Disentangling Mixtures of Unknown Causal Interventions

Abhinav Kumar[*1] **Gaurav Sinha**[2]

[1]Paypal, Hyderabad, Telangana, India
[2]Adobe Research, Bangalore, Karnataka, India

Abstract

In many real-world scenarios, such as gene knockout experiments, targeted interventions are often accompanied by unknown interventions at off-target sites. Moreover, different units can get randomly exposed to different unknown interventions, thereby creating a mixture of interventions. Identifying different components of this mixture can be very valuable in some applications. Motivated by such situations, in this work, we study the problem of identifying all components present in a mixture of interventions on a given causal Bayesian Network. We construct an example to show that, in general, the components are not identifiable from the mixture distribution. Next, assuming that the given network satisfies a *positivity* condition, we show that, if the set of mixture components satisfy a mild *exclusion* assumption, then they can be uniquely identified. Our proof gives an efficient algorithm to recover these targets from the exponentially large search space of possible targets. In the more realistic scenario, where distributions are given via finitely many samples, we conduct a simulation study to analyze the performance of an algorithm derived from our identifiability proof.

1 INTRODUCTION

Motivation Causal Bayesian Networks (CBN) (Pearl [2009], Spirtes [2010]), have become the popular choice to model causal relationships in many real-world systems. These models can simulate the effects of external interventions that forcibly fix target system variables to desired target values. The simulation is done via the $do()$ operator (Pearl [2009]) wherein the CBN is altered by breaking incoming edges of the target variables and fixing them to desired target values. Pre-estimating the effect of interventions can help in decision making, for example, interventions on a CBN describing gene interactions can guide gene editing experiments.

However, real-world interventions are not always precise and mistakenly end up intervening other unintended targets. For example, gene knockout experiments via the CRISPR-Cas9 gene-editing technology perform unintended cleavage at unknown genome sites (Fu et al. [2013], Wang et al. [2015]). Moreover, the unintended intervention targets[1] can themselves be noisy i.e. different individuals targeted by the same intervention might undergo completely different off-target interventions. For example, Aryal et al. [2018] demonstrated that same gene editing experiment (using CRISPR-Cas9) on mice embryos exhibited different unintended cleavage for different mice. In such situations, units (samples) that underwent different unintended interventions are not segregated and therefore the generated distribution becomes a mixture of individual interventional distributions. We ask the following natural question.

Question 1.1. *Given access to a mixture of interventional distributions, under what conditions can one identify all the intervention targets?*

Our Contributions First, we model the situation of identifying hidden off-target interventions as the problem of identifying individual components of a mixture of interventions. We assume an underlying CBN and model interventions via the $do()$ operator described above. Second, by constructing examples, we show that, in general for a given CBN and an input mixture distribution, components of the mixture might

[*]This work was done during Abhinav Kumar's internship under the guidance and mentorship of Gaurav Sinha at Adobe Research, Bangalore, Karnataka, India.

[1]we use terms targets and components interchangeably

Accepted for the 37th Conference on Uncertainty in Artificial Intelligence (UAI 2021).

not be unique. Using this, we motivate the need for a mild *positivity* assumption (Assumption 3.2) on the distribution generated by the CBN and a mild and reasonable *exclusion* assumption (Assumption 3.1) on the structure of the intervention components present in the mixture. Third, we prove that, given access to a CBN satisfying *positivity* and any input mixture having intervention components satisfying *exclusion*, such intervention components generating the mixture can be uniquely identified from its distribution. Fourth, given oracle access to marginals of the distributions generated by the CBN and the mixture, our identifiability proof gives an efficient algorithm to recover target components from an exponentially large space of possible components. Finally, in Section 5, we conduct a simulation study to analyze the performance of an algorithm (Algorithm 1 in Appendix D) directly inspired from our identifiability proof, but with access to only finitely many samples. Even though the goal of our paper is to prove identifiability of these intervention targets, our simulations indicate that our algorithm is promising in the realistic situation of finitely many samples.

Related Prior Work Recently Squires et al. [2020] considered the problem of causal discovery using unknown intervention targets, and, as a crucial intermediate step, prove identifiability of these targets. They also design two algorithms UT-IGSP and JCI-GSP (based on the Joint Causal Inference framework in Mooij et al. [2020]) to recover these targets from data. As discussed in our motivation, in many real situations, such as Aryal et al. [2018], the off-target effects are themselves noisy and end up creating mixtures of multiple unknown interventions. Since Squires et al. [2020] assumes separate access to each unknown intervention, their algorithm cannot be used in our situation. Another line of work related to ours is the study of mixtures of Bayesian Networks. Perfect interventions i.e. $do()$ operators on the CBNs create new interventional CBNs (Definition 1.3.1 in Pearl [2009]) and therefore the input mixture in our setup is actually a mixture of Bayesian Networks. This is a more general problem and was tackled first in Thiesson et al. [1998]. They developed an Expectation-Maximization (EM) based heuristic to find individual Bayesian Network components. However, they do not investigate identifiability of the components. In our setting, we care about identifiability since the components correspond to the unknown interventions. Along with recovering the individual components of a mixture, there is also growing interest in developing techniques to understand conditional independence (CI) relationships among the variables in the mixture data. For example, some recent works try to build other graphical representations, from which the CI relationships in the mixture can be easily understood (Spirtes [1994], Ramsey et al. [2011], Strobl [2019a,b], Saeed et al. [2020]). Even though these new representations can identify some aspects of the components, none of these works prove or discuss the uniqueness and identifiability of the components, which is the main interest of our work. Finally, we would like to mention that the general area of causal discovery and inference using different kinds of unknown interventions has received a lot of attention lately (Eaton and Murphy [2007], Squires et al. [2020], Jaber et al. [2020], Mooij et al. [2020], Rothenhäusler et al. [2015]). Even though many of these do not align with goal of our paper, the growing interest in this area highlights seriousness of the issue of unintended stochasticity in targeted interventions and the desire to design algorithms robust to them.

2 PRELIMINARIES

Notation We use capital letters (e.g. X) to represent random variables and the corresponding lower case letter x to denote the assignment $X = x$. The set of values taken by random variable X will be denoted by C_X. Unless otherwise specified, all random variables in this paper are discrete and have finite support i.e. $|C_X| < \infty$. A tuple or set of random variables is denoted by capital bold face letter (e.g. $\boldsymbol{X}$) and the corresponding lower case bold faced letter $\boldsymbol{x}$ will denote the assignment $\boldsymbol{X} = \boldsymbol{x}$. Let, $C_{\boldsymbol{X}} = \prod_{X_i \in \boldsymbol{X}} C_{X_i}$ denote the set of all possible values that can be taken by $\boldsymbol{X}$. Probability of $\boldsymbol{X}$ taking the value $\boldsymbol{x}$ is denoted by $\mathbb{P}(\boldsymbol{X} = \boldsymbol{x})$ or equivalently as $\mathbb{P}(\boldsymbol{x})$ and probability of $\boldsymbol{X} = \boldsymbol{x}$ given $\boldsymbol{Y} = \boldsymbol{y}$ is denoted as $\mathbb{P}(\boldsymbol{X} = \boldsymbol{x} | \boldsymbol{Y} = \boldsymbol{y})$ or equivalently with $\mathbb{P}(\boldsymbol{x}|\boldsymbol{y})$. We will use $[n]$ to denote the set $\{1, 2, ..., n\}$, $[m, n]$ to denote set $\{m, m+1, \ldots, n\}$, calligraphic capital letters e.g. $\mathcal{S}$ to denote sets. Size of any set $\mathcal{S}$ is denoted by $|\mathcal{S}|$. $\mathbb{R}, \mathbb{R}_+$ and $\mathbb{R}_{\geq 0}$ will denote the set of real numbers, positive real numbers and non-negative real numbers respectively.

