Disentangling Mixtures of Unknown Causal Interventions Supplementary Material

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

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

A PROOFS FROM SECTION 4

In this section, we provide missing proofs of the lemmas stated in Section 4.

A.1 PROOF OF LEMMA 4.1

Note that we have assumed $a_i > 0$ for all $i \in [k]$. We iterate from $i = 1$ to $k-1$ and apply the following row transformations on our matrix.

$$R_i \mapsto R_i - (\frac{a_i}{a_k})R_k$$

This results in the following linear system.

$$\begin{bmatrix} c & 0 & . & -\frac{a_1}{a_k}c \\ 0 & c & . & -\frac{a_2}{a_k}c \\ . & . & . & . \\ . & . & . & . \\ -a_k & -a_k & . & c-a_k \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ x_k \end{bmatrix} = \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ . \\ . \\ \tilde{b}_k \end{bmatrix}$$

where $\tilde{b}_i = (b_i - \frac{a_i}{a_k}b_k)$ for all $i \in [k-1]$ and $\tilde{b}_k = b_k$. Since $c > 0$, this matrix is easily seen to have rank $\geq k-1$. Using $c = \sum_{i=1}^{k} a_i$ we can easily check that the last row R_k is $\frac{-a_k}{c}(R_1 + \ldots + R_{k-1})$, implying that it has rank $k-1$. Since the system is assumed to have at least one solution, it actually has infinitely many solutions. The null space of this matrix is the one dimensional space spanned by $\boldsymbol{w} = (\frac{a_1}{a_k}, \ldots, \frac{a_k}{a_k})^T$ which has all positive entries since $a_i > 0, i \in [k]$. Assume there are two distinct solutions $\boldsymbol{u} = (u_1, \ldots, u_k)^T$ and $\boldsymbol{v} = (v_1, \ldots, v_k)^T$ in $\mathbb{R}^k_{\geq 0}$ such that both have at least one of their co-ordinates 0, then $\boldsymbol{u} - \boldsymbol{v}$ belongs to the null space i.e. $\boldsymbol{u} - \boldsymbol{v} = \lambda \boldsymbol{w}$ for some non-zero scalar λ. If the same co-ordinate of $\boldsymbol{u}, \boldsymbol{v}$ are 0 i.e. for some $i \in [k]$, $u_i = v_i = 0$, then since $\frac{a_i}{a_k}$ is non-zero $\lambda = 0 \Rightarrow \boldsymbol{u} = \boldsymbol{v}$, a contradiction. If different co-ordinates of $\boldsymbol{u}, \boldsymbol{v}$ are 0, say $u_1 = v_2 = 0$, then since $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^k_{\geq 0}$, $u_1 - v_1$ is negative and $u_2 - v_2$ is positive. This is not possible since both these quantities should have the same sign as λ, as all co-ordinates of $\boldsymbol{w}$ are strictly positive. Therefore we arrive at a contradiction and there is a unique solution.

Having proved this uniqueness, finding the solution is easy. We perform the above-mentioned row transformations and obtain the general solution. Then for each $i \in [k]$, we set $x_i = 0$ and try to solve for the other variables $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k \in \mathbb{R}_{\geq 0}$. By the above argument, we will get a valid solution for only one such i, which we return as the unique solution. Clearly it takes $k^{O(1)}$ time.

A.2 PROOF OF LEMMA 4.2

1. Since $V_1 \prec \ldots \prec V_{N+1}$ is a topological order, marginalizing over V_{N+1} in the factorization $\mathbb{P}(\boldsymbol{V}) = \prod_{i=1}^{N+1} \mathbb{P}(V_i|\boldsymbol{pa}(V_i))$ will give the factorization $\mathbb{P}(\boldsymbol{V}_N) = \prod_{i=1}^{N} \mathbb{P}(V_i|\boldsymbol{pa}(V_i))$ which is the factorization over $\mathcal{G}_N$.

2. Let $\mathcal{T} = \{(\pi_1, \boldsymbol{t}_1), \ldots, (\pi_m, \boldsymbol{t}_m)\}$ be a set of intervention tuples generating $\mathbb{P}_{mix}(\boldsymbol{V})$ and $C_{V_{N+1}} = \{v^1, \ldots, v^k\}$. Marginalizing with respect to V_{N+1} for a single target $\boldsymbol{t}_i, i \in [m]$ looks like,

$$\sum_{l=1}^{k} \mathbb{P}_{\boldsymbol{t}_i}(\boldsymbol{V}_N, v^l) = \begin{cases} \mathbb{P}_{\boldsymbol{t}_i}(\boldsymbol{V}_N) & : V_{N+1} \notin T_i \\ \mathbb{P}_{\boldsymbol{t}_i \setminus \{v^j\}}(\boldsymbol{V}_N) & : v^j \in \boldsymbol{t}_i \end{cases} \tag{1}$$

   Applying this marginalization to Equation 2, i.e., $\mathbb{P}_{mix}(\boldsymbol{V}) = \sum_{i=1}^{m} \pi_i \mathbb{P}_{\boldsymbol{t}_i}(\boldsymbol{V})$, gives a convex linear combination of different $\mathbb{P}_{\boldsymbol{s}}(\boldsymbol{V}_N)$, where $\boldsymbol{s}$ are values of some set of variables $\boldsymbol{S} \subset \boldsymbol{V}_N$, implying that $\mathbb{P}_{mix}(\boldsymbol{V}_N)$ is a mixture of interventions on $\mathcal{G}_N$.

3. This is straight-forward. To query $\mathbb{P}(v_1, \ldots, v_N)$, we query $\mathbb{P}(v_1, \ldots, v_N, v_{N+1})$ for all $v_{N+1} \in C_{V_{N+1}}$ and sum them up. The same can be done to create access for $\mathbb{P}_{mix}(\boldsymbol{V}_N)$. Since we are summing at most k_{max} terms, in $O(k_{max})$ time we can simulate access to both $\mathbb{P}(\boldsymbol{V}_N)$, $\mathbb{P}_{mix}(\boldsymbol{V}_N)$.

