
Causal Discovery in Mixtures of Populations

Bijan H. Mazaheri¹

Spencer Gordon²

Yuval Rabani³

Leonard Schulman⁴

¹Thayer School of Engineering, Dartmouth College, Hanover, New Hampshire, USA

²Fragment Data Technologies, San Francisco, California, USA

³Department of Computer Science, Hebrew University of Jerusalem, Jerusalem, Israel

⁴Computing and Mathematical Sciences, California Institute of Technology, Pasadena, California, USA

Abstract

Causal discovery aims to learn causal structures up to certain symmetries. Diverse populations or changing environments give rise to heterogeneous data in the following sense: each population/environment is a “source” which idiosyncratically determines the forms of causal effects. From this perspective, the source is a latent common cause for every observed variable. While some methods for causal discovery can work around latent confounding in special cases, a global confounder poses a significant challenge. The only known ways to deal with latent global confounding involve making assumptions that limit structural equations and/or noise functions. We demonstrate that globally confounded causal structures can still be identified with arbitrary structural equations and noise functions, so long as the number of latent classes remains small relative to the size and sparsity of the underlying DAG. The approach relies on agglomerating variables into large-enough matrices of moments, whose ranks directly reveal graphical properties of the causal structure. We also provide a statistical test to test the rank of these matrices.

1 INTRODUCTION

Many approaches to studying causal systems use structural causal models (SCMs) to graphically model causal relationships in a directed acyclic graph (DAG) [Pearl, 2009]. In an SCM, $A \rightarrow B$ indicates “ A has a direct causal effect on B .”

Formally, “d-separation rules”¹ give graphical criteria for statistical independence and dependence under the assumptions of faithfulness (independence implies d-separation)

and the causal Markov condition (d-separation implies independence). Constraint-based causal discovery utilizes this connection to uncover information about an unknown causal structure [Spirtes et al., 2000].

A critical assumption for many causal discovery algorithms is “causal sufficiency,” i.e., no unobserved confounding. In most real-world settings, a causal system spans both observed and latent variables. To address this, approaches such as FCI [Spirtes, 2001] have been developed to uncover equivalence classes of causal structures and the existence of pairwise confounding. In the parametric setting, more “pervasive” confounding has been studied under some parametric restrictions on its influence [Frot et al., 2019, Cai et al., 2023, Agrawal et al., 2023].

A “flipped” version of this problem is studied in causal representation learning (CRL) [Schölkopf et al., 2021, Squires et al., 2023] and latent variable models [Silva et al., 2006, Anandkumar et al., 2013, Xie et al., 2020, Huang et al., 2022], which seek to uncover the structure of *latent* variables using *known* (empty or sparse) structure within the *observed* variables. A question of particular theoretical interest in both problems is that of “identifiability,” i.e., the conditions under which a structure is uniquely recoverable.

There has been limited exploration into how known relationships *between latent and observed variables* can be utilized to learn more about a system. The simplest setting is causal discovery under pervasive confounding. Pervasive confounding is generally detrimental to identifiability because it exerts a broad confounding influence on the system. However, this known pervasiveness also means that the system contains significant information about the confounding, which makes its effects easier to learn and remove.

The current approaches to causal discovery in the presence of pervasive confounding are parametric and provide limited insight into the identifiability trade-offs involved. No approaches have yet addressed the issue of a *discrete* universal (i.e., edges to all observable variables) latent confounder. Discrete confounding allows for a simpler quantification of

¹See Pearl [2009] for a review of d-separation or Appendix B for a summary of important results used in this manuscript.

the “amount” of unobserved confounding: the number of latent populations or mixture components. Quantifying the relationship between confounding and the resulting identifiability requirements on the observed system is of particular theoretical interest.

Practically, discrete confounding is related to the study of mixture models. Combined data from multiple sources or populations often gives rise to such a setting. In many mixture problems, data is also discrete, such as the study of single-nucleotide polymorphisms (or SNPs) in heterogeneous populations.

1.1 PROBLEM STATEMENT

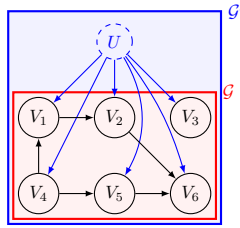


Figure 1: The goal is to learn the graph structure $\mathcal{G}$ without observing U .

Suppose we augment a DAG $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ with an unobserved U that has k “latent classes”, each of which exerts a distinctive signal on all the observed variables $\mathbf{V}$. The result is a mixture model whose DAG, $\mathcal{G}' = (\mathbf{V} \cup \{U\}, \mathbf{E}')$ includes additional arrows from U to every $V \in \mathbf{V}$. See Figure 1 for an example. We will refer to $\mathcal{G}$ as the observed sub-graph, and we will use $\Delta := \max_{V \in \mathbf{V}} \deg^{\mathcal{G}}(V)$ to denote its maximum (in plus out) degree.

The goal is to uncover the observed sub-graph’s structure $\mathcal{G}$ up to a Markov equivalence class (MEC) (i.e., a CP-DAG) using statistics gained from the “observed marginal distribution,” i.e., $\Pr(\mathbf{V})$ marginalized over U . Notice that U confounds all pairs of variables in $\mathbf{V}$, which requires it to be in *every* separating set. This means that the correspondence between conditional independence and d-separation is no longer adequate to make any deductions about causal structure.

Assumptions We will assume (1) that the distribution is faithful² with respect to $\mathcal{G}'$, (2) that U is discrete with a known number of latent classes k , and (3) that there are no latent variables other than U (i.e., causal sufficiency holds for $\mathcal{G}'$). Throughout, all variables are discrete: the observables $\mathbf{V}$ are categorical (we will focus on binary $\mathbf{V}$) and U is also categorical with k classes. k therefore represents the “complexity” of unobserved confounding, as

²The precise assumption is a slight extension of faithfulness to the mixture setting, discussed later.

more latent classes are capable of exerting “more complex” signals. Since our deductions rely on each value of U having distinct effects on the observables, the most difficult case arises for Bernoulli observable variables expressed by a single parameter.

We make no parametric assumptions (e.g. linearity of the structural equations or Gaussian noise) on the observed portion of the graph. The only mild parametric assumption is the categorical nature of U . Our approach does not require Bernoulli observables, but is built around this especially difficult case to achieve more general identifiability results.

1.2 CONTRIBUTIONS

Tests of Rank. Conditional independence testing is considered a crucial component of constraint-based methods for non-parametric causal discovery [Spirtes et al., 2000, Squires and Uhler, 2023]. By studying a problem for which there are *no conditional independencies*, we show that conditional independence tests are a special case of a broader set of tests that may be modified based on parametric assumptions about unobserved confounding. These modifications can be made while avoiding parametric assumptions on the observable variables. In this paper, we use a test for matrix rank and develop a hypothesis test for it based on the work of Ratsimalahelo [2001]. We demonstrate that this approach is superior to thresholding singular values, as in Anandkumar et al. [2012a], and note that it may be useful in other settings that use matrix rank, such as Squires et al. [2023]. The test can be accessed using `pip install probrank`.

Agglomeration in Structure Learning. Tests of matrix rank are usually used in “large alphabet” settings, where random variables can be reduced to categorical variables with large cardinality [Anandkumar et al., 2010, 2012b,a]. To extend the use of these tests to a “small alphabet” setting (i.e., Bernoulli observables), we show how to group or “agglomerate” simple variables into ones with larger alphabets. This process is nontrivial, since the properties of these agglomerated tests must then be disentangled to make conclusions about the original Bernoulli random variables. As such, the paper makes significant theoretical contributions to the conditional independence (and matrix rank) properties of agglomerated variables, which may have use in areas of causal abstraction [Beckers and Halpern, 2019, Beckers et al., 2020] and causal representation learning [Schölkopf et al., 2021].

Algorithmic and Theoretical Contributions. Our main result is, to our knowledge, the first algorithm for identifying causal structures in this mixture setting without parametric assumptions on the observed variables. The correctness of this algorithm and the uniqueness of its output provide sufficient conditions for identifying causal structures from data across multiple populations.

The identification of a causal structure (or equivalence class) is based on the ability to eliminate candidate structures based on the observed properties of the probability distribution. A larger k therefore reduces identifiability by increasing the support of valid probability distributions. For example, $k \geq 2^{|V|}$ is sufficient to model *any* marginal distribution on Bernoulli variables V , even with a fully underpowered $E = \emptyset$. To see this, assign one value of u to each of the $2^{|V|}$ possible assignments to the $|V|$ observed binary variables. By changing the relative probabilities of $U = u$, we may specify any marginal distribution. Since any distribution can be generated in this way (even when the graph is empty), such a large k clearly prohibits identifiability.

Identification of a causal structure also depends on the number of observed variables $|V|$ and the sparsity of $\mathcal{G}$. A larger $|V|$ provides more information about U , and a sparser $\mathcal{G}$ further restricts the possible distributions generated (e.g., a complete graph may generate any marginal distribution even with $k = 1$). A key information-theoretic question is how the identifiability requirements for $\mathcal{G}$ scale relative to k , $|V|$, and the maximum degree Δ . To this end, the conditions under which our algorithm recovers a unique (and correct) MEC—established later via Theorem 1—also yield the following identifiability result.

Corollary 1. Consider $\mathcal{G} = (V, E)$ with mixture source $U \in \{1, \dots, k\}$ and degree bound Δ . If

$$|V| \geq (\Delta^3 + 2\Delta^2 + 4\Delta + 2)\lceil \lg(k+1) \rceil + 2\Delta^2 + 2\Delta^3,$$

then $\mathcal{G}$ is generically³ identifiable up to its MEC.

In practice, identifiability often occurs in far smaller graphs, as later experiments with $k = 2$ and only 7 vertices demonstrate. While loose, this is the best and only known bound to guarantee identifiability.

We achieve this identifiability result by developing an algorithm that repeatedly identifies a mixture of k product distributions (which we call k -MixProd) using an oracle that takes time τ . We also use an oracle that can solve for non-negative rank in time ρ . Our algorithm runs in time $|V|^{\mathcal{O}(\Delta^2 \lg(k))} \rho + \mathcal{O}((\Delta^2 \lg(k)) + |E|)k2^{\Delta^2} \tau$. Both Corollary 1 and the associated runtime are a consequence of the algorithm and its requirements for its success, as outlined in Theorem 1. See Appendix E for further discussion on the runtime of ρ and τ .

1.3 RELATED WORKS

Causal discovery. The PC algorithm was the first causal discovery algorithm to use conditional independence, as outlined by [Spirtes et al. \[2000\]](#). Many causal discovery algorithms have since been developed, summarized in [Squires](#)

and [Uhler \[2023\]](#). A number of algorithms address the presence of latent confounding using one of two assumptions that “limit” the confounding influence: (1) low out-degree latents or (2) parametric limitations in the random variables.

The first type of assumption involves limiting the *out-degree* of latent confounding. For example, the FCI algorithm can detect the presence of unobserved confounders that act on *only two* observed variables [[Spirtes et al., 1993](#), [Spirtes, 2001](#)]. [Richardson and Spirtes \[2002\]](#)’s seminal work introduced ancestral graphs for the general study of this setting. It is important to note that ancestral graphs are larger equivalence classes than MECs. As such, these approaches recover the structure at a coarser resolution, and do so under assumptions that do not hold in the mixture setting.

Global Confounding. Recent work has studied “global” or “pervasive” confounding [[Gordon et al., 2023](#), [Frot et al., 2019](#), [Agrawal et al., 2023](#)]. Pervasive confounders affect all observable variables in a system and are often observed when data are gathered over a large area or over a long period of time. Such confounding is graphically modeled as a single unobserved variable that links every observable variable (see Figure 1). This structure d-connects all possible pairs of vertices, making all variables statistically dependent even after conditioning on any other variables, which rules out the use of the PC algorithm.

Pervasively confounded DAGs cannot be learned using only conditional independence tests. Instead, these settings require parametric assumptions that restrict the space of probability distributions. For example, [Frot et al. \[2019\]](#) was able to show superior performance in settings with large-degree confounders with linear relationships and additive sub-Gaussian or elliptical noise. Other settings include linear structural equations with non-Gaussian additive noise [[Cai et al., 2023](#)] and non-linear structural equations within a finite-dimensional Hilbert space [[Agrawal et al., 2023](#)].