Bayesian Network Let $\mathcal{G} = \{\boldsymbol{V}, \mathcal{E}\}$ be a directed acyclic graph (DAG) with node set $\boldsymbol{V} = \{V_1, \ldots, V_n\}$ where each node V_i represents a random variable. $\mathcal{G}$ is called a Bayesian Network if the following factorization of the joint probability of $\boldsymbol{V}$ holds.

$$\mathbb{P}(\boldsymbol{v}) = \prod_{V_i \in \boldsymbol{V}} \mathbb{P}(v_i | \boldsymbol{pa}(v_i))$$

where $\boldsymbol{pa}(V_i)$ are parent nodes of V_i.

A *causal Bayesian Network* is a Bayesian Network where all edges denote direct causal relationships. It allows for modeling effect of external actions called "interventions", by appropriate modification of the Bayesian Network. A formal definition of causal Bayesian Networks can be found in Definition 1.3.1, Pearl [2009].

Interventions: As mentioned above, these capture

external actions on a system under consideration, for example, dosage of medicines administered to a patient, providing subsidies to poorer sections of the population, etc. A natural way to model them in causal Bayesian Networks is to perform the act of *causal surgery*, wherein, incoming edges into the node(s) to be intervened are removed and the node(s) is forcibly fixed to the desired value. As described in Definition 1.3.1, Pearl [2009], the new network thus obtained is treated as the Bayesian Network modelling effect of the intervention. Formally, following the notation in Pearl [2009], if we perform intervention on nodes $\boldsymbol{X} \subseteq \boldsymbol{V}$ with a desire to set it to value $\boldsymbol{x}^* \in C_{\boldsymbol{X}}$, then the effect of this intervention (also known as *interventional distribution*) is a probability distribution on $\boldsymbol{V}$ denoted as $\mathbb{P}(\boldsymbol{v}|do(\boldsymbol{x}^*))$ (or $\mathbb{P}_{\boldsymbol{x}^*}(\boldsymbol{v})$). In the intervened Bayesian Network, conditional probability distributions (CPD) $\mathbb{P}(X_i|\boldsymbol{pa}(X_i))$ of all $X_i \in \boldsymbol{X}$ that are intervened and set to x_i^*, changes to the Kronecker delta function δ_{x_i,x_i^*} i.e. $\mathbb{P}(X_i = x_i|\boldsymbol{pa}(X_i)) = 1$ if $x_i = x_i^*$ else it is 0. The CPD of the non-intervened nodes i.e. $\boldsymbol{V} \setminus \boldsymbol{X}$ remains unchanged. Hence the interventional distribution factorizes as:

$$\mathbb{P}_{\boldsymbol{x}^*}(\boldsymbol{v}) = \prod_{V_i \notin \boldsymbol{X}} \mathbb{P}(v_i|\boldsymbol{pa}(v_i)) \prod_{V_i \in \boldsymbol{X}} \delta_{v_i,x_i^*}$$

Such interventions are called *perfect interventions*. They capture many real-world situations like *gene-editing-experiments*, where a certain target gene is spliced out and replaced with the desired gene. Other kinds of interventions such as *imperfect, uncertain* e.t.c. have been defined in literature (Section 2 in Eaton and Murphy [2007]). However, in this paper we only deal with perfect interventions.

3 PROBLEM FORMULATION AND MAIN THEOREM

As motivated in Section 1, the intended interventions performed during an experiment often have hidden off-target effects, which could themselves be stochastic, leading to different hidden treatments on different individuals. We can model such a situation as an unknown mixture of different off-target interventions. Here is a formal definition.

Definition 3.1 (Mixture of Interventions)**.** Let $\mathcal{G} = \{\boldsymbol{V}, \mathcal{E}\}$ be a causal Bayesian Network. A probability distribution $\mathbb{P}_{mix}(\boldsymbol{V})$ is called a mixture of interventions if for some $m \in \mathbb{N}$, there exist subsets $\boldsymbol{T}_1, \ldots, \boldsymbol{T}_m \subseteq \boldsymbol{V}$, corresponding values $\boldsymbol{t}_i \in \mathcal{C}_{\boldsymbol{T}_i}$, and positive scalar weights $\pi_i \in \mathbb{R}_+$, $i \in [m]$, such that

$$\mathbb{P}_{mix}(\boldsymbol{V}) = \sum_{i=1}^{m} \pi_i \mathbb{P}_{\boldsymbol{t}_i}(\boldsymbol{V})$$

where $\boldsymbol{t}_i \neq \boldsymbol{t}_j$ for all $i \neq j \in [n]$[2]. We allow $\boldsymbol{T_i} = \varnothing$, in which case, $\mathbb{P}_{\boldsymbol{t}_i}(\boldsymbol{V})$ is defined as $\mathbb{P}(\boldsymbol{V})$. Note that for $\mathbb{P}_{mix}$ to be a valid distribution $\sum_{i=1}^{m} \pi_i = 1$. We refer to the set $\mathcal{T} = \{(\boldsymbol{t}_i, \pi_i), i \in [m]\}$ as a set of *intervention tuples* generating the mixture.

Uniqueness and Identifiability : In our mixture model, each of the targets $\boldsymbol{t}_i$, corresponds to an intervention that intentionally or unintentionally transpired in the experiment. Since our ultimate goal is to recover them from the mixture distribution (see Question 1.1), the problem only makes sense if they "uniquely" define the mixture. Formally, there should not exist two distinct sets of intervention tuples $\mathcal{T}_1 = \{(\boldsymbol{t}_1^1, \pi_1^1), \ldots, (\boldsymbol{t}_n^1, \pi_1^1)\}$ and $\mathcal{T}_2 = \{(\boldsymbol{t}_1^2, \pi_1^2), \ldots, (\boldsymbol{t}_m^2, \pi_m^2)\}$ which generate the same mixture distribution, i.e.,

$$\mathbb{P}_{mix}(\boldsymbol{V}) = \sum_{i=1}^{n} \pi_i^1 \mathbb{P}_{\boldsymbol{t}_i^1}(\boldsymbol{V}) = \sum_{j=1}^{m} \pi_j^2 \mathbb{P}_{\boldsymbol{t}_j^2}(\boldsymbol{V})$$

An immediate next question is that of "identifiability". Given access to a causal Bayesian Network and the joint distribution $\mathbb{P}(\boldsymbol{V})$ it captures, does there exist an algorithm, that takes as input the mixture distribution $\mathbb{P}_{mix}(\boldsymbol{V})$ and exactly recovers the unknown set of intervention tuples that generated $\mathbb{P}_{mix}(\boldsymbol{V})$? In the general case, the answer to both these questions is no! Using a very simple network, with just one node, we show that mixture distributions need not be unique, motivating the need for more assumptions. More complicated examples with multiple nodes can be easily created in the same way, but, for a cleaner presentation we stick to this example since its purpose is to only motivate an assumption we make next.

