

# Testing Causal Models with Hidden Variables in Polynomial Delay via Conditional Independencies

Hyunchai Jeong<sup>\*1</sup>, Adiba Ejaz<sup>\*2</sup>, Jin Tian<sup>3</sup>, Elias Bareinboim<sup>2</sup>

<sup>1</sup>Purdue University

<sup>2</sup>Columbia University

<sup>3</sup>Mohamed bin Zayed University of Artificial Intelligence

jeong3@purdue.edu, adiba.ejaz@cs.columbia.edu, jin.tian@mbzuai.ac.ae, eb@cs.columbia.edu

## Abstract

Testing a hypothesized causal model against observational data is a key prerequisite for many causal inference tasks. A natural approach is to test whether the conditional independence relations (CIs) assumed in the model hold in the data. While a model can assume exponentially many CIs (with respect to the number of variables), testing all of them is both impractical and unnecessary. Causal graphs, which encode these CIs in polynomial space, give rise to local Markov properties that enable model testing with a significantly smaller subset of CIs. Model testing based on local properties requires an algorithm to list the relevant CIs. However, existing algorithms for realistic settings with hidden variables and non-parametric distributions can take exponential time to produce even a single CI constraint. In this paper, we introduce the  $c$ -component local Markov property (C-LMP) for causal graphs with hidden variables. Since C-LMP can still invoke an exponential number of CIs, we develop a *polynomial delay* algorithm to list these CIs in poly-time intervals. To our knowledge, this is the first algorithm that enables poly-delay testing of CIs in causal graphs with hidden variables against arbitrary data distributions. Experiments on real-world and synthetic data demonstrate the practicality of our algorithm.

**Code** — <https://github.com/CausalAILab/ListConditionalIndependencies>

## 1 Introduction

Causal models are the daily bread of many fields of research (Pearl 2000; Spirtes, Glymour, and Scheines 2001), but tools for testing them are lacking. In various studies, researchers posit a causal model and use it to compute causal effects from data (Tennant et al. 2020; Hoover 1990; King et al. 2004; Sverchkov and Craven 2017; Robins, Hernan, and Brumback 2000; Rotmensch et al. 2017). The model imposes testable constraints on the statistics of the data collected. Before using the model for causal inference, it’s crucial to test if these constraints are met, and adjust the model as needed (Pearl 1995, 2000; Bareinboim and Pearl 2016; Malinsky 2024; Ankan and Textor 2022).

Causal directed acyclic graphs (DAGs) are one popular model for causal assumptions (Pearl 2000; Spirtes, Glymour,

and Scheines 2001). Conditional independencies (CIs) are the most basic constraint that a causal DAG imposes on observational data. The study of CIs in the context of graphical models dates back to at least the 1980’s (Pearl 1988; Dawid 1979; Spirtes et al. 1998; Pearl 1998; Pearl and Meshkat 1999; Pearl 2000). A classic problem in this line of research is: *given observational data and a hypothesized causal graph, do all the CIs implied by this graph hold in the data?* If the answer is no, the DAG may be revised.

A key idea in the early literature of graphical models was to use a DAG to represent the constraints of probability distributions. A multivariate probability distribution may encode exponentially many CIs with respect to the number of variables. A DAG can encode these CIs in polynomial space. The *d-separation* criterion allows us to derive the CIs encoded in a DAG (Pearl 1988). The *global Markov property* of a DAG is the set of all CIs encoded in it (Pearl 1988). There is also a well-known *local Markov property* for DAGs (Pearl 1988; Lauritzen et al. 1990), which states that each variable must be conditionally independent of its non-descendants given its parents. Since the CI relation is a semi-graphoid, the linearly many CIs of the local Markov property together imply the exponentially many CIs of the global Markov property. This means that to test a DAG against observational data, it suffices to perform a linear number of CI tests as given by the local Markov property. For concreteness, consider the DAG  $\mathcal{G}^1$  in Fig. 1a and assume all variables  $\{A, B, \dots, H, U_1, U_2, U_3\}$  are observed. Though  $\mathcal{G}^1$  encodes 35787 CIs, only 11 need testing by the local Markov property. For example, if we test that  $F \perp\!\!\!\perp \{A, B\} \mid \{C\}$ , we do not need to test that  $F \perp\!\!\!\perp \{A\} \mid \{B, C\}$ , since the former implies the latter by the weak union axiom.

Unobserved confounding is a widespread phenomenon in real-world settings (Fisher 1936). It occurs when a hidden variable causally affects two or more observed variables. The local Markov property can be used to test *Markovian* causal DAGs, which represent models without unobserved confounding. However, it cannot be used to test *non-Markovian* DAGs, which represent models with unobserved confounding. This is because if the parents of a variable are partially unobserved, we cannot test CIs that require conditioning on these parents (Fig. 1a). Since the assumption of no unobserved confounding rarely holds in practice, alternative ways to test non-Markovian DAGs have been developed (Tian and

<sup>\*</sup>These authors contributed equally.

Pearl 2002b; Kang and Tian 2009; Geiger and Meek 1998, 1999; Richardson 2003, 2009; Hu and Evans 2023). Despite their power, these works either (a) make strong assumptions on the DAG or probability distribution, or (b) do not provide an algorithm to query their required CI tests in poly-time intervals, with naive algorithms taking exponential time to output a single CI constraint.

**Summary of contributions.** We give the first efficient algorithm for testing causal DAGs with hidden variables via conditional independencies. This enables researchers to test their causal assumptions using observational data prior to inference. Importantly, our approach extends to arbitrary data distributions and networks of unobserved confounding.

This result builds on a newer, fine-grained characterization of CIs in graphs based on a new construct called ancestral  $c$ -components (i.e., connected components in the bidirected skeleton). In particular, we show that  $O(n2^s)$  CI tests (Prop. 1) are required to test a DAG on  $n$  variables whose largest  $c$ -component has size  $s$ . This is an exponential improvement over naively testing all  $\Theta(4^n)$  CI constraints encoded in the DAG. The upshot is largest for DAGs with many variables but small  $c$ -components. For instance, the DAG  $\mathcal{G}^2$  in Fig. 1b implies 753 CIs, but only 5 really need testing. More specifically, our contributions are as follows:

1. We introduce the  $c$ -component local Markov property, or C-LMP (Def. 5). We show that C-LMP and the global Markov property are equivalent, admitting the same set of probability distributions for a given DAG. We then show an important property of C-LMP: a one-to-one mapping between the CI constraints it invokes and *ancestral  $c$ -components* (Thm. 2).
2. Building on this characterization, we develop the first algorithm (LISTCI) capable of listing all testable CI constraints of C-LMP in *polynomial delay* (Thm. 3). On a DAG with  $n$  nodes and  $m$  edges, LISTCI takes  $O(n^2(n+m))$  time to return each new CI constraint, if one exists, or exit when it has exhausted all CI constraints.

Experiments with synthetic data and a real-world protein signaling dataset (Sachs et al. 2005) corroborate the theoretical findings. For the sake of space, proofs are provided in Appendix C.

## 2 Preliminaries

**Notation.** We use capital letters to denote variables ( $X$ ), small letters for their values ( $x$ ), and bold letters for sets of variables ( $\mathbf{X}$ ) and their values ( $\mathbf{x}$ ). The probability distribution over a set of variables  $\mathbf{X}$  is denoted by  $P(\mathbf{X})$ . We consistently use  $P(\mathbf{x})$  as abbreviations for probabilities  $P(\mathbf{X} = \mathbf{x})$ . For disjoint sets of variables  $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ , we use  $\mathbf{X} \perp\!\!\!\perp \mathbf{Y} \mid \mathbf{Z}$  to denote that  $\mathbf{X}$  and  $\mathbf{Y}$  are conditionally independent given  $\mathbf{Z}$ .

**Structural causal models.** The basic framework of our analysis rests on *structural causal models* (SCMs) (Pearl 2000, Def. 7.1.1). An SCM  $\mathcal{M}$  is a quadruple  $\mathcal{M} = \langle \mathbf{V}, \mathbf{U}, \mathcal{F}, P(\mathbf{u}) \rangle$  where  $\mathbf{V}$  and  $\mathbf{U}$  are sets of endogenous and exogenous variables, respectively.  $\mathcal{F}$  is a set of functions: each  $V \in \mathbf{V}$  is a function  $f_V(\mathbf{PA}_V, \mathbf{U}_V)$  of its endogenous and exogenous parents,  $\mathbf{PA}_V \subseteq \mathbf{V}$  and  $\mathbf{U}_V \subseteq \mathbf{U}$

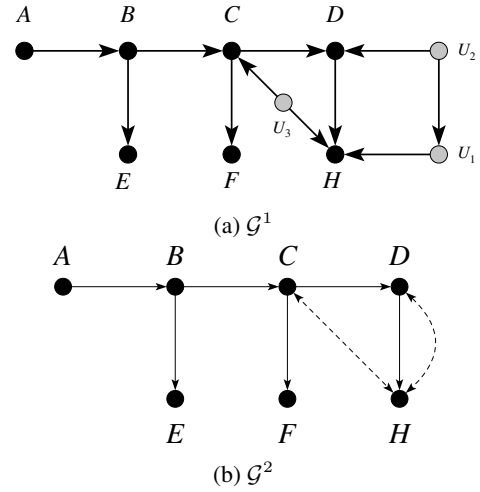


Figure 1: (a) A causal DAG  $\mathcal{G}^1$  in which the local Markov property implies the CI:  $H \perp\!\!\!\perp \{A, B, C, E, F\} \mid \{D, U_1, U_3\}$ . If  $U_1$  and  $U_3$  are unobserved, we cannot test this CI. (b) We project  $\mathcal{G}^1$  onto its observed variables to get  $\mathcal{G}^2$ . In  $\mathcal{G}^2$ , the  $c$ -component local Markov property invokes the testable CI:  $H \perp\!\!\!\perp \{A, E, F\} \mid \{B, C, D\}$ .

respectively.  $P(\mathbf{u})$  is a joint distribution over  $\mathbf{U}$ . Each SCM  $\mathcal{M}$  induces an observed distribution  $P(\mathbf{v})$  over  $\mathbf{V}$ . An SCM is said to be *Markovian* if  $\mathbf{U}_V, \mathbf{U}_W$  are independent for every distinct  $V, W \in \mathbf{V}$ , and *non-Markovian* otherwise. For a more detailed survey on SCMs, we refer to (Pearl 2000; Bareinboim et al. 2022).