The setting studied in this paper involves a discrete unobserved pervasive confounder that imposes a “mixture” or “latent class”. Methods designed for continuous latent confounders are ill-equipped to exploit the special properties of mixture models. Furthermore, the methods rely on parametric assumptions about the observed variables. Since our approach requires no parametric assumptions on the observed variables, it provides a clearer picture of the identifiability conditions in a more general setting.

k -MixProd. Almost all research on mixtures with categorical data leverages the assumption of mutually independent observed variables within the source distributions [[Freund and Mansour, 1999](#), [Cryan et al., 2001](#), [Chaudhuri and Rao, 2008](#), [Feldman et al., 2008](#), [Rabani et al., 2014](#), [Chen and Moitra, 2019](#), [Gordon et al., 2021, 2024](#)] (see also the earlier seminal work of [Kearns et al. \[1994\]](#)). We call this the k -MixProd problem (where k denotes the number of latent

³Generic identification means that the unidentifiable cases have Lebesgue measure 0 in the space of parameters.

classes), whose identifiability is discussed in Appendix C.

k-MixProd assumes that the causal model within each source is an empty graph and aims to learn the conditional distributions within each mixture component. Early work by Anandkumar et al. [2012a] and Gordon et al. [2023] broadened this class of independence assumptions, exploring Markov random fields and Bayesian networks, respectively. A key assumption of Gordon et al. [2023] and *k*-MixProd is that the Bayesian DAG structure within each component of the mixture is known. This paper will develop an algorithm to recover this structure, thereby implicitly providing conditions for the identifiability of Bayesian DAG mixture models with unknown structure.

Latent Variable Models. A related and complementary problem in causal discovery is learning a causal graph over many latent variables from their observable children. Many classic approaches to this problem use “rank constraints,” which are rank deficiencies on covariance matrices between sets of variables [Silva et al., 2006, Anandkumar et al., 2013, Xie et al., 2020]. This has been extended to hierarchical latent variable models [Huang et al., 2022]. Structure-based approaches have also been used to discover the presence and cardinality of hidden variables [Elidan et al., 2000, Elidan and Friedman, 2005], though these target a different setting from ours. Causal representation learning has also used rank deficiencies [Squires et al., 2023]. In fact, a number of very recent works have employed tests of rank to learn structures in latent variables under a “purity” assumption, in which there are no dependencies between the observed random variables [Chen et al., 2024, Kong et al., 2024, Lee and Gu, 2026]—equivalently, the mutual-independence assumption of *k*-MixProd, in which the observed variables are independent given the latent source. The hypothesis test that we develop may be useful for all of these problems. In addition, the algorithmic tools we develop to handle dependencies among observed variables may help move these problems beyond the assumptions of “purity.”

Network-level Differences. Causal discovery becomes more difficult when allowing for network-level differences within the mixture components, e.g., Saeed et al. [2020] and Varici et al. [2024] as well as Strobl [2023] for mixtures of cyclic DAGs. We focus on a shared graph structure, allowing for significantly fewer parametric assumptions.

2 PRELIMINARIES

Notation. We will use the capital Latin alphabet to denote random variables, which are vertices on our DAG. When referring to sets of these variables, we will use bolded font, e.g. $\mathbf{X} = \{X_1, X_2, \dots\}$. To refer to components of a graph, we will use the following operators: $\mathbf{PA}(V)$, $\mathbf{CH}(V)$ will refer to the parents and children of

V . $\mathbf{AN}(V)$, $\mathbf{DE}(V)$ will refer to the ancestors and descendants of V . $\mathbf{AN}(V) \cup \{V\}$ and $\mathbf{DE}(V) \cup \{V\}$ are denoted using $\mathbf{AN}^+(V)$, $\mathbf{DE}^+(V)$ respectively. $\mathbf{MB}(V) := \mathbf{PA}(V) \cup \mathbf{CH}(V) \cup \mathbf{PA}(\mathbf{CH}(V)) \setminus \{V\}$ will refer to the Markov boundary (i.e. the unique minimal Markov blanket) of V [Pearl, 2009]. $\mathbf{NB}_\ell(V)$ refers to the distance ℓ neighborhood of V . For all of these sets, we will use a bar to denote the complement, e.g., $\overline{\mathbf{X}} = V \setminus \mathbf{X}$.

As these operators act on graphs, they can specify the graph structure being used in the superscript, e.g. $\mathbf{PA}^{\mathcal{G}}(V)$. Unless otherwise specified, operators should be assumed to apply to the observed subgraph $\mathcal{G}$ and not $\mathcal{G}'$, i.e., $U \notin \mathbf{PA}^{\mathcal{G}}(V)$. We will also occasionally write tuples to indicate the intersection of the sets for two vertices, e.g. $\mathbf{CH}(V, W) = \mathbf{CH}(V) \cap \mathbf{CH}(W)$. Finally, these operators can also act on sets to indicate the union of the operation, e.g.

$$\mathbf{PA}(\mathbf{X}) = \bigcup_{X \in \mathbf{X}} \mathbf{PA}(X) \setminus \mathbf{X}. \quad (1)$$

We make use of lowercase letters to denote assignments, for example, x will denote the assignment $X = x$. This can also be applied to operators, e.g. $\mathbf{mb}(V)$ denotes an assignment to $\mathbf{MB}(V)$. Assignments can be specified by a set of assignments in the subscript, e.g. $\mathbf{mb}_c(V)$ obtains assignments for $\mathbf{MB}(V) \subseteq \mathbf{C}$ from the assignments of $\mathbf{c}$ to $\mathbf{C}$. $\text{rk}_+(\mathcal{M})$ denotes the nonnegative rank of matrix $\mathcal{M}$.

Constraint-Based Causal Discovery. When two variables are found to be conditionally independent given another set, e.g. $A \perp\!\!\!\perp B \mid \mathbf{C}$, we say that $\mathbf{C}$ is a “separating set” for A, B , which acts as a witness for the absence of an edge between them. This critical observation allows the discovery of a “skeleton,” i.e., a causal graph containing all causal adjacencies with no orientations. In addition, separating sets also allow in-degree > 2 vertices with non-adjacent parents to have their parental edges oriented. These are called v-structures, unshielded colliders, or immoralities.

These partial DAG structures become “completed partial DAGs,” or CP-DAGs [Andersson et al., 1997], by further orienting edges according to the “Meek rules” [Meek, 1995] that enforce acyclicity and no additional unshielded colliders. CP-DAGs represent a set of possible DAG structures that agree with the observed conditional independence, which is called a Markov equivalence class (MEC).

Outline of Algorithm. Our approach is based on the PC-algorithm [Spirtes et al., 2000], which works in two phases. Phase I will begin with a complete graph and remove edges between variables when we find evidence of non-adjacency (using rank constraints). This phase will only remove edges between groupings of variables, so its termination will not guarantee that we have discovered all possible non-adjacencies. Instead, a provably small subset of the graph will have *false positive* edges.

In Phase II, we will use the structure we have uncovered to induce instances of k -mixtures of products, a famous and long-studied problem [Gordon et al., 2021, 2024, Allman et al., 2009], which we will abbreviate k -MixProd. An oracle solver for this problem will then identify the joint probability distribution between subsets of $\mathbf{V}$ and the latent class U . Access to this joint probability distribution allows the rest of the structure to be resolved using conditional independence tests on distributions that are conditioned on U .

Phase III mirrors the last phase of the PC algorithm: identifying immoralities using non-adjacencies and separating sets, and then propagating orientations according to Meek rules. This phase is identical to the PC algorithm, so its details are omitted from this manuscript.

3 RANK TESTS

This section will introduce “rank tests” which will serve as a replacement for conditional independence tests as a test for d-separation or d-connectedness. These serve as a generalization of rank-deficiencies of covariance matrices noted in Spirtes et al. [2000]. The use of rank was also mentioned in Anandkumar et al. [2012b], but not formalized. Rank constraints induced by hidden variables of restricted cardinality have also been studied by Zjawin et al. [2021].

3.1 k -MIXTURE INDEPENDENCE

To determine non-adjacency, we will leverage the signal U leaves on the marginal probability distributions of variables that are independent conditional on U . First, we interpret the marginal probability distribution as a matrix.

Definition 1. Given two discrete variables $X, Y \in \mathbf{V}$ with $|X| = n$ and $|Y| = m$, define the “probability matrix” $\mathcal{M}[X, Y] \in [0, 1]^{n \times m}$ to be

$$\mathcal{M}[X, Y]_{x,y} := \Pr(x, y), \quad (2)$$

where $x \in \{1, \dots, n\}$ and $y \in \{1, \dots, m\}$. Similarly, for $C \subseteq \mathbf{V}$, define

$$\mathcal{M}[X, Y \mid c]_{x,y} := \Pr(x, y \mid c). \quad (3)$$

These matrices have ranks that give valuable information about the graphical structure under latent class confounding. Our algorithm will use the following fact.

Lemma 1 (Rank Test). For V_i, V_j with cardinality $> k$ confounded by U in $\mathcal{G}'$, $V_i \perp\!\!\!\perp_d^{\mathcal{G}'} V_j \mid C$ if and only if (generically) $\text{rk}_+(\mathcal{M}[V_i, V_j \mid C]) \leq k$.

The cardinality requirement $|V_i|, |V_j| > k$ is needed only to ensure that d-connectedness forces $\text{rk}_+ > k$ (the reverse direction, via Lemma 2); the forward implication, that d-separation yields $\text{rk}_+ \leq k$, holds for any cardinality.

To prove this Lemma we must show both directions. First, we observe that we can decompose $\mathcal{M}[X, Y \mid c]$ as follows:

$$\mathcal{M}[X, Y \mid c] = \sum_u \Pr(u) \mathcal{M}[X, Y \mid c, u]. \quad (4)$$

$X \perp\!\!\!\perp Y \mid C, U$, so $\mathcal{M}[X, Y \mid c, u]$ can be written as the outer product of two vectors describing the probabilities of each variable. Therefore, we conclude that $\text{rk}_+(\mathcal{M}[X, Y \mid c, u]) = 1$. If $U \in [k]$, then summing k such terms gives $\text{rk}_+(\mathcal{M}[X, Y \mid c]) \leq k$.

Having shown that d-separation in $\mathcal{G}$ upper bounds the rank of probability matrices, we now seek a lower bound on the rank in the case of d-connectedness. This will require a “faithfulness-like” assumption that the $\mathcal{M}[X, Y \mid c, u]$ terms in Equation 4 are linearly independent, resulting in an overall $\text{rk}_+(\mathcal{M}[X, Y \mid c]) > k$ when X and Y are not independent conditioned on U . Lemma 2 shows that this condition holds generically.

Lemma 2. Consider a mixture of Bayesian network distributions, each of which is faithful to $\mathcal{G}$. If $X \not\perp\!\!\!\perp_d^{\mathcal{G}} Y \mid C$ and $|X| = n$, $|Y| = m$ with $n, m > k$, then for all c , $\text{rk}_+(\mathcal{M}[X, Y \mid c]) > k$ with full Lebesgue measure.

The proof of Lemma 2 (see Appendix I) involves applying faithfulness to each summand of Equation 4. To make these rank tests practical, Appendix D develops a hypothesis test for the rank of a noisy matrix of estimated probabilities, which we later test in Section 5.

4 ALGORITHM

We will now outline the first two phases of our algorithm, leaving out the final phase of orienting edges with respect to Meek’s rules. The first phase involves agglomerating sets of variables and applying rank tests. We then analyze the output $\mathcal{G}_1$, which has removed *most but not all* of the missing edges. We define FP edges as these “leftovers” and show that they are contained in a bounded-size subset. This fact allows us to resolve all of the non-adjacencies in $\mathcal{G}$ by setting up instances of k -MixProd in Phase II.

4.1 PHASE I: AGGLOMERATED RANK TESTS

Lemma 1 allows for a simple generalization of the PC algorithm provided that our probability matrix $\mathcal{M}[X, Y]$ is at least $k + 1$ by $k + 1$. Unfortunately, categorical variables ranging over smaller alphabets (such as the binary alphabets addressed by this paper) do not contain sufficient information to detect non-adjacency in cases of larger k . We resolve this problem by the agglomeration of $\lg(k + 1)$ -cardinality sets of small-alphabet (binary) variables into supervariables of larger cardinality (i.e., $2^{\lg(k+1)} = k + 1$).

Definition 2. Consider DAG $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, $V_i, V_j \in \mathbf{V}$, and sets $\mathbf{S}_i, \mathbf{S}_j \subseteq \mathbf{V} \setminus \{V_i, V_j\}$. We call the ordered pair $(\mathbf{S}_i^+, \mathbf{S}_j^+) = (\mathbf{S}_i \cup \{V_i\}, \mathbf{S}_j \cup \{V_j\})$ an **independence pre-serving agglomeration (IPA)** of (V_i, V_j) if, for some **IPA conditioning set** $\mathbf{C} \subset \mathbf{V}$,

$$\mathbf{S}_i^+ \perp_d^{\mathcal{G}} \mathbf{S}_j^+ \mid \mathbf{C}.$$

The creation of supervariables allows us to use conditional rank tests in place of conditional independence tests. This leads to a modified version of the PC algorithm that searches over pairs of supervariable agglomerations instead of pairs of vertices, given in Algorithm 1.

