Efficient Interventional Distribution Learning in the PAC Framework

Arnab Bhattacharyya
National University of Singapore

Sutanu Gayen
National University of Singapore

Saravanan Kandasamy
Cornell University

Vedant Raval
Indian Institute of Technology Delhi

N. V. Vinodchandran
University of Nebraska-Lincoln

Abstract

We consider the problem of efficiently inferring interventional distributions in a causal Bayesian network from a finite number of observations. Let $\mathcal{P}$ be a causal model on a set $\mathbf{V}$ of observable variables on a given causal graph G. For sets $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{V}$, and setting $\mathbf{x}$ to $\mathbf{X}$, $P_{\mathbf{x}}(\mathbf{Y})$ denotes the interventional distribution on $\mathbf{Y}$ with respect to an intervention $\mathbf{x}$ to variables $\mathbf{X}$. Shpitser and Pearl (AAAI 2006), building on the work of Tian and Pearl (AAAI 2001), proved that the *ID algorithm* is sound and complete for recovering $P_{\mathbf{x}}(\mathbf{Y})$ from observations.

We give the first provably efficient version of the ID algorithm. In particular, under natural assumptions, we give a polynomial-time algorithm that on input a causal graph G on observable variables $\mathbf{V}$, a setting $\mathbf{x}$ of a set $\mathbf{X} \subseteq \mathbf{V}$ of bounded size, outputs succinct descriptions of both an evaluator and a generator for a distribution $\hat{P}$ that is ε-close (in total variation distance) to $P_{\mathbf{x}}(\mathbf{Y})$ where $\mathbf{Y} = \mathbf{V} \setminus \mathbf{X}$, if $P_{\mathbf{x}}(\mathbf{Y})$ is identifiable.

We also show that when $\mathbf{Y}$ is an arbitrary subset of $\mathbf{V} \setminus \mathbf{X}$, there is no efficient algorithm that outputs an evaluator of a distribution that is ε-close to $P_{\mathbf{x}}(\mathbf{Y})$ unless all problems that have statistical zero-knowledge proofs, including the Graph Isomorphism problem, have efficient randomized algorithms.

1 INTRODUCTION

Density estimation and *parameter learning* are classical problems in statistics studied since the field's inception (e.g., Devroye and Györfi (1985); Scott (2015); Silverman (2018) and references therein). A more recent focus has been on designing distribution learning algorithms that are provably *computationally efficient*, especially in high-dimensional settings. The seminal work of Kearns, Mansour, Ron, Rubinfeld, Schapire, and Sellie (Kearns et al., 1994) considered distribution learning in the PAC (Probably Approximately Correct) framework and set forth two core computational requirements for such a learner: the representation output by the learner should be both an (approximate) (i) *evaluator*, and a (ii) *sampler/generator*. An approximate evaluator for a distribution P takes a domain element $\mathbf{v}$ and outputs the mass of another distribution $\hat{P}$ at $\mathbf{v}$, where $\hat{P}$ is a distribution close, in total variation distance, to P. Similarly, an approximate generator for P takes as input a random seed and outputs an element $\mathbf{v}$ distributed according to $\hat{P}$. The authors argue that it is desirable to learn a representation of a distribution that is both evaluator and generator, as they can be useful for various downstream inference tasks. We call this framework *effective PAC learning* framework in the discussion below.

Distribution learning is especially interesting when the algorithm cannot directly access the distribution to be learned. One of the most prominent example of such a situation arises in *causal effect estimation*. To estimate the effect of a treatment intervention on some outcome, the "gold standard" is to conduct a randomized experiment where a random subpopulation is forcibly treated. However, in many settings (e.g., medicine, economics), it is often not

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

feasible to conduct such experiments, and the only recourse is to make use of observational data and learn/deduct from it the effect of the treatment intervention.

In this work, we investigate the problem of efficiently learning interventional distributions from finite observational samples in the PAC framework discussed above where the goal is to output a representation that is both an approximate evaluator and an approximated generator.

Our discussion of causality and intervention will be in Pearl's language of *causal Bayesian networks* (Pearl, 2009). A causal Bayes net is a standard Bayes net that is reinterpreted causally. Specifically, it makes the assumption of *modularity*: for any variable X, the dependence of X on its parents is an autonomous mechanism that does not change even if other parts of the network are changed.