**Causal graphs.** The causal graph  $\mathcal{G}$  for an SCM  $\mathcal{M} = \langle \mathbf{V}, \mathbf{U}, \mathcal{F}, P(\mathbf{u}) \rangle$  is constructed as follows: (1) add a vertex for every  $V \in \mathbf{V}$  (2) add an edge  $V_i \rightarrow V_j$  for every  $V_i, V_j \in \mathbf{V}$  if  $V_i \in \mathbf{PA}_{V_j}$  (3) add a dashed bidirected edge between  $V_i, V_j$  if  $\mathbf{U}_i, \mathbf{U}_j$  are correlated or  $\mathbf{U}_i \cap \mathbf{U}_j \neq \emptyset$ .  $\mathcal{G}$  is said to be Markovian if it contains only directed edges, and semi-Markovian otherwise.

We denote the sets of parents, ancestors, and descendants of  $\mathbf{X}$  (including  $\mathbf{X}$  itself) in  $\mathcal{G}$  as  $Pa(\mathbf{X})$ ,  $An(\mathbf{X})$ , and  $De(\mathbf{X})$ , respectively. The set of non-descendants of  $\mathbf{X}$  in  $\mathcal{G}$  is denoted  $Nd(\mathbf{X}) = \mathbf{V} \setminus De(\mathbf{X})$ , which does not include  $\mathbf{X}$  itself. The set of spouses of  $\mathbf{X}$  in  $\mathcal{G}$  is  $Sp(\mathbf{X}) = \bigcup_{X \in \mathbf{X}} \{Y \mid Y \leftrightarrow X\}$ .  $\mathbf{X}$  is said to be an *ancestral set* if it contains its own ancestors, i.e.,  $\mathbf{X} = An(\mathbf{X})$ . We use  $\mathcal{G}_{\mathbf{X}}$  to denote the induced subgraph of  $\mathcal{G}$  on  $\mathbf{X} \subseteq \mathbf{V}$ . A subscript  $\mathcal{G}'$ , e.g.,  $An(\mathbf{X})_{\mathcal{G}'}$  indicates that the set is computed from the subgraph  $\mathcal{G}'$ . We omit the subscript when clear from context. An ordering  $\mathbf{V}^{\prec}$  on variables  $\mathbf{V}$  is said to be consistent with  $\mathcal{G}$  (i.e., a topological ordering) if for any  $X, Y \in \mathbf{V}$ ,  $X \prec Y$  implies  $Y \notin An(X)_{\mathcal{G}}$ . Let  $\mathbf{V}^{\leq X} = \{Y \mid Y \prec X \text{ or } Y = X\}$ .

*Semi-Markovianity vs Non-Markovianity.* A non-Markovian causal DAG  $\mathcal{G}$  can be constructed for a non-Markovian SCM by making the exogenous variables  $\mathbf{U}$  explicit. A non-Markovian DAG with arbitrary hidden variables can be ‘projected’ onto a semi-Markovian causal DAG  $\mathcal{G}'$  which imposes exactly the same CI constraints over the observed variables (Tian and Pearl 2002b). In  $\mathcal{G}'$ ,

each unobserved variable is (i) a parent of at most two observed variables and (ii) made implicit by adding a dashed bidirected edge between its two children. The complexity of the latent structure is irrelevant to the CIs over observed variables. Therefore, we work with semi-Markovian graphs for model testing.

***d*-separation.** A node  $W$  on a path  $\pi$  is said to be a collider on  $\pi$  if  $W$  has converging arrows into  $W$  in  $\pi$ , e.g.,  $\rightarrow W \leftarrow$  or  $\leftrightarrow W \leftarrow$ .  $\pi$  is said to be blocked by a set  $\mathbf{Z}$  if there exists a node  $W$  on  $\pi$  satisfying one of the following two conditions: 1)  $W$  is a collider, and neither  $W$  nor any of its descendants are in  $\mathbf{Z}$ , or 2)  $W$  is not a collider, and  $W$  is in  $\mathbf{Z}$  (Pearl 1988). Given disjoint sets  $\mathbf{X}, \mathbf{Y}$ , and  $\mathbf{Z}$  in  $\mathcal{G}$ ,  $\mathbf{Z}$  is said to *d-separate*  $\mathbf{X}$  from  $\mathbf{Y}$  in  $\mathcal{G}$  if and only if  $\mathbf{Z}$  blocks every path from a node in  $\mathbf{X}$  to a node in  $\mathbf{Y}$  according to the *d*-separation criterion (Pearl 1988). If  $\mathbf{Z}$  *d*-separates  $\mathbf{X}$  from  $\mathbf{Y}$  in  $\mathcal{G}$  (written  $\mathbf{X} \perp_d \mathbf{Y} \mid \mathbf{Z}$ ), then  $\mathbf{X}$  is conditionally independent of  $\mathbf{Y}$  given  $\mathbf{Z}$  in any observational distribution consistent with  $\mathcal{G}$  (Pearl 1988; Richardson 2003).

**Definition 1.** (C-component) (Tian and Pearl 2002a) A set of variables  $\mathbf{C} \subseteq \mathbf{V}$  in a causal graph  $\mathcal{G}$  is said to be a confounded component (c-component, for short) if there is a path of only bidirected edges connecting any  $V_i, V_j \in \mathbf{C}$ , and  $\mathbf{C}$  is maximal.

For a variable  $X \in \mathbf{V}$ ,  $\mathcal{C}(X)_{\mathcal{G}}$  denotes the c-component containing  $X$  in  $\mathcal{G}$ .

Previously, we have referred to the set of all CIs encoded in a DAG. We define this formally.

**Definition 2.** (Global Markov Property (GMP)) (Pearl 1988; Geiger, Verma, and Pearl 1989) A probability distribution  $P(\mathbf{v})$  over a set of variables  $\mathbf{V}$  is said to satisfy the global Markov property for a causal graph  $\mathcal{G}$  if, for arbitrary disjoint sets  $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subset \mathbf{V}$  with  $\mathbf{X}, \mathbf{Y} \neq \emptyset$ ,

$$\mathbf{X} \perp_d \mathbf{Y} \mid \mathbf{Z} \implies \mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z} \text{ in } P(\mathbf{v}).$$

Various local Markov properties have been developed which identify a subset of the CIs invoked by GMP that imply all others. A prominent example is the local Markov property for Markovian DAGs.

**Definition 3** (The Local Markov Property (LMP) (Pearl 1988; Lauritzen et al. 1990; Lauritzen 1996)<sup>1</sup>). A probability distribution  $P(\mathbf{v})$  over a set of variables  $\mathbf{V}$  is said to satisfy the local Markov property for a given Markovian DAG  $\mathcal{G}$  if, for any variable  $X \in \mathbf{V}$ ,

$$X \perp Nd(\{X\}) \setminus Pa(\{X\}) \mid Pa(\{X\}) \setminus \{X\} \text{ in } P(\mathbf{v}).$$

**Example 1.** Consider Fig. 1b.  $\{C, D, H\}$  is a c-component, and  $\mathcal{C}(H)_{\mathcal{G}^2} = \{C, D, H\}$ . Since  $\{B, C, D\}$  *d*-separates  $H$  from  $\{A, E, F\}$  in  $\mathcal{G}^2$ ,  $\mathcal{G}^2$  implies the CI:  $H \perp \{A, E, F\} \mid \{B, C, D\}$ .

### 3 The C-component Local Markov Property

In this section, we motivate and introduce the c-component local Markov property for causal DAGs with unobserved

<sup>1</sup>Note that this property is referred to as the *directed local Markov property* in (Lauritzen et al. 1990).

confounders. In Sec. 3.1, we demonstrate the limitations of the traditional local Markov property (LMP) when applied to non-Markovian DAGs. In Sec. 3.2, to solve this problem, we present the c-component local Markov property (C-LMP) for semi-Markovian DAGs and establish its equivalence with GMP. In Sec. 3.3, we provide a useful property of C-LMP that makes its CIs amenable to listing.

#### 3.1 A Naive Approach to Testing Non-Markovian Graphs

First, we show the limitations of the well-known LMP (Def. 3) in testing non-Markovian DAGs. For each variable  $X$  in a given graph, LMP states that  $X$  is independent of its non-descendants conditioning on its parents. Intuitively, the parents of  $X$  form a minimal set separating  $X$  from its non-descendants.