Algorithm 1: Phase I

Input: The marginal probability distribution $\Pr(\mathbf{V})$, marginalized over U .
Output: An undirected graph $\mathcal{G}_1 = (\mathbf{V}, \mathbf{E}_1)$ and a separating set $\mathbf{C}_{ij}$ for each detected non-adjacency.
 Begin with a complete undirected graph $\mathcal{G}_1 = (\mathbf{V}, \mathbf{V} \times \mathbf{V})$ and $d_{\max} \leftarrow |\mathbf{V}| - 1$.
for $\ell = 0$ **to** $\ell = d_{\max}$ **do**
 for $\mathbf{C} \subset \mathbf{V}$ and $|\mathbf{C}| = \ell$ **do**
 for $\mathbf{S}, \mathbf{S}' \subseteq \mathbf{V} \setminus \mathbf{C}$, with
 $|\mathbf{S}| = |\mathbf{S}'| = \lceil \lg(k) \rceil + 1$ **do**
 if arbitrary assignment c has
 $\text{rk}_+(\mathcal{M}[\mathbf{S}, \mathbf{S}' \mid c]) \leq k$ **then**
 Remove edges between $\mathbf{S}$ and $\mathbf{S}'$ in $\mathcal{G}_1$.
 $\mathbf{C}_{i,j} \leftarrow \mathbf{C}$ for each $V_i \in \mathbf{S}, V_j \in \mathbf{S}'$
 Update Δ to the max degree of $\mathcal{G}_1$.
 end
end
end

Lemma 3. Algorithm 1 utilizes $|\mathbf{V}|^{\mathcal{O}(\Delta^2 \log(k))}$ non-negative rank tests.

Proof. Lemma 11 (see Appendix B) tells us that the maximum size of a separating set is $\alpha := (\lceil \lg(k) \rceil + 1)\Delta^2$, so we need to check $\binom{|\mathbf{V}|}{\alpha} + \binom{|\mathbf{V}|}{\alpha-1} + \dots + \binom{|\mathbf{V}|}{1}$ possible separating sets, which is $|\mathbf{V}|^{\mathcal{O}(\Delta^2 \log(k))}$. We must iterate over all possible supervariables for each separating sets, which is upper bounded by $\binom{|\mathbf{V}|}{2(\lceil \lg(k) \rceil + 1)}$, which is $|\mathbf{V}|^{\mathcal{O}(\log(k))}$. $\square$

The exponent of $\mathcal{O}(\Delta^2 \log(k))$ in Lemma 3 can be replaced with $|\mathbf{V}|$, a trivial maximum size of a separating set.

4.1.1 FP Edges

Phase I of our algorithm removes an edge between two non-adjacent variables through a rank test so long as there

exists an IPA for the non-adjacency. Not all non-adjacencies will contain an IPA, so the adjacency graph $\mathcal{G}_1$ contains a *superset* of the true adjacencies.

Definition 3. $\mathbf{E}_1 \setminus \mathbf{E}$ are **false positive (FP)** edges.

To see why FP edges exist, it helps to isolate the vertices that lie below both endpoints.

Definition 4. Let $\mathbf{D}_{ij} := \mathbf{DE}(V_i, V_j)$ be the common descendants of V_i and V_j , and let $\mathbf{A}_{ij} := \mathbf{V} \setminus \mathbf{D}_{ij}$ be the remaining vertices (which include V_i and V_j).

Conditioning on a common child of V_i and V_j (any descendant of one) opens the collider $V_i \rightarrow \cdot \leftarrow V_j$, so such a vertex can never belong to a separating set. It cannot be agglomerated into $\mathbf{S}_i^+$ or $\mathbf{S}_j^+$ either: separating a descendant of a common child from the opposite endpoint would require conditioning on an ancestor of that “collider descendant,” which would open up the collider path between V_i and V_j . Consequently, if too many vertices are common children or a descendant of one, no IPA exists, and an FP edge remains after Phase I. Figure 2 shows an extreme case, in which $\mathbf{A}_{ij} = \{V_i, V_j\}$ leaves nothing to agglomerate.

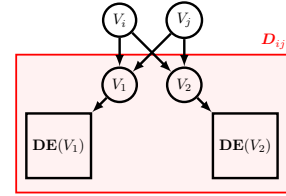


Figure 2: An FP edge remains between V_i and V_j after Phase I when every other vertex is a common descendant $\mathbf{D}_{ij}$ (here, a common child or one of its descendants). Although V_i and V_j are d-separated by $\mathbf{C} = \emptyset$, no IPA can be formed: the only vertices left to agglomerate lie in $\mathbf{D}_{ij}$, and conditioning on any of them would open a collider. Equivalently, $\mathbf{A}_{ij} = \{V_i, V_j\}$ is too small. The population variable U is omitted to avoid clutter.

This illustrates that FP edges can occur for pairs of vertices with too many common descendants, leaving too few vertices in $\mathbf{A}_{ij}$ to form an IPA (shown in Figure 2).

4.1.2 Using non-descendants to form IPAs

It turns out that the majority of pairs of vertices do not have many common descendants and can be agglomerated. To show this, we take $\mathbf{S}_i, \mathbf{S}_j \subseteq \mathbf{A}_{ij}$. The following lemma shows this restriction is *safe*: whenever the two supervariables can be separated at all, they can be separated using a conditioning set drawn entirely from $\mathbf{A}_{ij}$. Confining the conditions of existence to $\mathbf{A}_{ij}$ therefore never costs us a witness of non-adjacency, and—since $\mathbf{A}_{ij}$ excludes every common descendant—such a witness never conditions on an open collider.

Lemma 4. Let V_i, V_j be non-adjacent and let $S_i, S_j \subseteq A_{ij}$. If S_i^+ and S_j^+ are d -separated by any set, then they are d -separated by some $C \subseteq A_{ij}$; in particular, C contains no common descendant of V_i, V_j , so (S_i^+, S_j^+) is an IPA.

Proof. An ancestor of a non-common-descendant is again a non-common-descendant, so A_{ij} is ancestrally closed; hence $\text{AN}^+(S_i^+ \cup S_j^+) \subseteq A_{ij}$. If S_i^+ and S_j^+ are d -separated by some set, they are separated in the moral graph $(\mathcal{G}[\text{AN}^+(S_i^+ \cup S_j^+)])^{(m)}$ by a set $C \subseteq \text{AN}^+(S_i^+ \cup S_j^+)$; by Lemma 12 this C is a separating set in $\mathcal{G}$. Since $C \subseteq \text{AN}^+(S_i^+ \cup S_j^+) \subseteq A_{ij} = V \setminus D_{ij}$, it contains no common descendant of V_i, V_j . $\square$

Lemma 4 shows that working inside A_{ij} never removes our ability to separate, so the sole remaining obstruction to forming an IPA is having too few vertices in A_{ij} . Lemma 5 quantifies how many are needed, and since A_{ij} contains the non-descendants of both V_i and V_j , it is too small only when both endpoints have few non-descendants—that is, when both lie in the “early vertices” H introduced next.

Lemma 5. An IPA S_i^+, S_j^+ for $V_i, V_j \in V$ exists so long as $|A_{ij}| \geq (2 + \Delta^2) \lceil \lg(k+1) \rceil - 2$.

To utilize Lemma 5, we define a special set of vertices, which has an upper-bounded cardinality.

Definition 5. We define the early vertices,

$$H := \{V \in V \text{ s.t. } |\overline{\text{DE}}(V)| < (2 + \Delta^2) \lceil \lg(k+1) \rceil - 2\}.$$

Lemma 6. $|H| \leq (2 + \Delta^2) \lceil \lg(k+1) \rceil - 2$.

This definition is carefully crafted so that any pair of vertices with at least one vertex out of H has enough non-descendants to augment both vertices to an agglomeration set of at least $\lceil \lg(k+2) \rceil$ variables. Such a witness to the non-adjacency of these pairs means that they do not have FP edges.

Lemma 7. After Phase I (Algorithm 1), all false positive edges lie within the early vertices, i.e., $E_1 \setminus E \subseteq H \times H$.

4.2 PHASE II: CORRECT FP EDGES

Recall that the marginal probability distribution cannot use independence tests to discover non-adjacency because U confounds all pairs of vertices. An important observation is that the within-source distribution $\Pr(V | u)$ would not suffer from this limitation because it would allow us to query conditional independence that conditions on the unobserved U .

Phase II will make use of this observation by selecting subsets of variables $T \subseteq V$ on which to obtain $\Pr(T | u)$ using techniques from discrete mixture models. We will

then apply regular conditional independence checks on the recovered $\Pr(T | u)$ to detect FP edges. We will use a separate $T_{ij} \ni V_i, V_j$ agglomeration to verify each edge $(V_i, V_j) \in E_1$, though this process can likely be optimized.

We will use a lemma by Allman et al. [2009], which is a direct consequence of a result by Kruskal [1977] on the uniqueness of decomposing order-3 tensors into rank one components.

Lemma 8 (Allman et al. [2009]). Consider the discrete mixture source $U \in \{1, \dots, k\}$ and discrete variables X_1, X_2, X_3 with cardinality $\kappa_1, \kappa_2, \kappa_3$ that are mutually independent given U (i.e., $X_a \perp\!\!\!\perp X_b | U$ for all $a \neq b$). The mixture is generically identifiable (with full Lebesgue measure on the parameter space) if

$$\min(\kappa_1, k) + \min(\kappa_2, k) + \min(\kappa_3, k) \geq 2k + 2.$$

This result has been used extensively to uncover model parameters when three observables are known to be conditionally independent on an unknown mixing variable [Anandkumar et al., 2014]. The conditions for identifiability are therefore quite mild — Phase I only needs to uncover enough sparsity to d -separate three sufficiently large independent agglomerations, one of which will be T_{ij} with the intention of verifying or removing $V_i \rightarrow V_j$. Conveniently, the constrained nature of our FP edges means that the graph $\mathcal{G}_1$ is sufficiently sparse to allow such constructions.

T_{ij} must be designed to include enough information to discover a non-adjacency between V_i, V_j . In other words, we need to ensure that, if V_i and V_j are not adjacent, then T_{ij} contains a separating set $C \subset T_{ij}$ such that $V_i \perp_d^{\mathcal{G}} V_j | C$. Ensuring that T_{ij} also has the distance-1 neighborhoods of V_i, V_j is enough to satisfy this requirement.

Definition 6. Given vertices V_i, V_j , let T_{ij} be the set containing V_i, V_j and all vertices that are distance 1 in $\mathcal{G}_1$ from V_i or V_j .

Lemma 9. If vertices V_i, V_j are nonadjacent, the set T_{ij} contains a valid separating set $C \subseteq T_{ij}$ such that $V_i \perp_d V_j | C$.

Lemma 9 guarantees that the conditional probability distribution $\Pr(T_{ij} | u)$ has sufficient information to verify or falsify the adjacency of V_i and V_j .

The rest of the construction of the k -MixProd instances is left to Appendix F. Generally, it involves ensuring that the recovered $\mathcal{G}_1$ from Phase I is sparse enough to d -separate all T_{ij} from two other supervariables of sufficient cardinality. Lemma 15 tells us the number of vertices that are needed to set up an instance of k -MixProd. Algorithm 2 performs FP-edge correction with the statistics recovered from k -MixProd oracles. Theorem 1 summarizes the results proved in Appendix F.

Theorem 1. *Phase II requires $(2\Delta + \Delta^3)\lceil \lg(k+1) \rceil + 2(\Delta^2 + \Delta + 1)\lceil \lg(k) \rceil + 2\Delta^2 + 2\Delta^3$ vertices and solves k -MixProd $\mathcal{O}(k|E|2^{\Delta^2})$ times.*

As with Lemma 3, the Δ^2 in the runtime of Theorem 1 comes from the maximum size of a separating set and is therefore trivially bounded by $|V|$. A complete worked example walking through all phases on the graph of Figure 1 is given in Appendix A.

5 EMPIRICAL RESULTS

Our theoretical results guarantee success with infinite data. We now employ empirical tests for our hypothesis-based rank test and investigate the sensitivity of Phase I. Real-world datasets with a known ground truth are limited in this setting. As such, we utilize synthetic data on a broad class of functions and noise, the details of which are deferred to Appendix G. The synthetic approach is argued for in Poinso et al. [2025]. All experiments are run using code provided at this [GitHub link](#). Use `pip install probrank` to install the rank test.

5.1 TEST 1: RANK HYPOTHESIS TEST

This test compares the hypothesis test developed in Section D to a naive thresholding of singular values as in Anandkumar et al. [2012a]. We generate data from two graphs:

1. “Connected” $\mathcal{G}^c$: $V_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_4$
2. “Split” $\mathcal{G}^s$: $V_1 \rightarrow V_2 \quad V_3 \rightarrow V_4$

If we agglomerate our vertices into $S_i^+ = \{V_1, V_2\}$ and $S_j^+ = \{V_3, V_4\}$, then these two DAGs differ in that $S_i^+ \not\perp_d^{\mathcal{G}^c} S_j^+$ and $S_i^+ \perp_d^{\mathcal{G}^s} S_j^+$. Distinguishing between these two DAGs is notably difficult because S_i^+ and S_j^+ are only “loosely” connected by $V_2 \rightarrow V_3$.

We varied the number of samples from these distributions from 1000 to 9000 and studied the distributions of p-values and $k + 1$ th singular values across 200 runs. The results are reported in Figure 3, showing that the hypothesis test has a significant difference in p-values beyond 2000 samples (relative to the difference in the $k + 1$ th singular values). It is worth noting that it is almost impossible to choose a threshold for the $k + 1$ th singular value ahead of time, whereas hypothesis tests give a meaningful significance.