We fix some basic notation for causal Bayes nets to ground the subsequent discussion. The underlying structure of causal Bayes net $\mathcal{P}$ is a directed acyclic graph G. The graph G consists of $n + h$ nodes where n nodes correspond to the *observable* variables $\mathbf{V}$ while the h additional nodes correspond to the *hidden variables* $\mathbf{U}$. We assume that the observable variables take values over a finite alphabet Σ.

The observational distribution P on $\mathbf{V}$ is obtained by interpreting $\mathcal{P}$ as a standard Bayes net over $\mathbf{V} \cup \mathbf{U}$ and marginalizing to $\mathbf{V}$. An *intervention* is specified by a subset $\mathbf{X} \subseteq \mathbf{V}$ of variables and an assignment[1] $\mathbf{x} \in \Sigma^{|\mathbf{X}|}$. In the interventional distribution, the variables $\mathbf{X}$ are fixed to $\mathbf{x}$, while each variable $W \in (\mathbf{V} \cup \mathbf{U}) \setminus \mathbf{X}$ is sampled as it would have been in the original Bayes net, according to the conditional distribution $W \mid \pi(W)$, where $\pi(W)$ (parents of W) consist of either variables previously sampled in the topological order of G or variables in $\mathbf{X}$ set by the intervention. The marginal of the resulting distribution to $\mathbf{V}$ is the interventional distribution denoted by $P_{\mathbf{x}}(\mathbf{V})$[2]. In general, for any subset $\mathbf{Y} \subseteq \mathbf{V}$ of observables variables we can define the interventional distribution $P_{\mathbf{x}}(\mathbf{Y})$.

The general problem of interest for us is to efficiently learn $P_{\mathbf{x}}(\mathbf{Y})$ for an arbitrary $\mathbf{X}$ and $\mathbf{Y}$ in the PAC framework. If we leave aside considerations of efficiency, this problem was solved in the prior works of Tian (2002), Shpitser and Pearl (2006) and, independently, Huang and Valtorta (2006). We focus here on the work of Shpitser and Pearl (2006). They presented a characterization of the set of graphs G for which $P_{\mathbf{x}}(\mathbf{Y})$ is statistically identifiable from the observational distribution P. In particular, they isolated a 'blocking structure' which they call a *hedge* and showed that $P_{\mathbf{x}}(\mathbf{Y})$ is statistically identifiable if and only if G does not contain a hedge with respect to $\mathbf{X}$ and $\mathbf{Y}$. They gave an efficient algorithm, called the *ID algorithm*, to detect a hedge if it exists. If there are no hedges, the ID algorithm explicitly generates a formula for $P_{\mathbf{x}}(\mathbf{Y})$ in terms of conditional probabilities of P. The ID algorithm is described in Section 2.

Representation of the interventional distribution output by the ID algorithm require *unbounded observational samples* and strict positivity (all conditional probabilities should be strictly > 0) condition to compute any fixed interventional probability. In particular, it does not provide any guarantee that $P_{\mathbf{x}}(\mathbf{Y})$ is learned up to bounded error in total variation distance, either as an evaluator or as a generator, using a bounded number of samples and time. Designing an effective PAC learning algorithm for interventional distributions is the main focus of the paper.

1.1 OUR CONTRIBUTIONS

Our first contribution can be stated as the following theorem. For the definition of a *c-component*, refer to Section 2.

Theorem 1.1 (Informal). *Let $\mathcal{P}$ be a causal Bayes net where the underlying DAG G has in-degree at most d and c-component size at most k. There is an algorithm that given such a causal Bayes net G, a subset $\mathbf{X} \subset \mathbf{V}$ of size at most ℓ such that $P_{\mathbf{x}}(\mathbf{Y})$ is identifiable from P where $\mathbf{Y} = \mathbf{V} \setminus \mathbf{X}$, and ε; with constant probability, outputs (learns) a distribution $\hat{P}$ as an approximate evaluator and generator so that $d_{\mathrm{TV}}(P_{\mathbf{x}}(\mathbf{Y}), \hat{P}) \leq \varepsilon$. The algorithm uses $m = \tilde{O}\left(\frac{nk^{O(k)}|\Sigma|^{O(k\ell+kd)}}{\alpha^{k\ell}\varepsilon^2}\right)$ samples and $O(m(n+|\Sigma|^{kd+k}))$ time, where α is the minimum probability for any non-empty event defined by the variables in c-components intersecting $\mathbf{X}$ and such c-components' parents.*

The formal version of the theorem, Theorem 3.2, is presented in Section 3. Note that unlike in the ID algorithm, we do not require P to be a strictly positive distribution. Indeed, if we let α above be a lower bound on $P(\mathbf{v})$, then $\alpha \leq |\Sigma|^{-n}$, and hence, the sample/time complexity guaranteed by the theorem

[1] Consistent with the convention in the causality literature, we will use a lower case letter (e.g., $\mathbf{x}$) to denote an assignment to the subset of variables corresponding to its upper case counterpart (e.g., $\mathbf{X}$).