**Example 2.** Consider Fig. 1a. The DAG  $\mathcal{G}^1$  contains only directed edges; assuming all variables are observed,  $\mathcal{G}^1$  is Markovian. LMP invokes 11 CIs for  $\mathcal{G}^1$ :  $A \perp \{U_1, U_2, U_3\}$ ,  $B \perp \{U_1, U_2, U_3\} \mid \{A\}$ ,  $C \perp \{A, E, U_1, U_2\} \mid \{B, U_3\}$ ,  $D \perp \{A, B, E, F, U_1, U_3\} \mid \{C, U_2\}$ ,  $E \perp \{A, C, D, F, H, U_1, U_2, U_3\} \mid \{B\}$ ,  $F \perp \{A, B, E, D, H, U_1, U_2, U_3\} \mid \{C\}$ ,  $H \perp \{A, B, C, E, F, U_2\} \mid \{D, U_1, U_3\}$ ,  $U_1 \perp \{A, B, C, D, E, F, U_3\} \mid \{U_2\}$ ,  $U_2 \perp \{A, B, C, E, F, U_3\}$ ,  $U_3 \perp \{A, B, E, U_1, U_2\}$ . All 11 CIs are testable using samples from the distribution  $P(a, b, c, d, e, f, h, u_1, u_2, u_3)$ .  $\square$

LMP fails trivially for semi-Markovian DAGs since, for example, a variable may be connected to a non-descendant by a bidirected edge. One could think to instead apply LMP to the ‘unprojected’ non-Markovian DAG underlying the given semi-Markovian DAG. The non-Markovian DAG would contain no bidirected edges since the unobserved parents are made explicit. However, LMP does not extend to non-Markovian DAGs either, as we show in the following example.

**Example 3.** Continuing Ex. 2. Assume we are given the non-Markovian DAG  $\mathcal{G}^1$  shown in Fig. 1a. If  $U_1, U_2$  and  $U_3$  are unobserved, only samples from  $P(\mathbf{v}) = \int_{u_1, u_2, u_3} P(a, b, c, d, e, f, h, u_1, u_2, u_3) du_1 du_2 du_3$  are available, where  $\mathbf{V} = \{A, B, C, D, E, F, H\}$  denotes the observed variables. All 11 CIs invoked by LMP for  $\mathcal{G}^1$ , listed in Ex. 2, require samples from  $P(a, b, c, d, e, f, h, u_1, u_2, u_3)$ . Hence, none of these CIs can be tested using  $P(\mathbf{v})$ .

One approach to try salvaging these 11 CIs is to consider only those CIs in which  $\{U_1, U_2, U_3\}$  appear before the conditioning bar. In such CIs,  $\{U_1, U_2, U_3\}$  can be removed using the decomposition axiom. However, only two of the 11 CIs can be modified in this way, i.e.,

$$E \perp \{A, C, D, F, H\} \mid \{B\}, \quad (1)$$

$$F \perp \{A, B, E, D, H\} \mid \{C\}. \quad (2)$$

These two CIs do not suffice to derive the GMP for  $\mathcal{G}^1$ . To witness, consider a graph  $\mathcal{G}'$  over the same variables as  $\mathcal{G}^1$  but with only one edge  $H \rightarrow A$ . Say we have an observational distribution  $P(\mathbf{v})$  faithfully induced by  $\mathcal{G}'$ . Then, the CIs in

Eqs. (1,2) both hold in  $P(\mathbf{v})$ . However,  $\mathcal{G}^1$  implies that

$$H \perp\!\!\!\perp \{A, E, F\} \mid \{B, C, D\} \quad (3)$$

which does not hold in  $P(\mathbf{v})$  since  $\mathcal{G}'$  contains an edge  $H \rightarrow A$ . Only testing the two CIs in Eqs. (1,2) would lead to the false conclusion that  $P(\mathbf{v})$  is consistent with  $\mathcal{G}^1$ . As a result, it is insufficient to use only those CIs which invoke  $\{U_1, U_2, U_3\}$  outside the conditioning set.  $\square$

As in the example, to test a non-Markovian DAG, one can not simply ‘filter out’ CIs that require conditioning on unobserved variables. This is because such CIs can entail testable CIs over the observed variables. The remaining option is to derive all these entailed CIs using the semi-graphoid axioms, and test those which invoke only observed variables. This is the GMP (Def. 2) of the non-Markovian DAG, which can invoke  $\Theta(4^n)$  CIs for a DAG with  $n$  observed variables (Prop. C.3.1). This approach fails to exploit any locality in the graph, and requires a prohibitive number of CI tests, many of which are redundant. This suggests the need for alternative compatibility properties for semi-Markovian (equivalently, non-Markovian) DAGs. We next introduce our contribution, the c-component local Markov property.

### 3.2 C-LMP: A Local Markov Property for Semi-Markovian DAGs

In a semi-Markovian graph, the observed parents of a variable do not suffice to separate it from its non-descendants. Therefore, a surrogate of the parents is needed to restore locality. The construct of a c-component (Def. 1) was introduced for this purpose (Bareinboim et al. 2022), which we explain via an example.

**Example 4.** Continuing Ex. 2, assume  $\{U_1, U_2, U_3\}$  are unobserved in  $\mathcal{G}^1$  (Fig. 1a). The second graph  $\mathcal{G}^2$  (Fig. 1b) is the semi-Markovian projection of  $\mathcal{G}_1$ . Note that the conditional independence  $H \perp\!\!\!\perp \{A, B, C, E, F\} \mid \{D, U_1, U_3\}$  cannot be tested from the data since  $\{U_1, U_3\}$  are not observed. This means that a different conditioning set is needed to make  $H$  independent of its observed non-descendants.

One might condition on the observed descendants of  $\{U_1, U_3\}$  that are closest to  $U_1, U_3$ , i.e.,  $\{C, D\}$ . These variables are not separable from  $H$  without conditioning on  $U_1, U_2$  or  $U_3$ , which is not an option.  $\{C, D\}$  have bidirected edges to  $H$  in  $\mathcal{G}^2$ , the semi-Markovian projection of  $\mathcal{G}^1$ .  $\{C, D\}$  are now active on any paths on which  $\{C, D\}$  they are colliders: for instance, on the paths  $E \leftarrow B \rightarrow C \leftarrow U_3 \rightarrow H$  and  $A \rightarrow B \rightarrow C \leftarrow U_3 \rightarrow H$ . To block some of these paths, we also condition on the (remaining) observed parents of  $\{C, D\}$ , i.e.,  $\{B\}$ . Firstly, conditioning on  $\{C, D\}$  already makes  $B$  and its ancestors active on any paths where they are colliders; secondly,  $B$  is connected to  $H$  when conditioning on  $\{C, D\}$ . Therefore, conditioning on  $\{B\}$  does not introduce any new active paths to  $X$ . Conditioning on  $\{B\}$  additionally blocks paths to  $H$  containing  $B$  on which  $B$  is not a collider. Therefore, we have the conditioning set  $Pa(\mathbf{C}) \setminus \{H\} = \{B, C, D\}$ . The CI over observables  $H \perp\!\!\!\perp \{A, E, F\} \mid \{B, C, D\}$  is thus derived.  $\square$

Ex. 4 is relatively simple since the c-component of  $H$  is used to generate the given CI. However, the c-components of a variable do not always give rise to CIs.

**Example 5.** Consider, as an example DAG, a bidirected path of the form  $V_1 \leftrightarrow V_2 \cdots \leftrightarrow V_n$  on variables  $\mathbf{V}$ . For each  $V_i$ , the c-component including  $V_i$  is the entire graph. Therefore, conditioning on the c-component results in the ‘vacuous’ CI:  $V_i \perp\!\!\!\perp \emptyset \mid \mathbf{V} \setminus V_i$ . Clearly, from this set of vacuous CIs, we cannot derive non-vacuous CIs encoded the graph, such as those of the form  $V_i \perp\!\!\!\perp \{V_j\}, \forall i, j$  s.t.  $|i - j| > 1$  (e.g.,  $V_1 \perp\!\!\!\perp \{V_3\}$ ).  $\square$

A useful insight due to (Richardson 2003) is that subsets of a variable’s c-component can give rise to distinct ‘surrogates’ for its parents and hence distinct CIs. This is because conditioning on a certain variable in a c-component closes some paths while opening others. We generalize c-components to *ancestral c-components* to define these ‘surrogates.’<sup>2</sup>

**Definition 4.** (Ancestral C-component (AC)) Given a causal graph  $\mathcal{G}$  and a consistent ordering  $\mathbf{V}^<$ , let  $X$  be a variable in  $\mathbf{V}^<$ . A set of variables  $\mathbf{C}$  is said to be an ancestral c-component relative to  $X$  if there exists an ancestral set  $\mathbf{S} \subseteq \mathbf{V}^{\leq X}$  containing  $X$  such that  $\mathcal{C}(X)_{\mathcal{G}_{\mathbf{S}}} = \mathbf{C}$ . The collection of all such  $\mathbf{C}$  is denoted:

$$\mathcal{AC}_X = \{\mathbf{C} \mid \mathbf{C} \text{ is an ancestral c-component relative to } X\}.$$

Unlike c-components, there may be many ancestral c-components with respect to a given variable.

**Example 6.** Consider the graph  $\mathcal{G}$  in Fig. 2 with ordering  $A \prec B \prec \cdots \prec X \prec J \prec K$ . For the variable  $X$ ,  $\{X\}$  is an AC relative to  $X$  induced by the ancestral set  $\mathbf{S} = \{X\}$ ;  $\{B, X\}$  is an AC relative to  $X$  induced by the ancestral set  $\mathbf{S} = \{B, C, D, E, X\}$ .  $\{X, A, D, E\}$  is not an AC relative to  $X$  since the exclusion of  $B$  and/or  $H$  disconnects the variables in question. For the variable  $J$ ,  $\{J\}$  is not an AC relative to  $J$  since it excludes the ancestor  $X$  to which  $J$  is connected by a bidirected edge;  $\{X, J\}$  is an AC induced by the ancestral set  $\{X, J\}$ .  $\square$

We use ACs to define the c-component local Markov property, which generalizes LMP to semi-Markovian DAGs using this new notion of local independence.