A.3 PROOF OF LEMMA 4.3

This follows by the marginalization equation (Equation 1 in Appendix A.2).

A.4 PROOF OF LEMMA 4.5

Let $\boldsymbol{s} \in \mathcal{S}_r$ where $r > i$. This means that $\boldsymbol{s}$ is either $\boldsymbol{s}_r$ or $\boldsymbol{s}_r \cup \{v\}$ for some $v \in C_{V_{N+1}}$. We show that $\mathbb{P}_{\boldsymbol{s}_r}(\boldsymbol{v}_{i,l}) = \mathbb{P}_{\boldsymbol{s}_r}(\boldsymbol{s}_i \cup \boldsymbol{s}_{-i} \cup \{v^l\}) = 0$ for all $v^l \in C_{V_{N+1}}$. The proof for $\boldsymbol{s} = \boldsymbol{s}_r \cup \{v\}$ is identical. Since $i < r$, we get that $\boldsymbol{s}_r \not\subseteq \boldsymbol{s}_i$. Now there are two cases, either the set of variables $\boldsymbol{S}_r \subseteq \boldsymbol{S}_i$ or $\boldsymbol{S}_r \not\subseteq \boldsymbol{S}_i$. In the first case, since $\boldsymbol{s}_r \not\subseteq \boldsymbol{s}_i$ we get that there is some variable $V_j \in \boldsymbol{S}_r, \boldsymbol{S}_i$ such that different values v_j^r and v_j^i belong to $\boldsymbol{s}_r$ and $\boldsymbol{s}_i$ respectively implying that $\mathbb{P}_{\boldsymbol{s}_r}(\boldsymbol{s}_i \cup \boldsymbol{s}_{-i} \cup \{v\}) = 0$. In the second case, there is some variable $V_j \in \boldsymbol{S}_r$ ($j \in [N]$) that is not in $\boldsymbol{S}_i$. Note that the "missing value" $\bar{v}_j \in C_{V_j}$ (i.e. the one that is missing from all targets $\boldsymbol{s}_j, j \in [q]$) belongs to $\boldsymbol{s}_{-i}$ (since $V_j \notin \boldsymbol{S}_i$) but it cannot belong to $\boldsymbol{s}_r$ (since it is missing from all $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_q$) $\Rightarrow \mathbb{P}_{\boldsymbol{s}_r}(\boldsymbol{s}_i \cup \boldsymbol{s}_{-i} \cup \{v\}) = 0$.

A.5 PROOF OF LEMMA 4.6

Let $\boldsymbol{s} \in \mathcal{S}_i$. Note that $\boldsymbol{s}$ is either $\boldsymbol{s}_i$ or $\boldsymbol{s}_i \cup \{v\}$ for some $v \in C_{V_{N+1}}$. Using Equation 1 we get that for all $\boldsymbol{s} \in \mathcal{S}_i$, marginalization of V_{N+1} in $\pi_{\boldsymbol{s}} \mathbb{P}_{\boldsymbol{s}}(\boldsymbol{V})$ gives $\pi_{\boldsymbol{s}} \mathbb{P}_{\boldsymbol{s}_i}(\boldsymbol{V}_N)$. Marginalizing V_{N+1} in Equation 3, converts the left hand side to $\mathbb{P}_{mix}(\boldsymbol{V}_N)$ and right-hand side to $\sum_{i=1}^{q}(\sum_{\boldsymbol{s} \in \mathcal{S}_i} \pi_{\boldsymbol{s}}) \mathbb{P}_{\boldsymbol{s}_i}(\boldsymbol{V}_N)$ giving a set of intervention tuples that generates $\mathbb{P}_{mix}(\boldsymbol{V}_N)$. By the inductive hypothesis, $\mathbb{P}_{mix}(\boldsymbol{V}_N)$ is generated by the unique set of intervention tuples $\mathcal{S}$ satisfying Assumption 3.1. Since the intervention targets in $\mathcal{S}$ and the ones in the set of intervention tuples we just obtained are the same, using uniqueness of $\mathcal{S}$ we get that $\mu_i = \sum_{\boldsymbol{s} \in \mathcal{S}_i} \pi_{\boldsymbol{s}}$.

A.6 PROOF OF LEMMA 4.7

Note that since V_{N+1} is the last node in the topological order, using the definition of interventions, we can conclude that,

$$\mathbb{P}_{\boldsymbol{s}_i \cup \{v^j\}}(\boldsymbol{s}_i \cup \boldsymbol{s}_{-i} \cup \{v^l\}) = \mathbb{P}_{\boldsymbol{s}_i}(\boldsymbol{s}_i \cup \boldsymbol{s}_{-i}) \delta_{v^j, v^l}$$

where $v^l, v^j \in C_{V_{N+1}} = \{v^1, \ldots, v^k\}$. Recall that $\boldsymbol{v}_{i,l} = \boldsymbol{s}_i \cup \boldsymbol{s}_{-i} \cup \{v^l\}$. Now, on substituting for Δ using Equation 4 into Equation 5, we obtain,

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

$$\sum_{j \in [k]} \pi_{\boldsymbol{s}_i \cup \{v^j\}} \Big(\mathbb{P}_{\boldsymbol{s}_i}(\boldsymbol{s}_i \cup \boldsymbol{s}_{-i}) \delta_{v^j, v^l} - \mathbb{P}_{\boldsymbol{s}_i}(\boldsymbol{v}_{i,l}) \Big)$$

Note that all the unknown variables are on the right-hand side of this equation. Varying $l \in [k]$, gives us a linear system of equations satisfied by scalars $\pi_{\boldsymbol{s}_i \cup \{v^l\}}$.