[2] In the literature the 'do' notation $\mathsf{do}(\mathbf{x})$ is also used to denote the intervention process. The resulting interventional distribution is denoted by $P(\mathbf{V} \mid \mathsf{do}(\mathbf{x}))$.

will be exponential. So, it is crucial for efficiency that α be a lower bound on the probability of events defined by only a bounded number of variables.

Indeed, our sample complexities are polynomial when the in-degree and c-component size are constants and depends on them exponentially in general. We would like to note that recent works on learning causal models (Acharya et al., 2018; Bhattacharyya et al., 2020a) also make these assumptions in order to derive finite sample bounds. The bound on in-degree is indeed necessary due to the lower bound of Bhattacharyya et al. (2020a). The bound on the c-component size is necessary in general even for estimating the observational marginal over a single c-component of size k. Consider the following graph where a single unobservable is acting on k observables. In such cases, arbitrary distributions can be generated over a support of size $|\Sigma|^k$ whose estimation will require exponentially many samples.

Theorem 1.1 is proved by showing how to implement the ID algorithm while guaranteeing the bound on the distance between the learned and true interventional distributions. This analysis relies on recent work of Bhattacharyya et al. (2020b,c) which examined fixed-structure learning of discrete Bayes nets without hidden variables. However, unlike the 'chain-like' factorization $P(\mathbf{v}) = \prod_i P(v_i \mid \mathbf{v}_{\mathbf{Pa}(i)})$ of the probability mass function for a Bayes net, the interventional probability distribution may not admit a nice factorization (see the examples in Appendix D). Our algorithm exposes a decomposition of $P_{\mathbf{x}}(\mathbf{V})$ into a product of interventional distributions that are either on a small sample space or have a 'chain-like' factorization, and we show both can be learned efficiently using samples from P. The sample complexity of Theorem 1.1 is nearly optimal in terms of n and ε but it remains an interesting open problem to improve the dependence on d, k, ℓ, α and Σ.

One drawback of Theorem 1.1 is that it considers the interventional distribution on $\mathbf{V} \setminus \mathbf{X}$, whereas the causal effect on only a subset $\mathbf{Y}$ of outcome variables may be relevant and is considered and characterized by earlier work (Tian, 2002; Shpitser and Pearl, 2006; Huang and Valtorta, 2006). We show that unfortunately, we cannot hope for a learning algorithm for $P_{\mathbf{x}}(\mathbf{Y})$ in the effective PAC framework, analogous to Theorem 1.1, for an arbitrary subset $\mathbf{Y}$. In particular, we show that the existence of a learning algorithm that outputs an approximate evaluator representation of $P_{\mathbf{x}}(\mathbf{Y})$ for an arbitrary $\mathbf{Y} \subseteq \mathbf{V}$ will lead to efficient randomized algorithms for all problems in the complexity class SZK. The complexity class SZK contains hard computational problems including the Graph Isomorphism problem and it is believed that these problems do not have efficient randomized algorithms. Indeed, we establish the hardness even when the intervention set $\mathbf{X}$ is empty (i.e., when the goal is to learn the marginal on a subset of variables) and the Bayes net has in-degree at most 2. This result is stated below.

Theorem 1.2 (Informal). *Suppose there is a randomized polynomial-time algorithm that on input a Bayes net (without hidden variables) distribution P on $\mathbf{V}$, $\mathbf{Y} \subseteq \mathbf{V}$, and ε; outputs a representation R of a distribution $\hat{P}$ so that (1) $d_{\mathrm{TV}}(P(\mathbf{Y}), \hat{P}) \leq \varepsilon$, and (2) for every $\mathbf{y}$, $\hat{P}(\mathbf{y})$ can be evaluated (or even multiplicatively $(1 \pm \varepsilon)$-approximated) efficiently using R. Then all problems that have statistical zero knowledge (the complexity class* SZK*), including the Graph Isomorphism problem, can be solved in randomized polynomial time.*

Thus the problem of outputting an evaluator representation of a distribution that approximates the effect of an interventional distribution on an arbitrary subset, even when it is identifiable in the sense of the ID algorithm, is computationally hard.