**Definition 5.** (The C-component Local Markov Property (C-LMP)) A probability distribution  $P(\mathbf{v})$  over a set of variables  $\mathbf{V}$  is said to satisfy the c-component local Markov property for a causal graph  $\mathcal{G}$  with respect to the consistent ordering  $\mathbf{V}^<$ , if, for any variable  $X \in \mathbf{V}^<$  and ancestral c-component  $\mathbf{C} \in \mathcal{AC}_X$  relative to  $X$ ,

$$X \perp\!\!\!\perp \mathbf{S}^+ \setminus Pa(\mathbf{C}) \mid (Pa(\mathbf{C}) \setminus \{X\}) \text{ in } P(\mathbf{v}), \text{ where} \\ \mathbf{S}^+ = \mathbf{V}^{\leq X} \setminus De(Sp(\mathbf{C}) \setminus Pa(\mathbf{C})).$$

**Example 7.** Continuing Ex. 6. We give a few examples of CIs invoked by C-LMP for the variable  $X$ .

<sup>2</sup>Ancestral c-components can be shown to be equivalent to *ancestrally closed districts* as defined in (Richardson 2009). We thank Robin Evans for bringing this to our attention.

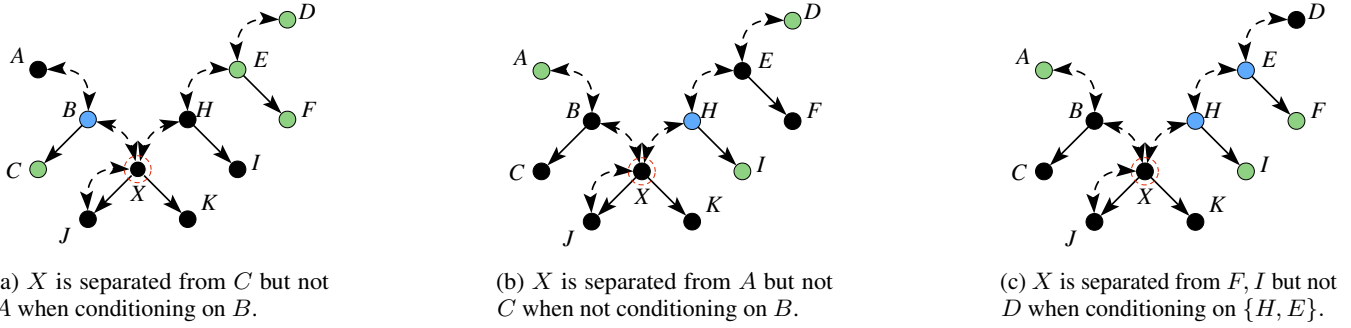


Figure 2: Three ACs relative to the variable  $X$  in the (same) causal DAG  $\mathcal{G}$ . Assume an ordering  $A \prec B \prec \dots \prec X \prec J \prec K$ . The ACs relative to  $X$  (excluding  $\{X\}$  itself), shown in blue, separate it from the variables shown in green.

1. The AC  $\mathbf{C} = \{X, B\}$  gives the CI  $X \perp\!\!\!\perp \{C, D, E, F\} \mid \{B\}$  (Fig. 2a), since

$$Pa(\mathbf{C}) = Pa(\{X, B\}) = \{X, B\}$$

$$\begin{aligned} \mathbf{S}^+ &= \mathbf{V}^{\leq X} \setminus De(Sp(\{X, B\} \setminus Pa(\{X, B\}))) \\ &= \{A, B, C, D, E, F, H, I, X\} \setminus De(\{A, H\}) \\ &= \{A, B, C, D, E, F, H, I, X\} \setminus \{A, H, I\} \\ &= \{B, C, D, E, F, X\} \end{aligned}$$

2. The AC  $\mathbf{C} = \{X, H\}$  gives the CI  $X \perp\!\!\!\perp \{A, D, I\} \mid \{H\}$  (Fig. 2b), since

$$Pa(\mathbf{C}) = Pa(\{X, H\}) = \{X, H\}$$

$$\begin{aligned} \mathbf{S}^+ &= \mathbf{V}^{\leq X} \setminus De(Sp(\{X, H\} \setminus Pa(\{X, H\}))) \\ &= \{A, B, C, D, E, F, H, I, X\} \setminus De(\{B, E\}) \\ &= \{A, B, C, D, E, F, H, I, X\} \setminus \{B, C, E, F\} \\ &= \{A, D, H, I, X\}. \end{aligned}$$

3. The AC  $\mathbf{C} = \{X, H, E\}$  gives the CI  $X \perp\!\!\!\perp \{A, F, I\} \mid \{H, E\}$  (Fig. 2c), since

$$Pa(\mathbf{C}) = Pa(\{X, H, E\}) = \{X, H, E\}$$

$$\begin{aligned} \mathbf{S}^+ &= \mathbf{V}^{\leq X} \setminus De(Sp(\{X, H, E\} \setminus Pa(\{X, H, E\}))) \\ &= \{A, B, C, D, E, F, H, I, X\} \setminus De(\{B, D\}) \\ &= \{A, B, C, D, E, F, H, I, X\} \setminus \{B, C, D, E\} \\ &= \{A, E, F, H, I, X\} \end{aligned}$$

□

As a sanity check, let us examine the CIs C-LMP implies for a Markovian DAG  $\mathcal{G}$ , where all c-components are singletons. There is exactly one AC  $\mathbf{C} = \{X\}$  relative to a given variable  $X$ . Moreover,  $Pa(\mathbf{C}) = Pa(\{X\})$ ,  $Sp(\{X\}) = \emptyset$  and  $\mathbf{S}^+ = \mathbf{V}^{\leq X} \setminus De(\emptyset) = \mathbf{V}^{\leq X}$ . Therefore, the CI invoked by C-LMP for  $X$  is

$$X \perp\!\!\!\perp \mathbf{V}^{\leq X} \setminus Pa(\{X\}) \mid Pa(\{X\}) \setminus \{X\} \quad (4)$$

Thus, C-LMP reduces to the local well-numbering Markov property (Lauritzen et al. 1990) for a Markovian DAG  $\mathcal{G}$ .<sup>3</sup>

<sup>3</sup>The local well-numbering Markov property was shown to be equivalent to LMP in (Lauritzen et al. 1990). A subtle difference is that LMP tests the independence of  $X$  from its all non-descendants, not just  $\mathbf{V}^{\leq X}$  for a given ordering  $\mathbf{V}^{\prec}$ .

In semi-Markovian DAGs, c-components are not necessarily singletons. Comparing the CIs invoked by LMP and C-LMP for a given variable  $X$ , we see that C-LMP generalizes two concepts:

1. The conditioning set  $Pa(\{X\}) \setminus \{X\}$  stated in LMP is replaced with  $Pa(\mathbf{C}) \setminus \{X\}$  in C-LMP, using an AC  $\mathbf{C}$  relative to  $X$ .
2. The conditioning set  $Pa(\mathbf{C}) \setminus \{X\}$  renders  $X$  independent of  $\mathbf{S}^+ \setminus Pa(\mathbf{C})$  where  $\mathbf{S}^+ = \mathbf{V}^{\leq X} \setminus De(Sp(\mathbf{C}) \setminus Pa(\mathbf{C}))$ , as stated by C-LMP. The set  $\mathbf{S}^+ \setminus Pa(\mathbf{C})$  replaces the set  $Nd(\{X\}) \setminus Pa(\{X\})$  in LMP.

In Ex. 4, we gave intuition for the generalised conditioning set  $Pa(\mathbf{C}) \setminus \{X\}$  (Case 1). Next, we explain the construction of  $\mathbf{S}^+$  (Case 2) used to compute the maximal set of variables in  $\mathbf{V}^{\leq X}$  that are independent of  $X$  given  $Pa(\mathbf{C}) \setminus \{X\}$ . Consider what happens to a variable  $Y \in \mathbf{V}^{\leq X} \setminus Pa(\mathbf{C})$  when conditioning on  $Pa(\mathbf{C}) \setminus \{X\}$ .

- If  $Y$  is a descendant (or an ancestor) of some node  $W \in Pa(\mathbf{C})$ , we have a directed path  $\pi$  from  $W$  to  $Y$  (or vice-versa). Conditioning on  $Pa(\mathbf{C}) \setminus \{X\}$  blocks  $\pi$  (since  $Y \notin Pa(\mathbf{C})$ ), and hence any path from  $X$  to  $Y$  which contains  $\pi$  as a sub-path. For example, in Fig. 2a, taking  $\mathbf{C} = \{X, B\}$ ,  $W = B$  and  $Y = C$ , conditioning on  $\{B\}$  blocks the path  $X \leftrightarrow B \rightarrow C$ .
- If  $Y$  is connected by a bidirected path to some node in  $\mathbf{C}$ , but  $Y$  is not in  $Sp(\mathbf{C})$ , then some node  $V \in Sp(\mathbf{C}) \setminus Pa(\mathbf{C})$  ‘intercepts’ this path, i.e.,  $V$  is a closed collider and thus blocks the path from  $X$  to  $Y$ . For example, in Fig. 2a, taking  $\mathbf{C} = \{X, B\}$ ,  $V = H$  and  $Y = E$ ,  $H$  blocks the path  $X \leftrightarrow H \leftrightarrow E$ .
- If  $Y$  is in  $Sp(\mathbf{C}) \setminus Pa(\mathbf{C})$ , is an active bidirected path from  $X$  to  $Y$  when conditioning on  $\mathbf{C} \setminus \{X\}$ . For example, in Fig. 2a, taking  $\mathbf{C} = \{X, B\}$  and  $Y = A$ , conditioning on  $\{B\}$  opens the path  $X \leftrightarrow B \leftrightarrow A$ .

