TECHNICAL REPORT
R-28-L
November 2017

Experimental Design for Learning Causal Graphs with Latent Variables

Murat Kocaoglu[*]
Department of Electrical and Computer Engineering
The University of Texas at Austin, USA
mkocaoglu@utexas.edu

Karthikeyan Shanmugam[*]
IBM Research NY, USA
karthikeyan.shanmugam2@ibm.com

Elias Bareinboim
Department of Computer Science and Statistics
Purdue University, USA
eb@purdue.edu

Abstract

We consider the problem of learning causal structures with latent variables using interventions. Our objective is not only to learn the causal graph between the observed variables, but to locate unobserved variables that could confound the relationship between observables. Our approach is stage-wise: We first learn the observable graph, i.e., the induced graph between observable variables. Next we learn the existence and location of the latent variables given the observable graph. We propose an efficient randomized algorithm that can learn the observable graph using $\mathcal{O}(d \log^2 n)$ interventions where d is the degree of the graph. We further propose an efficient deterministic variant which uses $\mathcal{O}(\log n + l)$ interventions, where l is the longest directed path in the graph. Next, we propose an algorithm that uses only $\mathcal{O}(d^2 \log n)$ interventions that can learn the latents between both non-adjacent and adjacent variables. While a naive baseline approach would require $\mathcal{O}(n^2)$ interventions, our combined algorithm can learn the causal graph with latents using $\mathcal{O}(d \log^2 n + d^2 \log(n))$ interventions.

1 Introduction

Causality shapes how we view, understand, and react to the world around us. It is arguably a key ingredient in building intelligent systems that are autonomous and can act efficiently in complex environments. Not surprisingly, the task of automating the learning of cause-and-effect relationships have attracted great interest in the artificial intelligence and machine learning communities. This effort has led to a general theoretical and algorithmic understanding of the assumptions under which cause-and-effect relationships can be inferred from data. These results have started to percolate through the applied fields ranging from genetics to medicine, from psychology to economics [5, 26, 33, 25].

The endeavour of algorithmically learning causal relations may have started from the independent discovery of the IC [35] and PC algorithms [33], which almost identically, and contrary to previously held beliefs, showed the feasibility of recovering these relations from purely observational, non-experimental data. A plethora of methods followed this breakthrough, and now we understand, at least in principle, the limits of what can be inferred from purely observational data, including (not exhaustively) [31, 14, 21, 27, 19, 13]. There are a number of assumptions that have been considered about the data-generating model when attempting to unveil the causal structure. One of the most

[*]Equal contribution.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

popular assumptions is that the data-generating model is *causally sufficient*, which means that no latent (unmeasured) variable affects more than one observed variable. In practice, this is a very stringent condition since the existence of latents affecting more than one observed variable, and generating what is called *confounding bias*, is one of the main concerns of empirical scientists. The problem of causation is deemed challenging in most of the empirical fields because scientists recognize that not all the variables influencing the observed phenomenon can be measured. The general question that arises is then how much of the observed behavior of the system is truly causal, or whether it is due to some external, unobserved forces [26, 5].

To account for the latent variables in the context of structural learning, the IC* [35] and FCI [33] algorithms were introduced, which showed the possibility of recovering causal structures *even when* latent variables may be confounding the observed behavior [2]. One of the main challenges faced by these algorithms is that although some ancestral relations as well as certain causal edges can be learned [36, 7], many observationally equivalent architectures cannot be distinguished. Despite the practical challenges when collecting the data (e.g., finite samples, selection bias, missing data), we now have a complete characterization of what structures are recoverable from observational data based on conditional independence constraints [33, 2, 37]. Inferences will be constrained within an equivalence class. Initial works leveraged ideas of experimental design and the availability of interventional data to move from the equivalence class to a specific graph, but almost exclusively considering causally sufficient systems [9, 15, 11, 12, 30, 18].

For causally insufficient systems, there is a growing interest in identifying experimental quantities and structures based on partially observed interventional data [4, 32, 29, 28, 24, 16, 8, 34, 22], but without the goal of designing the optimal set of interventions. Perhaps the most relevant paper to our setup is [23]. Authors identify the experiments needed to learn the causal graph under latents, given the output of FCI algorithm. However, they are not interested in minimizing the number of experiments.

In this paper, we propose the first efficient non-parametric algorithm for learning a causal graph with latent variables. It is known that $\log(n)$ interventions are necessary (across all graphs) and sufficient to learn a causal graph without latent variables [12], and we show, perhaps surprisingly, that there exists an algorithm that can learn any causal graph with latent variables which requires $\mathrm{poly}(\log n)$ interventions when the observable graph is sparse. More specifically, our contributions are as follow:

- We introduce a deterministic [3] algorithm that can learn any causal graph and the existence and location of the latent variables using $\mathcal{O}(d\log(n)+l)$ interventions, where d is the largest node degree and l is the longest directed path of the causal graph.
- We design a randomized algorithm that can learn the observable graph and all the latent variables using $\mathcal{O}(d\log^2(n)+d^2\log(n))$ interventions with high probability, where d is the largest node degree.

The first algorithm is useful in practical settings where the longest directed path is not very deep, e.g., $\mathcal{O}(\log(n))$. This includes bipartite, time-series, and relational type of domains where the underlying causal topology is somewhat sparse. As an example application, consider the problem of inferring the causal effect of a set of genes on a set of phenotypes, that could be cast as learning a bipartite causal system. For the more general setting, we introduce a randomized algorithm that with high probability is capable of unveiling the true causal structure.

Background

We assume for simplicity that all the random variables are discrete. We use the language of Structural Causal Models (SCM) [26, pp. 204-207]. Formally, an SCM $\mathcal{M}$ is a 4-tuple $\langle \mathcal{U}, \mathcal{V}, \mathcal{F}, P(u)\rangle$, where $\mathcal{U}$ is a set of exogenous (unobserved, latent) variables, $\mathcal{V}$ is a set of endogenous (measured) variables. We partition the set of exogenous variables into two disjoint sets: Exogenous variables with one observable child, denoted by $\mathcal{E}$, exogenous variables with two observable children, denoted by $\mathcal{L}$. $\mathcal{F}=\{f_i\}$ is a collection of functions such that each endogenous variable $V_i\in\mathcal{V}$ is determined by a function $f_i\in F$: Each f_i is a mapping from the respective domain of the exogenous variables associated with V_i and a set of observable variables associated with V_i, called PA_i, into V_i. The

[2] Hereafter, *latent variable* refers to any unmeasured variable that affects more than one observed variable.

[3] We assume access to an oracle that outputs a size-$\mathcal{O}(d^2\log(n))$ independent set cover for the non-edges of a given graph. This oracle can be implemented using another randomized algorithm as we explain in Section 5.

set of exogenous variables associated with V_i can be divided into two classes, the one with a single observable child, denoted by $\mathcal{E}_i \in \mathcal{E}$, and those with two observable children, denoted by $\mathcal{L}_i \subseteq \mathcal{L}$. Hence f_i maps from the domain of $\mathcal{E}_i \cup PA_i \cup \mathcal{L}_i$ to V_i. The entire set $\mathcal{F}$ forms a mapping from $\mathcal{U}$ to $\mathcal{V}$. The uncertainty is encoded through a product probability distribution over the exogenous variables $P(\mathcal{E}, \mathcal{L})$. For simplicity we refer to $\mathcal{L}$ as the set of latents, and $\mathcal{E}$ as the set of exogenous variables.

Within the structural semantics, performing an action $S = s$ is represented through the do-operator, $do(S = s)$, which encodes the operation of replacing the original equation of S by the constant s and induces a submodel $\mathcal{M}_S$ (also for when S is not a singleton). We denote the post-interventional distribution by $P_S(\cdot)$. For a detailed discussion on the properties of structural models, we refer readers to [5, 23, 24, Ch. 7]. Define $D_\ell = (\mathcal{V} \cup \mathcal{L}, E_\ell)$ to be the causal graph with latents. We define the observable graph to be the induced subgraph on $\mathcal{V}$ which is $D = (\mathcal{V}, E)$.