5.2 TEST 2: VARYING DENSITY

In our second test, we explore the role of density in accurately detecting graph adjacency. For this test, we sample random Erdős-Renyi undirected graph structures on 7 vertices and orient them according to a random permutation of the vertices which determines the topological ordering. We vary the edge-probability from .1 to .9 in .1 increments,

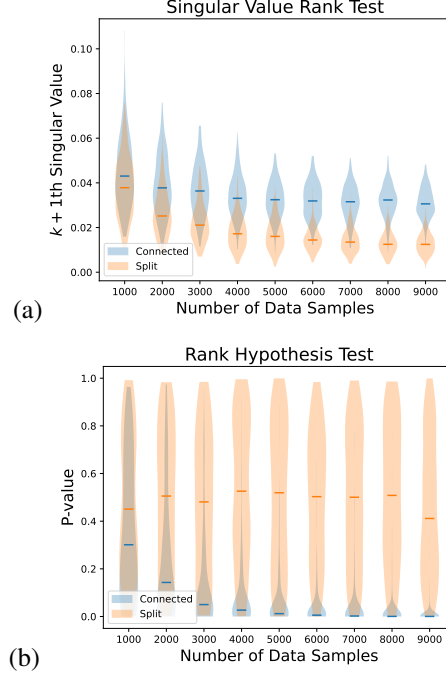


Figure 3: The results of Test 1. We compare (a) the singular values and (b) the p-values of our hypothesis test for the connected (blue) and split (orange) models.

sampling 20 graph structures for each. Among these graphs, we draw 10,000 datapoints and study the percentage of correctly recovered edges (significance threshold of .7 to account for many tests). Results are given in Figure 4 (a).

Strong performance is shown at low-density graphs, which represent the regime of theoretical study in this paper. As the graph density increases, we fail to preserve edges (lower blue marks) and incorrectly report edges where none exist (lower orange marks). At high densities, our accuracy for detecting true edges returns to higher levels, but at the cost of occasionally adding false positive edges. Figure 4 (b) compares overall adjacency accuracy against PC and FCI, using the implementation of Zheng et al. [2024], with the trivial “always-adjacent” baseline (dashed, equal to the realized graph density) shown for reference. Adjacency accuracy must be read relative to this baseline, since on denser ground-truth graphs a larger fraction of pairs are genuinely adjacent and the baseline itself rises with density. Neither PC nor FCI models the latent global confounder U , and empirically both return graphs of roughly constant, moderate density—on the order of 14 to 20 of the 21 possible edges—largely insensitive to the true sparsity. On sparse graphs this amounts to near-complete output (about 20 edges when fewer than 2 are real at density 0.1), so their accuracy only marginally exceeds the baseline; their improvement on denser graphs is driven mainly by the coincidence of a fixed dense output with an increasingly dense ground truth, rather than by recovery of sparse structure. Our method is

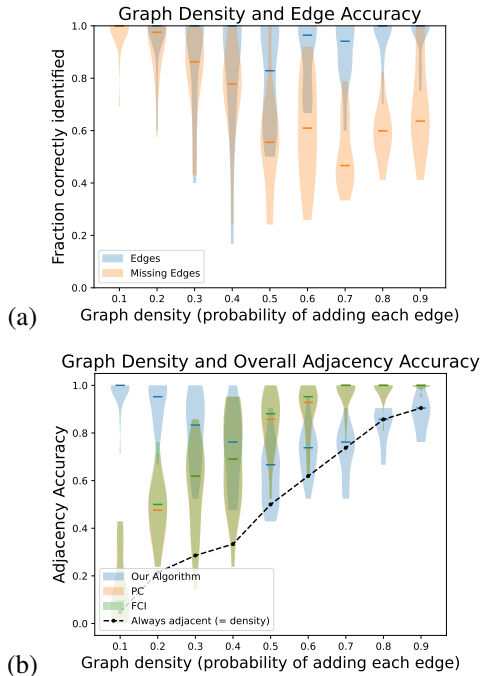


Figure 4: The results of Test 2. In (a), blue gives the percentage of correctly identified edges, and orange is the percentage of correctly removed edges. In (b), our method is compared to PC and FCI; the dashed line is the trivial “always-adjacent” baseline (equal to the graph density), shown for reference. PC and FCI return graphs of roughly constant moderate density regardless of the truth—near-complete on sparse graphs—so they exceed the baseline substantially only once the ground truth is itself dense. Our method is strongest in the sparse regime covered by our theory.

designed to remove exactly this confounder-induced dependence and attains the highest accuracy in the sparse regime, where PC and FCI are effectively at chance relative to the baseline. On denser graphs our accuracy is governed by a p -value threshold that trades removal of spurious edges against retention of true ones. Selecting this threshold is complicated by multiplicity: each candidate pair is subjected to many rank tests—over agglomerations, conditioning sets, and their assignments—so a fixed per-test level does not correspond to a fixed family-wise level. A principled choice, for instance, a correction that scales with the number of tests applied per pair, remains open.

6 DISCUSSION

This paper gives the first identifiability result and algorithm for causal DAG discovery under confounding by a mixture or latent class. It does so when the observables are Bernoulli, which contain only one parameter. This scarcity of parameters creates a unique identifiability problem, bringing about the need for “agglomeration.” In principle, our

approach can be adapted for easier settings, e.g., discrete observables with higher cardinality alphabets or continuous observables with higher moments, *without* the need to agglomerate (thereby making the problem significantly easier). Our result illustrates the relationship between the amount of observed information (given by the number of Bernoulli random variables and their graphical sparsity) and the amount of unobserved information (given by the number of latent classes k). Corollary 1 only supplies sufficient conditions, and we leave the question of necessity as an open question.

The majority of our approach relies on the assumption of a known k . This restriction is unavoidable in all mixture settings (e.g., k -means). In practice, such a k must be determined through trial and error. If k is too large, then the resulting DAG will be very sparse. If k is too small, then the resulting DAG will be very densely connected. Appendix H repeats our experiments with (incorrectly) larger and smaller k , verifying that this property is very sensitive to the choice of k . This means that a practitioner could, in principle, tune k until they receive results that are not close to cliques or empty graphs.

Fortunately, k is sometimes known approximately. k can also be estimated under certain circumstances, such as when some observables are *known* to be independent (conditioned on U). In this case, a rank test can be used to determine k —i.e., the rank of the probability matrix between two conditionally independent variables will be rank k under the previously discussed faithfulness assumptions.

We also assumed that the latent classes obeyed the *same* DAG structure. This assumption can be relaxed: so long as there is enough shared sparsity from Phase I, we can detect statistical dependencies within *at least one* graph structure. Phase II can then detect mild class-specific differences in graphical structure. This relaxation will, however, change the number of required vertices in nontrivial ways.

The results in this paper focus on discrete observables. Caution should be exercised when discretizing continuous results, as pointed out by Sun et al. [2025]. A more general approach could modify the probability matrix (Definition 1) to a matrix of moments, e.g. $\mathcal{M}[\mathbf{X}, \mathbf{Y}]_{ij} = E[X_i Y_j]$, utilizing kernel mean embedding [Muandet et al., 2017] to compute conditional expectations.

Finally, it is worth noting that the hypothesis test of rank developed in Appendix D may have uses for causal representation learning (CRL) [Squires et al., 2023] and latent variable models [Silva et al., 2006, Anandkumar et al., 2013, Xie et al., 2020, Huang et al., 2022]. Additionally, the notion of agglomeration may have implications for the abstraction of coarse-grained concepts within CRL problems.

Author Contributions

B. Mazaheri conceived the idea, implemented the experiments, and wrote the paper. S. Gordon, Y. Rabani, and L. Schulman assisted with writing, editing, and verifying the results.

Acknowledgements

B. M. is partially supported by the Advanced Research Concepts (ARC) COMPASS program, sponsored by the Defense Advanced Research Projects Agency (DARPA) under agreement number HR001-25-3-0212.

L. S. is partially supported by NSF grant CCF-2321079.

References

- Silvia Acid and Luis M De Campos. An algorithm for finding minimum d-separating sets in belief networks. In *Proceedings of the Twelfth international conference on Uncertainty in artificial intelligence*, pages 3–10, 1996.
- Raj Agrawal, Chandler Squires, Neha Prasad, and Caroline Uhler. The DeCAMFounder: nonlinear causal discovery in the presence of hidden variables. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, page qkad071, 07 2023. ISSN 1369-7412. doi: 10.1093/jrssb/qkad071. URL <https://doi.org/10.1093/jrssb/qkad071>.
- E. S. Allman, C. Matias, and J. A. Rhodes. Identifiability of parameters in latent structure models with many observed variables. *Ann. Statist.*, 37(6A):3099–3132, 2009. doi: 10.1214/09-AOS689.
- A. Anandkumar, V. Tan, and A. Willsky. High dimensional structure learning of ising models on sparse random graphs. 2010.
- A. Anandkumar, D. Hsu, F. Huang, and S. M. Kakade. Learning high-dimensional mixtures of graphical models. *arXiv preprint arXiv:1203.0697*, 2012a.
- A. Anandkumar, D. J. Hsu, and S. M. Kakade. A method of moments for mixture models and hidden Markov models. In *Proc. 25th Ann. Conf. on Learning Theory - COLT*, volume 23 of *JMLR Proceedings*, pages 33.1–33.34, 2012b. URL <http://proceedings.mlr.press/v23/anandkumar12/anandkumar12.pdf>.
- Animashree Anandkumar, Daniel Hsu, Adel Javanmard, and Sham Kakade. Learning linear bayesian networks with latent variables. In *International Conference on Machine Learning*, pages 249–257. PMLR, 2013.
- Animashree Anandkumar, Rong Ge, Daniel J Hsu, Sham M Kakade, Matus Telgarsky, et al. Tensor decompositions for learning latent variable models. *J. Mach. Learn. Res.*, 15(1):2773–2832, 2014.
- Steen A Andersson, David Madigan, and Michael D Perlman. A characterization of markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541, 1997.
- Sander Beckers and Joseph Y. Halpern. Abstracting Causal Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):2678–2685, July 2019. ISSN 2374-3468. doi: 10.1609/aaai.v33i01.33012678. URL <https://ojs.aaai.org/index.php/AAAI/article/view/4117>.
- Sander Beckers, Frederick Eberhardt, and Joseph Y. Halpern. Approximate Causal Abstractions. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, pages 606–615. PMLR, August 2020. URL <https://proceedings.mlr.press/v115/beckers20a.html>.
- Ruichu Cai, Zhiyi Huang, Wei Chen, Zhifeng Hao, and Kun Zhang. Causal discovery with latent confounders based on higher-order cumulants. In *International conference on machine learning*, pages 3380–3407. PMLR, 2023.
- K. Chaudhuri and S. Rao. Learning mixtures of product distributions using correlations and independence. In *Proc. 21st Ann. Conf. on Learning Theory - COLT*, pages 9–20. Omnipress, 2008. URL <http://colt2008.cs.helsinki.fi/papers/7-Chaudhuri.pdf>.
- S. Chen and A. Moitra. Beyond the low-degree algorithm: mixtures of subcubes and their applications. In *Proc. 51st Ann. ACM Symp. on Theory of Computing*, pages 869–880, 2019. doi: 10.1145/3313276.3316375.
- Zhengming Chen, Ruichu Cai, Feng Xie, Jie Qiao, Anpeng Wu, Zijian Li, Zhifeng Hao, and Kun Zhang. Learning Discrete Latent Variable Structures with Tensor Rank Conditions. *Advances in Neural Information Processing Systems*, 37:17398–17427, December 2024. doi: 10.5222/079017-0553. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/1f530eef1ae1d4d4f4e0f51437976395-Abs tract-Conference.html.
- M. Cryan, L. Goldberg, and P. Goldberg. Evolutionary trees can be learned in polynomial time in the two state general Markov model. *SIAM J. Comput.*, 31(2):375–397, 2001. doi: 10.1137/S0097539798342496.
- Jingqiu Ding, Tommaso d’Orsi, Chih-Hung Liu, David Steurer, and Stefan Tiegel. Fast algorithm for overcomplete order-3 tensor decomposition. In *Conference on Learning Theory*, pages 3741–3799. PMLR, 2022.