Example 3.1. Consider a causal Bayesian Network with a single binary variable $\boldsymbol{V} = \{V_1\}$, i.e. $C_{V_1} = \{0, 1\}$ and denote $\mathbb{P}(V_1 = 0), \mathbb{P}(V_1 = 1)$ by p_0, p_1 respectively. Define the mixture,

$$\mathbb{P}_{mix}(V_1) = \pi_0 \mathbb{P}_0(V_1) + \pi_1 \mathbb{P}_1(V_1) + (1 - \pi_0 - \pi_1)\mathbb{P}(V_1)$$

On setting $V_1 = 0$ and then $V_1 = 1$ in the above equation, and rearranging the terms, we obtain

$$\begin{bmatrix} 1 - p_0 & -p_0 \\ p_0 - 1 & p_0 \end{bmatrix} \begin{bmatrix} \pi_0 \\ \pi_1 \end{bmatrix} = \begin{bmatrix} \mathbb{P}_{mix}(V_1 = 0) - p_0 \\ \mathbb{P}_{mix}(V_1 = 1) - p_1 \end{bmatrix}$$

The above 2×2 matrix is singular and has rank 1 i.e. the system does not have a unique solution. In fact, when $0 < p_0 < 1$,

$$\pi_0 = \frac{\mathbb{P}_{mix}(V_1 = 0) - p_0 + p_0 t}{1 - p_0}, \quad \pi_1 = t$$

[2] if $t_i = t_j$, then $(\pi_i + \pi_j)\mathbb{P}_{\boldsymbol{t}_i}(\boldsymbol{V})$ is one component.

are all valid solutions whenever $t \leq 1 - \mathbb{P}_{mix}(V_1 = 0)$ and $t \geq \max\{\frac{p_0 - \mathbb{P}_{mix}(V_1=0)}{p_0}, 0\}$. Therefore, uniqueness of intervention tuples does not hold in general.

Even though the example looks very simple, it captures the main reason behind the non-identifiability of the set of intervention tuples. Exactly like the above example, for any mixture, we can obtain systems of linear equations by evaluating marginal probabilities of $\mathbb{P}_{mix}$ for different settings of $\boldsymbol{V}$. Our goal then would be to find settings which help us solve these systems uniquely and recover the set of intervention tuples. Unfortunately, in this process, similar to the above example, the linear systems will have dependent equations and therefore infinitely many solutions. To get over this issue, we focus our attention on sets of intervention tuples, where, for each variable there exists some value that is missing from all of it's intervention targets. In, our main theorem, we show that any mixture generated by such a set cannot be generated by any other set of this kind. Next, we formally state the assumption and then discuss why it is extremely mild and reasonable in most real situations.

Assumption 3.1 (Exclusion)**.** Let $\mathcal{T}$ be a set of intervention tuples as defined in Definition 3.1. We say that $\mathcal{T}$ satisfies *exclusion*, if for all $V_i \in \boldsymbol{V}$, there exists $\bar{v}_i \in C_{V_i}$ such that $\bar{v}_i \notin \boldsymbol{t}$ for any target $\boldsymbol{t}$ belonging to any tuple in $\mathcal{T}$. We say that a mixture of interventions $\mathbb{P}_{mix}(\boldsymbol{V})$ satisfies *exclusion* if some set of intervention tuples $\mathcal{T}$ generating it satisfies exclusion.

Remark. *This assumption puts only a mild constraint on the set of mixtures we consider. For example, in a network with n nodes and each node having $\leq k$ possible values, excluding a fixed value of each node, can still generate arbitrary mixtures over $\Omega(k^n)$ allowed targets. Without exclusion, there are $O((k+1)^n)$ possible targets that generate the mixtures. Therefore the reduction is minimal compared to the size of the space of targets we are searching in. In real-world applications, it's common for nodes to have a large number of possible values. Therefore, for each node, the possibility of off-target interventions impacting all values becomes unlikely. We also emphasize that the values missing from the targets can be different for different input mixtures and are not known to our algorithms. Our identifiability algorithm only uses existence of such missing values making its interpretation even more general.*

Even though the above assumption helps us tackle the singularity problem outlined in Example 3.1, it is not enough to guarantee uniqueness of intervention tuples in general. We also assume a simple "positivity" assumption on the causal Bayesian Network, which demands that the joint probability $\mathbb{P}(\boldsymbol{v}) > 0$ for any setting $\boldsymbol{V} = \boldsymbol{v}$. In fact, using the same example as above (Example 3.1), we show that not assuming $p_0, p_1 > 0$, can lead to multiple set of intervention tuples satisfying Assumption 3.1 and generating the same mixture. To see this, we consider the input mixture $\mathbb{P}_{mix}(\boldsymbol{V}) = \mathbb{P}(\boldsymbol{V})$. The set of intervention tuples $\mathcal{T}_1 = \{(\varnothing, 1)\}$ for it clearly satisfies Assumption 3.1 as intervention targets $(V_1 = a)$ and $(V_1 = b)$ are excluded. Now, if $p_1 = 0$, then $\mathbb{P}_0(V_1) = \mathbb{P}(V_1)$ and for any $\pi_0 \in [0, 1]$, we can trivially write

$$\mathbb{P}_{mix}(V_1) = \pi_0 \mathbb{P}_0(V_1) + (1 - \pi_0)\mathbb{P}(V_1)$$

implying that $\mathcal{T}_2 = \{(V_1 = 0, \pi_0), (\varnothing, 1 - \pi_0)\}$ is another set of intervention tuples for $\mathbb{P}_{mix}$, implying non-uniqueness. Here is the statement of our assumption.

Assumption 3.2 (Positivity)**.** Let $\boldsymbol{V}$ be the set of nodes in our causal Bayesian Network and $\mathbb{P}(\boldsymbol{V})$ be the corresponding joint probability distribution. We assume that $\mathbb{P}(\boldsymbol{v}) > 0$ for all $\boldsymbol{v} \in C_{\boldsymbol{V}}$.

Remark. *As a straight forward consequence of this assumption, for every random variable $V_i \in \boldsymbol{V}$, we can show that the conditional probability distributions are positive as well i.e. $\mathbb{P}(v_i | \boldsymbol{pa}(v_i)) > 0$ for all $v_i \in C_{V_i}$ and setting $\boldsymbol{pa}(v_i)$ of the parents. This positivity assumption is commonly assumed in many works related to causal graphs. e.g. Hauser and Bühlmann [2012] assume positivity throughout their discussion when characterizing the Interventional Markov Equivalence class.*

Having stated these assumptions, we are now ready to state the main theorem of this paper. A detailed proof is provided in Section 4.