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

In the above system, we have renamed the known values as follows. For $l \in [k]$, denote

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

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

$$c = \mathbb{P}_{\boldsymbol{s}_i}(\boldsymbol{s}_i \cup \boldsymbol{s}_{-i})$$

All a_l's are probabilities from interventional distributions and can be computed as product of conditional probabilities. Thus, by Assumption 3.2, $a_l > 0$ for all $l \in [k]$. It's easy to see that $c = \sum_{l \in [k]} a_l$, by the sum rule of probability. By statement of Lemma 4.3, for each $i \in [q]$ and $l \in [k]$, $\pi_{\boldsymbol{s}_i \cup \{v^l\}} \geq 0$. Since we are only considering set of intervention tuples which satisfy Assumption 3.1, there is some $l \in [k]$ such that $\pi_{\boldsymbol{s}_i \cup \{v^l\}} = 0$. On constraining the variables $x_1, \ldots, x_k$ in the above system to these conditions (i.e. $x_i \geq 0$ for all $i \in [q]$ and $x_i = 0$ for some $i \in [q]$), by Lemma 4.1 we are guaranteed a unique solution. Therefore there is a unique tuple $(\pi_{\boldsymbol{s}_i \cup \{v^1\}}, \ldots, \pi_{\boldsymbol{s}_i \cup \{v^k\}})$ satisfying these requirements $\Rightarrow$ Equation 5 has a unique solution which is easily computed in $k^{O(1)}$ time using the technique described in proof of Lemma 4.1.

B NON-NECESSITY OF ASSUMPTION 3.2

With the help of an example we argue that Assumption 3.2 is not necessary in Theorem 3.1.

Example B.1. Consider a causal Bayesian Network

$$V_1 \to V_2$$

defined over two binary variable $\boldsymbol{V} = \{V_1, V_2\}$ with $C_{V_i} = \{0, 1\}$, $i \in [2]$. Further, define CPDs, $\mathbb{P}(V_1 = 1) = 0.5$, $\mathbb{P}(V_2 = 1 | V_1 = 0) = 0.5$, and $\mathbb{P}(V_2 = 1 | V_1 = 1) = 0$. Clearly $\mathbb{P}(V_2 = 1, V_1 = 1) = 0$ implying that this CBN doesn't satisfy Assumption 3.2.

Let $\mathbb{P}_{mix}(\boldsymbol{V})$ be a mixture distribution defined as

$$\frac{1}{2}\mathbb{P}(\boldsymbol{V} | do(V_1 = 0)) + \frac{1}{2}\mathbb{P}(\boldsymbol{V} | do(V_1 = 0, V_2 = 0))$$

This mixture satisfies Assumption 3.1. Our algorithm first marginalizes on V_2 and tries to find the unique set of intervention targets for $\mathbb{P}_{mix}(V_1)$. For this sub-problem, all steps of Algorithm 1 go through (since the distribution $\mathbb{P}(V_1)$ satisfies positivity), and the correct components get identified. Note that, for this sub-problem the algorithm identifies that $\mathbb{P}_{mix}(V_1) = \mathbb{P}(V_1 | do(V_1 = 0))$. Now it tries to lift this computed target $(V_1 = 0)$ to targets for the full mixture $\mathbb{P}_{mix}(\boldsymbol{V})$.

Since the algorithm does not try to lift the target $(V_1 = 1)$ (as it was not found as a target for $\mathbb{P}_{mix}(V_1)$), it does not require $\mathbb{P}(V_2|V_1 = 1)$ to be non-zero. This can be easily checked in the lifting process described in Section 4.2.1. We do not repeat the steps of our algorithm here and encourage the reader to work through the lifting steps outlined in Section 4.2.1 and obtain unique solutions proving our point that Assumption 3.2 is not necessary and can be weakened.

C WORKED-OUT EXAMPLES

In this section, we illustrate the workings of Algorithm 1 (in the main paper) using two worked-out examples. Example C.1 is simpler and uses a mixture distribution on a CBN with just two nodes. It does not really require all of the crucial ideas from the lifting procedure described in Section 4.2.1. However, we believe it is important since it gives a good broad understanding of the entire algorithm. Example C.2 is complicated enough (using a mixture distribution on a CBN with three nodes) to highlight some of the key novelties of our lifting procedure in Section 4.2.1. We urge the reader to first work through Example C.1 and then through Example C.2 to get a full understanding of the critical ideas that make our proof of Theorem 3.1 work.

Example C.1. Consider a causal Bayesian Network

$$V_1 \to V_2$$

defined over two binary variable $\boldsymbol{V} = \{V_1, V_2\}$ with $C_{V_i} = \{0, 1\}$, $i \in [2]$. Further, define CPDs, $\mathbb{P}(V_1 = 1) = 0.5$, $\mathbb{P}(V_2 = 1|V_1) = 0.5$ for both $V_1 = 0$ and $V_1 = 1$. Clearly, this CBN satisfies Assumption 3.2. Let $\mathbb{P}_{mix}(\boldsymbol{V})$ be a mixture distribution defined as

$$\frac{1}{2}\mathbb{P}(\boldsymbol{V}|do(V_1 = 0)) + \frac{1}{2}\mathbb{P}(\boldsymbol{V}|do(V_1 = 0, V_2 = 0))$$

This mixture satisfies Assumption 3.1. On marginalizing variable V_2, the mixture becomes $\mathbb{P}_{mix}(V_1) = \mathbb{P}(V_1|do(V_1 = 0))$ which also satisfies Assumption 3.1, as a mixture of interventions on the CBN comprising of the single variable V_1. A general mixture distribution on variable V_1 looks like,

$$\pi_0\mathbb{P}(V_1|do(V_1 = 0)) + \pi_1\mathbb{P}(V_1|do(V_1 = 0)) + (1 - \pi_0 - \pi_1)\mathbb{P}(V_1)$$

