{G}^{m\star}$, where $R_i(\mathcal{W}^m, \mathcal{X}^m, \mathcal{Y}^m, \mathcal{Z}^m)$ still fails and every cluster has arrows on at most three vertices. Finally, by removing all vertices that are not incident to any arrow in $\mathcal{G}^{m\star}$, we obtain $\mathcal{G}^{m\star}_{\leq 3}$, a graph compatible with $\mathcal{G}^C{}_{\leq 3}$ in which $R_i(\mathcal{W}^m_{\leq 3}, \mathcal{X}^m_{\leq 3}, \mathcal{Y}^m_{\leq 3}, \mathcal{Z}^m_{\leq 3})$ does not hold.

Therefore, there exists $\mathcal{G}^m_{\leq 3} \in C(\mathcal{G}^C_{\leq 3})$ in which $R_i(\mathcal{W}^m_{\leq 3}, \mathcal{X}^m_{\leq 3}, \mathcal{Y}^m_{\leq 3}, \mathcal{Z}^m_{\leq 3})$ does not hold. □

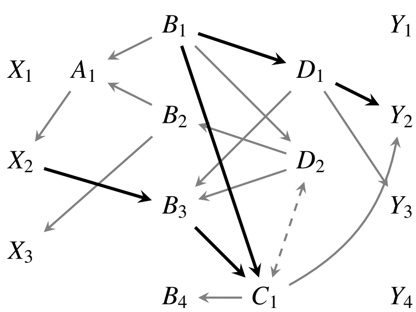
We come back to the example presented in Figure 3 to illustrate the successive steps of this proof in Figure 9. Particularly, Figure 9a is the initial graph, then in Figure 9b we consider an analogous structure of interest such that $\#\mathcal{NF} \leq 1$. Then, in Figure 9c, we consider an analogous structure of interest such that $\mathcal{F} \subseteq \{V_1\}$. Finally, in Figure 9d we move every unused gray arrows such that only three vertices of B^m are incident to any arrows. From this graph, we deduce $\mathcal{G}^m_{\leq 3}$ in Figure 9e on which d-connection with structures of interests are equivalent. We apply equivalent rules to get 9f, which corresponds to Figure 3.



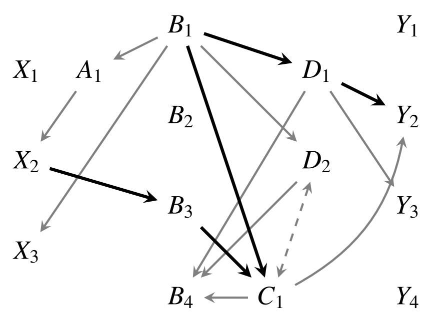
(a) $\mathcal{G}^m \in \mathcal{C}(\mathcal{G}^C)$



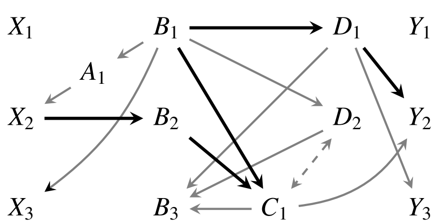
(b) Application of Proposition 6 to the arrow $B_3 \to C_1$, thereby adding $B_2 \to C_1$. Note that now $\#\mathcal{NF} = 1$.



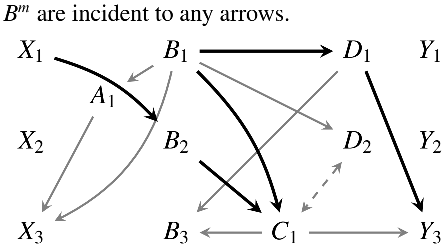
(c) Application of Proposition 6 to shift $B_2 \to C_1$ to $B_1 \to C_1$. Note that now $\mathcal{F} \subseteq \{V_1\}$.



(d) Application of Propositions 6 and 7 to reposition all unused (gray) arrows. Note that only three vertices of B^m are incident to any arrows.



(e) Remove some vertices that are not incident to any arrow to get $\mathcal{G}^m_{\leq 3}$. Note that $\mathcal{G}^m_{\leq 3} \in \mathcal{C}(\mathcal{G}^C_{\leq 3})$.



(f) Application of Propositions 6 and 7 to get the same graph of Figure 3d.

Figure 9: Figure 9a shows the graph $\mathcal{G}^m$ containing the structure of interest σ^m (in bold black), which connects X^m and Y^m under $C^m \cup A^m$. Figures 9b, 9c and 9d illustrate the successive transformations of $\mathcal{G}^m$ and σ^m (as carried out in the proof of Theorem 4) for the cluster B^C. Figure 9e shows the last step of the proof. Figure 9f shows how to transform the graph in Figure 9e to get the graph in Figure 3d.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Main claims are listed in the introduction at line 62. They correspond to the outline of the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in the conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recog nition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: All assumptions are described and all proofs are given in the Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short Sketch of proof to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [NA]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [NA]

Justification: The paper does not include experiments requiring code.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [NA]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [NA]

   Justification: The paper does not include experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper conform, in every respect, with the NeurIPS Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The paper is a theoretical study of completeness of calculus.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.