Theorem 3.1. *Let $\mathcal{G} = \{\boldsymbol{V}, \mathcal{E}\}$ be a causal Bayesian Network and $\mathbb{P}(\boldsymbol{V})$ be the associated joint probability distribution satisfying Assumption 3.2. Let $\mathbb{P}_{mix}(\boldsymbol{V})$ (Definition 3.1) be any mixture of interventions that satisfies Assumption 3.1. The following are true.*

1. *There exists a unique set of intervention tuples $\mathcal{T} = \{(\boldsymbol{t_1}, \pi_1), \ldots, (\boldsymbol{t_m}, \pi_m)\}$ satisfying Assumption 3.1, such that*

$$\mathbb{P}_{mix}(\boldsymbol{V}) = \sum_{i=1}^{m} \pi_i \mathbb{P}_{\boldsymbol{t_i}}(V).$$

2. *Given access to $\mathcal{G}$, $\mathbb{P}(\boldsymbol{V})$ and $\mathbb{P}_{mix}(\boldsymbol{V})$, there exists an algorithm, that runs in time $n * (m * k_{max})^{O(1)}$, and, outputs the set of intervention tuples $\mathcal{T}$ (satisfying Assumption 3.1) generating it. Here n is the number of nodes in $\mathcal{G}$, m is the size of set $\mathcal{T}$ and k_{max} is the maximum number of distinct values that any node can take.*

Remark. *Though Assumption 3.2 is a sufficient conditions for Theorem 3.1, it is not necessary. In Example B.1 (Appendix B), we give an example that does not satisfy this assumption but is uniquely generated by a set of intervention tuples satisfying Assumption 3.1.*

4 PROOF OF MAIN THEOREM

In this section, we provide rigorous proof to both parts of Theorem 3.1 together. Our uniqueness proof (for Part 1) is constructive and gives an algorithm as described in Part 2. Our proof goes via an induction argument on the number of nodes n present in the given Bayesian Network. There are many lemmas stated throughout the proof. For a cleaner exposition, all of their proofs are provided in Appendix A.

4.1 BASE CASE ($n = 1$)

Consider a causal Bayesian Network $\mathcal{G} = (V, \mathcal{E})$ with only one vertex V and no edges (i.e. $\mathcal{E} = \varnothing$), such that $\mathbb{P}(V)$ satisfies Assumption 3.2. Let $C_V = \{v^1, \ldots, v^k\}$ be the set of values that V can take. Therefore, by Assumption 3.2, $\mathbb{P}(v^i) > 0$ for all $i \in [k]$. Next, consider any mixture of interventions $\mathbb{P}_{mix}(V)$ that satisfies Assumption 3.1. Writing the most general form of $\mathbb{P}_{mix}$, i.e. allowing for scalar weights to be ≥ 0, we can write,

$$\mathbb{P}_{mix}(V) = \pi_0 \mathbb{P}_{t_0}(V) + \pi_1 \mathbb{P}_{t_1}(V) + \ldots + \pi_k \mathbb{P}_{t_k}(V),$$

where $t_0 = \varnothing, t_1 = v^1, \ldots, t_k = v^k$. By the notation in Definition 3.1, $\mathbb{P}_{\varnothing}(V) = \mathbb{P}(V)$. Subtracting $\mathbb{P}(V)$ from both sides and setting $\pi_0 = 1 - \sum_{i=1}^{k} \pi_i$, we get,

$$\mathbb{P}_{mix}(V) - \mathbb{P}(V) = \sum_{i=1}^{k} \pi_i (\mathbb{P}_{v^i}(V) - \mathbb{P}(V)).$$

Recall, from the definition of interventions in Section 3, for any $v^j \in C_V$, $\mathbb{P}_{v^i}(v^j) = \delta_{v^i, v^j}$. Substituting $V = v^1, \ldots, v^k$ and using $\mathbb{P}_{v^i}(v^j) = \delta_{v^i, v^j}$, gives us k linear equations which can be written in the following matrix form:

$$\begin{bmatrix} 1-a_1 & -a_1 & . & . & -a_1 \\ -a_2 & 1-a_2 & . & . & -a_2 \\ . & . & . & . & . \\ -a_k & -a_k & . & . & 1-a_k \end{bmatrix} \begin{bmatrix} \pi_1 \\ \pi_2 \\ . \\ \pi_k \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ . \\ b_k \end{bmatrix} \quad (1)$$

where $b_i = \mathbb{P}_{mix}(v^i) - \mathbb{P}(v^i)$ and $a_i = \mathbb{P}(v^i) > 0$ (Assumption 3.2). Any set of intervention tuples $\mathcal{T}$ generating $\mathbb{P}_{mix}(V)$ can be obtained as a solution to the above system. Since, in Part 1, we restrict our focus to $\mathcal{T}$ that satisfy Assumption 3.1, we know there exists some $i \in [k]$, such that $\pi_i = 0$. In the following lemma, we show that such a system under these assumptions has a unique solution when $\pi_1, \ldots, \pi_k \in \mathbb{R}_{\geq 0}$. Proof of this lemma is presented in Appendix A.1.

Lemma 4.1. *Consider the following linear system.*

$$\begin{bmatrix} c-a_1 & -a_1 & . & . & -a_1 \\ -a_2 & c-a_2 & . & . & -a_2 \\ . & . & . & . & . \\ -a_k & -a_k & . & . & c-a_k \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ . \\ x_k \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ . \\ b_k \end{bmatrix}$$

Assume that $a_1, \ldots, a_k > 0$, $\sum_{i=1}^{k} a_i = c$ and it has at least one solution. Then, rank of the above matrix is $k-1$ and there are infinitely many solutions. Under the assumption that $\boldsymbol{x} \in \mathbb{R}_{\geq 0}$ and $x_i = 0$ for some $i \in [k]$, the solution becomes unique. Given access to a_is, b_is and c, there exists an algorithm that computes this solution in $k^{O(1)}$ time.

It's easy to see that Equation 1 satisfies all requirements of Lemma 4.1, implying the base case of our induction proof.

Inductive hypothesis ($n = N$)**:**Assume, Theorem 3.1 is true for all causal Bayesian Networks on N nodes, that satisfy Assumption 3.2 and input mixtures that satisfy Assumption 3.1.

4.2 INDUCTION STEP ($n = N + 1$)**:**

Assuming the above inductive hypothesis, we show that Theorem 3.1 is true for all causal Bayesian Networks on $N+1$ nodes, and mixture on interventions on it, satisfying Assumptions 3.2 and 3.1 respectively. Let $\boldsymbol{V} = \{V_1, \ldots, V_{N+1}\}$, $\mathbb{P}(\boldsymbol{V})$ be the distribution of $\boldsymbol{V}$ and $\mathbb{P}_{mix}(\boldsymbol{V})$ be any mixture of interventions that satisfies Assumption 3.1. We wish to show that there is a unique set of intervention tuples satisfying Assumption 3.1 that generates $\mathbb{P}_{mix}(\boldsymbol{V})$. Without loss of generality let $V_1 \prec \ldots \prec V_{N+1}$ be a topological order for $\mathcal{G}$. We will now marginalize on V_{N+1} to reduce our problem to the $n = N$ case, so that we can use the inductive hypothesis. The following lemma is required to make this argument. We present it's proof in Appendix A.2.