Varying V_1 in $\{0, 1\}$ gives the system of equations,

$$\begin{bmatrix} 1 - 0.5 & -0.5 \\ -0.5 & 1 - 0.5 \end{bmatrix} \begin{bmatrix} \pi_0 \\ \pi_1 \end{bmatrix} = \begin{bmatrix} \mathbb{P}_{mix}(V_1 = 0) - 0.5 \\ \mathbb{P}_{mix}(V_1 = 1) - 0.5 \end{bmatrix}$$

Lemma 4.1 highlights why such a system has a unique non-negative solution if we assume that at least one of π_0, π_1 is 0, i.e. the set of intervention tuples we are trying to construct satisfy Assumption 3.1. Lemma 4.1 also provides an efficient algorithm to find the unique solution giving $\mathbb{P}_{mix}(V_1) = \mathbb{P}(V_1|do(V_1 = 0))$. Now we have one intervention target $(V_1 = 0)$ at hand, which we will try to lift. Note that, the possible lifts of such a target are $(V_1 = 0), (V_1 = 0, V_2 = 0), (V_1 = 0, V_2 = 1)$. Our next step will search within this space of targets and try to complete the construction. A general solution for mixtures within this space would look like

$$\mu_0\mathbb{P}(\boldsymbol{V}|do(V_1 = 0)) + \mu_1\mathbb{P}(\boldsymbol{V}|do(V_1 = 0, V_2 = 0)) + \mu_2\mathbb{P}(\boldsymbol{V}|do(V_1 = 0, V_2 = 1))$$

where $\mu_i \geq 0, i \in \{0, 1, 2\}$. So we want to find μ_0, μ_1, μ_2 such that the above general solution becomes equal to $\mathbb{P}_{mix}(\boldsymbol{V})$. Since we already know that $\mathbb{P}_{mix}(V_1) = \mathbb{P}(V_1|do(V_1 = 0))$, marginalizing on V_2 implies that $\mu_0 + \mu_1 + \mu_2 = 1$. We substitute $\mu_0 = 1 - \mu_1 - \mu_2$, equate the above general mixture to $\mathbb{P}_{mix}(\boldsymbol{V})$, and re-arrange terms to get,

$$\begin{aligned}\mathbb{P}_{mix}(\boldsymbol{V}) - \mathbb{P}(\boldsymbol{V}|do(V_1 = 0)) = \mu_1(\mathbb{P}(\boldsymbol{V}|do(V_1 = 0, V_2 = 0)) - \mathbb{P}(\boldsymbol{V}|do(V_1 = 0))) \\ + \mu_2(\mathbb{P}(\boldsymbol{V}|do(V_1 = 0, V_2 = 1)) - \mathbb{P}(\boldsymbol{V}|do(V_1 = 0)))\end{aligned}$$

We evaluate this mixture at settings $V_1 = 0, V_2 = 0$, and $V_1 = 0, V_2 = 1$, to get a system of linear equations in (μ_1, μ_2),

$$\begin{bmatrix} 1 - 0.5 & -0.5 \\ -0.5 & 1 - 0.5 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} = \begin{bmatrix} \mathbb{P}_{mix}(0, 0) - 0.5 \\ \mathbb{P}_{mix}(0, 1) - 0.5 \end{bmatrix}$$

Again, since we are solving for a set of intervention tuples satisfying Assumption 3.1, one of μ_1, μ_2 would be 0. This used with Lemma 4.1 gives the unique solution $\mu_1 = 0.5, \pi_2 = 0 \Rightarrow \mu_0 = 0.5$, thereby identifying the correct set of intervention tuples.

We would like to note that the example we illustrated above is rather simple and does not capture some main non-trivial aspects of our Algorithm. However, we think it is important as a warm up exercise. A more involved example that would bring out the crucial ideas of our proof is presented below in Example C.2.

Example C.2. Consider a causal Bayesian Network defined over three binary variables $\boldsymbol{V} = \{V_1, V_2, V_3\}$ taking values in $C_{V_i} = \{0, 1\}$, $i \in [3]$, with $\boldsymbol{pa}(V_1) = \varnothing$, $\boldsymbol{pa}(V_2) = \{V_1\}$ and $\boldsymbol{pa}(V_3) = \{V_1, V_2\}$. Let CPDs be such that Assumption 3.2 is satisfied. Consider a mixture distribution

$$\mathbb{P}_{mix}(\boldsymbol{V}) = \mu_0 \mathbb{P}(\boldsymbol{V}|do(V_1 = 0)) + \mu_1 \mathbb{P}(\boldsymbol{V}|do(V_1 = 0, V_2 = 0)) + \mu_2 \mathbb{P}(\boldsymbol{V}|do(V_1 = 0, V_2 = 0, V_3 = 0))$$

with positive scalars μ_0, μ_1, μ_2 satisfying $\mu_0 + \mu_1 + \mu_2 = 1$. Marginalizing on V_3, gives,

$$\mathbb{P}_{mix}(V_1, V_2) = \pi_0 \mathbb{P}(V_1, V_2|do(V_1 = 0)) + \pi_1 \mathbb{P}(V_1, V_2|do(V_1 = 0, V_2 = 0))$$