In practice, we use an independent random variable W_i taking values uniformly at random in the state space of V_i, to implement an intervention $do(V_i)$. A conditional independence statement, e.g., X *is independent from* Y *given* $Z \subset \mathcal{V}$ *with respect to causal model* $\mathcal{M}_S$, in shown by $(X \perp\!\!\!\perp Y|Z)_{\mathcal{M}_S}$, or $(X \perp\!\!\!\perp Y|Z)_S$ when the causal model is clear from the context. These conditional independencies are with respect to the post-interventional joint probability distribution $P_S(\cdot)$. In this paper, we assume that an oracle to conditional independence (CI) tests is available.

The *mutilated* or *post-interventional causal graph*, denoted $D_\ell[S] = (\mathcal{V} \cup \mathcal{L}, E_\ell[S])$, is identical to D_ℓ except that all the incoming edges incident on any vertex in the interventional set S is absent, i.e., $E_\ell[S] = E_\ell - \{(Y, V) : V \in S,\ (Y, V) \in E_\ell\}$. We define the *transitive closure*, denoted D_{tc}, of an observable causal DAG D as follows: If there is a directed path from V_i to V_j in D, there is a directed edge from V_i to V_j in D_{tc}. Essentially, a directed edge in D_{tc} represents an ancestral relation in D.

For any DAG $D = (V, E)$, a set of nodes $S \subset V$ d-separates two nodes a and b if and only if S blocks all paths between a and b. 'Blocking' is a graphical criterion associated with d-separation [4]. A probability distribution is said to be faithful (or stable) to a graph, if and only if every conditional independence statement can be read off from the graph using d-separation, see [26, Ch. 2] for a review. We assume that faithfulness holds in the observational and post-interventional distributions following [12].

Results and outline of the paper

The skeleton of the proposed learning algorithms can be split into 3 steps, namely:

$$\emptyset \xrightarrow{\text{(a)}} \text{Transitive Closure} \xrightarrow{\text{(b)}} \text{Observable graph} \xrightarrow{\text{(c)}} \text{Observable graph with Latent variables}$$

Each step requires different tools and graph theoretic concepts:

(a) We use a pairwise independence test under interventions that reveals the ancestral relations. This is combined in an efficient manner with separating systems to discover the transitive closure of D in $\mathcal{O}(\log n)$ interventions.

(b) We rely on the transitive reduction of directed acyclic graphs that can be efficiently computed only from their transitive closure. A key property we observe is that the *transitive reduction reveals a subset of the true edges*. For our randomized algorithm, we use a sequence of transitive reductions computed from transitive closures (obtained using step (a)) of different post-interventional graphs.

(c) Given the observable graph, it is possible to discover latents between non-adjacent nodes using CI tests under suitable interventions. We use an edge-clique cover on the complement graph to optimize the number of experiments. For latents between adjacent nodes, we use a relatively unknown test called the do-see test, i.e., leveraging the equivalence between observing and intervening on the node. We implement it using induced matching cover of the observable graph.

The modularity of our approach allows us to solve subproblems: given the ancestral graph, we can use (b) to discover the observable graph D. If D is known, we can learn the latents with (c). Some pictorial illustrations of the main results in the technical sections are found in the full version [20].

2 Identifying the Observable Graph: A simple baseline

We discuss a natural and a simple deterministic baseline algorithm that finds the observable graph with experiments when confounders are present. To our knowledge, a *provably complete* algorithm

[4] For convenience, detailed definitions of blocking and non-blocking paths are provided in the full version [20].

that recovers the observable graph under this setting and is superior than this simple baseline in the worst case is not known. We start from the following observation. Suppose $X \to Y$ where X, Y are observable variables and let L be a latent variable such that $L \to X,\ L \to Y$. Consider the post interventional graph $D_\ell[\{X\}]$ where we intervene on X. It is easy to see that, X and Y are dependent in the post interventional graph too because of the direct causal relationship. However, if X is not a parent of Y, then in the post interventional graph $D_\ell[\{X\}]$ even with or without the latent L between X and Y, X is independent of Y since X is intervened on.

It is possible to recreate this condition between any target variable Y and any one of its direct parents X when many other observable variables are involved. Simply, we consider the post-interventional graph where we intervene on all observable variables but Y. In $D_\ell[V - \{Y\}]$, Y and X are dependent if and only if $X \to Y$ is a directed edge in the observable graph D, because every variable except X becomes independent of all other variables in the post interventional graph. Therefore, one needs n interventions, each of size $n-1$ to find out the parent set of every node. We basically show in the next two sections that when the graph D has constant degree, it is enough to do $O(\log^2(n))$ interventions representing the first provably exponential improvement.

3 Learning Ancestral Relations

In this section, we show that *separating systems* can be used to construct sequences of pairwise CI tests to discover the transitive closure of the observable causal graph, i.e., the graph that captures all ancestral relations. The following lemma relates post-interventional statistical dependencies with the ancestral relations in the graph with latents.

Lemma 1. *[Pairwise Conditional Independence Test] Consider a causal graph with latents D_ℓ. Consider an intervention on the set $S \subset \mathcal{V}$ of observable variables. Then, under the post-interventional faithfulness assumption, for any pair $X_i \in S, X_j \in \mathcal{V} \backslash S$, $(X_i \not\perp\!\!\!\perp X_j)_{D_\ell[S]}$ if and only if X_i is an ancestor of X_j in the post-interventional observable graph $D[S]$.*

Lemma 1 constitutes, for any ordered pair of variables (X_i, X_j) in the observable graph D, a test for whether X_i is an ancestor of X_j or not. Note that a single test is not sufficient to discover the ancestral relation between a pair (X_i, X_j), e.g., if $X_i \to X_k \to X_j$ and $X_i, X_k \in S, X_j \notin S$, the ancestral relation will not be discovered. This issue can be resolved by using a sequence of interventions guided by a separating system, and later finding the transitive closure of the learned graph.

Separating systems were first defined by [17], and has been subsequently used in the context of experimental design [10]. A separating system on a ground set S is a collection of subsets of S, $\mathcal{S} = \{S_1, S_2 \ldots\}$ such that for every pair (i, j), there is a set that contains only one, i.e., $\exists k$ such that $i \in S_k, j \notin S_k$ or $j \in S_k, i \notin S_k$. We require a stronger notion which is captured by a strongly separating system.

Definition 1. *An (m, n) strongly separating system is a family of subsets $\{S_1, S_2 \ldots S_m\}$ of the ground set $[n]$ such that for any two pairs of nodes i and j, there is a set S in the family such that $i \in S,\ j \notin S$ and also another set S' such that $i \notin S',\ j \in S'$.*

Similar to separating systems, one can construct strongly separating systems using $\mathcal{O}(\log(n))$ subsets:

Lemma 2. *An (m, n) strong separating system exists on a ground set $[n]$ where $m \leq 2\lceil \log n \rceil$.*

We propose Algorithm 1 to discover the ancestral relations between the observable variables. It uses the subsets of a strongly separating system on the ground set of all observable variables as intervention sets, to assure that the ancestral relation between every ordered pair of observable variables is tested. The following theorem shows the number of experiments and the soundness of Algorithm 1.

Theorem 1. *Algorithm 1 requires only $2\lceil \log n \rceil$ interventions and conditional independence tests on samples obtained from each post-interventional distribution and outputs the transitive closure D_{tc}.*

4 Learning the Observable Graph

We introduce a deterministic and a randomized algorithm for learning the observable causal graph D from ancestral relations. D encodes every direct causal connection between the observable nodes.

Algorithm 1 LearnAncestralRelations- Given access to a conditional independence testing oracle (CI oracle), query access to samples from any post-interventional causal model derived out of $\mathcal{M}$ (with causal graph D_ℓ), outputs all ancestral relationships between observable variables, i.e., D_{tc}

function LEARNANCESTRALRELATIONS($\mathcal{M}$)
  $E = \emptyset$.
  Consider a strongly sep. system of size $\leq 2 \log n$ on the ground set $\mathcal{V}$ - $\{S_1, S_2..S_{2\lceil \log n \rceil}\}$.
  for i in $[1 : 2\lceil \log n \rceil]$ **do**
    Intervene on the set S_i of nodes.
    for $X \in S_i,\ Y \notin S_i,\ Y \in \mathcal{V}$ **do**
      Use samples from $\mathcal{M}_{S_i}$ and use the CI-oracle to test the following.
      if $(X \not\perp\!\!\!\perp Y)_{D_\ell[S]}$ **then**
        $E \leftarrow E \cup (X, Y)$.
      end if
    end for
  end for
  return The transitive closure of the graph $(\mathcal{V}, E)$
end function

4.1 A Deterministic Algorithm

Based on Section 3, assume that we are given the transitive closure of the observable graph. We show in Lemma 3 that, when the intervention set contains all parents of X_i, the only variables dependent with X_i in the post-interventional observable graph are the parents of X_i in the observable graph.