Lemma 4.2. *Let $\boldsymbol{V}_N = \{V_1, \ldots, V_N\}$,*

1. *$\mathbb{P}(\boldsymbol{V}_N)$ is generated by the CBN $\mathcal{G}_N = \mathcal{G} \setminus \{V_{N+1}\}$. and satisfies Assumption 3.2.*
2. *$\mathbb{P}_{mix}(\boldsymbol{V}_N)$ can be written as a mixture of interventions on $\mathcal{G}_N$ that satisfies Assumption 3.1.*
3. *Given access to $\mathbb{P}(\boldsymbol{V}), \mathbb{P}_{mix}(\boldsymbol{V})$, in $O(k_{max})$ time we can create access to $\mathbb{P}(\boldsymbol{V}_N), \mathbb{P}_{mix}(\boldsymbol{V}_N)$, by marginalizing on V_{N+1}.*

Using the inductive hypothesis with this claim, we get that there exists a unique set of intervention tuples

$\mathcal{S} = \{(\boldsymbol{s}_1, \mu_1), \ldots, (\boldsymbol{s}_q, \mu_q)\}$[3] satisfying Assumption 3.1 that generates $\mathbb{P}_{mix}(\boldsymbol{V}_N)$, i.e.,

$$\mathbb{P}_{mix}(\boldsymbol{V}_N) = \sum_{j=1}^{q} \mu_j \mathbb{P}_{\boldsymbol{s}_j}(\boldsymbol{V}_N),$$

The induction hypothesis also implies that $\mathcal{S}$ can be computed in $N * (q * k_{max})^{O(1)}$ time using access to $\mathbb{P}(\boldsymbol{V}_N)$ and $\mathbb{P}_{mix}(\boldsymbol{V}_N)$. The next step in our proof then is to show that, for a given $\mathcal{G}$, $\mathbb{P}(\boldsymbol{V})$ and $\mathbb{P}_{mix}(\boldsymbol{V})$, the set of intervention tuples $\mathcal{S}$ can be uniquely lifted to a set $\mathcal{T}$ of intervention tuples that satisfies Assumption 3.1 and generates $\mathbb{P}_{mix}(\boldsymbol{V})$. We also show that using access to $\mathcal{G}$, $\mathbb{P}(\boldsymbol{V})$ and $\mathbb{P}_{mix}(\boldsymbol{V})$, the lifting process runs in $(m * k_{max})^{O(1)}$ time implying that $\mathcal{T}$ can be computed in $(N+1) * (m * k_{max})^{O(1)}$ time.

4.2.1 Lifting $\mathcal{S}$

In this section we lift the set of intervention tuples $\mathcal{S}$ generating $\mathbb{P}_{mix}(\boldsymbol{V}_N)$ uniquely to a set of intervention tuples satisfying Assumption 3.1 generating $\mathbb{P}_{mix}(\boldsymbol{V})$. Let $\mathcal{T} = \{(\boldsymbol{t}_1, \pi_1), \ldots, (\boldsymbol{t}_m, \pi_m)\}$ be any arbitrary set of intervention tuples satisfying Assumption 3.1 that generates $\mathbb{P}_{mix}(\boldsymbol{V})$, i.e.

$$\mathbb{P}_{mix}(\boldsymbol{V}) = \sum_{i=1}^{m} \pi_i \mathbb{P}_{\boldsymbol{t}_i}(\boldsymbol{V}). \tag{2}$$

First, we give a lemma that connects targets $\boldsymbol{t}_1, \ldots \boldsymbol{t}_m$ inside $\mathcal{T}$ with targets $\boldsymbol{s}_1, \ldots \boldsymbol{s}_q$ inside $\mathcal{S}$. We present it's proof in Appendix A.3.

Lemma 4.3. *For every $\boldsymbol{t}_i, i \in [m]$, there is some $\boldsymbol{s}_j, j \in [q]$ such that, either $\boldsymbol{t}_i = \boldsymbol{s}_j$ or $\boldsymbol{t}_i = \boldsymbol{s}_j \cup \{v\}$ for some v in $C_{V_{N+1}}$.*

For $j \in [q]$, we define sets $\mathcal{S}_j = \{\boldsymbol{s}_j, \boldsymbol{s}_j \cup \{v^1\}, \ldots \boldsymbol{s}_j \cup \{v^k\}\}$ where $C_{V_{N+1}} = \{v^1, \ldots, v^k\}$. Since the targets $\boldsymbol{s}_j, j \in [q]$ are distinct, the sets $\mathcal{S}_j, j \in [q]$ are disjoint. Lemma 4.3 implies that

$$\{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_m\} \subset \mathcal{S}_1 \cup \ldots \mathcal{S}_q$$

Since $\mathcal{T}$ was arbitrary, for every such $\mathcal{T}$, there exist non-negative scalars $\pi_{\boldsymbol{s}}$, $\boldsymbol{s} \in \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_q$ such that Equation 2 can be written as,

$$\mathbb{P}_{mix}(\boldsymbol{V}) = \sum_{j=1}^{q} \sum_{\boldsymbol{s} \in \mathcal{S}_j} \pi_{\boldsymbol{s}} \mathbb{P}_{\boldsymbol{s}}(\boldsymbol{V}) \tag{3}$$

Any solution of Equation 3 with $\pi_{\boldsymbol{s}} \geq 0$ gives a set of intervention tuples for $\mathbb{P}_{mix}$. We show that there is a unique such set which satisfies Assumption 3.1.

[3]For $i \in [q]$, $\boldsymbol{s}_i$ are values taken by variables $\boldsymbol{S}_i \subset \boldsymbol{V}$

Lemma 4.4. *Let $\pi_{\boldsymbol{s}}$, $\boldsymbol{s} \in \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_q$ be some non-negative solution to Equation 3 and the set $\mathcal{T} = \{(\boldsymbol{s}, \pi_{\boldsymbol{s}}) : \pi_{\boldsymbol{s}} > 0\}$ be the corresponding set of intervention tuples. There exists a unique $\mathcal{T}$ that satisfies Assumption 3.1.*

Proof. We show that enforcing Assumption 3.1 uniquely determines all $\pi_{\boldsymbol{s}}$ as solutions to a sequence of linear equations, implying that there is a unique $\mathcal{T}$ that satisfies Assumption 3.1. To construct this sequence, we need an ordering on $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_q$. So, without loss of generality, we assume that for $j_1 \leq j_2$, $\boldsymbol{s}_{j_2} \not\subseteq \boldsymbol{s}_{j_1}$. The linear equations are created by using specific settings for $\boldsymbol{V}$ in Equation 3 which enable us to decompose the linear system into a sequence of simpler systems i.e. one for each $\mathcal{S}_i$. We propose these settings next and explain why and how they work. Since $\mathcal{S}$ satisfies Assumption 3.1, there exists $\bar{v}_i \in V_i, i \in [N]$ such that for all $j \in [q]$, $\bar{v}_i \notin \boldsymbol{s}_j$. For $j \in [q]$, we define $\boldsymbol{s}_{-j} = \{\bar{v}_i : V_i \notin \boldsymbol{S}_j\}$[4] and for every $l \in [k]$, create settings

$$\boldsymbol{v}_{j,l} = \boldsymbol{s}_j \cup \boldsymbol{s}_{-j} \cup \{v^l\}$$

where $C_{V_{N+1}} = \{v^1, \ldots, v^k\}$. The following lemma is used to decompose the system of equations into simpler systems. Proof is presented in Appendix A.4.