Our inductive hypothesis assumes that this mixture on smaller number of nodes is generated by a unique set of intervention tuples that satisfies Assumption 3.1 and also that this set can be efficiently computed. This will give us access to the two scalars π_0, π_1 and to the two targets $(V_1 = 0)$ and $(V_1 = 0, V_2 = 0)$ (we call them the currently computed targets). We need to lift these targets to targets for the original mixture distribution like we did in Example C.1. However, the situation is not as simple here. Note that $(V_1 = 0)$ can be lifted to one of the three targets $(V_1 = 0), (V_1 = 0, V_3 = 0), (V_1 = 0, V_3 = 1)$. Similarly $(V_1 = 0, V_2 = 0)$ can be lifted to one of the three targets $(V_1 = 0, V_2 = 0), (V_1 = 0, V_2 = 0, V_3 = 0), (V_1 = 0, V_2 = 0, V_3 = 1)$. So there are 6 possible targets in the original mixture and therefore a general solution for our mixture can be written using 6 new variables (say $\delta_0, \delta_1, \delta_2, \delta_3, \delta_4, \delta_5$) such that,

$$\begin{aligned}\mathbb{P}_{mix}(\boldsymbol{V}) = \delta_0 \mathbb{P}(\boldsymbol{V}|do(V_1 = 0)) + \delta_1 \mathbb{P}(\boldsymbol{V}|do(V_1 = 0, V_3 = 0)) + \delta_2 \mathbb{P}(\boldsymbol{V}|do(V_1 = 0, V_3 = 1)) \\ + \delta_3 \mathbb{P}(\boldsymbol{V}|do(V_1 = 0, V_2 = 0)) + \delta_4 \mathbb{P}(\boldsymbol{V}|do(V_1 = 0, V_2 = 0, V_3 = 0)) + \delta_5 \mathbb{P}(\boldsymbol{V}|do(V_1 = 0, V_2 = 0, V_3 = 1))\end{aligned} \tag{2}$$

where all δ_i are non-negative By marginalizing on V_3, and using the solution we got from the inductive hypothesis (like we did in Example C.1), we can show that,

$$\pi_0 = \delta_0 + \delta_1 + \delta_2, \quad \pi_1 = \delta_3 + \delta_4 + \delta_5$$

Now the two main non trivial ingredients needed from here are:

- Deciding the order in which the currently computed targets should be lifted, and
- Deciding the settings for $\boldsymbol{V}$ that would give linear systems where we can argue about unique solutions like in Example C.1.

For the first one, we lift the currently computerd targets in an order which does not violate set inclusion for these targets, i.e. we first lift $(V_1 = 0)$ and then lift $(V_1 = 0, V_2 = 0)$. This can be done by considering any extension of the set inclusion partial order on these targets. Then for lifting the target $(V_1 = 0)$, we choose to evaluate on the settings $\boldsymbol{v}_1 = (V_1 = 0, V_2 = 1, V_3 = 0), \boldsymbol{v}_2 = (V_1 = 0, V_2 = 1, V_3 = 1)$. Here, we pick the value of V_2 that is missing from the currently computed target under consideration i.e. $(V_1 = 0)$. There will always be one such missing value (it follows from Assumption 3.1). Evaluating on these settings simplifies our equation drastically. For $l \in [2]$, we get,

$$\mathbb{P}_{mix}(\boldsymbol{v}_l) = \delta_0 \mathbb{P}(\boldsymbol{v}_l|do(V_1 = 0)) + \delta_1 \mathbb{P}(\boldsymbol{v}_l|do(V_1 = 0, V_3 = 0)) + \delta_2 \mathbb{P}(\boldsymbol{v}_l|do(V_1 = 0, V_3 = 1))$$

Basically all possible lifts of the other currently computed target i.e. $(V_1 = 0, V_2 = 0)$ vanish and we have a much simpler system of equations at hand. From here the solution follows exactly like the previous example. We substitute $\delta_0 = \pi_0 - \delta_1 - \delta_2$ and rearrange to get a linear system in 2 equations and 2 variables δ_1, δ_2. Similar to the argument made in Example C.1, at least one of δ_1, δ_2 will be 0 and therefore this system has a unique

solution (using Lemma 4.1) giving values of $\delta_0, \delta_1, \delta_2$. These can then be substituted back in Equation 2 reducing the number of variables to 3 (i.e. $\delta_3, \delta_4, \delta_5$). Again we substitute $\delta_3 = \pi_1 - \delta_4 - \delta_5$ and reduce the equation to just two unknowns. Finally by using settings $\boldsymbol{v}_1 = (V_1 = 0, V_2 = 0, V_3 = 0), \boldsymbol{v}_2 = (V_1 = 0, V_2 = 0, V_3 = 1)$ in Equation 2 we will be left with 2 equations in 2 variables. Exactly like our argument for lifting of $(V_1 = 0)$ (i.e. using Lemma 4.1), we can show that, when at least one of δ_4, δ_5 is 0, this system has a unique solution as well. Lemma 4.1 would efficiently give us the values of δ_4, δ_5 and therefore of δ_3. Thus, we uniquely identify a set of intervention tuples that satisfies Assumption 3.1 and generates $\mathbb{P}_{mix}(\boldsymbol{V})$.

D FINITE SAMPLE ALGORITHM

In a real world scenario, we will only have finitely many samples from the distributions $\mathbb{P}(\boldsymbol{V})$ and $\mathbb{P}_{mix}(\boldsymbol{V})$. We still assume access to the underlying causal graph G (the CPDs might not be known though). In this situation, we modify Algorithm 1 (from the main paper) slightly to make it work with finitely many samples. The resulting algorithm is presented in Algorithm 1 (Appendix D). Let the sets containing the samples be $\mathcal{B} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_M\}$ where $\boldsymbol{b}_i \sim \mathbb{P}(\boldsymbol{V})$ and $\mathcal{B}_{mix} = \{\boldsymbol{b}_1^{mix}, \ldots, \boldsymbol{b}_M^{mix}\}$ where $\boldsymbol{b}_j^{mix} \sim \mathbb{P}_{mix}(\boldsymbol{V})$. As a preprocessing step, we estimate the distributions $\mathbb{P}(\boldsymbol{V})$ and $\mathbb{P}_{mix}(\boldsymbol{V})$ as $\hat{\mathbb{P}}(\boldsymbol{V})$ and $\hat{\mathbb{P}}_{mix}(\boldsymbol{V})$ (respectively) using samples in $\mathcal{B}$ and $\mathcal{B}_{mix}$ respectively. $\hat{\mathbb{P}}(\boldsymbol{V})$ is estimated by estimating all the CPDs using maximum likelihood estimation (MLE). In our implementation, we use a function from the pgmpy library (Ankan and Panda [2015]), to compute these MLE estimates. We enforce Assumption 3.2 on $\hat{\mathbb{P}}(\boldsymbol{V})$, by enforcing it on all it's CPDs, using a small positive parameter δ (chosen by us). When $\hat{\mathbb{P}}(v_i|\boldsymbol{pa}(v_i)) = 0$ for some v_i and some setting of parents $\boldsymbol{pa}(v_i)$, we update,