Analogous to how, for a given variable  $X$ , different conditioning sets give rise to different CIs from  $X$ , different ACs also give rise to different CIs from  $X$ . The upshot of defining ACs is that they carve out a relatively small set of CIs (commonly known as a ‘basis’ (Bareinboim et al. 2022)) from which all CIs encoded in the given graph can be derived.

The main result of this section, given below, establishes that GMP and C-LMP are equivalent.

**Theorem 1** (Equivalence of C-LMP and GMP). *Let  $\mathcal{G}$  be a causal graph and  $\mathbf{V}^<$  a consistent ordering. A probability distribution over  $\mathbf{V}$  satisfies the global Markov property for  $\mathcal{G}$  if and only if it satisfies the  $c$ -component local Markov property for  $\mathcal{G}$  with respect to  $\mathbf{V}^<$ .*

As a corollary of Thm. 1, we can conclude that C-LMP is equivalent to Richardson’s ordered local Markov property (Richardson 2003), since the latter is equivalent to GMP (Richardson 2003, Thm. 2, Section 3.1).

**Corollary 1** (Equivalence of C-LMP and the Ordered Local Markov Property). *Let  $\mathcal{G}$  be a causal graph and  $\mathbf{V}^<$  a consistent ordering. A probability distribution over  $\mathbf{V}$  satisfies the ordered local Markov property (Richardson 2003) for  $\mathcal{G}$  with respect to  $\mathbf{V}^<$  if and only if it satisfies the  $c$ -component local Markov property for  $\mathcal{G}$  with respect to  $\mathbf{V}^<$ .*

In Appendix B, we further develop the connection between C-LMP and the ordered local Markov property. In fact, in Thm. B.2.1, we show that these two properties induce the exact same set of CIs for a given DAG and a consistent ordering. Thm. B.2.1 thus provides another way to obtain Thm. 1 as a corollary.

The equivalence of C-LMP and GMP means that the CIs invoked by C-LMP for a given causal DAG can be used to test the DAG against observational data.

### 3.3 Uniqueness Property of C-LMP

By definition, each CI invoked by C-LMP is generated from an AC. We further show that each CI can be generated from exactly one AC.

**Theorem 2** (Unique AC for each CI Invoked by C-LMP). *Let  $\mathcal{G}$  be a causal graph,  $\mathbf{V}^<$  a consistent ordering, and  $X$  a variable in  $\mathbf{V}^<$ . For every conditional independence relation invoked by the  $c$ -component local Markov property of the form  $X \perp\!\!\!\perp \mathbf{W} \mid \mathbf{Z}$ , there is exactly one ancestral  $c$ -component  $\mathbf{C} \in \mathcal{AC}_X$  such that  $\mathbf{W} = \mathbf{V}^{\leq X} \setminus ((De(Sp(\mathbf{C}) \setminus Pa(\mathbf{C}))) \cup Pa(\mathbf{C}))$  and  $\mathbf{Z} = Pa(\mathbf{C}) \setminus \{X\}$ .*

The one-to-one correspondence between ACs and CIs invoked by C-LMP allows us to give bounds on the latter number that are tight in the exponent.

**Proposition 1** (Number of CIs Invoked by C-LMP). *Given a causal graph  $\mathcal{G}$  and a consistent ordering  $\mathbf{V}^<$ , let  $n$  and  $s \leq n$  denote the number of variables and the size of the largest  $c$ -component in  $\mathcal{G}$  respectively. Then, the  $c$ -component local Markov property for  $\mathcal{G}$  with respect to  $\mathbf{V}^<$  invokes  $O(n2^s)$  conditional independencies implied by  $\mathcal{G}$  over  $\mathbf{V}$ . Moreover, there exists a graph  $\mathcal{G}$  and a consistent ordering  $\mathbf{V}^<$  for which the property induces  $\Omega(2^n)$  conditional independencies.*

This result shows that C-LMP offers an exponential improvement on the  $\Theta(4^n)$  CIs invoked by GMP. However, C-LMP can still invoke an exponential number of CIs. For example, in  $\mathcal{G}^{ex}$  (Fig. 3a) with  $2n$  nodes, there are  $2^{n+(n-3)}$  CIs invoked by C-LMP.

The main upshot of the one-to-one correspondence between ACs and CIs invoked by C-LMP is that to list such

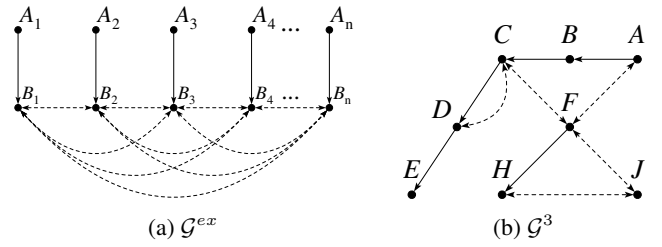


Figure 3: (a) An example showing that C-LMP may invoke an exponential number of CIs. (b) A causal graph used to show the execution of LISTCI in Ex. 9.

CIs, it suffices to enumerate ACs. We study the problem of listing CIs in the next section.

## 4 Listing CIs

Our goal in this section is to develop an algorithm that lists CIs invoked by C-LMP. In the worst case, there may exist exponentially many such CIs, requiring exponential time to list them all. In such cases, we look for algorithms that run in polynomial delay (Johnson, Yannakakis, and Papadimitriou 1988). Poly-delay algorithms output the first solution (or indicate none is available) in poly-time, and take poly-time to output each consecutive solution.

However, not all CIs invoked by C-LMP are useful for model testing. C-LMP invokes some ‘vacuous’ CIs of the form  $X \perp\!\!\!\perp \emptyset \mid \mathbf{Z}$ , which do not need testing. Therefore, we constrain the problem by requiring that we list only *non-vacuous* CIs, as defined below.

**Definition 6** (Vacuous CI and Admissible AC (AAC)). *Given a conditional independence relation invoked by C-LMP of the form  $X \perp\!\!\!\perp \mathbf{W} \mid Pa(\mathbf{C}) \setminus \{X\}$ , where  $\mathbf{W} = \mathbf{S}^+ \setminus Pa(\mathbf{C})$  (by Def. 5), if  $\mathbf{W} \neq \emptyset$ , the conditional independence relation is said to be non-vacuous and  $\mathbf{C}$  is said to be an admissible ancestral  $c$ -component relative to  $X$ .*

**Example 8.** Consider the causal graph  $\mathcal{G}^3$  (Fig. 3b). The AC  $\{J\}$  relative to  $J$  is admissible. Given  $\mathbf{S}^+ = \mathbf{V} \setminus \{F, H\}$ , we have  $\mathbf{W} = \mathbf{S}^+ \setminus \{J\} = \{A, B, C, D, E\}$ . However, the AC  $\{F, J\}$  relative to  $J$  is not admissible. Since  $\mathbf{S}^+ = \{F, J\}$ ,  $\mathbf{W} = \mathbf{S}^+ \setminus \{F, J\} = \emptyset$ .  $\square$

Listing only non-vacuous CIs is important since C-LMP may invoke exponentially many vacuous CIs. To witness, consider a bidirected clique on  $n$  nodes such that no two variables are independent of each other given any conditioning set. Every set  $\mathbf{Z} \subseteq \mathbf{V} \setminus \{X\}$  forms an AC, resulting in  $\Omega(2^n)$  vacuous CIs (see Ex. D.3.1 in Appendix D.3 for details).

Our bounds on the number of CIs invoked by C-LMP are also tight for the number of non-vacuous CIs (Prop. 1). We develop the algorithm LISTCI (Alg. 1) to list all non-vacuous CIs invoked by C-LMP in poly-delay.