Lemma 3. *For variable X_i, consider an intervention on S where $Pa_i \subset S$. Then $\{X_j \in S : (X_i \not\perp\!\!\!\perp X_j)_{D[S]}\} = Pa_i$.*

Let the longest directed path of D_{tc} be r. Consider the partial order $<_{D_{\text{tc}}}$ implied by D_{tc} on the vertex set $\mathcal{V}$. Define $\{T_i : i \in [r+1]\}$ as the unique partitioning of vertices of D_{tc} where $T_i <_{D_{\text{tc}}} T_j, \forall i < j$ and each node in T_i is a set of mutually incomparable elements. In other words, T_i are the set of nodes at layer i of the transitive closure graph D_{tc}. Define $\mathcal{T}_i = \cup_{k=1}^{i-1} T_k$. We have the following observation: $Pa_i \subset \mathcal{T}_i$. This paves the way for Algorithm 2 that leverages Lemma 3.

Algorithm 2 LearnObservableGraph/Deterministic - Given the ancestral graph, access to a conditional independence testing oracle (CI oracle) and outputs the graph induced on observable nodes.

function LEARNOBSERVABLEGRAPH/DETERIMINISTIC($\mathcal{M}$)
  $E = \emptyset$.
  for i in $\{r+1, r, r-1, \ldots, 2\}$ **do**
    Intervene on the set $\mathcal{T}_i$ of nodes.
    Use samples from $\mathcal{M}_{\mathcal{T}_i}$ and use the CI-oracle to test the following.
    for X in T_i **do**
      if $(X \not\perp\!\!\!\perp Y)_{D_\ell[\mathcal{T}_i]}$ **then**
        $E \leftarrow E \cup (X, Y)$.
      end if
    end for
  end for
  return Observable graph
end function

The correctness of Algorithm 2 follows from Lemma 3, which is stated explicitly in the sequel.

Theorem 2. *Let r be the length of the longest directed path in the causal graph D_ℓ. Algorithm 2 requires only r interventions and conditional independence tests on samples obtained from each one of the post-interventional distributions and outputs the observable graph D.*

4.2 A Randomized Algorithm

We propose a randomized algorithm that repeatedly uses the ancestor graph learning algorithm from Section 3 to learn the observable graph [5]. A key structure that we use is the transitive reduction:

[5]Note that this algorithm does not require learning the ancestral graph first.

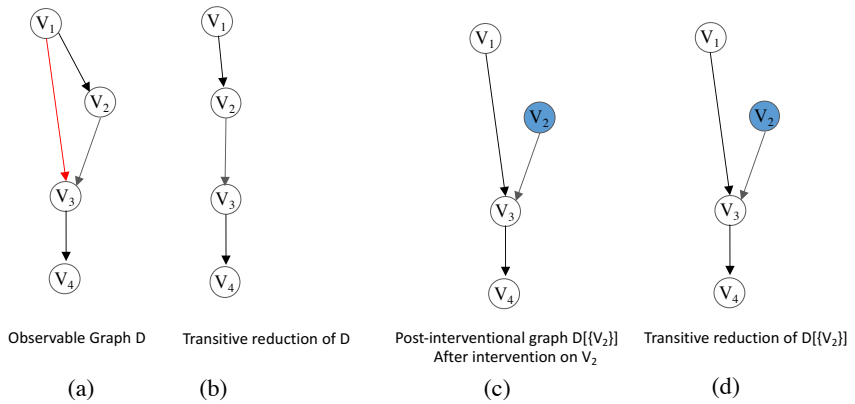


Figure 1: Illustration of Lemma 5 - (a) An example of an observable graph D without latents (b): Transitive reduction of D. The highlighted red edge (V_1, V_3) has not been revealed under the operation of transitive reduction. c) Intervention on node V_2 and its post interventional graph $D[\{V_2\}]$ d) Since all parents of V_3 above V_1 in the partial order have been intervened on, by Lemma 5, the edge (V_1, V_3) is revealed in the transitive reduction of $D[\{V_2\}]$.

Definition 2 (Transitive Reduction)**.** *Given a directed acyclic graph $D = (V, E)$, let its transitive closure be D_{tc}. Then $\text{Tr}(D) = (V, E_r)$ is a directed acyclic graph with minimum number of edges such that its transitive closure is identical to D_{tc}.*

Lemma 4. *[1] $\text{Tr}(D)$ is known to be unique if D is acyclic. Further, the set of directed edges of $\text{Tr}(D)$ is a subset of the directed edges of D, i.e., $E_r \subset E$. Computing $\text{Tr}(D)$ from D takes the same time as transitive closure of a DAG D, which takes time $\text{poly}(n)$.*

We note that $\text{Tr}(D) = \text{Tr}(D_{\text{tc}})$. Now, we provide an algorithm that outputs an observable graph based on samples from the post-interventional distribution after a sequence of interventions. Let us assume an ordering π on the observable vertices $\mathcal{V}$ that satisfies the partial order relationships in the observable causal graph D. The key insight behind the algorithm is given by the following Lemma.

Lemma 5. *Consider an intervention on a set $S \subset \mathcal{V}$ of nodes in the observable causal graph D. Consider the post-interventional observable causal graph $D[S]$. Suppose for a specific observable node V_i, $V_i \in S^c$. Let Y be a direct parent of V_i in D such that all the direct parents of V_i above Y in the partial order[6] $\pi(\cdot)$ is in S, i.e., $\{X : \pi(X) > \pi(Y),\ (X, V) \in D\} \subseteq S$. Then, $\text{Tr}(D[S])$ will contain the directed edge (Y, V_i) and it can be computed from $\text{Tr}((D[S])_{\text{tc}})$*

We illustrated Lemma 5 through an example in Figure 1. The red edge in Figure 1(a) is not revealed in the transitive reduction. The edge is revealed when computing the transitive reduction of the post-interventional graph $D[\{V_2\}]$. This is possible because all parents of V_3 above V_1 in the partial order (in this case node V_2) have been intervened on.

Lemma 5 motivates Algorithm 3. The basic idea is to intervene in randomly, then compute the transitive closure of the post-interventional graph using the algorithm in the previous section, compute the transitive reduction, and then accumulate all the edges found in the transitive reduction at every stage. We will show in Theorem 3 that with high probability, the observable graph can be recovered.

Theorem 3. *Let $d_{\max}$ be greater than the maximum in-degree in the observable graph D. Algorithm 3 requires at most $8cd_{\max}(\log n)^2$ interventions and CI tests on samples obtained from post-interventional distributions, and outputs the observable graph with probability at least $1 - \frac{1}{n^{c-2}}$.*

Remark. The above algorithm takes as input a parameter $d_{\max}$ that needs to be estimated. One practical option is to gradually increase $d_{\max}$ and run Algorithm 3.

[6]The nodes above with respect to the partial order of a graph are those that are closer to the source nodes.