$$\hat{\mathbb{P}}(V_i|\boldsymbol{pa}(v_i)) \leftarrow \hat{\mathbb{P}}(V_i|\boldsymbol{pa}(v_i)) + \delta$$

for all values of V_i, and then re-normalize to make it a probability distribution again. Marginal $\hat{\mathbb{P}}_{mix}(\boldsymbol{V} = \boldsymbol{v})$ is calculated using relative frequency of the occurence of $\boldsymbol{V} = \boldsymbol{v}$ in the samples inside $\mathcal{B}_{mix}$. These estimated distributions are then used as inputs in Algorithm 1 (Appendix D). We use another small positive parameter ϵ as input to Algorithm 1 (Appendix D) which prunes each recovered set of intervention tuples computed during the algorithm, by only keeping mixing coefficients greater that ϵ. It's easy to see that the time complexity of Algorithm 1 (Appendix D), including the estimation of probabilities from samples is:

$$\left(\frac{Nk_{max}^d M}{\epsilon}\right)^{O(1)}$$

Here N is number of nodes in the causal graph G, d is the maximum in-degree of any node in G, k_{max} is maximum number of values that any node in G can take and M is the number of samples present in $\mathcal{B}$ and $\mathcal{B}_{mix}$.

Remark. *Since our algorithm's run-time depends on ϵ, we need to carefully select it's value. Setting it too small could increase the run time whereas setting it too big could lead to wrongfully pruning intervention targets (with significant mixing proportions) present in the mixture.*

E ADDITIONAL SIMULATIONS

E.1 EFFECT OF GRAPH SIZE

Figure 1 (Appendix E) shows the variation of performance of Algorithm 1 (Appendix D) keeping the number of samples fixed at $\sim 10^6$. We observe that recall decreases and root-mean-squared error in mixing coefficient increases very quickly as the number of nodes increases in the graph. 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.

E.2 EFFECT OF GRAPH TYPE

In Figure 2 (Appendix E), we demonstrate performance of Algorithm 1 (Appendix D) for CBNs generated from two different family of random graphs (Erdös-Rényi (ER) and Scale-Free (SF)). We observe no significant

Algorithm 1: DISENTANGLE-FINITE

input : $\boldsymbol{V}$, Causal Graph G, $\hat{\mathbb{P}}(\boldsymbol{V}), \hat{\mathbb{P}}_{mix}(\boldsymbol{V}), \epsilon$
output : Set of intervention tuples $\mathcal{T}$