Remark 1.3. Using notations of Theorem 1.1, the lower bound of Theorem 1.2 corresponds to the case $d = 2, k = 1, l = 0$ and $\alpha = 1$.

1.2 RELATED WORK

Pearl (1989) introduced Causal Bayesian Networks to formally define interventions and causal effects. Tian and Pearl (2002) first studied the problem of identification of causal effects from observations, and gave graphical characterizations for such identifications when the intervention is atomic, i.e. consisting of a single variable and we are interested to determine the effect of this variable on the rest of the variables. They also gave an expression for such an atomic intervention in terms of observational quantities whenever the later is identifiable. Subsequently, Shpitser and Pearl (2006) and Huang and Valtorta (2006) independently generalized Tian and Pearl (2002) for non-atomic interventions on any subset of variables, giving graphical characterizations for identifiability and an expression of the causal effect whenever it is identifiable.

Recently, there has been a surge of interest in designing efficient estimators for causal effects to accompany the above identifiability results. For estimating the *average treatment effect*, classic methods are the inverse probability weighting and regression-based estimators. These estimators can be combined and made "doubly robust" in the form of the *augmented inverse probability weighted (AIPW)* estimator. The AIPW has been systematically extended to apply to causal Bayes nets using the framework of semiparametric theory; see the works of Henckel et al. (2019), Rotnitzky and Smucler (2020); Jung et al. (2020a, 2021); Bhattacharya et al. (2020). Also, Chernozhukov et al. (2018) quantitatively studied the double robustness of such semiparametric estimators. This line of work, though close to ours in terms of motivation, differs in two significant ways: (i) our algorithm learns a description of the full interventional distribution, rather than just its mean, and (ii) we provide PAC-style finite sample bounds instead of a convergence statement in the form of asymptotic normality. The recent work by Kennedy et al. (2021) does study density estimation of the interventional distribution but again does not provide finite sample bounds.

Finite sample bounds for causal inference have been studied only recently. Acharya et al. (2018) gave finite bounds for goodness-of-fit testing for non-parametric causal models using observation and experimental data. In Bhattacharyya et al. (2020a), the authors gave finite sample bounds for learning *atomic* interventions. In this setting, Tian and Pearl (2002) gave a simple formula for the intervention as a function of the joint distribution whenever it is identifiable. In contrast, an intervention involving multiple variables is more involved for deciding identifiability. In this case, Shpitser and Pearl (2006) gave a recursive procedure that generates a formula for the intervention whenever it is identifiable. We give a finite sample bound for this problem. The finite sample and time bounds shown in this paper can be seen as a finite-sample version of Shpitser and Pearl (2006), generalizing Bhattacharyya et al. (2020a) for learning non-atomic interventions on the rest of the graph using observational samples.

At a high-level, we partition the random variables of the interventional distribution into two sub-distributions: those which share the c-component with an intervention and those which do not. The final distribution is a product of these two sub-distributions. Our technique for learning the former sub-distribution involves a clear understanding of how the said recursive procedure unrolls at every call and to provide it with the required joint probability distribution needed at every recursive call. Our technique for learning the later sub-distribution involves a procedure for learning Bayesian networks from samples and has some similarity to the algorithm given for the atomic case in Bhattacharyya et al. (2020a).

Testing various properties of polynomial-time samplable distributions such as uniformity, entropy, and closeness is a fundamental problem that characterizes several zero-knowledge complexity classes (Watson, 2016; Sahai and Vadhan, 2003; Malka, 2015). We show our hardness result employing a connection between testing polynomial-time samplable distributions and testing Bayesian networks. Bayesian network is an important statistical model for which finite sample bounds for testing and learning have been given recently (Bhattacharyya et al., 2020b; Canonne et al., 2020; Bhattacharyya et al., 2020c).

2 PRELIMINARIES

Notation. We use bold and non-bold fonts to denote sets and singleton variables respectively. We use capital and small letters to denote variables and values taken by them respectively.