- Gal Elidan and Nir Friedman. Learning Hidden Variable Networks: The Information Bottleneck Approach. *Journal of Machine Learning Research*, 6(4):81–127, 2005. ISSN 1533-7928. URL <http://jmlr.org/papers/v6/elidan05a.html>.
- Gal Elidan, Noam Lotner, Nir Friedman, and Daphne Koller. Discovering Hidden Variables: A Structure-Based Approach. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000. URL https://proceedings.neurips.cc/paper_files/paper/2000/file/95e6834d0a3d99e9ea8811855ae9229d-Paper.pdf.
- J. Feldman, R. O’Donnell, and R. A. Servedio. Learning mixtures of product distributions over discrete domains. *SIAM J. Comput.*, 37(5):1536–1564, 2008. doi: 10.1137/060670705.
- Y. Freund and Y. Mansour. Estimating a mixture of two product distributions. In *Proc. 12th Ann. Conf. on Computational Learning Theory*, pages 53–62, July 1999. doi: 10.1145/307400.307412.
- Benjamin Frot, Preetam Nandy, and Marloes H Maathuis. Robust causal structure learning with some hidden variables. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 81(3):459–487, 2019.
- Spencer L Gordon, Bijan Mazaheri, Yuval Rabani, and Leonard J Schulman. Source identification for mixtures of product distributions. In *Proc. 34th Ann. Conf. on Learning Theory - COLT*, volume 134 of *Proc. Machine Learning Research*, pages 2193–2216. PMLR, 2021. URL <http://proceedings.mlr.press/v134/gordon21a.html>.
- Spencer L Gordon, Bijan Mazaheri, Yuval Rabani, and Leonard J Schulman. Causal inference despite limited global confounding via mixture models. In *2nd Conference on Causal Learning and Reasoning*, 2023.
- Spencer L Gordon, Erik Jahn, Bijan Mazaheri, Yuval Rabani, and Leonard J Schulman. Identification of mixtures of discrete product distributions in near-optimal sample and time complexity. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 2071–2091. PMLR, 2024.
- Biwei Huang, Charles Jia Han Low, Feng Xie, Clark Glymour, and Kun Zhang. Latent hierarchical causal structure discovery with rank constraints. *Advances in neural information processing systems*, 35:5549–5561, 2022.
- M. Kearns, Y. Mansour, D. Ron, R. Rubinfeld, R. Schapire, and L. Sellie. On the learnability of discrete distributions. In *Proc. 26th Ann. ACM Symp. on Theory of Computing*, pages 273–282, 1994. doi: 10.1145/195058.195155.
- Lingjing Kong, Guangyi Chen, Biwei Huang, Eric Xing, Yuejie Chi, and Kun Zhang. Learning Discrete Concepts in Latent Hierarchical Models. *Advances in Neural Information Processing Systems*, 37:36938–36975, December 2024. doi: 10.52202/079017-1165. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/411fa9d368b5485be4c6bb62615b365e-Abstract-Conference.html.
- Joseph B Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18(2):95–138, 1977.
- Steffen L Lauritzen, A Philip Dawid, Birgitte N Larsen, and H-G Leimer. Independence properties of directed markov fields. *Networks*, 20(5):491–505, 1990.
- Seunghyun Lee and Yuqi Gu. On Theoretical Identifiability of Discrete Latent Causal Graphical Models, January 2026. URL <http://arxiv.org/abs/2505.18410>. arXiv:2505.18410 [stat].
- C. Meek. Causal inference and causal explanation with background knowledge. In *Proc. Eleventh Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 403–411. Morgan Kaufmann, 1995.
- Ankur Moitra. An almost optimal algorithm for computing nonnegative rank. *SIAM Journal on Computing*, 45(1): 156–173, 2016.
- Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. Kernel Mean Embedding of Distributions: A Review and Beyond. *Foundations and Trends in Machine Learning*, 10(1-2):1–141, June 2017. ISSN 1935-8237. doi: 10.1561/22000000060. URL <https://doi.org/10.1561/22000000060>.
- Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.
- Judea Pearl. *Causality*. Cambridge university press, 2009.
- Audrey Poinot, Panayiotis Panayiotou, Alessandro Leite, Nicolas Chesneau, Özgür Şimşek, and Marc Schoenauer. Position: Causal Machine Learning Requires Rigorous Synthetic Experiments for Broader Adoption. June 2025. URL <https://openreview.net/forum?id=Nr2ulBN50q>.
- Yuval Rabani, Leonard J Schulman, and Chaitanya Swamy. Learning mixtures of arbitrary distributions over large discrete domains. In *Proc. 5th Conf. on Innovations in Theoretical Computer Science*, pages 207–224, 2014. doi: 10.1145/2554797.2554818.

- Zaka Ratsimalahelo. Rank test based on matrix perturbation theory. Technical report, EERI Research Paper Series, 2001.
- Thomas Richardson and Peter Spirtes. Ancestral graph markov models. *The Annals of Statistics*, 30(4):962–1030, 2002.
- Basil Saeed, Snigdha Panigrahi, and Caroline Uhler. Causal structure discovery from distributions arising from mixtures of dags. In *International Conference on Machine Learning*, pages 8336–8345. PMLR, 2020.
- Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.
- Ricardo Silva, Richard Scheines, Clark Glymour, Peter Spirtes, and David Maxwell Chickering. Learning the structure of linear latent variable models. *Journal of Machine Learning Research*, 7(2), 2006.
- Peter Spirtes. An anytime algorithm for causal inference. In *International Workshop on Artificial Intelligence and Statistics*, pages 278–285. PMLR, 2001.
- Peter Spirtes, Clark Glymour, and Richard Scheines. Discovery algorithms for causally sufficient structures. *Causation, prediction, and search*, pages 103–162, 1993.
- Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000.
- Chandler Squires and Caroline Uhler. Causal structure learning: A combinatorial perspective. *Foundations of Computational Mathematics*, 23(5):1781–1815, 2023.
- Chandler Squires, Anna Seigal, Salil S Bhate, and Caroline Uhler. Linear causal disentanglement via interventions. In *International Conference on Machine Learning*, pages 32540–32560. PMLR, 2023.
- Eric V Strobl. Causal discovery with a mixture of dags. *Machine Learning*, 112(11):4201–4225, 2023.
- Boyang Sun, Yu Yao, Guang Yuan Hao, Yumou Qiu, and Kun Zhang. A conditional independence test in the presence of discretization. In *13th International Conference on Learning Representations, ICLR 2025*, pages 39668–39700. International Conference on Learning Representations, ICLR, 2025.
- B. Tahmasebi, S. A. Motahari, and M. A. Maddah-Ali. On the identifiability of finite mixtures of finite product measures. (Also in “On the identifiability of parameters in the population stratification problem: A worst-case analysis,” Proc. ISIT’18 pp. 1051-1055), 2018. URL <https://arxiv.org/abs/1807.05444>.
- Burak Varici, Dmitriy Katz, Dennis Wei, Prasanna Sattigeri, and Ali Tajer. Separability analysis for causal discovery in mixture of dags. *Transactions on Machine Learning Research*, 2024.
- Feng Xie, Ruichu Cai, Biwei Huang, Clark Glymour, Zhifeng Hao, and Kun Zhang. Generalized independent noise condition for estimating latent variable causal graphs. *Advances in neural information processing systems*, 33:14891–14902, 2020.
- Yujia Zheng, Biwei Huang, Wei Chen, Joseph Ramsey, Mingming Gong, Ruichu Cai, Shohei Shimizu, Peter Spirtes, and Kun Zhang. Causal-learn: Causal discovery in python. *Journal of Machine Learning Research*, 25(60):1–8, 2024.
- Beata Zjawin, Elie Wolfe, and Robert Spekkens. Restricted Hidden Cardinality Constraints in Causal Models. *Electronic Proceedings in Theoretical Computer Science*, 343: 119–131, September 2021. doi: 10.4204/EPTCS.343.6.

Causal Discovery in Mixtures of Populations (Supplementary Material)

Bijan H. Mazaheri¹

Spencer Gordon²

Yuval Rabani³

Leonard Schulman⁴

¹Thayer School of Engineering, Dartmouth College, Hanover, New Hampshire, USA

²Fragment Data Technologies, San Francisco, California, USA

³Department of Computer Science, Hebrew University of Jerusalem, Jerusalem, Israel

⁴Computing and Mathematical Sciences, California Institute of Technology, Pasadena, California, USA

A WORKED EXAMPLE

We illustrate all three phases on the small graph $\mathcal{G}$ of Figure 1, with a binary mixture source ($k = 2$) and Bernoulli observables $V_1, \dots, V_6$. The observed edges of $\mathcal{G}$ are $V_4 \rightarrow V_1$, $V_1 \rightarrow V_2$, $V_2 \rightarrow V_6$, $V_4 \rightarrow V_5$, and $V_5 \rightarrow V_6$; the vertex V_3 has no observed neighbors. The mixture source U points to every V_i , so in the observed marginal *every* pair of variables is statistically dependent and no conditional independence holds. The maximum degree is $\Delta = 2$.

Why a naive approach fails. Because U confounds all pairs, a standard conditional-independence test never declares any pair independent, and the PC algorithm would return the complete graph. Our rank tests instead recover the structure by detecting, for suitable agglomerations, that the relevant probability matrix has nonnegative rank at most $k = 2$.

Phase I: agglomerated rank tests. Since the observables are binary and $k = 2$, a single pair of variables yields only a 2×2 probability matrix, which cannot exceed rank 2 and is therefore uninformative. We instead form supervariables of $\lceil \lg(k+1) \rceil = 2$ binary variables each, giving a 4×4 probability matrix.

Consider testing the non-adjacency of V_1 and V_5 . In $\mathcal{G}$ they are d-separated by $\{V_4\}$: the fork $V_1 \leftarrow V_4 \rightarrow V_5$ is blocked by conditioning on V_4 , and the path $V_1 \rightarrow V_2 \rightarrow V_6 \leftarrow V_5$ is blocked at the unconditioned collider V_6 . We agglomerate $\mathcal{S}_i^+ = \{V_1, V_2\}$ and $\mathcal{S}_j^+ = \{V_5, V_3\}$ with conditioning set $\mathcal{C} = \{V_4\}$. One checks that $\mathcal{S}_i^+ \perp\!\!\!\perp_d^{\mathcal{G}} \mathcal{S}_j^+ \mid \mathcal{C}$: every path between the two groups passes through the collider V_6 or through V_4 , both of which are blocked. Hence $(\mathcal{S}_i^+, \mathcal{S}_j^+)$ is an IPA. By Lemma 1, for any assignment $V_4 = v$,

$$\text{rk}_+(\mathcal{M}[\{V_1, V_2\}, \{V_5, V_3\} \mid V_4 = v]) \leq k = 2,$$

because conditioning on U as well makes the two groups independent, so the matrix decomposes into a sum of $k = 2$ rank-one terms (Equation 4). Detecting this low rank lets Phase I delete all four candidate *cross-group* edges at once— V_1 – V_5 , V_1 – V_3 , V_2 – V_5 , V_2 – V_3 —none of which is a true edge, while the within-group edge V_1 – V_2 is untouched. Repeating over agglomerations and conditioning sets removes every non-edge that admits an IPA, leaving the skeleton $\{V_4$ – V_1 , V_1 – V_2 , V_2 – V_6 , V_4 – V_5 , V_5 – $V_6\}$ with V_3 isolated.

A caveat the example makes concrete. Our worst-case guarantee (Corollary 1) does *not* apply at this size: with $\Delta = 2$, $k = 2$ it would require 76 vertices. Indeed, every vertex here is an “early vertex” (Definition 5), since each has fewer than $(2 + \Delta^2)\lceil \lg(k+1) \rceil - 2 = 10$ non-descendants, so Lemma 7 gives no useful restriction on FP edges. Nevertheless, generically the rank tests still recover the correct skeleton, consistent with the experiments of Section 5. This is precisely the gap between the (loose) worst-case bound and typical behavior.

Phase II: k -MixProd correction (when needed). Had Phase I left a false-positive edge between two early vertices V_i, V_j (e.g., because too many candidate separating sets lay among the collider descendants of V_i, V_j), we would verify it as follows. Form $\mathcal{T}_{ij}$ from V_i, V_j and their $\mathcal{G}_1$ -neighbors (Definition 6); find two further agglomerations $\mathcal{X}_1, \mathcal{X}_2$ at distance > 2 from $\mathcal{T}_{ij}$ and from each other; and condition on $\mathcal{Z}_{ij} = \text{NB}_2(\mathcal{X}_1) \cup \text{NB}_2(\mathcal{X}_2)$ so that $\mathcal{T}_{ij}, \mathcal{X}_1, \mathcal{X}_2$ become mutually

independent given U (Lemma 14). A k -MixProd oracle applied to these three groups recovers $\Pr(\mathcal{T}_{ij} \mid u, \mathbf{z}_{ij})$. Because $\mathcal{Z}_{ij}$ may include collider descendants of V_i, V_j , we deconfound using Lemma 17 to obtain the unconditioned $\Pr(\mathcal{T}_{ij} \mid u)$, then run an ordinary conditional-independence test *within* each class u . If $V_i \perp\!\!\!\perp V_j \mid \mathcal{C}, u$ for all u , the edge is removed. For the graph above no FP edge arises, so Phase II is vacuous here; the construction on a larger graph is shown in Figure 5.

Phase III: orientation. On the recovered skeleton, V_2 and V_5 are non-adjacent yet share the child V_6 , forming the unshielded collider $V_2 \rightarrow V_6 \leftarrow V_5$, which we orient. The remaining edges V_4-V_1 , V_4-V_5 , and V_1-V_2 are forced neither by an immorality nor by a Meek rule, and so remain undirected within the CP-DAG (the isolated V_3 contributes nothing). The output is therefore the MEC of $\mathcal{G}$.