Algorithm 3 LearnObservable- Given access to a conditional independence testing oracle (CI oracle), a parameter $d_{\max}$ outputs induced subgraph between observable variables, i.e. D

function LEARNOBSERVABLE/RANDOMIZED($\mathcal{M}, d_{\max}$)
  $E = \emptyset$.
  for i in $[1 : c * 4 * \mathrm{d}_{max} \log n]$ **do**
    $S = \emptyset$.
    for $V \in \mathcal{V}$ **do**
      $S \leftarrow S \cup V$ randomly with probability $1 - 1/d_{\max}$.
    end for
    $\hat{D}_S = \mathrm{LearnAncestralRelations}(\mathcal{M})$. Let $\hat{D} = (\mathcal{V}, \hat{E})$.
    Compute the transitive reduction of $\hat{D}(\mathrm{Tr}(\hat{D}_S))$ according to the algorithm in [1].
    Add the edges of the transitive reduction to the set E if not already there, i.e. $E \leftarrow E \cup \hat{E}$.
  end for
  return The directed graph $(\mathcal{V}, E)$.
end function

5 Learning Latents from the Observable Graph

The final stage of our framework is learning the existence and location of latent variables given the observable graph. We divide this problem into two steps – first, we devise an algorithm that can learn the latent variables between any two variables that are non-adjacent in the observable graph; later, we design an algorithm that learns the latent variables between every pair of adjacent variables.

5.1 Baseline Algorithm for Detecting Latents between Non-edges

Consider two variables X and Y such that $X \leftarrow L \rightarrow Y$ and where L is a latent variable. Clearly, to distinguish it from the case where X and Y are disconnected and have no latents, one needs check if $X \not\perp\!\!\!\perp Y$ or not. This is a conditional independence test. For any non edge (X, Y) in the observable graph D, when the observable graph D is known, to check for latents between them, when other variables and possible confounders are around, one has to simply intervene on the rest of the $n - 2$ variables and do a independence test between X and Y in the post interventional graph. This requires a distinct intervention for every pair of variables. If the observable graph has maximum degree $d = o(n)$, this requires $\Theta(n^2)$ interventions. We will reduce this to $O(d^2 \log n)$ interventions which is an exponential improvement for constant degree graphs.

5.2 Latents between Non-adjacent Nodes

We start by noting the following fact about causal systems with latent variables:

Theorem 4. *Consider two non-adjacent nodes X_i, X_j. Let S be the union of the parents of X_i, X_j, $S = Pa_i \cup Pa_j$. Consider an intervention on S. Then we have $(X_i \not\perp\!\!\!\perp X_j)_{\mathcal{M}_S}$ if and only if there exists a latent variable $L_{i,j}$ such that $X_j \leftarrow L_{i,j} \rightarrow X_i$. The statement holds under an intervention S such that $Pa_i \cup Pa_j \subset S,\ X_i, X_j \notin S$.*

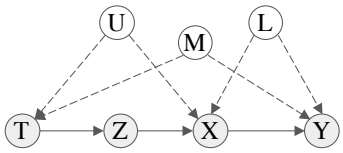
The above theorem motivates the following approach: For a set of nodes which forms an independent set, an intervention on the union of parents of the nodes of the independent set allows us to learn the latents between any two nodes in the independent set. We leverage this observation using the following lemma on the number of such independent sets needed to cover all non-edges.

Lemma 6. *Consider a directed acyclic graph $D = (V, E)$ with degree (out-degree+in-degree) d. Then there exists a randomized algorithm that returns a family of $m = \mathcal{O}(4e^2(d+1)^2 \log(n))$ independent sets $\mathcal{I} = \{I_1, I_2, \ldots, I_m\}$ that cover all non-edges of D: $\forall i, j$ such that $(X_i, X_j) \notin E$ and $(X_j, X_i) \notin E$, $\exists k \in [m]$ such that $X_i \in I_k$ and $X_j \in I_k$, with probability at least $1 - \frac{1}{n^2}$.*

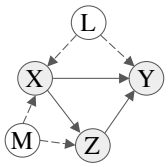
Note that this is a randomized construction and we are not aware of any deterministic construction. Our deterministic causal learning algorithm requires oracle access to such a famiy of independent sets, whereas our randomized algorithm can directly use this randomized construction. Now, we use this observation to construct a procedure to identify latents between non-edges (see Algorithm 4). The following theorem about its performance follows from Lemma 6 and Theorem 4.

Algorithm 4 LearnLatentNonEdge- Given access to a CI oracle, observable graph D with max degree d (in-degree+out-degree), outputs all latents between non-edges

function LEARNLATENTNONEDGE($\mathcal{M}, d_{\max}$)
  $L = \emptyset$.
  Apply the randomized algorithm in Lemma 6 to find a family of independent sets $\mathcal{I} = \{I_1, I_2, \ldots, I_m\}$ that cover all non-edges in D such that $m \leq 4e^2(d+1)^2 \log(n)$.
  for $j \in [1:m]$ **do**
    Intervene on the parent set of the nodes in I_j.
    for every pair of nodes X, Y in I_j **do**
      if $(X \not\perp\!\!\!\perp Y)_{D_\ell[I_j]}$ **then**
        $L \leftarrow L \cup \{X, Y\}$.
      end if
    end for
  end for
  return The set of non-edges L.
end function



G1: do(PaX) is needed



G2: do(PaY) is needed

Figure 2: Left: A graph where intervention on the parents of X is needed for do-see test to succeed. Right: A graph where intervention on the parents of Y is needed for do-see test to succeed.

Theorem 5. *Algorithm 4 outputs a list of non-edges L that have latent variables between them, given the observable graph D, with probability at least* $1 - \frac{1}{n^2}$. *The algorithm requires* $4e^2(d+1)^2 \log(n)$ *interventions where* d *is the max-degree (in-degree+out-degree) of the observable graph.*

5.3 Latents between Adjacent Nodes

We construct an algorithm that can learn latent variables between the variables adjacent in the observable graph. Note that the approach of CIT testing in the post-interventional graph is not helpful. Consider the variables $X \to Y$. To see the effect of the latent path, one needs to cut the direct edge from X to Y. This requires intervening on Y. However, such an intervention disconnects Y from its latent parent. Thus we resort to a different approach compared to the previous stages and exploit a different characterization of causal Bayesian networks called a 'do-see' test.

A do-see test can be described as follows: Consider again a graph where $X \to Y$. If there are no latents, we have $\mathbb{P}(Y|X) = \mathbb{P}(Y|\text{do}(X))$. Assume that there is a latent variable Z which causes both X and Y, then excepting the pathological cases[7], $\mathbb{P}(Y|X) \neq \mathbb{P}(Y|\text{do}(X))$.

Figure 2 illustrates the challenges associated with a do-see test in bigger graphs with latents. Graphs $G1$ and $G2$ are examples where parents of both nodes involved in the test need to be included in the intervention set for the Do-see test to work. In $G1$, suppose we condition on X, as required by the 'see' test. This opens up a non-blocking path $X - U - T - M - Y$. Since $X \to Y$ is not the only d-connecting path, it is not necessarily true that $\mathbb{P}(Y|X) = \mathbb{P}(Y|\text{do}(X))$. Now suppose we perform the do-see test under the intervention $\text{do}(Z)$. Then the aforementioned path is closed since X is not a descendant of T in the post interventional graph. Hence we have $\mathbb{P}(Y|X, \text{do}(Z)) = \mathbb{P}(Y|\text{do}(X, Z))$. Similarly $G2$ shows that intervening on the parent set of Y is also necessary.

We have the following theorem, which shows that we can perform the do-see test between X, Y under $\text{do}(Pa_X, Pa_Y)$:

[7]These cases are fully identified in the full version [20].

Theorem 6. *[Interventional Do-see test] Consider a causal graph D on the set of observable variables $\mathcal{V} = \{V_i\}_{i\in[n]}$ and latent variables $L = \{L_i\}_{i\in[m]}$ with edge set E. If $(V_i, V_j) \in E$, then*

$$\Pr(V_j|V_i = v_i, do(Pa_i = pa_i, Pa_j = pa_j)) = \Pr(V_j|do(V_i = v_i, Pa_i = pa_i, Pa_j = pa_j)),$$

iff $\nexists k$ such that $(L_k, V_i) \in E$ and $(L_k, V_j) \in E$, where Pa_i is the set of parents of V_i in V. Quantities on both sides are invariant irrespective of additional interventions elsewhere.

Next we need a subgraph structure to perform multiple do-see tests at once in order to efficiently discover the latents between the adjacent nodes. Performing the test for every edge would take $\mathcal{O}(n)$ even in graphs with constant degree. We use strong edge coloring of sparse graphs.