Lemma 4.5. *For $i \in [q]$, $l \in [k]$ and $\boldsymbol{s} \in \mathcal{S}_{i+1} \cup \ldots \cup \mathcal{S}_q$,*

$$\mathbb{P}_{\boldsymbol{s}}(\boldsymbol{v}_{i,l}) = 0$$

Using this in Equation 3, leaves us with the following simpler system for every $i \in [q]$,

$$\mathbb{P}_{mix}(\boldsymbol{v}_{i,l}) - \sum_{j=1}^{i-1} \sum_{\boldsymbol{s} \in \mathcal{S}_j} \pi_{\boldsymbol{s}} \mathbb{P}_{\boldsymbol{s}}(\boldsymbol{v}_{i,l}) = \sum_{\boldsymbol{s} \in \mathcal{S}_i} \pi_{\boldsymbol{s}} \mathbb{P}_{\boldsymbol{s}}(\boldsymbol{v}_{i,l}) \tag{4}$$

Suppose all $\pi_{\boldsymbol{s}}, \boldsymbol{s} \in \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_{i-1}$ have been determined. Then the left had side of this equation is completely known and has no unknown variables. We denote it by Δ going forward. Therefore, by varying $l \in [k]$, we have k equations in $k+1$ variables $\pi_{\boldsymbol{s}}, s \in \mathcal{S}_i$. In the next lemma, we will obtain a linear equation satisfied by these $k+1$ variables and reduce the system to k equations in k variables. On marginalizing over V_{N+1} in Equation 3, we get

Lemma 4.6. *For all $i \in [q]$, the following holds.*

$$\mu_i = \sum_{\boldsymbol{s} \in \mathcal{S}_i} \pi_{\boldsymbol{s}}$$

Proof of Lemma 4.6 is presented in Appendix A.5. By making the substitution from this lemma above into

[4]$\boldsymbol{s}_j$ corresponds to set of variables $\boldsymbol{S}_j \subset \boldsymbol{V}$.

Equation 4, we get the equation

$$\Delta - \mu_i \mathbb{P}_{\boldsymbol{s}_i}(\boldsymbol{v}_{i,l}) = \sum_{l \in [k]} \pi_{\boldsymbol{s}_i \cup \{v^l\}} (\mathbb{P}_{\boldsymbol{s}_i \cup \{v^l\}}(\boldsymbol{v}_{i,l}) - \mathbb{P}_{\boldsymbol{s}_i}(\boldsymbol{v}_{i,l})) \tag{5}$$

that gives a system of k equations in k variables when we vary $l \in [k]$. Clearly we are looking for non-negative solutions for $\pi_{\boldsymbol{s}_i \cup \{v^l\}}$, $l \in [k]$. When we enforce Assumption 3.1, there is some $l \in [k]$ such that $\pi_{\boldsymbol{s}_i \cup \{v^l\}} = 0$. In Lemma 4.7, we show that we can uniquely solve Equation 5 for such $\pi_{\boldsymbol{s}_i \cup \{v^l\}}$, $l \in [k]$.

Lemma 4.7. *For every $i \in [q]$, Equation 5 has a unique solution when we enforce that $\pi_{\boldsymbol{s}_i \cup \{v^l\}}$, $l \in [k]$ are non-negative and at least one of them is 0.*

We present a proof of this Lemma in Appendix A.6. This lemma implies that under enforcement of Assumption 3.1, all targets $\in \mathcal{S}_i$ (and their respective mixing coefficients) that appear in $\mathbb{P}_{mix}(\boldsymbol{V})$ get uniquely identified. Using this technique from $i = 1$ to q, any set of intervention tuples satisfying Assumption 3.1 that generates $\mathbb{P}_{mix}(\boldsymbol{V})$ gets uniquely identified. Therefore, there is a unique set of intervention tuples $\mathcal{T}$ that generates $\mathbb{P}_{mix}$ and satisfies Assumption 3.1. □

The lifting of targets in $\mathcal{S}_i$ is done in Lemma 4.7 using technique from Lemma 4.1 which takes $(k_{max})^{O(1)}$ time. This is repeated for all $i \in [q]$, therefore, we spend $(q * k_{max})^{O(1)}$ time. It's easy to see that $q < m$ and so using the induction hypothesis the set of intervention tuples is computed in $(N+1)*(m*k_{max})^{O(1)}$ time, completing the induction step. We describe our complete algorithm in Algorithm 1. It's correctness and time complexity follows from the discussion in this section. For better understanding, in Examples C.1 and C.2 (Appendix C), we provide two worked out examples on small problem instances, that illustrate important aspects of our algorithm.

Algorithm 1: DISENTANGLE

input : Variables $\boldsymbol{V} = (V_1, \ldots, V_{N+1})$, CBN $\mathcal{G}$, Distributions $\mathbb{P}(\boldsymbol{V}), \mathbb{P}_{mix}(\boldsymbol{V})$

output : Set of intervention tuples $\mathcal{T}$

1. When $|\boldsymbol{V}| = 1$, setup the linear system in Equation 1 and solve it using technique described in Lemma 4.1 to obtain a set $\mathcal{T}$ of intervention tuples. **return** $\mathcal{T}$.
2. Let $V_1 \prec \ldots \prec V_{N+1}$ denote a topological order in $\mathcal{G}$. Marginalize on V_{N+1} to create access to $\mathbb{P}_{mix}(\boldsymbol{V}_N)$ and $\mathbb{P}(\boldsymbol{V}_N)$ where $\boldsymbol{V}_N = (V_1, \ldots, V_N)$. Construct $\mathcal{G}_N = \mathcal{G} \setminus \{V_{N+1}\}$. Recursively call this algorithm with inputs $\mathcal{G}_N$, $\mathbb{P}(\boldsymbol{V}_N)$, $\mathbb{P}_{mix}(\boldsymbol{V}_N)$, to compute the unique set of intervention tuples $\mathcal{S} = \{(\boldsymbol{s}_1, \mu_1), \ldots, (\boldsymbol{s}_q, \mu_q)\}$ that satisfies Assumption 3.1 and generates $\mathbb{P}_{mix}(\boldsymbol{V}_N)$. Let $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_q$ be ordered such that $i \leq j$ implies that $\boldsymbol{s}_j \not\subseteq \boldsymbol{s}_i$. For each $i \in [N]$, by inspecting $\boldsymbol{s}_j, j \in [q]$, identify $\bar{v}_i \in C_{V_i}$ such that $\bar{v}_i \notin \boldsymbol{s}_j$ for any $j \in [q]$. Define $\boldsymbol{s}_{-j} = \{\bar{v}_i : V_i \notin S_j\}$. Let $C_{V_{N+1}} = \{v^1, \ldots, v^k\}$. For each $i \in [q]$ and $l \in [k]$, create setting $\boldsymbol{v}_{i,l} = \boldsymbol{s}_i \cup \boldsymbol{s}_{-i} \cup \{v^l\}$.
3. For each fixed $i \in [q]$, evaluate distributions for different $\boldsymbol{v}_{i,l}$, $l \in [k]$, to setup the system of equations described in Equation 5. Solve the system using the technique outlined in proof of Lemma 4.7 (which in turn uses Lemma 4.1). At the end of this process collect all the intervention tuples thus obtained (for all $i \in [q]$), in the set $\mathcal{T}$. **return** $\mathcal{T}$.

5 SIMULATION STUDY

We conduct a simulation study to experimentally analyze performance of Algorithm 1 (Appendix D) which modifies Algorithm 1 (Section 4) to make it work with finitely many samples from $\mathbb{P}_{mix}(\boldsymbol{V})$, $\mathbb{P}(\boldsymbol{V})$.