B D-SEPARATION

We will rely on the concepts of **d-separation**, **active paths**, and **separating sets**. Active paths are defined relative to sets of variables to be conditioned on (“conditioning sets”).

A key concept in active paths is that of a collider, C , which takes the form $V_1 \rightarrow C \leftarrow V_2$ along an undirected path. Undirected paths through unconditioned variables with no colliders are considered active because they can “carry dependence” from one end of the path to the other. If a non-collider along one of these non-collider paths is conditioned on, then the path is inactive.

Collider steps of paths behave in the opposite fashion: paths with an unconditioned collider are inactive, whereas conditioning on a collider or any descendant of that collider “opens” the active path.

When two variables have an active path between them, we say that they are d-connected. If there are no active paths between the variables, we say they are d-separated. A separating set between two variables is a set of variables which, when conditioned on, break all active paths between those variables. While only loosely described here, a full precise definition of d-separation can be found in Pearl [1988] and Pearl [2009] (for a more extensive study).

Pearl [1988] uses structural causal models to justify the *local Markov condition*, which means that d-separation always implies independence and allows DAG structures to be factorized. It is possible that two d-connected variables by chance exhibit some unexpected *statistical* independence. This complication is often assumed away using “faithfulness” [Spirtes et al., 2000], which ensures that d-connectedness implies statistical dependence. Together, the local Markov condition and faithfulness give a correspondence statistical dependence and the graphical conditions of the causal DAG which can be leveraged for causal structure learning.

The following fact will be useful when arguing for the existence of separating sets.

Lemma 10 (Pearl [2009]). *If vertices V_i, V_j are nonadjacent in $\mathcal{G}$, either $\text{PA}^{\mathcal{G}}(V_i)$ or $\text{PA}^{\mathcal{G}}(V_j)$ are a valid separating set for V_i, V_j .*

We will often want to bound the cardinality of separating sets relative to the degree bound of the graph (Δ). When dealing with a separating set between two vertices V_i, V_j , Lemma 10 implies a simple upper bound of Δ . Separating sets for *sets* (or agglomerations) of vertices are significantly more complicated because conditioning may d-separate some pairs of vertices while d-connecting others.

To unify the treatment of separating sets, we will make use of **moral graphs**, which can be thought of as undirected equivalents of DAGs [Lauritzen et al., 1990]. We will denote the moral graph of $\mathcal{G}$ as $\mathcal{G}^{(m)}$. To transform $\mathcal{G}$ into $\mathcal{G}^{(m)}$, we add edges between all immoralities, i.e., nonadjacent vertices with a common child, sometimes called an unshielded collider. After this, we change all directed edges to undirected edges.

A very useful fact from Lauritzen et al. [1990] (also Eq. 1 in Acid and De Campos [1996]) is that all separating sets $C \subseteq V$ for $S, S' \subseteq V$ in $\mathcal{G}$ are also separating sets in the moralized subgraph on $\text{AN}^+(S \cup S' \cup C)$, denoted $(\mathcal{G}[\text{AN}^+(S \cup S' \cup C)])^{(m)}$. This transforms complicated active path analysis into simple connectedness arguments on undirected moral graphs (of special subgraphs of $\mathcal{G}$). A convenient consequence of this transformation, which we will use throughout the paper, is Lemma 11.

Lemma 11. *If $\mathcal{G} = (V, E)$ has degree bound Δ , then the size of the minimal separating set between $S, S' \subseteq V$ is no larger than $\min(|S|, |S'|)\Delta^2$.*

Lemma 11 is a consequence of the maximum increase in the degree of the moral graph.

Proof. A key observation is that the moral graph of a subgraph of $\mathcal{G}$ has no additional edges relative to $\mathcal{G}^{(m)}$. That is, if $\mathcal{G}[\mathbf{W}] = (\mathbf{W}, \mathbf{F})$ is a subgraph of $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ then the corresponding edge-sets of the moral graphs obey $\mathbf{F}^{(m)} \subseteq \mathbf{E}^{(m)}$ because adding vertices cannot have invalidated any previously contained immoralities.

Abbreviate $(\mathcal{G}[\mathbf{AN}^+(\mathbf{S} \cup \mathbf{S}' \cup \mathbf{C})])^{(m)}$ as $\mathcal{G}_C^{(m)}$. Even though we do not know what $\mathbf{C}$ is, we know that the 1-neighborhood of $\mathbf{S}$ in $\mathcal{G}_C^{(m)}$ suffices as a separating set. $\mathcal{G}^{(m)}$ has all of the edges of $\mathcal{G}_C^{(m)}$, so

$$\mathbf{NB}_1^{\mathcal{G}_C^{(m)}}(\mathbf{S}) \subseteq \mathbf{NB}_1^{\mathcal{G}^{(m)}}(\mathbf{S}). \quad (5)$$

Note that $\mathbf{NB}_1^{\mathcal{G}^{(m)}}(\mathbf{S})$ is not necessarily a separating set for $\mathbf{S}, \mathbf{S}'$ in $\mathcal{G}^{(m)}$, in fact it may include some vertices in $\mathbf{S}$ itself. However, the size of the separating set is bounded by $|\mathbf{NB}_1^{\mathcal{G}^{(m)}}(\mathbf{S})|$, which is no larger than $|\mathbf{S}|\Delta^2$. As we chose $\mathbf{S}$ arbitrarily, this bound also holds for $\mathbf{S}'$. $\square$

Another helpful result from moral graphs is Lemma 12, which will help us when we prove the existence of separating sets.

Lemma 12 (Corollary of Theorem 1 in [Acid and De Campos \[1996\]](#)). *For DAG $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, and $\mathbf{S}, \mathbf{S}' \subseteq \mathbf{V}$, separating sets $\mathbf{S}, \mathbf{S}'$ in $(\mathcal{G}[\mathbf{AN}^+(\mathbf{S} \cup \mathbf{S}')]^{(m)})$ are also separating sets in $\mathcal{G}$.*

C IDENTIFYING MIXTURES OF DISCRETE PRODUCTS

The main tool used in [Gordon et al. \[2023\]](#) is a solution to identifying discrete k -mixtures of product distributions (k -**MixProd**) - i.e. $\mathbf{X} = X_1, \dots, X_n$ and latent global confounder or “source” U such that $X_i \perp\!\!\!\perp X_j \mid U$ for all i, j . The key complexity parameter for identifiability is k , the cardinality of the support of U .¹

At its core, k -MixProd shows how coincidences of multiple independent events reveal information about their confoundedness. Of course, it is possible for U with sufficiently large k to completely control the distribution on $\mathbf{X}$. For example, for binary $X_i \in \{0, 1\}$, a cardinality of $k = 2^n$ would be sufficient to assign each binary sequence in $\mathbf{X}$ to a latent class in U . Such a powerful U could generate *any* desired probability distribution on $\mathbf{X}$ by simply controlling the probability distribution on U . Limiting k , however, limits the space of marginal probability distributions on $\mathbf{X}$, eventually giving rise to identifiability.

Under a cardinality bound k on the support of U , [Allman et al. \[2009\]](#) showed that $n \geq \Omega(\log(k))$ is sufficient for the generic identification of k -MixProd. In other words, other than a Lebesgue measure 0 set of exceptions, most instances of k -MixProd have a one-to-one correspondence with their observed statistics (the probability distribution on $\mathbf{X}$ marginalized over U) and generating model (up to a set of $k!$ models with permuted labels of U).

[Tahmasebi et al. \[2018\]](#) demonstrated that a linear lower bound ($n \geq 2k - 1$), in conjunction with a separation condition in the distributions of $X_i \mid U$, is sufficient to guarantee identifiability. The best-known algorithm for identification is given in [Gordon et al. \[2024\]](#), which nearly matches the known lower bounds for sample complexity. This paper will use the result from [Allman et al. \[2009\]](#), but our methods easily extend to stronger identifiability conditions with modifications in the sparsity requirements.

D HYPOTHESIS TEST FOR RANK

We build a test for the null-hypothesis that $\mathcal{A} := \mathcal{M}[V, V' \mid C]$ has rank $\leq k$. For this derivation, we use $V_i = V$ and $V_j = V'$ to free up i, j for indexing entries of matrices. At a high level, the test asks whether the smallest singular values of $\hat{\mathcal{A}}$ (those beyond the first k) are jointly indistinguishable from zero given sampling noise; if so, $\mathcal{A}$ has rank $\leq k$. In the notation that follows, $\mathcal{B}$ denotes the (rescaled) estimation error and $\varepsilon(N)$ its scale, $\mathcal{U}_2$ and $\mathcal{V}_2$ collect the left/right singular vectors associated with the “extra” singular values beyond rank k , $\mathcal{L}$ is the diagonal matrix of those extra singular values (vectorized as $\hat{l}$), and f is the degrees of freedom of the limiting χ^2 distribution. This section will closely follow [Ratsimalahelo \[2001\]](#)’s test of the rank of a matrix $\mathcal{A}$ under an asymptotically normal perturbation

$$\hat{\mathcal{A}} = \mathcal{A} + \varepsilon(N)\mathcal{B} \quad (6)$$

with $\varepsilon(N) \propto 1/\sqrt{N}$ for N samples.

¹We will refer to the cardinality of the support of discrete random variables as their cardinality.

To apply the test of rank to our setting, we break down $\hat{\mathcal{A}}$ into the sum of indicator functions which are applied to each of N data entries.

$$\mathcal{C}(v, v')_{i,j} := \frac{1}{N} \mathbb{1}[v = i] \mathbb{1}[v' = j], \quad (7)$$

$$\hat{\mathcal{A}}_{i,j} = \sum_{v,v' \in \text{Data}} \mathcal{C}(v, v') \quad (8)$$

From this decomposition, $\varepsilon(N)\mathcal{B} = \mathcal{A}_{i,j} - \hat{\mathcal{A}}_{i,j}$ approaches normal with variance proportional to $1/N$ for larger N . This means that the covariance of $\mathcal{B}$ can be approximated using the covariance matrix for the entries of $\mathcal{C}$, which are Bernoulli random variables with probability estimates given by the entries of $\hat{\mathcal{A}}$.

$$\text{Cov}(\mathcal{C}_{ij}, \mathcal{C}_{i'j'}) = \begin{cases} \mathcal{A}_{ij}(1 - \mathcal{A}_{ij}) & \text{if } i = i', j = j' \\ -\mathcal{A}_{ij}\mathcal{A}_{i'j'} & \text{otherwise.} \end{cases} \quad (9)$$

First, take the SVD of the empirical $m \times n$ matrix, $\hat{\mathcal{A}} = \mathcal{U}\mathcal{D}\mathcal{V}^\top$, with the singular values ordered decreasingly along the diagonal of $\mathcal{D}$. Let $\mathcal{U}_2$ (columns $k+1, \dots, m$ of $\mathcal{U}$) and $\mathcal{V}_2$ (columns $k+1, \dots, n$ of $\mathcal{V}$) collect the *trailing* left and right singular vectors—those associated with the “extra” singular values beyond the first k . Let $\mathcal{L} := \mathcal{U}_2^\top \hat{\mathcal{A}} \mathcal{V}_2$ be the (diagonal) matrix of those extra singular values, and vectorize it to $\hat{l}$ by stacking its columns.

[Ratsimalahelo \[2001\]](#)’s estimator uses the covariance matrix of the entries of $\hat{\mathcal{B}} - \mathcal{B}$, denoted Σ . For our setting, we substitute the covariance matrix for the entries of $\mathcal{C}$ whose empirical estimates are given by Equation 9.

$$\Sigma_{i+jm, i'+j'm} := \text{Cov}(\mathcal{C}_{ij}, \mathcal{C}_{i'j'}). \quad (10)$$

Now, with $(\cdot)^\dagger$ indicating the Moore-Penrose pseudoinverse and $\otimes$ indicating the Kronecker product, define the projected covariance

$$\hat{\mathcal{Q}} := (\mathcal{V}_2^\top \otimes \mathcal{U}_2^\top) \hat{\Sigma} (\mathcal{V}_2 \otimes \mathcal{U}_2). \quad (11)$$

Following [Ratsimalahelo \[2001\]](#), the statistic $N\hat{l}^\top \hat{\mathcal{Q}}^\dagger \hat{l}$ converges to χ_f^2 under N samples, where the degrees of freedom $f = (m-k)(n-k)$ count the tested “extra” singular directions (for the square $m \times m$ case, $f = (m-k)^2$). Equivalently, f is the rank of $\hat{\mathcal{Q}}$. Note that the pseudoinverse is taken *after* projecting the covariance onto the extra singular directions, not before. This gives us a hypothesis test that is specifically designed for the rank test from this section. This approach is tested and compared to thresholding the $k+1$ th singular value in Section 5.

E FURTHER RUNTIME DISCUSSION

The non-negative rank of a $k+1 \times k+1$ matrix (as used for our algorithm) can be solved in time $k^{\mathcal{O}(k^2)}$ [[Moitra, 2016](#)]. In the absence of non-negative rank tests, [Anandkumar et al. \[2012a\]](#) demonstrated that regular rank tests generally work well in place of non-negative rank tests in practice. We will develop a hypothesis test for matrix-rank that requires $\mathcal{O}(k^6)$ operations to invert a $(k+1)^2 \times (k+1)^2$ covariance matrix of the estimated elements of our $k+1 \times k+1$ matrix.