1. When $|\boldsymbol{V}| = 1$, setup the linear system in Equation 1 (say $\boldsymbol{Ax} = \boldsymbol{b}$) using the estimated distributions. Similar to the technique described in Lemma 4.1, set one variable to 0 at a time giving solution $(\pi_1, \ldots, \pi_k)$ corresponding to targets $(\boldsymbol{t}_1, \ldots, \boldsymbol{t}_k)$ as described in Section 4.1. For every variable that is set to 0, create a set $\mathcal{T} = \{(\boldsymbol{t}_i, \pi_i) : i \in [k]\}$ containing the solution. For every such $\mathcal{T}$, iterate through the tuples $(\boldsymbol{t}_i, \pi_i)$ in it. If some $\pi_i < 0$, set $\pi_i \leftarrow 0$. Compute the score $r(\mathcal{T}) = \|\boldsymbol{A\pi} - \boldsymbol{b}\|^2$, where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k)$. Next, select $\mathcal{T}$ with the smallest value of $r(\mathcal{T})$. For this selected $\mathcal{T}$, check if $1 - \sum_{i=1}^{k} \pi_i < \epsilon$. If yes, renormalize $\pi_i \leftarrow \pi_i / (\sum_{i=1}^{k} \pi_i)$. If no, add the tuple $(\boldsymbol{t}_0, 1 - \sum_{i=1}^{k} \pi_i)$ to $\mathcal{T}$. Only keep the tuples with strictly positive mixing coefficients i.e. $\mathcal{T} \leftarrow \{(\boldsymbol{t}_i, \pi_i) \in \mathcal{T} : \pi_i > 0\}$. **return** $\mathcal{T}$.
2. Let $V_1 \prec \ldots \prec V_{N+1}$ denote a topological order in G. Marginalize on V_{N+1} to create access to $\hat{\mathbb{P}}_{mix}(\boldsymbol{V}_N)$ and $\hat{\mathbb{P}}(\boldsymbol{V}_N)$ where $\boldsymbol{V}_N = (V_1, \ldots, V_N)$. Construct $G_N = G \setminus \{V_{N+1}\}$. Recursively call this algorithm with inputs G_N, $\hat{\mathbb{P}}(\boldsymbol{V}_N)$, $\hat{\mathbb{P}}_{mix}(\boldsymbol{V}_N)$, and obtain a set of intervention tuples $\mathcal{S} = \{(\boldsymbol{s}_1, \mu_1), \ldots, (\boldsymbol{s}_q, \mu_q)\}$. 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 all $i \in [N]$, by inspecting $\boldsymbol{s}_j$, 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 (say $\boldsymbol{Ax} = \boldsymbol{b}$) described in Equation 5. Similar to the technique described in Lemma 4.7 (which in turn uses Lemma 4.1), set one variable to 0 at a time giving solution $(\pi_{\boldsymbol{s}_i \cup \{v^1\}}, \ldots, \pi_{\boldsymbol{s}_i \cup \{v^k\}})$ corresponding to targets $(\boldsymbol{s}_i \cup \{v^1\}, \ldots, \boldsymbol{s}_i \cup \{v^k\})$ as described in Section 4.2. For every variable that is set to 0, create a set $\mathcal{T} = \{(\boldsymbol{s}_i \cup \{v^l\}, \pi_{\boldsymbol{s}_i \cup \{v^l\}}) : l \in [k]\}$ containing the solution. For every such $\mathcal{T}$, iterate through the tuples $(\boldsymbol{s}_i \cup \{v^l\}, \pi_{\boldsymbol{s}_i \cup \{v^l\}})$ in it. If some $\pi_{\boldsymbol{s}_i \cup \{v^l\}} < 0$, set $\pi_{\boldsymbol{s}_i \cup \{v^l\}} \leftarrow 0$. Compute the score $r(\mathcal{T}) = \|\boldsymbol{A\pi} - \boldsymbol{b}\|^2$, where $\boldsymbol{\pi} = (\pi_{\boldsymbol{s}_i \cup \{v^1\}}, \ldots, \pi_{\boldsymbol{s}_i \cup \{v^k\}})$. Next, select $\mathcal{T}$ with the smallest value of $r(\mathcal{T})$. For this selected $\mathcal{T}$, check if $\mu_i - \sum_{l=1}^{k} \pi_{\boldsymbol{s}_i \cup \{v^l\}} < \epsilon$. If yes, renormalize $\pi_{\boldsymbol{s}_i \cup \{v^l\}} \leftarrow (\mu_i \times \pi_{\boldsymbol{s}_i \cup \{v^l\}}) / (\sum_{l=1}^{k} \pi_{\boldsymbol{s}_i \cup \{v^l\}})$. If no, add the tuple $(\boldsymbol{s}_i, \mu_i - \sum_{l=1}^{k} \pi_{\boldsymbol{s}_i \cup \{v^l\}})$ to $\mathcal{T}$. At the end of this process, collect all the intervention tuples thus obtained (for all $i \in [q]$), in the set $\mathcal{T}$.
4. Find the excluded value of node V_{N+1}, i.e. the value which is not present in any target in $\mathcal{T}$. If no such value exists, find $v \in C_{V_{N+1}}$ which minimizes $\sum_{i=1}^{q} \pi_{\boldsymbol{s}_i \cup \{v\}}$. For each $i \in [q]$, set $\pi_{\boldsymbol{s}_i \cup \{v\}} \leftarrow 0$. For each $i \in [q]$, renormalize the mixing coefficients $\pi_{\boldsymbol{s}_i \cup \{v^l\}} \leftarrow (\pi_{\boldsymbol{s}_i \cup \{v^l\}} \times \mu_i) / (\sum_{l=1}^{k} \pi_{\boldsymbol{s}_i \cup \{v^l\}})$. Only keep the tuples with strictly positive mixing coefficients in $\mathcal{T}$ i.e. $\mathcal{T} \leftarrow \{(\boldsymbol{s}, \pi_{\boldsymbol{s}}) \in \mathcal{T} : \pi_{\boldsymbol{s}} > 0\}$. **return** $\mathcal{T}$

difference in performance for these models 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.

F EVALUATION METRICS

Let $\mathcal{T}$ denote the actual set of intervention targets and $\widehat{\mathcal{T}}$ denote the set of intervention targets computed by our algorithm. Let $\pi_{\boldsymbol{t}}$, $\widehat{\pi}_{\boldsymbol{s}}$ denote mixing coefficients of target $\boldsymbol{t}$, $\boldsymbol{s}$ in $\mathcal{T}$ and $\widehat{\mathcal{T}}$ respectively. We use the following evaluation metrics to evaluate the performance of our algorithm.

1. **Recall**: Proportion of number of targets in $\mathcal{T}$ that were correctly identified in $\widehat{\mathcal{T}}$

$$\text{Recall} = \frac{|\mathcal{T} \cap \widehat{\mathcal{T}}|}{|\mathcal{T}|}.$$

2. **Root Mean Squared Error**: Root-mean-squared error (RMSE) in the mixing coefficients.

$$\text{RMSE} = \sqrt{\frac{\sum\limits_{\boldsymbol{t}\in\mathcal{T}\cap\widehat{\mathcal{T}}}(\pi_{\boldsymbol{t}}-\widehat{\pi}_{\boldsymbol{t}})^2+\sum\limits_{\boldsymbol{t}\in(\mathcal{T}\setminus\widehat{\mathcal{T}})}(\pi_{\boldsymbol{t}})^2 + \sum\limits_{\boldsymbol{t}\in(\widehat{\mathcal{T}}\setminus\mathcal{T})}(\widehat{\pi}_{\boldsymbol{t}})^2}{|\mathcal{T}\cup\widehat{\mathcal{T}}|}}$$

3. **False-Positive RMSE**: RMSE in the mixing coefficients of the incorrectly identified targets.

$$\text{FP-RMSE} = \sqrt{\frac{\sum_{\boldsymbol{t}\in(\widehat{\mathcal{T}}\setminus\mathcal{T})}(\widehat{\pi}_{\boldsymbol{t}})^2}{|\widehat{\mathcal{T}}\setminus\mathcal{T}|}}$$

4. **False-Negative RMSE**: RMSE in the mixing coefficients of targets not identified.

$$\text{FN-RMSE} = \sqrt{\frac{\sum_{\boldsymbol{t}\in(\mathcal{T}\setminus\widehat{\mathcal{T}})}(\pi_{\boldsymbol{t}})^2}{|\mathcal{T}\setminus\widehat{\mathcal{T}}|}}$$

References

Ankur Ankan and Abinash Panda. pgmpy: Probabilistic graphical models using python. In *Proceedings of the 14th Python in Science Conference (SCIPY 2015)*. Citeseer, 2015.