Definition 3. *A strong edge coloring of an undirected graph with k colors is a map $\chi : E \to [k]$ such that every color class is an induced matching. Equivalently, it is an edge coloring such that any two nodes adjacent to distinct edges with the same color are non-adjacent.*

Graphs of maximum degree d can be strongly edge-colored with at most $2d^2$ colors.

Lemma 7. *[6] A graph of maximum degree d can be strongly edge-colored with at most $2d^2$ colors. A simple greedy algorithm that colors edges in sequence achieves this.*

Now observe that a color class of the edges forms an induced matching. We show that due to this, the 'do' part (RHS of Theorem 6) of all the do-see tests in a color class can be performed with a single intervention while the 'see' part (RHS of Theorem 6) can be again performed with another intervention. We argue that we need exactly two different interventions per color class. The following theorem uses this property to prove correctness of Algorithm 5.

Algorithm 5 LearnLatentEdge- Observable graph D with max degree d (in-degree+out-degree), outputs all latents between edges

function LEARNLATENTEDGE($\mathcal{M}, d$)
  $L = \emptyset$.
  Apply the greedy algorithm in Lemma 7 to color the edges of D with $k \leq 2d^2$ colors.
  for $j \in [1 : k]$ **do**
    Let A_j be the nodes involved with the edges that form color class j. Let P_j be the union of parents of all nodes in A_j except the nodes in A_j.
    Let the set of tail nodes of all edges be T_j.
    Following loop requires the intervention on the set $T_j \cup P_j$, i.e. $\mathrm{do}(\{T_j, P_j\})$.
    for Every directed edge (V_t, V_h) in color class j **do**
      Calculate $S(V_t, V_h) = P(V_h|\mathrm{do}(T_j, P_j))$ using post interventional samples.
    end for
    Following loop requires the intervention on the set P_j.
    for Every directed edge (V_t, V_h) in color class j **do**
      Calculate $S'(V_t, V_h) = P(V_h|V_t, \mathrm{do}(P_j))$ using post interventional samples.
      if $S'(V_t, V_h) \neq S(V_t, V_h)$ **then**
        $L \leftarrow L \cup (V_t, V_h)$
      end if
    end for
  end for
  return The set of edges L that have latents between them.
end function

Theorem 7. *Algorithm 5 requires at most $4d^2$ interventions and outputs all latents between the edges in the observable graph.*

6 Conclusions

Learning cause-and-effect relations is one of the fundamental challenges in science. We studied the problem of learning causal models with latent variables using experimental data. Specifically, we introduced two efficient algorithms capable of learning direct causal relations (instead of ancestral relations) and finding the existence and location of potential latent variables.

References

[1] Alfred V. Aho, Michael R Garey, and Jeffrey D. Ullman. The transitive reduction of a directed graph. *SIAM Journal on Computing*, 1(2):131–137, 1972.

[2] Ayesha R. Ali, Thomas S. Richardson, Peter L. Spirtes, and Jiji Zhang. Towards characterizing markov equivalence classes for directed acyclic graphs with latent variables. In *Proc. of the Uncertainty in Artificial Intelligence*, 2005.

[3] Noga Alon. Covering graphs by the minimum number of equivalence relations. *Combinatorica*, 6(3):201–206, 1986.

[4] E. Bareinboim and J. Pearl. Causal inference by surrogate experiments: z-identifiability. In Nando de Freitas and Kevin Murphy, editors, *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 113–120, Corvallis, OR, 2012. AUAI Press.

[5] E. Bareinboim and J. Pearl. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113:7345–7352, 2016.

[6] Julien Bensmail, Marthe Bonamy, and Hervé Hocquard. Strong edge coloring sparse graphs. *Electronic Notes in Discrete Mathematics*, 49:773–778, 2015.

[7] Sofia Borboudakis, Giorgos andTriantafillou and Ioannis Tsamardinos. Tools and algorithms for causally interpreting directed edges in maximal ancestral graphs. In *Sixth European Workshop on Probabilistic Graphical Models*, 2012.

[8] Tom Claassen and Tom Heskes. Causal discovery in multiple models from different experiments. In *Advances in Neural Information Processing Systems*, pages 415–423, 2010.

[9] Frederick Eberhardt. Phd thesis. *Causation and Intervention (Ph.D. Thesis)*, 2007.

[10] Frederick Eberhardt and Richard Scheines. Interventions and causal inference. *Philosophy of Science*, 74(5):981–995, 2007.

[11] Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 13(1):2409–2464, 2012.

[12] Alain Hauser and Peter Bühlmann. Two optimal strategies for active learning of causal networks from interventional data. In *Proceedings of Sixth European Workshop on Probabilistic Graphical Models*, 2012.

[13] Christina Heinze-Deml, Marloes H. Maathuis, and Nicolai Meinshausen. Causal structure learning. *Annual Review of Statistics and Its Applications*, 2017, To appear.

[14] Patrik O Hoyer, Dominik Janzing, Joris Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In *Proceedings of NIPS 2008*, 2008.

[15] Antti Hyttinen, Frederick Eberhardt, and Patrik Hoyer. Experiment selection for causal discovery. *Journal of Machine Learning Research*, 14:3041–3071, 2013.

[16] Antti Hyttinen, Patrik O Hoyer, Frederick Eberhardt, and Matti Jarvisalo. Discovering cyclic causal models with latent variables: A general sat-based procedure. *arXiv preprint arXiv:1309.6836*, 2013.

[17] Gyula Katona. On separating systems of a finite set. *Journal of Combinatorial Theory*, 1(2):174–194, 1966.

[18] Murat Kocaoglu, Alexandros G. Dimakis, and Sriram Vishwanath. Cost-optimal learning of causal graphs. In *ICML'17*, 2017.

[19] Murat Kocaoglu, Alexandros G. Dimakis, Sriram Vishwanath, and Babak Hassibi. Entropic causal inference. In *AAAI'17*, 2017.

[20] Murat Kocaoglu*, Karthikeyan Shanmugam*, and Elias Bareinboim. Experimental design for learning causal graphs with latent variables. Technical Report R-28, AI Lab, Purdue University, `https://www.cs.purdue.edu/homes/eb/r28.pdf`, 2017.

[21] Po-Ling Loh and Peter Bühlmann. High-dimensional learning of linear causal networks via inverse covariance estimation. *Journal of Machine Learning Research*, 5:3065–3105, 2014.

[22] Sara Magliacane, Tom Claassen, and Joris M Mooij. Joint causal inference on observational and experimental datasets. *arXiv preprint arXiv:1611.10351*, 2016.

[23] Stijn Meganck, Sam Maes, Philippe Leray, and Bernard Manderick. Learning semi-markovian causal models using experiments. In *Proceedings of The third European Workshop on Probabilistic Graphical Models , PGM 06*, 2006.

[24] Pekka Parviainen and Mikko Koivisto. Ancestor relations in the presence of unobserved variables. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2011.

[25] J. Pearl, M. Glymour, and N.P. Jewell. *Causal Inference in Statistics: A Primer*. Wiley, 2016.

[26] Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2009.

[27] Jonas Peters and Peter Bühlman. Identifiability of gaussian structural equation models with equal error variances. *Biometrika*, 101:219–228, 2014.

[28] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference using invariant prediction: identification and confidence intervals. *Statistical Methodology, Series B*, 78:947 – 1012, 2016.

[29] Bernhard Schölkopf, David W. Hogg, Dun Wang, Daniel Foreman-Mackey, Dominik Janzing, Carl-Johann Simon-Gabriel, and Jonas Peters. Removing systematic errors for exoplanet search via latent causes. In *Proceedings of the 32 nd International Conference on Machine Learning*, 2015.

[30] Karthikeyan Shanmugam, Murat Kocaoglu, Alex Dimakis, and Sriram Vishwanath. Learning causal graphs with small interventions. In *NIPS 2015*, 2015.

[31] S Shimizu, P. O Hoyer, A Hyvarinen, and A. J Kerminen. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7:2003—2030, 2006.

[32] Ricardo Silva, Richard Scheines, Clark Glymour, and Peter Spirtes. Learning the structure of linear latent variable models. *Journal of Machine Learning Research*, 7:191–246, 2006.