Simulation Setup For each simulation setting (N, M)[5] we randomly sample a directed acyclic graph on N nodes (each having 3 categories), from the Scale-Free (SF) model (Barabási and Albert [1999]), with number of edges chosen uniformly randomly from $[N, 5N]$. For each graph, we model the CPD of each node as a multinoulli distribution with Dirichlet priors having fixed parameter $\alpha = 2$ for all categories. This is done to conform with Assumption 3.2. This generates our causal Bayesian Network $\mathcal{G}$. We generate a set $\mathcal{B}$ of M samples using ancestral sampling on this network and use this as input for our algorithm. To create a mixture, we first choose an integer m uniformly randomly from the set $[4, 16]$ and use it as the number of interventions in the mixture. Then we iterate from 1 to m to build each intervention target of the mixture. First, we choose the size of the target by picking an integer r uniformly randomly from the set $\{0, \ldots, N\}$. Then we uniformly randomly choose an $r-$sized subset of $[N]$, defining variables in the target. For each of these variables, we choose a category uniformly randomly and remove it from consideration (to satisfy Assumption 3.1). From the remaining categories, we uniformly randomly select one for each variable in the target and use it to define the intervention. Finally, we generate m scalar weights for mixing coefficients such that they sum to 1. To make sure that these coefficients are not too small, we generate them with Dirichlet priors with

[5] N is number of nodes, M is number of samples

all parameter values fixed to 2. We create a set $\mathcal{B}_{mix}$ containing M samples from this mixture model and use it as input for the algorithm. We set parameters ϵ, δ required by our algorithm (see Appendix D) to 0.01 and $1/M$ respectively. The settings for N and M used in the experiments are $(N, M) \in \{4, 8, 12\} \times \{2^4, 2^5, \ldots, 2^{20}\}$ where $\times$ is the direct product of sets.

Results Discussion: Figure 1 presents four plots that demonstrate performance of our algorithm as sample size M varies in $\{2^4, 2^5, \ldots, 2^{20}\}$. We also vary the number of nodes N in $\{4, 8, 12\}$ and show separate plots for each N in each of the figures. The four plots in Figure 1, demonstrate four different accuracy metrics we describe in Appendix F. In Figure 1a, we plot the average recall of intervention targets as M increases. Recall for a single input instance is the number of intervention targets in the input that are identified in the output, as defined in Appendix F. Average recall is the average of this over all random instances generated in the simulation. We observe a general trend of increase in the recall as we increase the number of samples. Also, a relatively larger number of samples are required to achieve the same level of recall for mixtures generated from CBN with a large number of nodes as compared to smaller ones. This trend is expected as Algorithm 1 (Appendix D) estimates the intervention targets by sequentially adding nodes to them. Hence for larger-sized CBNs, the error accumulated is larger as compared to smaller ones.

In Figure 1b, we plot the average root-mean-squared error (RMSE) between the estimated and actual mixing coefficients. For each input, RMSE is calculated using the definition supplied in Appendix F. Then it is averaged over all the random input instances. We observe a fast decrease in the average RMSE as M increases. We also observe that the average RMSE is higher for higher N. This is also expected since for distributions on larger number of variables, more samples will be needed to estimate marginal probabilities accurately.

In Figure 1c, we plot average False-Positive RMSE (Section Appendix F) or FP-RMSE as M increases. For each input instance, FP-RMSE computes the RMSE in mixing proportions for components which are not present in actual target set but predicted by our algorithm. This is then averaged over all the random input instances. For each value of N, we observe a similar decreasing trend in this plot showing that incorrect targets in our output have very small mixing proportions (as sample size increases) and therefore even if they are present in the output their contribution is insignificant.

In Figure 1d, we plot average False-Negative RMSE (Section Appendix F) or FN-RMSE as M increases. For each input instance, FN-RMSE computes the RMSE in mixing proportions for components present in the actual target but not present in the output targets. This is then averaged over all the random input instances. Even though we observe a clear decreasing trend in this situation as well, the rate is much slower as N increases. This implies that the sample complexity of our algorithm is high and it might need too many samples to correctly identify the coefficients of targets present in the input. Reducing the sample complexity is an interesting research direction which we plan to pursue in a future work.

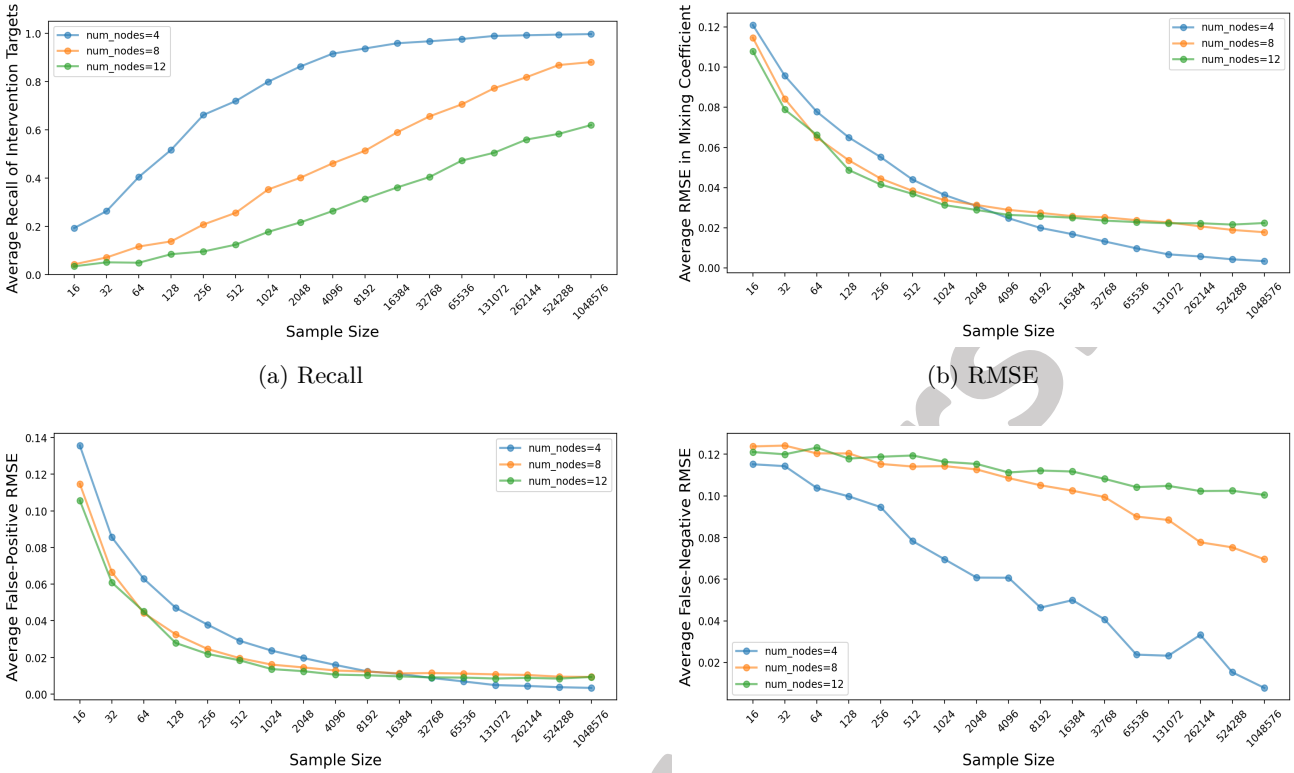
In Figure 2 (Appendix E), we demonstrate and compare performance of our Algorithm (as M, N increase), for CBNs generated using different random graph models (Scale-Free and Erdös-Rényi). We observe no significant difference in performance and make a conjecture that only high level graph parameters (such as number of nodes, edges, in-degree etc.) might be having an impact on performance and the topology (given these parameters) might not be that crucial.

To further understand the performance of our algorithm with respect to the number of nodes, in Figure 1 (Appendix E), we plot the Average Recall and Average RMSE as number of nodes varies from 4 to 32, for a fixed sample size of $\sim 10^6$. We observe that recall decreases and RMSE increases very quickly as number of nodes increase. Even though this is expected since error is accumulated as we successively add nodes and find new intervention targets, such performance for a very large sample size indicates bad dependence of sample complexity on the number of nodes. Improving this needs more exploration and is left for future work.