**Example 9.** Consider the causal graph  $\mathcal{G}^3$  (Fig. 3b) with  $\mathbf{V}^< = \{A, B, C, D, E, F, H, J\}$ . LISTCI( $\mathcal{G}^3, \mathbf{V}^<$ ) lists 11 non-vacuous CIs invoked by C-LMP:  $C \perp\!\!\!\perp \{A\} \mid \{B\}$ ,  $D \perp\!\!\!\perp \{A\} \mid \{B, C\}$ ,  $E \perp\!\!\!\perp \{A, B, C\} \mid \{D\}$ ,  $F \perp\!\!\!\perp \{B\} \mid \{A\}$ ,  $F \perp\!\!\!\perp \{E\} \mid \{A, B, C, D\}$ ,  $H \perp\!\!\!\perp \{A, B, C, D, E\} \mid \{F\}$ ,  $J \perp\!\!\!\perp$

---

**Algorithm 1: LISTCI** ( $\mathcal{G}, \mathbf{V}^{\prec}$ )

---

- 1: **Input:**  $\mathcal{G}$  a causal diagram;  $\mathbf{V}^{\prec}$  an ordering consistent with  $\mathcal{G}$ .
  - 2: **Output:** Listing non-vacuous CIs invoked by C-LMP for  $\mathcal{G}$  with respect to  $\mathbf{V}^{\prec}$ .
  - 3: **for** each  $X \in \mathbf{V}^{\prec}$  **do**
  - 4:    $\mathbf{I} \leftarrow \mathcal{C}(X)_{\mathcal{G}_{An(\{X\})}}$ ;  $\mathbf{R} \leftarrow \mathcal{C}(X)_{\mathcal{G}_{\mathbf{V}^{\leq X}}}$
  - 5:   LISTCIX( $\mathcal{G}_{\mathbf{V}^{\leq X}}, X, \mathbf{V}^{\leq X}, \mathbf{I}, \mathbf{R}$ )
- 

$\{A, B, C, D, E\}, J \perp\!\!\!\perp \{B\} \mid \{A, F\}, J \perp\!\!\!\perp \{B\} \mid \{A, F, H\}, J \perp\!\!\!\perp \{E\} \mid \{A, B, C, D, F\}, J \perp\!\!\!\perp \{E\} \mid \{A, B, C, D, F, H\}$ . After, LISTCI terminates as there are no more non-vacuous CIs.  $\square$

#### 4.1 Listing CIs for a Given Variable

The algorithm LISTCI iterates over each variable  $X \in \mathbf{V}^{\prec}$  and lists all non-vacuous CIs invoked by C-LMP for  $X$ . By Defs. 5, 6 and Thm. 2, listing non-vacuous CIs reduces to enumerating AACs. In this section, we show how to enumerate AACs relative to a given variable  $X \in \mathbf{V}^{\prec}$  using the procedure LISTCIX (Alg. 2).

LISTCIX adopts a divide-and-conquer strategy similar to the algorithm of (Takata 2010). LISTCIX implicitly constructs a binary search tree for  $X$  using a depth-first approach. Tree nodes of the form  $\mathcal{N}(\mathbf{I}', \mathbf{R}')$  represents the collection of all AACs  $\mathbf{C}$  with  $\mathbf{I}' \subseteq \mathbf{C} \subseteq \mathbf{R}'$ . The top-level call of LISTCIX, at the root node  $\mathcal{N}(\mathbf{I}, \mathbf{R})$ , represents all AACs  $\mathbf{C}$  relative to  $X$ . This is due to the construction on line 4 of LISTCI so that  $\mathbf{I}$  is contained in and  $\mathbf{R}$  contains all possible AACs relative to  $X$ . Thus, the top-level call can generate all CIs for  $X$ .

Subsequent recursive calls expand this tree by shrinking the range  $\mathbf{I}' \subseteq \mathbf{R}'$  one variable at a time. One requirement of the poly-delay property is that each AAC should appear exactly once in the enumeration. To expand the tree from  $\mathcal{N}(\mathbf{I}', \mathbf{R}')$ , LISTCIX constructs two ‘disjoint’ children (lines 10-11: a chosen variable  $S \in \mathbf{V}^{\prec}$  cannot be in any AAC from the left child but must be in every AAC from the right child).

As another requirement of the poly-delay property, we expand the tree from a node  $\mathcal{N}(\mathbf{I}', \mathbf{R}')$  if and only if the expansion is guaranteed to produce a non-vacuous CI. Equivalently, there must exist at least one AAC  $\mathbf{C}$  such that  $\mathbf{I}' \subseteq \mathbf{C} \subseteq \mathbf{R}'$ . If there is no such  $\mathbf{C}$ , we prune the tree and back-track to the previous tree node. To perform this check for the existence of an AAC in poly-time, LISTCIX calls the function FINDAAC (Alg. 3). We explain FINDAAC in the next subsection.

Finally, a leaf node is reached when  $\mathbf{I} = \mathbf{R}$ . LISTCIX outputs a non-vacuous CI generated from the AAC  $\mathbf{C} = \mathbf{I}$  using Def. 5.

**Example 10.** Expanding Ex. 9 to demonstrate the construction of the search tree  $\mathcal{T}^3$  (Fig. 4) generated by running LISTCI( $\mathcal{G}^3, \mathbf{V}^{\prec}$ ) for  $X = J$ . With  $\mathbf{I} = \{J\}$  and  $\mathbf{R} = \{A, C, D, F, H, J\}$  constructed on line 4, the initial search starts from the root node  $\mathcal{N}(\mathbf{I}, \mathbf{R})$  on line 5. On line 3 of LISTCIX, FINDAAC returns  $\{J\}$ . With  $S = F$  and

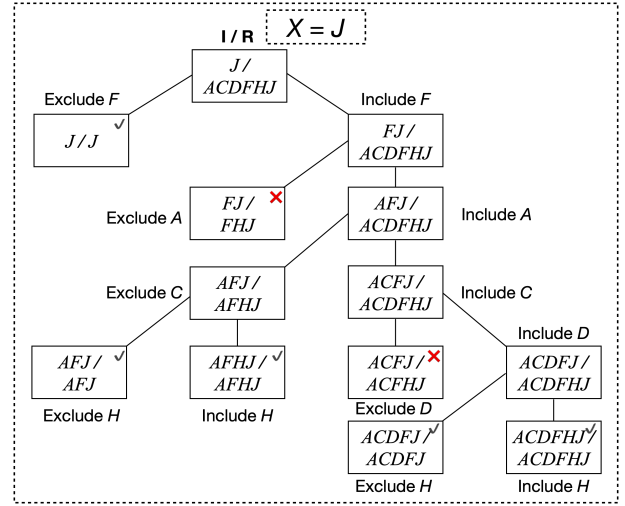


Figure 4:  $\mathcal{T}^3$  a search tree illustrating the running of LISTCI in Ex. 9 for  $X = J$ .

$\mathbf{R}' = \{J\}$ , the recursive call LISTCIX( $\mathcal{G}^3, J, \mathbf{V}^{\prec}, \mathbf{I}, \mathbf{R}'$ ) is made at line 10, spawning a child  $\mathcal{N}_1(\{J\}, \{J\})$ . The search continues from  $\mathcal{N}_1$ . FINDAAC returns  $\{J\}$ .  $\mathcal{N}_1$  is a leaf node, and LISTCIX outputs a CI:  $J \perp\!\!\!\perp \{A, B, C, D, E\}$  on line 6. The rest of the search tree for  $J$  is shown in  $\mathcal{T}^3$ . The full set of search trees is shown in Fig. D.3.1 in Appendix D.3.  $\square$

#### 4.2 Finding an AAC

In this section, we address the following subproblem, needed for LISTCIX to run in poly-delay: given a variable  $X \in \mathbf{V}^{\prec}$ , and two ACs  $\mathbf{I}, \mathbf{R}$  relative to  $X$ , how do we find an AAC  $\mathbf{C}$  such that  $\mathbf{I} \subseteq \mathbf{C} \subseteq \mathbf{R}$  (or indicate that there is none) in poly-time?

The poly-time constraint on solving this subproblem rules out the brute-force approach: namely, iterating over all subsets  $\mathbf{C}$  such that  $\mathbf{I} \subseteq \mathbf{C} \subseteq \mathbf{R}$  until we find some  $\mathbf{C}$  that is an AAC (or conclude that there is none). We find that this exponential search is not necessary. The key idea behind our solution is to reduce this search to a verification of whether a particular AAC is admissible.

To elaborate, assume there exists an AAC  $\mathbf{C}$  such that  $\mathbf{I} \subseteq \mathbf{C} \subseteq \mathbf{R}$ . It is possible that  $\mathbf{C} = \mathbf{I}$ , in which case we are done. Otherwise, consider what it means for  $\mathbf{I}$  not to be admissible. When conditioning on  $Pa(\mathbf{I}) \setminus \{X\}$ , every variable in  $\mathbf{V}^{\leq X}$  has an active path to  $X$ . This means that every such variable (outside the conditioning set) is a descendant of  $Sp(\mathbf{I}) \setminus Pa(\mathbf{I})$  (Def. 5). Since  $\mathbf{C}$  is admissible, then there must be some descendant of  $Sp(\mathbf{I}) \setminus Pa(\mathbf{I})$ , say  $D$ , which has an active path to  $X$  when conditioning on  $Pa(\mathbf{I}) \setminus \{X\}$  but not when conditioning on  $Pa(\mathbf{C}) \setminus \{X\}$ . We show that the AAC  $\mathbf{C}$  can exist if and only if there exists *any* set  $\mathbf{Z}$  separating  $X$  from some such  $D$ , with the restriction that  $Pa(\mathbf{I}) \setminus \{X, D\} \subseteq \mathbf{Z} \subseteq Pa(\mathbf{R}) \setminus \{X, D\}$ .  $\mathbf{Z}$  need not be an AAC. We can check if such  $\mathbf{Z}$  exists (line 6) in poly-time using the function FINDSEPARATOR (Fig. C.2.2 in Appendix C.2).