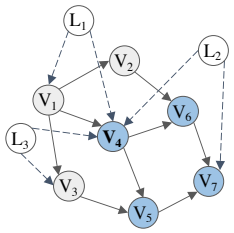
[33] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. A Bradford Book, 2001.

[34] Sofia Triantafillou and Ioannis Tsamardinos. Constraint-based causal discovery from multiple interventions over overlapping variable sets. *Journal of Machine Learning Research*, 16:2147–2205, 2015.

[35] Thomas Verma and Judea Pearl. An algorithm for deciding if a set of observed independencies has a causal explanation. In *Proceedings of the Eighth international conference on uncertainty in artificial intelligence*, 1992.

[36] Jiji Zhang. Causal reasoning with ancestral graphs. *J. Mach. Learn. Res.*, 9:1437–1474, June 2008.

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



(a)

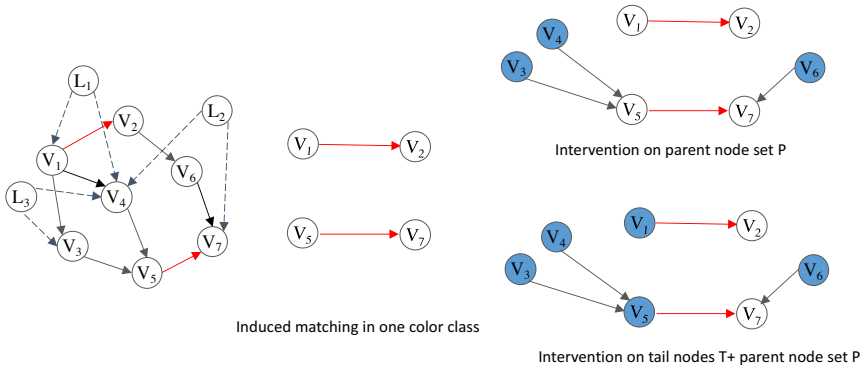
Ground Set \ Separating Sets	S_1	S_2	S_3
A	1	0	0
B	1	1	0
C	0	0	0
D	0	1	0
E	1	0	1
F	1	1	1

(b)

Ground Set \ Strongly Separating Sets	S_1	S_2	S_3	S_4	S_5	S_6
A	1	0	1	0	0	0
B	1	1	0	1	0	0
C	0	0	0	1	1	1
D	0	1	0	0	1	1
E	1	0	0	1	1	0
F	1	1	0	0	1	0

(c)

Figure 1: (a): Illustration of Lemma 1: Consider an intervention on V_4. In the post-interventional distribution, V_4 is dependent with only V_5, V_6, V_7, its descendants, despite latent connections. (b): A separating system on the ground set $\{A, B, C, D, E, F\}$. Each column is the element-set membership vector of the corresponding set. Notice that every pair of rows is distinct (b): A strongly separating system. Notice that, for every pair of rows i, j, there are two columns such that in one column row i is 1 while row j is 0 and vice versa for the other column.



(a) (b) (c)

Figure 2: Illustration of Theorem 7 and Algorithm 5 - (a) An example of an observable graph with latents with an induced matching highlighted (b): An induced matching (color class) under consideration in the outer loop of Algorithm 5. c) For every color class, only two interventions are needed. One intervenes on the observable parents of all nodes in the color class present outside it, i.e. nodes V_3, V_4 and V_6. The second intervention intervenes on the parent set along with the tail nodes in every edge of the color. This is sufficient to carry out all do-see tests in parallel for the color class.

Appendix

Illustrative Figures for Section 3

Figure 1(a) illustrates Lemma 1. Consider an intervention on V_4. This intervention disconnects V_4 from its parents, including the latent ones. In the post-interventional distribution, V_4 is d-connected to its descendants V_5, V_6, V_7. An example separating system construction is given in Figure 1(b). Figure 1(c) shows a strongly separating system construction.

In Fig. 4, we pictorially illustrate the key idea of using only two post-interventional distributions to carry out all do-see tests of Theorem 6 for all edges in a single color class simultaneously.

Blocking and Non-Blocking paths in DAGs

Consider any directed acyclic graph $D = (V, E)$. A path P from node v_0 to v_{k+1} is a sequence of nodes $P = v_0, v_1, v_2 \dots v_k, v_{k+1}$ such that for all $i \in [0 : k]$, either (v_i, v_{i+1}) or (v_{i+1}, v_i) is a directed edge in E. A node v_i with respect to a path P is said to be a collider if (v_{i+1}, v_i) and (v_{i-1}, v_i) exists in E, i.e. two directed edges collide at v_i. A path P between nodes a and b is said to be non-blocking with respect to a set S if and only if for every collider v on P, either v in S or a descendant of v is in S and no non collider v in P is in S. S is said to block a path P if it is not non-blocking.

Proof of Lemma 1

First, we prove the forward direction. Suppose that X_i and X_j are dependent under the post-interventional causal model $\mathcal{M}_S$. By the assumption of post-interventional faithfulness, this implies that there is a non-blocking path P between X_i and X_j in $D_\ell[S]$. Suppose V in P is a collider because either that or one of its descendant has to be in the conditioning set. Since, there is no conditioning set and we are testing only marginal dependence, there cannot be any collider V in P. Also note that there are no incoming edges into X_i in either $D_\ell[S]$ or $D[S]$ because there X_i is in the set S intervened on. Therefore, there cannot be any internal node $V \neq X_i, X_j$ in P with no incoming arrows because then either the path must have a collider or X_i must have incoming arrows. Since both these events are ruled out, no internal node V can have in-degree 0 or 2. The only option is for it to be a directed path from X_i to X_j in $D_\ell[S]$. This implies that no latent variable in $\mathcal{L}$ is a part of the path since latents have 0 in-degree. This implies that P is a directed path from X_i to X_j in $D[S]$ also. This proves one direction.

For the other direction, suppose there is a directed path from X_i to X_j in $D[S]$, it is still a directed path from X_i to X_j in $D_\ell[S]$ as no latents are involved. This implies that it is a non-blocking path between X_i to X_j. By the post-interventional faithfulness assumption, this implies that X_i and X_j are dependent in the causal model $\mathcal{M}_S$. This completes the proof in the other direction. □

Proof of Lemma 2

Consider the $\lceil \log n \rceil$ length binary expansions of numbers from $1 : n$. For every bit i, create a set S_i with the numbers where the i-th digit is 1 and another set S_i' with the numbers where the i-th digit is 0. The family of sets $\{S_i, S_i'\}$ if a strong separating system. It is easy to check the condition.

Proof of Theorem 1

It is enough to show that every directed edge e in the observable graph D is included at some step in E. Let the directed edge e be from U to V in the observable graph. Due to the strong separating system property, there is one intervention set S such that $U \in S,\ V \notin S$. Therefore, in that post interventional graph $D[S]$, U is an ancestor of V and therefore by Lemma 1, it is included in E after processing Line 8 for S. This implies that all directed edges of D (in addition to other ancestral relationships) are included in E. Therefore, the transitive closure at the end yields D_{tc}. □

Proof of Lemma 3

Consider the pair (X_i, X_j), where $X_i \notin S, X_j \in S, X_j \notin Pa_i$. In the post-interventional graph, X_j has no parents, including the possible latent variables. Any d-connecting path from X_j to X_i must end with an incoming arrow at X_i since any path that ends with an outgoing arrow at X_i is closed as it travels through a collider and all colliders are closed since no variable is conditioned in the graph. Thus any X_j not in the parent set of X_i is independent from X_i. Any parent of X_i will clearly be statistically dependent with X_i. □

Proof of Lemma 5

It is easy to show that $\mathrm{Tr}((D[S])_{\mathrm{tc}}) = \mathrm{Tr}(D[S])$ from the properties of $\mathrm{Tr}(\cdot)$. We will prove the rest of the implication by contradiction. Suppose (Y, V_i) is not a directed edge in $\mathrm{Tr}(D[S])$, then the ancestral relation (Y, V_i) needs to be accounted for by another directed path from Y to V_i in $\mathrm{Tr}(D[S])$. This is due to the definition of transitive reduction of $D[S]$ and the fact that V_i is connected to all its direct parents in the post interventional graph as V_i has not been intervened on. This implies that there is a directed path starting from Y and ending at some other parent $X \neq Y$ of V_i in $D[S]$. This implies that such a direct parent X has an incoming edge. This cannot happen since then by the partial ordering $\pi(\cdot)$, all direct parents of V_i above Y in the partial order have been intervened on and thereby leaving no incoming edges onto those nodes in $D[S]$. This implies a contradiction. This implies that the directed edge (Y, V_i) is present in $\mathrm{Tr}(D[S])$.