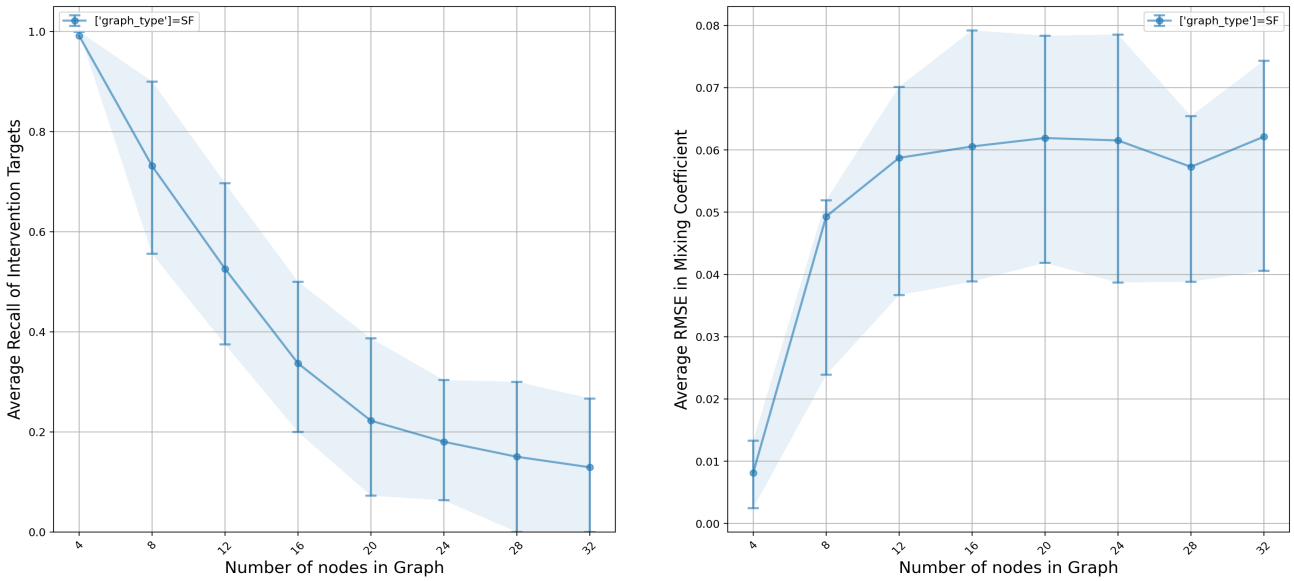

Figure 1: Performance of Algorithm 1 (Appendix D) as a function of number of nodes in the graph. The error bars show the 20% and 80% quantile respectively.

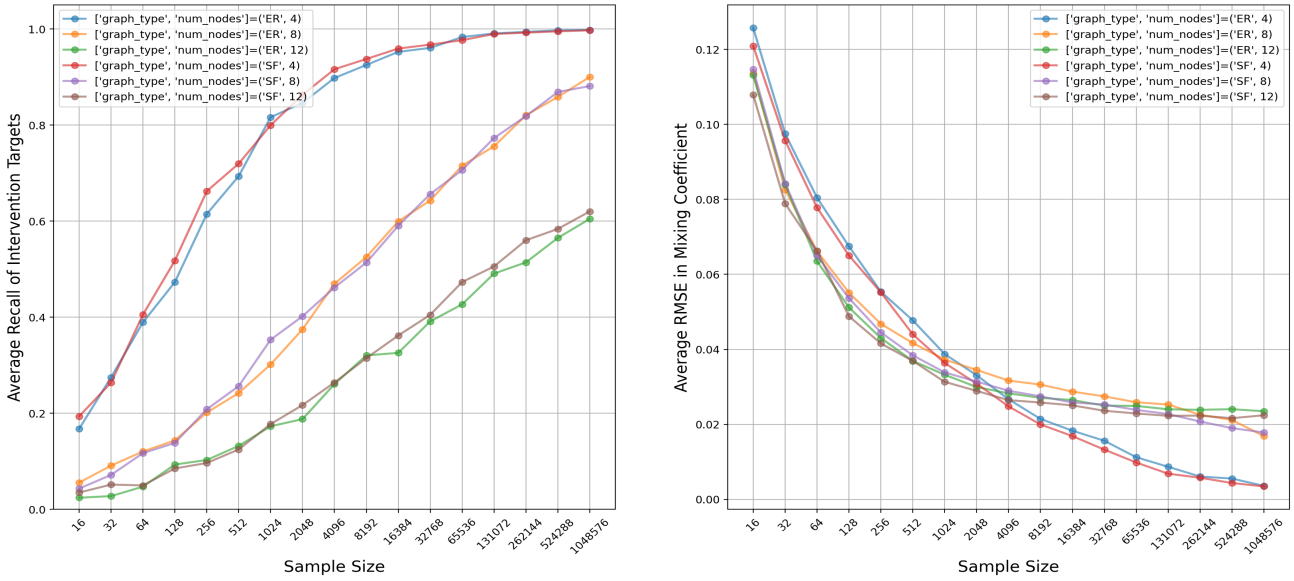

Figure 2: Comparison of performance of Algorithm 1 (Appendix D) for CBNs generated from Erdös-Rényi (ER) and Scale Free (SF) models.