---

**Algorithm 2: LISTCIX** ( $\mathcal{G}_{\mathbf{V}^{\leq X}}, X, \mathbf{V}^{\leq X}, \mathbf{I}, \mathbf{R}$ )

---

- 1: **Input:**  $\mathcal{G}_{\mathbf{V}^{\leq X}}$  a causal diagram;  $X$  a variable;  $\mathbf{V}^{\leq X}$  an ordering consistent with  $\mathcal{G}$ ;  $\mathbf{I}$  and  $\mathbf{R}$  ACs relative to  $X$ .
- 2: **Output:** Listing non-vacuous CIs invoked by C-LMP associated with  $X$  and AACs  $\mathbf{C}$  under the constraint  $\mathbf{I} \subseteq \mathbf{C} \subseteq \mathbf{R}$ .
- 3: **if** FINDAAC( $\mathcal{G}_{\mathbf{V}^{\leq X}}, X, \mathbf{V}^{\leq X}, \mathbf{I}, \mathbf{R}$ )  $\neq \perp$  **then**
- 4:     **if**  $\mathbf{I} = \mathbf{R}$  **then**
- 5:          $\mathbf{S}^+ \leftarrow \mathbf{V}^{\leq X} \setminus De(Sp(\mathbf{I}) \setminus Pa(\mathbf{I}))$
- 6:         Output  $X \perp\!\!\!\perp \mathbf{S}^+ \setminus Pa(\mathbf{I}) \mid Pa(\mathbf{I}) \setminus \{X\}$
- 7:         **return**
- 8:      $\mathbf{T} \leftarrow \mathbf{R} \cap (Sp(\mathbf{I}) \setminus \mathbf{I}), S \leftarrow$  Any node in  $\mathbf{T}$
- 9:      $\mathbf{I}' \leftarrow \mathcal{C}(X)_{\mathcal{G}_{An(\mathbf{I} \cup \{S\})}}, \mathbf{R}' \leftarrow \mathcal{C}(X)_{\mathcal{G}_{R \setminus De(\{S\})}}$
- 10:     LISTCIX( $\mathcal{G}_{\mathbf{V}^{\leq X}}, X, \mathbf{V}^{\leq X}, \mathbf{I}, \mathbf{R}'$ )
- 11:     LISTCIX( $\mathcal{G}_{\mathbf{V}^{\leq X}}, X, \mathbf{V}^{\leq X}, \mathbf{I}', \mathbf{R}$ )

---

**Example 11.** Expanding Ex. 10 to illustrate the usage of FINDAAC. Let  $X = J$ ,  $\mathbf{V}^{\prec} = \mathbf{V}^{\leq J}$ ,  $\mathbf{I} = \{J\}$ , and  $\mathbf{R} = \{A, C, D, F, H, J\}$ . FINDAAC( $\mathcal{G}^3, J, \mathbf{V}^{\leq J}, \mathbf{I}, \mathbf{R}$ ) returns  $\mathbf{C} = \{J\}$  since there exists an AAC  $\mathbf{C}$  relative to  $J$  with  $\mathbf{I} \subseteq \mathbf{C} \subseteq \mathbf{R}$ . With  $\mathbf{I} = \{F, J\}$  and  $\mathbf{R} = \{F, H, J\}$ , FINDAAC( $\mathcal{G}^3, J, \mathbf{V}^{\leq J}, \mathbf{I}, \mathbf{R}$ ) returns  $\perp$  since none of the ACs  $\mathbf{C}$  relative to  $J$  with  $\mathbf{I} \subseteq \mathbf{C} \subseteq \mathbf{R}$  are admissible.  $\square$

**Lemma 1** (Correctness of FINDAAC). *Given a causal graph  $\mathcal{G}$ , a consistent ordering  $\mathbf{V}^{\prec}$ , and a variable  $X \in \mathbf{V}^{\prec}$ , let  $\mathbf{I}, \mathbf{R}$  be ancestral  $c$ -components relative to  $X$  such that  $\mathbf{I} \subseteq \mathbf{R}$ . FINDAAC( $\mathcal{G}_{\mathbf{V}^{\leq X}}, X, \mathbf{V}^{\leq X}, \mathbf{I}, \mathbf{R}$ ) outputs an admissible ancestral  $c$ -component  $\mathbf{C}$  relative to  $X$  such that  $\mathbf{I} \subseteq \mathbf{C} \subseteq \mathbf{R}$  if such a  $\mathbf{C}$  exists, and  $\perp$  otherwise.*

**Lemma 2** (Correctness of LISTCIX). LISTCIX( $\mathcal{G}_{\mathbf{V}^{\leq X}}, X, \mathbf{V}^{\leq X}, \mathbf{I}, \mathbf{R}$ ) enumerates all and only all non-vacuous conditional independence relations invoked by the  $c$ -component local Markov property associated with  $X$  and admissible ancestral  $c$ -components  $\mathbf{C}$  relative to  $X$  where  $\mathbf{I} \subseteq \mathbf{C} \subseteq \mathbf{R}$ . Further, LISTCIX runs in  $O(n^2(n+m))$  delay where  $n$  and  $m$  represent the number of nodes and edges in  $\mathcal{G}$ , respectively.

Our results are summarized in the following theorem, which provides the soundness, completeness, and poly-delay complexity of the proposed algorithm.

**Theorem 3** (Correctness of LISTCI). *Let  $\mathcal{G}$  be a causal graph and  $\mathbf{V}^{\prec}$  a consistent ordering. LISTCI( $\mathcal{G}, \mathbf{V}^{\prec}$ ) enumerates all and only all non-vacuous conditional independence relations invoked by the  $c$ -component local Markov property in  $O(n^2(n+m))$  delay where  $n$  and  $m$  represent the number of nodes and edges in  $\mathcal{G}$ , respectively.*

## 5 Experiments

In this section, we first demonstrate the runtime of LISTCI on benchmark DAGs of up to 100 nodes from the bnlearn repository (Scutari 2010). Next, we apply LISTCI to model testing on a real-world protein signaling dataset with an expert-provided graph (Sachs et al. 2005). Third, we provide analysis of the total number of non-vacuous CIs invoked

---

**Algorithm 3: FINDAAC** ( $\mathcal{G}_{\mathbf{V}^{\leq X}}, X, \mathbf{V}^{\leq X}, \mathbf{I}, \mathbf{R}$ )

---

- 1: **Input:**  $\mathcal{G}_{\mathbf{V}^{\leq X}}$  a causal diagram;  $X$  a variable;  $\mathbf{V}^{\leq X}$  an ordering consistent with  $\mathcal{G}$ ;  $\mathbf{I}$  and  $\mathbf{R}$  ACs relative to  $X$ .
- 2: **Output:** An AAC  $\mathbf{C}$  relative to  $X$  under the constraint  $\mathbf{I} \subseteq \mathbf{C} \subseteq \mathbf{R}$ , if such  $\mathbf{C}$  exists;  $\perp$  otherwise.
- 3: **if** ISADMISSIBLE( $\mathcal{G}_{\mathbf{V}^{\leq X}}, X, \mathbf{V}^{\leq X}, \mathbf{I}$ ) **then**
- 4:     **return**  $\mathbf{I}$
- 5:     **for** each  $D \in De(Sp(\mathbf{I}) \setminus Pa(\mathbf{I}))$  **do**
- 6:          $\mathbf{Z} \leftarrow$  FINDSEPARATOR( $\mathcal{G}_{\mathbf{V}^{\leq X}}, \{X\}, \{D\}, Pa(\mathbf{I}), Pa(\mathbf{R})$ )
- 7:         **if**  $\mathbf{Z} \neq \perp$  **then**
- 8:             **return**  $\mathcal{C}(X)_{\mathcal{G}_{An(\mathbf{I} \cup \mathbf{Z})}}$
- 9:     **return**  $\perp$

---

by C-LMP, using LISTCI for the analysis. The details of the three experiments are shown in Appendix F.

**Experiment 1 (Comparison of LISTCI with other algorithms).** We compare the runtime of LISTCI with two baselines: LISTGMP (Fig. E.0.1 in Appendix E) and LISTCIBF (Alg. B.1.1 in Appendix B.1)<sup>4</sup>. LISTGMP lists all CIs invoked by GMP (Def. 2); LISTCIBF iterates over ancestral sets to list CIs invoked by the ordered local Markov property (Richardson 2003). The algorithms were run on DAGs that describe real-world scenarios from the bnlearn repository. Since the graphs are Markovian, non-Markovian graphs were generated by randomly assigning  $U\%$  of nodes to be unobserved for  $U \in \{0, 10, 20, \dots, 90\}$ . For each  $U$ , we generated 10 random samples. For a given graph, algorithm, and  $U$ , if any one sample timed out ( $> 1$  hour), no further samples are tested. Fig. 5 shows the average runtime of the algorithms, with further details in Fig. F.1.1.

The results corroborate our theoretical conclusion that LISTCI outperforms the other algorithms. For LISTGMP, the algorithm did not timeout on graphs with  $n < 10$  nodes. For LISTCIBF, we have mixed results. The algorithm did not time out for some graphs with up to  $n = 35$  nodes, but there were other graphs with  $n = 25$  where the algorithm did time out. For LISTCI, the algorithm did not timeout for many graphs up to  $n = 80$ , but did time out for some graphs with  $n = 70$ .

**Experiment 2 (Application to model testing).** A real-world protein signaling dataset (Sachs et al. 2005) has been used to benchmark causal discovery methods (Cundy, Grover, and Ermon 2021; Zantedeschi et al. 2023). The dataset (853 samples) comes with an expert-provided ground-truth DAG (11 nodes, 16 edges). Using LISTCI, we test to what extent this graph is compatible with the available data. We use a kernel-based CI test from the causal-learn package (Zheng et al. 2024) with p-value  $p = 0.05$  (for the null hypothesis of dependence).

<sup>4</sup>Our implementation of LISTCIBF can be improved by generating ancestral sets more efficiently. Regardless, we know LISTCI performs better in theory (Sec. 3), and have strong evidence that it is also better in practice to this more efficient implementation of LISTCIBF.