6.1 Proof of Theorem 3

Consider a directed edge (Y, V_i) in D. Let the number of direct parents of V_i above Y in the partial order be d_i. Clearly, $d_i \leq d_{\max}$. Observe that in one run of the inner for loop at Line 3, the probability V_i is excluded from S and that all direct parents of V_i above Y in the partial order are included in S is given by:

$$\begin{aligned}\Pr\left(V_i \notin S \cap \{X : \pi(X) \geq \pi(Y),\ (X, V_i) \in D\} \subseteq S\right) &= \frac{1}{d_{\max}}\left(1 - 1/d_{\max}\right)^{d_i} \\ &\geq \frac{1}{d_{\max}}\left(1 - 1/d_{\max}\right)^{d_{\max}} \\ &\overset{a}{\geq} \frac{1}{d_{\max}}\frac{1}{4} \end{aligned} \tag{1}$$

(a)- This is because $\frac{1}{4} \leq (1 - 1/n)^n \leq \frac{1}{e},\ \forall n \geq 2$ and $d_{\max} \geq 2$. Here, e is the base of the natural logarithm. Let $\mathcal{A}_i(Y)$ be the event: $V_i \notin S \cap \{X : \pi(X) \geq \pi(Y),\ (X, V_i) \in D\} \subseteq S$. By Lemma 5, the event $\mathcal{A}_i(Y)$ implies that the directed edge (Y, V_i) is included in the output and the output cannot contain any extra edges as edges set of $Tr(D[S])$ is contained in $D[S]$ which is contained in D. Now, in over $4cd_{\max}\log n$ runs of the outer for loop we upper bound the probability of failure, i.e. $(\mathcal{A}_i(Y))^c$ is true over all runs of the outer for loop.

$$\Pr((\mathcal{A}_i(Y))^c,\ \text{for all runs}) \leq (1 - \frac{1}{4d_{\max}})^{4cd_{\max}\log n} \leq \exp(-c\log n) \leq \frac{1}{n^c}. \tag{2}$$

Union bounding over all possible bad events for every pair Y, V_i in the graph D, the probability of failure is at most $\frac{1}{n^{c-2}}$.

6.2 Proof of Theorem 4

Under the interventional causal faithfulness assumption, we only need to show that, under the intervention $(Pa_i \cup Pa_j)$, two non-adjacent nodes V_i, V_j will be d-separated if and only if there is no latent variable that causes both. Consider any undirected path (a path that does not necessarily respect edge directions) between V_i and V_j in the post-interventional graph. For convenience, we say the path starts at V_i and ends at V_j without loss of generality. Since the observable parents are intervened on, any d-connecting path must start with either a child of V_i or a latent parent of V_i and end with either a child of V_j or a latent parent of V_j. If the path starts and ends with the children of V_i and V_j, then there must be a collider on the path, which closes the path since no variable is conditioned on in the graph. Consider a path that starts with a latent parent of V_i and ends with a latent parent of V_j. Since latent variables are non-adjacent, these latent variables can only be connected through their children. Hence, by the same argument that any path through the children of two variables must have a collider, the path between these two latent variables is closed, making the path between V_i and V_j closed. Consider a path that starts with a latent parent of V_i and ends with a child of V_j. The path should arrive at the child of V_j through one of its parents, as otherwise there will be a collider on the path by the same argument above. But then the child of V_j is a collider on this path, making the path closed. Hence, any path between V_i and V_j in the post-interventional graph is closed. This proves the first part of the Theorem.

Now, we show that the set S can actually larger without affecting anything. An intervention can only affect the descendant variables in the causal graph, since all the backdoor paths are closed. In the

post-interventional graph under (Pa_i, Pa_j), X_i and X_j do not have any ancestors other than the direct parents Pa_i and Pa_j. Hence, intervening on the variables in $S \backslash (Pa_i \cup Pa_j)$ does not affect the interventional distribution between X_i, X_j under $do(Pa_i, Pa_j)$.

6.3 Proof of Lemma 6

Consider the undirected version G of D. Consider the complement graph G^c. A set of independent sets $\mathcal{I} = \{I_1, I_2, \ldots, I_m\}$ in G that cover every non-edge in G is an edge-clique cover in the complement graph G^c. The minimum edge-clique cover is also known as the intersection number of the graph. When D has degree d, G^c has degree at least $n-d$. It was shown in [3] that the intersection number of graphs with degree at least $n-d$ is at most $2e^2(d+1)^2 \ln(n)$ by a probabilistic method argument that employs a randomized algorithm as follows: Choose every vertex independently with probability $\frac{1}{d+1}$ into a set S. Then prune S to delete vertices that are not connected to the rest of the vertices in S in G^c to obtain a clique S' in G^c. Repeat this $4e^2(d+1)^2 \ln(n)$ times to generate many cliques. By repeating the calculations in [3], it can be easily shown that the above randomized procedure succeeds with probability at least $1 - \frac{1}{n^2}$ in returning an edge clique cover of G^c.

6.4 Proof of Theorem 6

Proof that if there are no latents, then equality in Thm 6 holds. We use the notation L^i for the set of latent parents of V_i. Also, with a slight abuse of notation, we use Pa_j to refer to all observable parents of V_j except V_i. Suppose there does not exist a variable L_k that is the parent of both V_i and V_j ($L^i \cap L^j = \emptyset$). We can write $V_j = g(V_i, Pa_j, L_j)$.

$$\Pr(V_j | do(V_i = v_i, Pa_i = pa_i, Pa_j = pa_j)) \tag{3}$$

$$= \sum\nolimits_{l_j} \Pr(V_j | L^j = l_j, do(V_i = v_i, Pa_i = pa_i, Pa_j = pa_j)) \tag{4}$$
$$\Pr(L^j = l_j | do(V_i = v_i, Pa_i = pa_i, Pa_j = pa_j))$$

$$= \sum\nolimits_{l_j} \Pr(V_j | L^j = l_j, do(V_i = v_i, Pa_i = pa_i, Pa_j = pa_j)) \tag{5}$$
$$\Pr(L^j = l_j)$$

(4) is obtained through conditioning and marginalizing out the latent parents of X_j. (5) is due to the fact L^j are non-descendants of the set $\{X_i, Pa_i, Pa_j\}$. We also have,

$$\Pr(V_j | V_i = x_i, do(Pa_i = pa_i, Pa_j = pa_j))$$

$$= \sum\nolimits_{l_j} \Pr(V_j | L^j = l_j, V_i = v_i, do(Pa_i = pa_i, Pa_j = pa_j)) \tag{6}$$
$$\Pr(L^j = l_j | V_i = v_i, do(Pa_i = pa_i, Pa_j = pa_j))$$

$$= \sum\nolimits_{l_j} \Pr(V_j | L^j = l_j, do(V_i = v_i, Pa_i = pa_i, Pa_j = pa_j)) \tag{7}$$
$$\Pr(L^j = l_j | V_i = v_i, do(Pa_i = pa_i, Pa_j = pa_j))$$

$$= \sum\nolimits_{l_j} \Pr(V_j | L^j = l_j, do(V_i = v_i, Pa_i = pa_i, Pa_j = pa_j)) \tag{8}$$
$$\Pr(L^j = l_j)$$

(6) is obtained through conditioning and marginalizing out the other parents of Y. (7) is due to Lemma 8.