In this paper we only consider distributions over finite sample spaces. For two distributions P and Q over a sample space Ω, their total variation distance $d_{\mathrm{TV}}(P,Q) = \frac{1}{2}\sum_{i\in\Omega}|P_i - Q_i|$ and their KL distance $d_{\mathrm{KL}}(P,Q) = \sum_{i\in\Omega} P_i \ln\frac{P_i}{Q_i}$. Pinsker's inequality says $d_{\mathrm{TV}}(P,Q) \leq \sqrt{0.5\cdot d_{\mathrm{KL}}(P,Q)}$. A distribution Q is $(1\pm\varepsilon)$-approximate p.m.f. for another distribution P if $Q(x)\in[(1-\varepsilon)P(x),(1+\varepsilon)P(x)]$ for every x in the sample space.

We are interested in distributions over directed graphical models: Bayes nets and Causal Bayes nets. We will introduce necessary notation to define them. Let G be a directed acyclic graph on $\mathbf{V}$. For any subset $\mathbf{S}\subseteq\mathbf{V}$, $\mathbf{An}^+(\mathbf{S})$ and $\mathbf{Pa}^+(\mathbf{S})$ denotes the set of all observable ancestors and parents of $\mathbf{S}$ (including $\mathbf{S}$) respectively; $\mathbf{Pa}^-(\mathbf{S}) = \mathbf{Pa}^+(\mathbf{S})\setminus\mathbf{S}$; and $G[\mathbf{S}]$ denotes the induced subgraph of G over $\mathbf{S}$.

Definition 2.1. *A* Bayesian Network *P is a distribution that can be specified by a tuple $\langle\mathbf{V}, G, \{\Pr(V_i \mid \mathbf{pa}^-(V_i)) : V_i\in\mathbf{V}, \mathbf{pa}^-(V_i)\in\Sigma^{|\mathbf{Pa}^-(V_i)|}\}\rangle$ where: (i) $\mathbf{V} = (V_1,\ldots,V_n)$ is a set of variables over alphabet Σ, (ii) G is a directed acyclic graph with n nodes corresponding to the elements of $\mathbf{V}$, and (iii) $\Pr[V_i \mid \mathbf{pa}^-(V_i)]$ is the conditional distribution of variable V_i given that its parents $\mathbf{Pa}^-(V_i)$ in G take the values $\mathbf{pa}^-(V_i)$.*

The Bayesian Network $P = \langle\mathbf{V}, G, \{\Pr[V_i \mid \mathbf{pa}^-(V_i)]\}\rangle$ defines a probability distribution over $\Sigma^{|\mathbf{V}|}$, as follows. For all $\mathbf{v}\in\Sigma^{|\mathbf{V}|}$,

$$P(\mathbf{v}) = \prod_{V_i\in\mathbf{V}} \Pr[v_i \mid \mathbf{Pa}(V_i) = \mathbf{v}_{\mathbf{Pa}(V_i)}].$$

In this distribution, each variable V_i is independent of its non-descendants given its parents in G.

For a given $\mathbf{X} \subseteq \mathbf{V}$ and an assignment $\mathbf{x}$ to $\mathbf{X}$, the interventional distribution $P_{\mathbf{x}}(\mathbf{V})$ is the Bayes net distribution defined on the DAG where the set of incoming edges to $\mathbf{X}$ are removed and $\mathbf{X}$ is fixed to $\mathbf{x}$ with probability 1. All other variables follow the usual parent-child relation and factorization of P.

It is standard in the literature to assume that the unobservable variables in $\mathbf{U}$ have exactly two observable children. In this case, Causal Bayes nets can be represented using *Acyclic Directed Mixed Graph* (ADMG in short) representation over only the observable vertices $\mathbf{V}$. An ADMG consists of a set of observable variables and two kinds of edges: directed edges $E^{\rightarrow}$ and bi-directed edges $E^{\leftrightarrow}$. The directed edges $(X \rightarrow Y)$ denote parent-child relationship as in a DAG. The bi-directed edges $(X \longleftrightarrow Y)$ denote an indirect correlation between X and Y due to a hidden parent U. Due to the reduction of Verma and Pearl (1990) from general causal graphs to ADMGs that preserves important independence and structural properties, the results of this paper do work for general causal graphs when the effective in-degree and the c-component size of the original graph are still bounded. The reasoning is similar to the one in Appendix I of Acharya et al. (2018).