Limitations and Future Directions: The increasing trend in recall and decreasing trend in RMSE of mixing coefficients shows promise. But the current algorithm appears to be expensive in terms of sample complexity, especially for mixture generated from larger graphs as seen in Figures 1a, 1d and Figure 1 (Appendix E). Hence, it will be interesting to explore directions which could reduce sample complexity. We leave this for future work. Another limitation is the absence of baseline works to compare to. Since, ours is the first paper that proves identifiability of such mixtures and gives the first such algorithm, there are no prior works to compare against. In future, we plan to compare our algorithm on a related or downstream task that might have been explored in other works such as Thiesson et al. [1998], Squires et al. [2020], Jaber et al. [2020].

6 CONCLUSION

In this paper, we investigated the problem of identifying individual intervention targets from a mixture



(a) Recall

(b) RMSE

(c) False-Positive RMSE

(d) False-Negative RMSE

Figure 1: Performance of Algorithm 1 (Appendix D) as sample size and number of nodes increase

of interventions on a causal Bayesian Network. This problem is well motivated from the real-world scenario wherein experiments/interventions are accompanied by stochastic hidden off-target effects. We modeled this problem as a mixture of intervention distributions and constructed examples to show that, in general, it is impossible to identify all targets in it. Then, we proposed a mild *positivity* assumption on the underlying network and a very reasonable *exclusion* assumption on the intervention targets that can appear in the mixture distribution. Using these assumptions we proved that given access to the underlying CBN and the mixture distribution, there is a unique set of intervention targets that satisfies our *exclusion* assumption and also generates the mixture. Our uniqueness proof also provides an algorithm that uses access to the underlying distributions and efficiently identifies all the targets along with their coefficients in the mixture. In order to work with finitely many samples from the distributions, we created a small modification to our algorithm and validated it's performance using simulated experiments. We tested our algorithm and bench-marked its performance as the number of samples and nodes increased. As future work, we plan to investigate algorithms to recover targets in such mixtures using a smaller number of samples. Another interesting direction is to use limited access to the underlying CBN while recovering the targets. This can be very useful in situations where sufficient data or prior knowledge might not be available to pin down the CBN. Solving the identifiability problem when the CBN has unobserved confounders might be a good first step in this direction.

Acknowledgements

This work was done during an internship of the first author (Abhinav Kumar) under the guidance and mentorship of the second author (Gaurav Sinha), during the period January-August 2020. Abhinav Kumar would like to thank Adobe Research India for hosting him during this period and providing facilities for a fruitful and enjoyable research experience. Abhinav Kumar did this internship as part of his undergraduate thesis offered by his undergraduate institution BITS Pilani, Hyderabad. He would like to thank the institution for this opportunity. Gaurav Sinha would like to thank his mentees Aurghya Maiti, Pulkit Goel, Naman Poddar and Ayush Chauhan who were part of an older internship where the seed of this work was planted. The authors would like to thank anonymous reviewers for their very helpful comments which helped in greatly improving the presentation of this paper.

References

N.K. Aryal, A.R. Wasylishen, and G. Lozano. Crispr/cas9 can mediate high-efficiency off-target mutations in mice in vivo. *Cell Death and Disease*, 9, 2018. URL https://doi.org/10.1038/s41419-018-1146-0.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999. ISSN 0036-8075. doi: 10.1126/science.286.5439.509. URL https://science.sciencemag.org/content/286/5439/509.

Daniel Eaton and Kevin Murphy. Exact bayesian structure learning from uncertain interventions. In Marina Meila and Xiaotong Shen, editors, *PMLR*, volume 2 of *Proceedings of Machine Learning Research*, pages 107–114, San Juan, Puerto Rico, 21–24 Mar 2007. PMLR. URL http://proceedings.mlr.press/v2/eaton07a.html.

Yanfang Fu, Jennifer Foden, Cyd Khayter, Morgan Maeder, Deepak Reyon, J Joung, and Jeffry Sander. High-frequency off-target mutagenesis induced by crispr-cas nucleases in human cells. *Nature biotechnology*, 31, 06 2013. doi: 10.1038/nbt.2623.

Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *J. Mach. Learn. Res.*, 13(1):2409–2464, August 2012. ISSN 1532-4435.

Amin Jaber, Murat Kocaoglu, Karthikeyan Shanmugam, and Elias Bareinboim. Causal discovery from soft interventions with unknown targets: Characterization and learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9551–9561. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/6cd9313ed34ef58bad3fdd504355e72c-Paper.pdf.

Joris M. Mooij, Sara Magliacane, and Tom Claassen. Joint causal inference from multiple contexts. *Journal of Machine Learning Research*, 21(99):1–108, 2020. URL http://jmlr.org/papers/v21/17-123.html.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.

Joseph Ramsey, Peter Spirtes, and C Glymour. On meta-analyses of imaging data and the mixture of records. *NeuroImage*, 57:323–30, 07 2011. doi: 10.1016/j.neuroimage.2010.07.065.

Dominik Rothenhäusler, Christina Heinze, Jonas Peters, and Nicolai Meinshausen. Backshift: Learning causal cyclic graphs from unknown shift interventions. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1513–1521. Curran Associates, Inc., 2015.

Basil Saeed, Snigdha Panigrahi, and Caroline Uhler. Causal structure discovery from distributions arising from mixtures of dags. *CoRR*, abs/2001.11940, 2020. URL https://arxiv.org/abs/2001.11940.

P. Spirtes. Conditional independence in directed cyclic graphical models for feedback. In *Carnegie Mellon University*, 1994.

Peter Spirtes. Introduction to causal inference. *Journal of Machine Learning Research*, 11(54):1643–1662, 2010. URL http://jmlr.org/papers/v11/spirtes10a.html.

Chandler Squires, Yuhao Wang, and Caroline Uhler. Permutation-based causal structure learning with unknown intervention targets. In Jonas Peters and David Sontag, editors, *Proceedings of Machine Learning Research*, volume 124, pages 1039–1048, Virtual, 03–06 Aug 2020. PMLR. URL http://proceedings.mlr.press/v124/squires20a.html.

Eric V. Strobl. Improved causal discovery from longitudinal data using a mixture of dags. In Thuc Duy Le, Jiuyong Li, Kun Zhang, Emre Kıcıman Peng Cui, and Aapo Hyvärinen, editors, *PMLR*, volume 104 of *Proceedings of Machine Learning Research*, pages 100–133, Anchorage, Alaska, USA, 05 Aug 2019a. URL http://proceedings.mlr.press/v104/strobl19a.html.

Eric V. Strobl. The global markov property for a mixture of dags. *arXiv: Statistics Theory*, 2019b.

Bo Thiesson, Christopher Meek, David Maxwell Chickering, and David Heckerman. Learning mixtures of dag models. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, page 504–513, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 155860555X.

Xiaoling Wang, Yebo Wang, Xiwei Wu, Jinhui Wang, Yingjia Wang, Zhaojun Qiu, Tammy Chang, He Huang, Ren-Jang Lin, and Jiing-Kuan Yee. Unbiased detection of off-target cleavage by crispr-cas9 and talens using integrase-defective lentiviral vectors. *Nature biotechnology*, 33, 01 2015. doi: 10.1038/nbt.3127.