Lemma 8. *Let S_i, T_i be subsets of Pa_X such that $S_1 \cup S_2 = Pa_X, S_1 \cap S_2 = \emptyset$ and $T_1 \cup T_2 = Pa_X, T_1 \cap T_2 = \emptyset$. Then*

$$\Pr(X | S_1, do(S_2)) = \Pr(X | do(Pa_X)) = \Pr(X | Pa_X) = \Pr(X | T_1, do(T_2)). \tag{9}$$

Proof. The proof uses the invariance principle of causal Bayesian networks: The invariance principle (see Definition 1.3.1 (iii) in page 24 in ([26]) states that $\Pr(X | Pa_X = pa_X, do(Z = z)) = \Pr(X | Pa_X = pa_X)$ as long as $X \notin Z$ and $Z = z$ is consistent with $Pa_X = pa_X$. Let $Z = S_2$. Then $\Pr(X | Pa_X = pa_X, do(S_2 = s_2)) = \Pr(X | S_1 = s_1, do(S_2 = s_2))$, where $S_1 = Pa_X \backslash S_2$. Thus $\Pr(X | Pa_X = pa_x) = \Pr(X | S_1 = s_1, do(S_2 = s_2))$. From Property 1 in page 24 of

[Pearl2009], we have $\Pr(X|Pa_X = pa_X) = \Pr(X|do(S_1 = s_1, S_2 = s_2))$. Choosing T_1, T_2 instead of S_1, S_2 we can show that $\Pr(X|Pa_X = pa_X) = \Pr(X|T_1 = t_1, do(T_2 = t_2))$, which completes the proof. □

(8) is due to the following: For L^j, we have two possibilities: (i) : X_i is a non-descendant of L^j. Then the result is implied by the Markov condition. (ii). X_i is a descendant of L^j. Then there are directed paths from L^j to X_i. Note that all these paths must go through variables in Pa_i. Then, the result follows from the fact that $L^j \perp\!\!\!\perp X_i|do(Pa_i)$.

Outline of the proof that if there are latents, then equality in Thm 6 does not hold. Let us assume that the between two variables V_i and V_j a latent L_{ij} exists. Assume V_i is a parent of V_j. Suppose in contradiction, equality in Thm 6 holds. Then we have the following: Denote $\mathrm{do}(Pa_i = pa_i, Pa_j = pa_j)$ by the shorthand $\mathrm{do}(pa_{ij})$. Latent variable L_{ij} influences V_i and V_j and U_i is the exogenous variable tied to V_i. All other latents are denoted by L's and exogenous variables by U's. Consider the set of latents $\mathbf{L}_i$ which are related to V_i and let $\mathbf{l}_i$ be the values they take.

$$
\begin{aligned}
&\Pr(V_j|V_i = v_i, do(Pa_i = pa_i, Pa_j = pa_j)) = \\
&\Pr(V_j|do(V_i = v_i), do(Pa_i = pa_i, Pa_j = pa_j)) \\
&\Rightarrow \sum_{\{L_p=l_p, U_q=u_q\}_{p,q}} \Pr(V_j|V_i = v_i, \mathrm{do}(pa_{ij}), \{u_q, l_p\}) \Pr(\{u_q, l_p\}|V_i = v_i, \mathrm{do}(pa_{ij})) = \\
&\sum_{\{L_p=l_p, U_q=u_q\}_{p,q}} \Pr(V_j|\mathrm{do}(V_i = v_i), \mathrm{do}(pa_{ij}), \{u_q, l_p\}) \Pr(\{u_q, l_p\}|\mathrm{do}(V_i = v_i), \mathrm{do}(pa_{ij})) \\
&\overset{a}{\Rightarrow} \sum_{\{L_p=l_p, U_q=u_q\}_{p,q}} \Pr(V_j|\mathrm{do}(V_i = v_i), \mathrm{do}(pa_{ij}), \{u_q, l_p\}) \Pr(\mathbf{L}_i = \mathbf{l}_i, U_i = u_i|V_i = v_i, \mathrm{do}(pa_{ij})) \\
&\ldots \prod_{p:L_p \notin \mathbf{L}_i} \Pr(l_p) \prod_{q:U_q \neq U_i} \Pr(u_q) = \sum_{\{L_p=l_p, U_q=u_q\}_{p,q}} \Pr(V_j|\mathrm{do}(V_i = v_i), \mathrm{do}(pa_{ij}), \{u_q, l_p\}) \\
&\ldots \Pr(\mathbf{L}_i = \mathbf{l}_i, U_i = u_i) \prod_{p:L_p \notin \mathbf{L}_i} \Pr(l_p) \prod_{q:U_q \neq U_i} \Pr(u_q) \\
&\Rightarrow \sum_{\{L_p=l_p, U_q=u_q\}_{p,q}} \Pr(V_j|\mathrm{do}(V_i = v_i), \mathrm{do}(pa_{ij}), \{u_q, l_p\}) \Pr(\mathbf{L}_i = \mathbf{l}_i, U_i = u_i) \\
&\frac{\Pr(V_i = v_i|\mathbf{L}_i = \mathbf{l}_i, U_i = u_i, \mathrm{do}(pa_{ij}))}{\Pr(V_i = v_i|\mathrm{do}(pa_{ij})} \prod_{p:L_p \notin \mathbf{L}_i} \Pr(l_p) \prod_{q:U_q \neq U_i} \Pr(u_q) \\
&= \sum_{\{L_p=l_p, U_q=u_q\}_{p,q}} \Pr(V_j|\mathrm{do}(V_i = v_i), \mathrm{do}(pa_{ij}), \{u_q, l_p\}) \\
&\ldots \Pr(\mathbf{L}_i = \mathbf{l}_i, U_i = u_i) \prod_{p:L_p \notin \mathbf{L}_i} \Pr(l_p) \prod_{q:U_q \neq U_i} \Pr(u_q)
\end{aligned}
\tag{10}
$$

(a)- Once all hidden variables l_q, u_q(exogenous and latents are conditioned), then do operations and conditioning are identical. The distributions of latents is unaffected by interventions on observables. And latents and observables are independent. When hidden variables are conditioned on v_i and its parents are intervened on all the latents that are not related to v_i.

It seems like in both sides the ratio $\frac{\Pr(V_i=v_i|\mathbf{L}_i=\mathbf{l}_i, U_i=u_i, \mathrm{do}(pa_{ij}))}{\Pr(V_i=v_i|\mathrm{do}(pa_{ij})}$ appears which is a function of $\mathbf{l}_i, u_i$. For most functions (parameters) in the SCM, the ratios will be different from 1 and only with measure zero over the parameter space will equality hold despite the ratio being different. This gives rise to a contradiction.

6.5 Proof of Theorem 7

In this proof, when we refer to parents we refer to parent nodes from the observable graph only. Consider a color class j resulting from the strong-edge coloring of the observable graph D. Consider one

directed edge (V_t, V_h) belonging to the color class j. V_h is the vertex at the head of the edge while V_t is the vertex at the tail of the edge. First observe that $P(V_h|\mathrm{do}(T_j, P_j)) = P(V_h|\mathrm{do}(Pa_t, Pa_h, V_t))$. Here, Pa_h is the set of parent nodes of V_h not including V_t. Pa_t is the set of parent nodes of V_t. This is because V_h has no other parent other than V_t inside the color class due to the strong edge coloring property. Note that $V_t \in T_j$. Therefore, once all parents of V_h are intervened on, other interventions in the graph do not make any difference on the computation of $P(V_h|\mathrm{do}(Pa_t, Pa_h, V_t))$. Since there is no conditioning involved, Latents do not affect the equality.

Now, observe that $P(V_h|V_t, \mathrm{do}(Pa_t, Pa_h)) = P(V_h|V_t, \mathrm{do}(P_j))$. This is because all parents of V_h except V_t are in P_j due to the strong edge coloring property. All parents of V_t are in P_j due to the strong edge coloring property. Since parents of V_t are intervened on, the random variable V_t is independent of any other intervened variable in the system. Further, the joint distribution of (V_h, V_t) is invariant to conditioning on other intervened variables in the system as all parents of V_h, V_t have been intervened on (except the parent V_t of V_h). Intervention on non-parents of either V_t and V_h have no effect on the joint distribution of (V_h, V_t) on the post-interventional graph where Pa_h, Pa_t have been intervened on. Therefore, all quantities required for the do and see test can be calculated from just two interventions in the algorithm. This is true within a color class. This proves the theorem. The experimental budget is quite obvious from the algorithm. □