Our solution also requires solving k -MixProd on 3 variables of cardinality $\mathcal{O}(k)$, which corresponds to decomposing a $\mathcal{O}(k) \times \mathcal{O}(k) \times \mathcal{O}(k)$ tensor into rank 1 components. When the rank of such a tensor is known to be linear in k , this decomposition can be solved in $\mathcal{O}(k^{6.05})$ [[Ding et al., 2022](#)], though the problem generally suffers from instability, with a worst-case sample complexity that is exponential in k [[Rabani et al., 2014](#)]. This step may be considered optional, as it is used to refine a small number of “false-positive” adjacencies confined to a provably small subset of the DAG.

F UTILIZING $\mathcal{G}_1$ TO SET UP k -MIXPROD

The first step to recovering $\Pr(\mathcal{T}_{ij} \mid u)$ is to select some $\mathcal{Z}_{ij}$ and recover $\Pr(\mathcal{T}_{ij} \mid u, \mathcal{z}_{ij})$ using instances of k -MixProd induced on the conditional probability distribution $\Pr(\mathcal{V} \mid \mathcal{z}_{ij})$. Recall that k -MixProd requires three independent variables

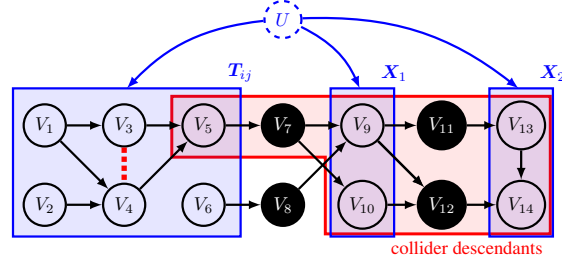


Figure 5: The given graph has an FP edge between V_3 and V_4 , indicated by a dashed line. This FP edge is induced by a large set of collider descendants, shown in red. Conditioning on $\mathbf{Z}_{ij} = \{V_7, V_8, V_{11}, V_{12}\}$ creates an instance of k -MixProd on $\mathbf{T}_{ij}, \mathbf{X}_1, \mathbf{X}_2$. Notice that $\mathbf{Z}_{ij}$ are all collider descendants of V_3, V_4 , which means that the conditional distribution, $\Pr(\mathbf{T}_{ij} \mid \mathbf{z}_{ij}, u)$, recovered by k -MixProd will not be sufficient for detecting the FP edge. This obstacle will be solved in Subsection F.2.

of sufficient cardinality. Hence, we must find $\mathbf{X}_1, \mathbf{X}_2, \mathbf{T}_{ij}$ which are sufficiently large, and d-separated from each other by $\mathbf{Z}_{ij}$ in $\mathcal{G}$. See Figure 5 for an example.

Of course, we have access to $\mathcal{G}_1$, not $\mathcal{G}$. $\mathcal{G}_1$ contains no orientations² and may contain extra false-positive edges. We will need to build a conditioning set $\mathbf{Z}_{ij}$ that guarantees an instance of k -MixProd. While Markov boundaries cannot be computed without $\mathcal{G}$, we can easily use $\mathcal{G}_1$ to find a superset that contains the Markov boundary of a given vertex.

Lemma 13. *The 2-neighborhood of $\mathbf{X} \subseteq \mathbf{V}$ in $\mathcal{G}_1$ contains $\text{MB}^{\mathcal{G}}(\mathbf{X})$.*

Proof. The distance between $X \in \mathbf{X}$, and $V \in \mathbf{V}$ in $\mathcal{G}_1$ is less than or equal to the distance in $\mathcal{G}$, because $\mathbf{E}_1 \supseteq \mathbf{E}$. This means that the 2-neighborhood of $\mathbf{X}$ in $\mathcal{G}_1$ includes the 2-neighborhood of $\mathbf{X}$ in $\mathcal{G}$. Furthermore because all vertices in $\text{MB}^{\mathcal{G}}(\mathbf{X})$ are distance ≤ 2 from at least one $X \in \mathbf{X}$, we have that $\text{MB}^{\mathcal{G}}(\mathbf{X})$ is contained in the 2-neighborhood of $\mathbf{X}$ in $\mathcal{G}_1$. $\square$

Recall that the definition of a Markov boundary of $\mathbf{X}$ is a set such that all supersets of the Markov boundary d-separate every other variable from $\mathbf{X}$. As such, three sets can be d-separated by conditioning on the Markov boundaries of just two of those sets. The following lemma makes this statement precise for 2-neighborhoods.

Lemma 14. *Suppose we have three disjoint sets $\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2$ such that for all $X_i \in \mathbf{X}_i, X_j \in \mathbf{X}_j$ ($i \neq j \in \{0, 1, 2\}$), X_i and X_j are distance > 2 . For an arbitrary (disjoint) choice of $\mathbf{Z}, \mathbf{Z}' \in \{\text{NB}_2(\mathbf{X}_0), \text{NB}_2(\mathbf{X}_1), \text{NB}_2(\mathbf{X}_2)\}$, then $\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2$ are mutually independent given $\mathbf{Z} \cup \mathbf{Z}'$.*

Proof. Because both sets are distance > 2 apart, the 2-neighborhoods do not overlap with either set. By Lemma 13, the conditioning set contains the Markov boundaries for both sets, so they are d-separated. $\square$

A worst-case analysis leads to the following minimum number of vertices for identifiability.

Lemma 15. *If $|\mathbf{V}| \geq (2\Delta + \Delta^3)\lceil \lg(k+1) \rceil + 2(\Delta^2 + \Delta + 1)\lceil \lg(k) \rceil + 2\Delta^2 + 2\Delta^3$, then there exists $\mathbf{X}_1, \mathbf{X}_2$ each with cardinality $\lceil \lg(k) \rceil$ such that $\mathbf{X}_1, \mathbf{X}_2, \mathbf{T}_{ij}$ are mutually independent given $\text{NB}_2(\mathbf{X}_1) \cup \text{NB}_2(\mathbf{X}_2)$.*

Proof. The proof will involve multiple applications of the fact that a degree bound Δ allows us to bound $|\text{NB}_1(\mathbf{X})| \leq \Delta|\mathbf{X}|$. We will also use $\text{NB}_2(\mathbf{X}) = \text{NB}_1(\text{NB}_1(\mathbf{X})) \cup \text{NB}_1(\mathbf{X})$.

The added complexity comes from the fact that our degree bound Δ applies to $\mathcal{G}$, not $\mathcal{G}_1$, which may have extra FP edges within the early vertices $\mathbf{H}$. The worst-case scenario involves an $\mathbf{H}$ that is made a clique in $\mathcal{G}_1$, with all possible FP edges present. As such, one can count the maximum number of vertices needed to find a $\mathbf{X}_1$ and $\mathbf{X}_2$ with the distance constraints needed for Lemma 14 (separation by distance 2).

²It is, in principle, possible to orient immoralities within $\mathcal{G}_1$ at this stage, but this gives no complexity improvements.

In order for X_1 and X_2 to be distance 2 from T_{ij} , $\text{NB}_2(T_{ij})$ must be disjoint from X_1 and X_2 . $\text{NB}_2(T_{ij})$'s cardinality can be bounded by first decomposing the 1-neighborhood into vertices in H and vertices not in H .

$$\text{NB}_1(T_{ij}) = (\text{NB}_1(T_{ij}) \cap H) \cup (\text{NB}_1(T_{ij}) \setminus H). \quad (12)$$

Notice that the first term is the full set H if H is a clique, since $T_{ij} \subseteq H$. The cardinality of the second term can be bounded using Δ , because it does not involve any potential for FP edges.

$$\text{NB}_1(T_{ij}) \leq |H| + \Delta|T_{ij}|. \quad (13)$$

When extending this neighborhood to the 2-neighborhood, we have already exhausted all of H , meaning that the degree bound of Δ still applies. The degree of the vertices in H is now $\Delta + 1$, meaning that those vertices have $(\Delta - 1)|H|$ distance 2 neighbors, plus no more than H distance 1 neighbors. Similarly, T_{ij} has no more than $\Delta^2 T_{ij}$ distance 2 neighbors and no more than ΔT_{ij} distance 1 neighbors. This gives,

$$\text{NB}_2(T_{ij}) + |T_{ij}| \leq (\Delta + 1 - 1)|H| + (\Delta^2 + \Delta + 1)|T_{ij}|. \quad (14)$$

Lemma 6 bounds the cardinality of H . The cardinality of T_{ij} is no more than two vertices for the V_i, V_j in question, and their distance one neighborhood, which includes connections to no more than $\Delta - 1$ additional vertices (see Definition 6), which means $|T_{ij}| \leq 2\Delta$. Combining these gives,

$$\text{NB}_2(T_{ij}) \leq \Delta((2 + \Delta^2)\lceil \lg(k + 1) \rceil - 2) + (\Delta^2 + \Delta + 1) \cdot 2\Delta = (2\Delta + \Delta^3)\lceil \lg(k + 1) \rceil + 2\Delta^2 + 2\Delta^3.$$

We need no more than an additional $(\Delta^2 + \Delta + 1)\lceil \lg(k) \rceil$ vertices for each $|X_1| = |X_2| = \lceil \lg(k) \rceil$ and their 2-neighborhoods. Summing all these together gives the total number of vertices needed. $\square$

Having now shown that two X_1, X_2 of cardinality $\lceil \lg(k) \rceil$ may be isolated, we note that these two variables can be agglomerated into super-variables S_1, S_2 with cardinality at least k each. Combined with T_{ij} , which may be agglomerated into a variable S_3 with at least cardinality 2 (trivially), all three S_1, S_2, S_3 are conditionally independent and meet the requirements of Lemma 8 for generic identification of $\Pr(T_{ij} \mid U)$. This proves the vertex requirement portion of Corollary 1 and Theorem 1.

F.1 ALIGNING MULTIPLE k -MIXPROD RUNS

k -MixProd distributions are symmetric with respect to the $k!$ permutations on the label of their source. For this reason, there is no guarantee that multiple calls to a k -MixProd solver will return the same permutation of source labels.

To solve this, Gordon et al. [2023] noticed that any two solutions to k -MixProd problems that share the same conditional probability distribution for at least one “alignment variable” can be “aligned” by permuting the source labels until the distributions on that variable match up. We will only need alignment along runs for different assignments to each Z_{ij} , used in the next section. Explicitly, two assignments z_{ij} and z'_{ij} , need least one $X^* \in \{T_{ij}, X_1, X_2\}$ such that $\text{mb}_{z_{ij}}(X^*)$ and $\text{mb}_{z'_{ij}}(X^*)$ are the same, in order for alignability to be satisfied.

To align sets of k -MixProd results which are not all pairwise alignable, Gordon et al. [2023] introduced the concept of an “alignable set of runs” for which chains of alignable pairs allow alignability.

Lemma 16. *The set of k -MixProd instances on the same T_{ij}, X_1, X_2 with all possible assignments z_{ij} to Z_{ij} is alignable.*

Proof. Any two runs with assignments z_{ij} and z'_{ij} that differ in their assignment to only one variable are alignable. Therefore, any two non-alignable runs can be aligned using a chain of Hamming-distance one alignments. $\square$

F.2 RECOVERING THE UNCONDITIONED WITHIN-SOURCE DISTRIBUTION

After all our calls to the k -MixProd oracle, we have access to $\Pr(T_{ij} \mid u, z_{ij})$ and $\Pr(u \mid z_{ij})$ for every assignment z_{ij} and u . $\Pr(T_{ij} \mid u, z_{ij})$ is insufficient to determine the adjacency of V_i, V_j because Z_{ij} may contain vertices in the collider descendants of V_i, V_j , prohibiting the discovery of a separating set within T_{ij} .

Instead, we must recover $\Pr(\mathbf{T}_{ij} \mid u)$, which is not conditioned on $\mathbf{Z}_{ij}$. To do this, we can apply the law of total probability over all possible assignments to $\mathbf{Z}_{ij}$.

$$\Pr(\mathbf{T}_{ij} \mid u) = \sum_{\mathbf{z}_{ij}} \Pr(\mathbf{z}_{ij} \mid u) \Pr(\mathbf{T}_{ij} \mid \mathbf{z}_{ij}, u) \quad (15)$$

We can obtain $\Pr(\mathbf{z}_{ij} \mid u)$ by using Bayes rule on the k -MixProd output, $\Pr(u \mid \mathbf{z}_{ij})$.

$$\Pr(\mathbf{z}_{ij} \mid u) = \frac{\Pr(u \mid \mathbf{z}_{ij}) \Pr(\mathbf{z}_{ij})}{\Pr(u)}. \quad (16)$$

$\Pr(\mathbf{z}_{ij})$ can be obtained by counting the frequency of $\mathbf{z}_{ij}$ in the data. In addition, $\Pr(u) = \sum_{\mathbf{z}_{ij}} \Pr(\mathbf{z}_{ij}) \Pr(u \mid \mathbf{z}_{ij})$ is computable by the law of total probability after the runs for each assignment $\mathbf{z}_{ij}$, have been aligned. Alternatively, we can observe that all $\Pr(\mathbf{z}_{ij} \mid u)$ have the same $\Pr(u)$ in their denominator, so we can scale these values so that $\sum_{\mathbf{z}_{ij}} \Pr(\mathbf{z}_{ij} \mid u) = 1$.