The notion of a *c-component*, introduced by Tian and Pearl (2002), plays a central role in the identification of causal effects.

Definition 2.2 (c-component)**.** *Let G be an ADMG. Then a set of vertices $\mathbf{C}$ of G is a* c-component *of G if every pair of vertices in $\mathbf{C}$ is connected by a path of only bi-directed edges.*

Definition 2.3 (c-component factorization)**.** *For any ADMG G over observables $\mathbf{V}$, the* c-component factorization *$C(G)$ is the partition of vertices $\mathbf{V}$ into $\{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_m\}$ such that the induced subgraphs $G[\mathbf{C}_i]$s are (maximal) c-components. Then the observational distribution of any causal Bayes net on G gets factorized as: $P[\mathbf{V}] = \prod_{i=1}^{m} P_{\mathbf{V}\setminus\mathbf{C}_i}[\mathbf{C}_i]$.*

Definition 2.4 (Effective parents of V_i in G)**.** *For any vertex V_i, let $\mathbf{C}$ be the c-component of G that contains V_i. Then the* effective parents *of V_i is $\mathbf{Pa}^{+}(\mathbf{C}) \cap \{V_1, V_2, \ldots, V_{i-1}\}$.*

Definition 2.5 (α-strong positivity for c-components)**.** *A c-component $\mathbf{C}$ is said to be α-strongly positive if for every assignment $\mathbf{z}$, $P(\mathbf{pa}^{+}(\mathbf{C}) = \mathbf{z}) \geq \alpha$.*

Other standard definitions of technical concepts related to causal Bayesian networks appear in the supplementary material.

2.1 IDENTIFICATION ALGORITHM REVISITED

Algorithm 1: ID$(\mathbf{y}, \mathbf{x}, P, G)$

```
Input : Subset $\mathbf{X} \subseteq \mathbf{V}$, $\mathbf{Y} = \mathbf{V} \setminus \mathbf{X}$,
        Assignments $\mathbf{x}, \mathbf{y}$, Observational
        distribution $P$, ADMG $G$
Output: $P_{\mathbf{x}}(\mathbf{y})$
1 if $\mathbf{x} = \emptyset$ then
      return $P(\mathbf{v})$
2 if $\mathbf{V} \setminus \mathbf{An}^{+}(\mathbf{Y})_G \neq \emptyset$ then
      return ID$(\mathbf{y}, \mathbf{x} \cap \mathbf{An}^{+}(\mathbf{Y})_G, \sum_{\mathbf{v}\setminus\mathbf{An}^{+}(\mathbf{Y})_G} P,$
      $G[\mathbf{An}^{+}(\mathbf{Y})_G])$.
3 if $C(G \setminus \mathbf{X}) = \{\mathbf{S}_1, \ldots, \mathbf{S}_k\}$ then
      return $\prod_i$ ID$(\mathbf{s}_i, \mathbf{v} \setminus \mathbf{s}_i, P, G)$.
4 if $C(G \setminus \mathbf{X}) = \{\mathbf{S}\}$ is singleton then
      (a) if $C(G) = G$ then
          return FAIL.
      b if $\mathbf{S} \in C(G)$ then
          return $\prod_{i|V_i\in\mathbf{S}} P(v_i \mid v_1, v_2, \ldots, v_{i-1})$.
      c if $\exists\mathbf{S}' : \mathbf{S} \subset \mathbf{S}' \in C(G)$ then
          return ID$(\mathbf{y}, \mathbf{x} \cap \mathbf{S}', \prod_{i|V_i\in\mathbf{S}'} P(V_i \mid$
          $(V_1, V_2, \ldots, V_{i-1}) \cap \mathbf{S}', (v_1, \ldots, v_{i-1}) \setminus$
          $\mathbf{S}'), \mathbf{S}'$ ).
```

Suppose we are given an ADMG G and an observational distribution $P(\mathbf{V})$. Let $\mathbf{X}, \mathbf{Y} \subseteq \mathbf{V}$ be disjoint subsets and $\mathbf{x}, \mathbf{y}$ be assignments to $\mathbf{X}$ and $\mathbf{Y}$. The goal of the *identification* question in general is to determine the probability $\Pr_{\mathbf{x}}(\mathbf{y})$. We restrict our attention to the case $\mathbf{Y} = \mathbf{V} \setminus \mathbf{X}$ due to the hardness result discussed in the supplementary material. Algorithm 1 is the modified ID algorithm of Shpitser and Pearl (2006) restricted to this case.