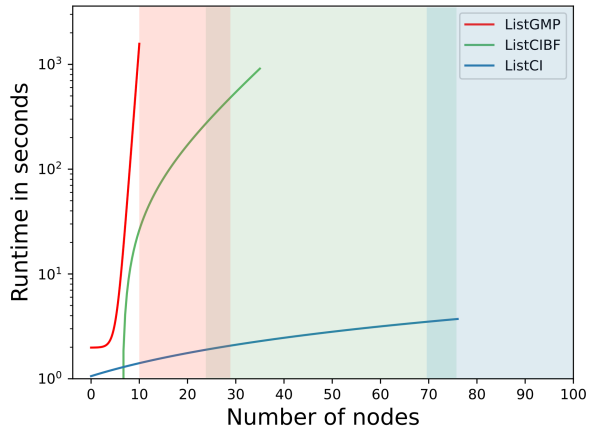


Figure 5: Plot of runtimes of the algorithms LISTGMP, LISTCIBF, and LISTCI on graphs of various sizes. A colored box indicates the interval of  $n$  on which the relevant algorithm has timed out on some graphs with  $n$  nodes. The y-axis uses a logarithmic scale.

For our chosen topological order, seven out of ten CIs invoked by C-LMP resulted in  $p > 0.05$ . This suggests the ground-truth DAG may need revision before use as a benchmark for structure learning. The exact local CIs that are violated may guide experts in this revision process.

**Experiment 3 (Analysis of C-LMP).** We use LISTCI to understand the total number of non-vacuous CIs invoked by C-LMP. Let CI denote this number. CI is also the number of CIs that need to be tested from a given semi-Markovian causal DAG. Based on experiments with random graphs shown in Appendix F.3, we conclude that the graph topology associated with c-components plays a major role in CI. More specifically, two factors related to c-components are of primary interest:

1.  $s \leq n$ : the size of the largest c-component, and
2. The sparsity of c-components, a proxy for which is the number of bidirected edges.

As we add bidirected edges, while c-components are sparse, CI increases exponentially with  $s$ , as given by the bound  $O(n2^s)$ . As c-components become more dense, CI decays exponentially with the number of bidirected edges. As an illustrative example, please refer to Fig. F.3.1 and the discussion on Case 1 in Appendix F.3.

## 6 Conclusions

In this paper, we introduced a new conditional independence property for causal models with unobserved confounders, namely, the *c-component local Markov property* (C-LMP, Def. 5). Given a DAG  $\mathcal{G}$ , C-LMP identifies a small subset of conditional independence constraints (CIs) that together imply all other CIs encoded in  $\mathcal{G}$ . We showed that C-LMP is equivalent to the global Markov property (Thm. 1), and that each CI that C-LMP invokes can be generated from a unique

ancestral c-component (Thm. 2). Building on this foundation, we developed the first algorithm LISTCI (Alg. 1) capable of listing all CIs invoked by C-LMP in polynomial delay (Thm. 3). Reducing the number of CI tests needed to evaluate a causal model is important as it improves runtime performance and helps mitigate concerns about statistical power and multiple hypothesis testing. We hope our work will help researchers test their causal assumptions using observational data prior to inference.

## Acknowledgements

This research is supported in part by the NSF, DARPA, ONR, AFOSR, DoE, Amazon, JP Morgan, and The Alfred P. Sloan Foundation. We thank Robin Evans and the anonymous reviewers for their thoughtful comments.

## References

- Ankan, A.; and Textor, J. 2022. A Simple Unified Approach to Testing High-Dimensional Conditional Independences for Categorical and Ordinal Data. arXiv:2206.04356.
- Bareinboim, E.; Correa, J. D.; Ibeling, D.; and Icard, T. 2022. On Pearl’s Hierarchy and the Foundations of Causal Inference. In *Probabilistic and Causal Inference: The Works of Judea Pearl*, 507–556. New York, NY, USA: Association for Computing Machinery, 1st edition.
- Bareinboim, E.; and Pearl, J. 2016. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113(27): 7345–7352.
- Cundy, C.; Grover, A.; and Ermon, S. 2021. BCD Nets: Scalable Variational Approaches for Bayesian Causal Discovery. arXiv:2112.02761.
- Dawid, A. P. 1979. Conditional independence in statistical theory. *Journal of the Royal Statistical Society, Series B*, 41(1): 1–31.
- Fisher, R. A. 1936. Design of Experiments. *British Medical Journal*, 1(3923): 554.
- Geiger, D.; and Meek, C. 1998. Graphical Models and Exponential Families. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*.
- Geiger, D.; and Meek, C. 1999. Quantifier Elimination for Statistical Problems. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*.
- Geiger, D.; Verma, T. S.; and Pearl, J. 1989. d-Separation: From Theorems to Algorithms. In *Proceedings, 5th Workshop on Uncertainty in AI*, 118–124. ISBN 9780444887382.
- Hoover, K. D. 1990. The logic of causal inference: Econometrics and the conditional analysis of causation. *Economics and Philosophy*, 6(2): 207–234.
- Hu, Z.; and Evans, R. 2023. Towards standard imsets for maximal ancestral graphs. arXiv:2208.10436.
- Johnson, D. S.; Yannakakis, M.; and Papadimitriou, C. H. 1988. On generating all maximal independent sets. *Information Processing Letters*, 27(3): 119–123.
- Kang, C.; and Tian, J. 2009. Markov properties for linear causal models with correlated errors. *The Journal of Machine Learning Research*, 10: 41–70.

- King, R. D.; Whelan, K. E.; Jones, F. M.; Reiser, P. G.; Bryant, C. H.; Muggleton, S. H.; Kell, D. B.; and Oliver, S. G. 2004. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427(6971): 247–252.
- Lauritzen, S. L. 1996. *Graphical Models*. Oxford: Clarendon Press.
- Lauritzen, S. L.; Dawid, A. P.; Larsen, B. N.; and Leimer, H. G. G. 1990. Independence Properties of Directed Markov Fields. *Networks*, 20(5): 491–505.
- Malinsky, D. 2024. A cautious approach to constraint-based causal model selection. arXiv:2404.18232.
- Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems*. San Mateo, CA: Morgan Kaufmann.
- Pearl, J. 1995. Causal diagrams for empirical research. *Biometrika*, 82(4): 669–688.
- Pearl, J. 1998. Graphs, causality, and structural equation models. *Sociological Methods and Research*, 27(2): 226–284.
- Pearl, J. 2000. *Causality: Models, Reasoning, and Inference*. New York, NY, USA: Cambridge University Press, 2nd edition.
- Pearl, J.; and Meshkat, P. 1999. Testing regression models with fewer regressors. In Heckerman, D.; and Whittaker, J., eds., *Artificial Intelligence and Statistics 99*, 255–259. San Francisco, CA: Morgan Kaufmann.
- Richardson, T. 2003. Markov properties for acyclic directed mixed graphs. *Scandinavian Journal of Statistics*, 30(1): 145–157.
- Richardson, T. S. 2009. A factorization criterion for acyclic directed mixed graphs. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, 462–470. Arlington, Virginia, USA: AUAI Press. ISBN 9780974903958.
- Robins, J. M.; Hernan, M. A.; and Brumback, B. 2000. Marginal structural models and causal inference in epidemiology.
- Rotmensch, M.; Halpern, Y.; Tlimat, A.; Horng, S.; and Sontag, D. 2017. Learning a health knowledge graph from electronic medical records. *Scientific reports*, 7(1): 5994.
- Sachs, K.; Perez, O.; Pe'er, D.; Lauffenburger, D. A.; and Nolan, G. P. 2005. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721): 523–529.
- Scutari, M. 2010. Learning Bayesian Networks with the bnlearn R Package. *Journal of Statistical Software*, 35(3): 1–22.
- Spirtes, P.; Glymour, C. N.; and Scheines, R. 2001. *Causation, Prediction, and Search*. MIT Press, 2nd edition. ISBN 9780262194402.
- Spirtes, P.; Richardson, T.; Meek, C.; Scheines, R.; and Glymour, C. N. 1998. Using path diagrams as a structural equation modelling tool. *Sociological Methods and Research*, 27(2): 182–225.
- Sverchkov, Y.; and Craven, M. 2017. A review of active learning approaches to experimental design for uncovering biological networks. *PLOS Computational Biology*, 13(6): 1–26.
- Takata, K. 2010. Space-optimal, backtracking algorithms to list the minimal vertex separators of a graph. *Discrete Applied Mathematics*, 158(15): 1660–1667.
- Tennant, P. W. G.; Murray, E. J.; Arnold, K. F.; Berrie, L.; Fox, M. P.; Gadd, S. C.; Harrison, W. J.; Keeble, C.; Ranker, L. R.; Textor, J.; Tomova, G. D.; Gilthorpe, M. S.; and Ellison, G. T. H. 2020. Use of directed acyclic graphs (DAGs) to identify confounders in applied health research: review and recommendations. *International Journal of Epidemiology*, 50(2): 620–632.
- Tian, J.; and Pearl, J. 2002a. A General Identification Condition for Causal Effects. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI 2002)*, 567–573. Menlo Park, CA: AAAI Press/The MIT Press.
- Tian, J.; and Pearl, J. 2002b. On the Testable Implications of Causal Models with Hidden Variables. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, 519–527. ISBN 1-55860-897-4.
- Zantedeschi, V.; Franceschi, L.; Kaddour, J.; Kusner, M. J.; and Niculae, V. 2023. DAG Learning on the Permutahedron. arXiv:2301.11898.
- Zheng, Y.; Huang, B.; Chen, W.; Ramsey, J.; Gong, M.; Cai, R.; Shimizu, S.; Spirtes, P.; and Zhang, K. 2024. Causal-learn: Causal discovery in python. *Journal of Machine Learning Research*, 25(60): 1–8.