Lemma 17. *We can compute $\Pr(\mathbf{T}_{ij} \mid u)$ using known quantities,*

$$\Pr(\mathbf{T}_{ij} \mid u) = \frac{\sum_{\mathbf{z}_{ij}} \Pr(u \mid \mathbf{z}_{ij}) \Pr(\mathbf{z}_{ij}) \Pr(\mathbf{T}_{ij} \mid \mathbf{z}_{ij}, u)}{\sum_{\mathbf{z}_{ij}} \Pr(\mathbf{z}_{ij}) \Pr(u \mid \mathbf{z}_{ij})}.$$

$\Pr(\mathbf{T}_{ij} \mid u)$ is a completely deconfounded distribution on which we can run the PC-algorithm. The full procedure is given in Algorithm 2, in which we use breadth first search (BFS) to find $\mathbf{T}_{ij}$, $\mathbf{X}_1$, $\mathbf{X}_2$ such that they are separated by distance 2 in $\mathcal{G}'$, followed by alignment and Lemma 17 in order to remove all of the false-positive edges from $\mathcal{G}_1$.

Algorithm 2: Phase II: Detection and correction of FP edges.

Input: $\Pr(\mathbf{V})$ marginalized over U , a black box solver for k -MixProd, and $\mathcal{G}_1 = (\mathbf{V}, \mathbf{E}_1)$ from the output of Algorithm 1.

Output: $\mathcal{G}_2 = (\mathbf{V}, \mathbf{E}_2)$, an undirected skeleton of $\mathcal{G}$ and separating sets for nonadjacencies (vertices not in $\mathbf{E}_2$).

Start with $\mathbf{E}_2 \leftarrow \mathbf{E}_1$.

```

for each  $\{V_i, V_j\} \in \mathbf{E}_1$  do
  Retrieve  $\mathbf{T}_{ij}, \mathbf{X}_1, \mathbf{X}_2, \mathbf{Z}_{ij}$  using BFS.
  for each assignment  $\mathbf{z}_{ij}$  do
    | Run the  $k$ -MixProd solver on  $\mathbf{T}_{ij}, \mathbf{X}_1, \mathbf{X}_2$  on  $\Pr(\mathbf{V} \mid \mathbf{z}_{ij})$ .
  end
  Perform alignment of the  $2^{\mathbf{z}_{ij}}$  runs to retrieve  $\Pr(\mathbf{T}_{ij} \mid \mathbf{Z}_{ij}, U)$ .
  Calculate  $\Pr(\mathbf{T}_{ij} \mid u)$  for every  $u$  using Lemma 17.
  Run PC or any other structure learning algorithm on  $\Pr(\mathbf{T}_{ij} \mid u)$  to find a separating set  $\mathbf{C}_{ij}$  (or verify adjacency)
  for  $V_i, V_j$ . If  $V_i \perp\!\!\!\perp V_j \mid \mathbf{C}_{ij}, u$  for all  $u$ , remove  $\{V_i, V_j\}$  from  $\mathbf{E}_2$  and store  $\mathbf{C}_{ij}$ .
end

```

The runtime of Lemma 2 is dominated by the number of k -MixProd runs, since BFS is $O(|\mathbf{V}| + |\mathbf{E}_1|)$, and the number of runs of k -MixProd are also linearly proportional to $|\mathbf{E}_1|$ while also incurring the time costs of running k -MixProd.

Lemma 18. *Algorithm 2 requires solving k -MixProd $\mathcal{O}((\Delta^2 \lg(k)) + |\mathbf{E}|)k2^{\Delta^2}$ times.*

Proof. This algorithm requires running k -MixProd for every possible assignment to the conditioning set $\mathbf{z}_{ij}$, for which we have $|\mathbf{Z}_{ij}| \leq (\lg(k) + 2)\Delta^2$ total binary variables. This gives an upper bound of $2k2^{\Delta^2}$ runs of k -MixProd for each edge in $\mathcal{G}_1$, of which there are no more than $|\mathbf{E}| + |\mathbf{H}|$ which is $\mathcal{O}(\Delta^2 \lg(k)) + |\mathbf{E}|$. $\square$

G SYNTHETIC DATA DETAILS

G.1 STRUCTURAL EQUATION SETUP

SCMs are made up of a graphical structure and accompanying structural equations. We focus our tests primarily on varying the graphical structure, using a standard set of structural equations on these graphs. Our U are generated using a fair coin ($k = 2$), and all other vertices are Bernoulli random variables with bias p_V determined by V 's parents (including U):

$$p_V = \frac{1 + \sum_{W \in \mathbf{PA}^{G'}(V)} W}{|\mathbf{PA}^{G'}(V)| + 2}. \quad (17)$$

Structural equations of this form have a reasonable strength between vertices that is decreased relative to the in-degree. Since vertices can be permuted, this definition describes the broad class of distributions with sufficient separation.

G.2 NON-SYMMETRIC STRUCTURAL EQUATIONS

The structural equation used above is symmetric: it depends on the parents of V only through their sum, not through which parents are active. To probe the nonparametric generality of our method, we additionally consider a non-symmetric family in which each parent is assigned an independent random weight. Concretely, for each vertex V we draw weights $w_{VW} \sim \text{Unif}[,5, 1]$ for each $W \in \mathbf{PA}^{G'}(V)$ and a bias $b_V \sim \text{Unif}[-1, 1]$, and set

$$p_V = \sigma \left(b_V + \sum_{W \in \mathbf{PA}^{G'}(V)} w_{VW} (2W - 1) \right), \quad (18)$$

where σ is the logistic function. Unlike the symmetric form, this family depends on *which* parents are active and is non-additive, exercising asymmetric dependence.

Figure 6 repeats the density sweep of Test 2 (Figure 4(a)) under this non-symmetric SEM, reporting the fraction of true edges and true non-edges recovered as a function of graph density; the structure-learning algorithm is unchanged, since it makes no parametric assumptions on the observables.

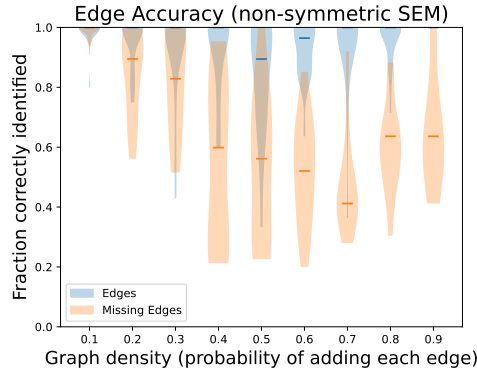


Figure 6: Edge-recovery accuracy under the non-symmetric weighted-logistic SEM, as a function of graph density (compare Figure 4(a) for the symmetric SEM).

H DEFERRED EXPERIMENTS

H.1 REPEATING TEST 2 WITH INCORRECT k

We repeated Test 2 with specifications of $k = 1$ (too small) and $k = 3$ (too large). The results are given in Figure 7. Overall, the method does not perform well – giving close to complete graphs when k is too small and close to empty graphs when k is too large. This makes tuning k rather easy, since a result of a very dense or very sparse graph suggests that a different k should be used.

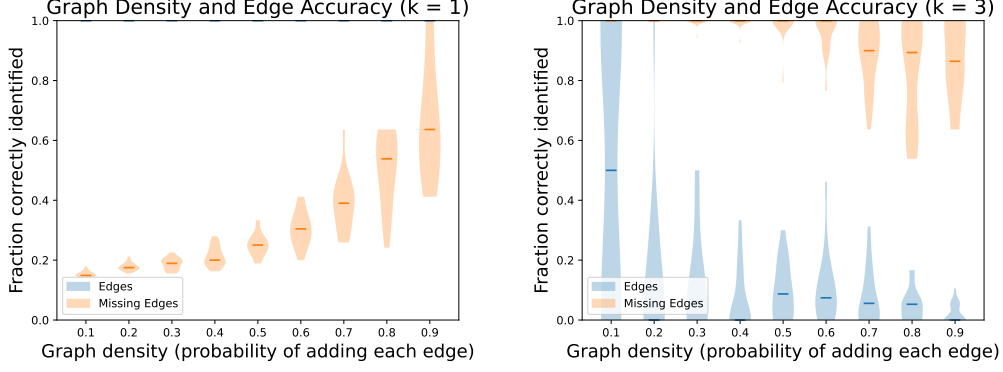


Figure 7: The results of Test 2 with incorrectly specified k .

I DEFERRED PROOFS

I.1 PROOF OF LEMMA 2

Proof. We will drop the conditioning on c in this proof for simplicity. Consider the sum

$$\sigma_j := \sum_{i=1}^j \Pr(u_i) \mathcal{M}[X, Y \mid u_i], \quad (19)$$

and note that $\sigma_k = \mathcal{M}[X, Y]$. Faithfulness with respect to $\mathcal{G}'$ tells us that there is some assignment, which we call u_1 , such that $X \not\perp\!\!\!\perp Y \mid u_1$. Hence $\text{rk}_+(\sigma_1) > 1$.

Now, we show inductively that $\text{rk}_+(\sigma_i) = \text{rk}_+(\sigma_{i-1}) + 1$ is a measure 1 event for $i = 1, \dots, k$. Denote $\mathcal{M}[X, Y \mid u_i] = v_i w_i^\top$ with column space v_i drawn from a subspace with non-zero measure on $\mathbb{R}^n$. The column space of σ_{i-1} is rank $\leq i-1 < m$, so it has measure zero on $\mathbb{R}^m$. Hence, v_i being in the column space of σ_{i-1} is a measure 0 event. We conclude that $\text{rk}_+(\sigma_{i-1} + \mathcal{M}^{u_i}[X, Y]) = \text{rk}_+(\sigma_{i-1}) + 1$ with measure 1. Inducting on i gives $\text{rk}_+(\sigma_k) > k$ with measure 1. $\square$

I.2 PROOF OF LEMMA 5

Proof. We can form S_i^+ out of V_i and $\lceil \lg(k+1) \rceil - 1$ arbitrary other vertices from A_{ij} . Now, consider subgraph on A_{ij} , denoted $\mathcal{G}_1[A_{ij}]$. Notice that d-separating other vertices in A_{ij} from S_i^+ does not require conditioning on any descendants of V_i or V_j . As such, we define a separating set on this subgraph,

$$C := \text{MB}^{\mathcal{G}[A_{ij}]}(S_i^+). \quad (20)$$

Since we know $|C| \leq \Delta^2 \lceil \lg(k+1) \rceil$, we need at least $\lceil \lg(k+1) \rceil - 1$ vertices in A_{ij} left to join with V_j and make S_j^+ . In summary, we have:

- $\lceil \lg(k+1) \rceil - 1$ vertices to augment S_i^+ .
- $\lceil \lg(k+1) \rceil - 1$ vertices to augment S_j^+ .
- $\Delta^2 \lceil \lg(k+1) \rceil$ vertices to d-separate S_i^+ and S_j^+ .

The sum of these three completes the proof, since S_i^+ was chosen arbitrarily and S_j^+ was formed with the leftover vertices. $\square$

I.3 PROOF OF LEMMA 6

Let $a = (2 + \Delta^2) \lceil \lg(k+1) \rceil - 2$ be equal to the number of non-descendants needed to define H (from Definition 5). Now, we need to show that there are no more than a vertices with fewer than a non-descendants. Index V according to a topological ordering and observe that vertices $V_1, \dots, V_a$ are non-descendants of vertices $V_{a+1}, \dots, V_{|V|}$. Hence $H \subseteq \{V_1, \dots, V_a\}$ meaning that the cardinality of X is no more than a .

I.4 PROOF OF LEMMA 7

Proof. A convenient consequence of Lemma 5 is that it guarantees the existence of IPAs everywhere except within a small subset of vertices. Let $\overline{\mathbf{DE}}(V) := V \setminus \mathbf{DE}(V)$ be the “non-descendants” of V . Note that $\mathbf{A}_{ij} = \overline{\mathbf{DE}}(V_i) \cup \overline{\mathbf{DE}}(V_j)$. This implies that

$$|\mathbf{A}_{ij}| \geq \max(|\overline{\mathbf{DE}}(V_i)|, |\overline{\mathbf{DE}}(V_j)|). \quad (21)$$

Hence, so long as at least one vertex has enough non-descendants, $\mathbf{A}_{ij}$ will be large enough to form an IPA. This set of vertices with enough non-descendants corresponds to the complement of the early vertices. $\square$

I.5 PROOF OF LEMMA 9

Proof. T_{ij} contains both $\mathbf{PA}(V_i)$ and $\mathbf{PA}(V_j)$, so Lemma 10 tells us that we contain a separating set. $\square$