Here we explain the steps of Algorithm 1 in detail. Let $\mathbf{X} \subseteq \mathbf{V}$, $\mathbf{Y} = \mathbf{V} \setminus \mathbf{X}$ and let $\mathbf{x}, \mathbf{y}$ be assignments to $\mathbf{X}$ and $\mathbf{Y}$. The algorithm accepts $\mathbf{x}, \mathbf{y}$, the observational distribution P and the ADMG G as inputs and outputs the interventional probability $P_{\mathbf{x}}(\mathbf{y})$ or return FAIL when $P_{\mathbf{x}}(\mathbf{y})$ is not uniquely determinable. The steps 1 and 2 of Algorithm 1 correspond to the base cases while 3 and 4 correspond to the non-trivial cases.

1. Step 1 handles the base case when $\mathbf{X}$ is an empty set. In this case, the algorithm directly outputs the observational probability $P(\mathbf{y})$.
2. Step 2 handles the base case when $\sigma := \mathbf{V} \setminus \mathbf{An}^{+}(\mathbf{Y})_G$ is non-empty. It is clear by definition that the vertices of σ is not affecting $\mathbf{Y}$. Therefore,

σ can be removed from the original graph G (and also from $\mathbf{X}$).

3. Suppose $\mathbf{S}_1, \ldots, \mathbf{S}_k$ (for $k > 1$) are the c-components of $G[\mathbf{V} \setminus \mathbf{X}]$. Then we get the following formula:

$$P_{\mathbf{x}}(\mathbf{y}) = \prod_{i \in [k]} P_{\mathbf{v} \setminus \mathbf{s}_i}(\mathbf{s}_i),$$

   where the product term is due to c-component factorization (Definition 2.3).

4. Suppose the c-component of $G[\mathbf{V} \setminus \mathbf{X}]$ is a singleton $\mathbf{S} = \{\mathbf{V} \setminus \mathbf{X}\}$. Here we have three cases.

   (a) Consider the case where the c-component of G is G itself. This results in the existence of hedge (with root set $\mathbf{S}$ and internal nodes $\mathbf{X}$, and some edges can be removed to make sure each internal node has exactly one outgoing edge. Please refer Shpitser and Pearl (2006) for the definition of Hedge). Hence, it is impossible to uniquely determine the required query.

   (b) Consider the case where $\mathbf{S}$ is a c-component of G. This means there is no-bidirected edge between $\mathbf{S}$ and $\mathbf{V} \setminus \mathbf{S}$. Due to the absence of bi-directed edges, no backdoor paths (Pearl (2009)) exist from $\mathbf{X}$ to $\mathbf{S}$, which results in the following formula:

$$P_{\mathbf{x}}(\mathbf{y}) = \prod_{i | V_i \in \mathbf{S}} P(v_i \mid v_1, \ldots, v_{i-1})$$

   whose correctness can be easily verified by Bayes rule and marginalization. (Note: Additionally, we can also remove some of the v_j's in the conditional terms on top of this expression by applying conditional independence properties of the model.)

   (c) This is the final case. Suppose there exists $\mathbf{S}' : \mathbf{S} \subset \mathbf{S}'$ which is a c-component of G. Here, similar to 4(b), since there is no bi-directed edge between $\mathbf{X} \setminus \mathbf{S}'$ and the rest of the vertices, it is possible to identify the distribution obtained after intervening on $\mathbf{x} \setminus \mathbf{s}'$ - using the following formula: $P_{\mathbf{x} \setminus \mathbf{s}'}(\mathbf{S}') = \prod_{V_i \in \mathbf{S}'} P(V_i | V_{[i-1]} \cap \mathbf{S}', (\mathbf{x} \setminus \mathbf{s}')_{V_{[i-1]} \setminus \mathbf{S}'})$ where $V_{[i-1]} = \{V_1, \ldots, V_{i-1}\}$.
   The idea here is to first intervene on $\mathbf{X} \setminus \mathbf{S}'$ and then try to recursively intervene on $\mathbf{X} \cap \mathbf{S}'$ on top of that – which is equivalent to intervening on $\mathbf{X}$. Hence the required query is obtained by determining $P_{\mathbf{x} \cap \mathbf{S}'}(\mathbf{y})$ in the graph $G[\mathbf{S}']$ with observational distribution $P_{\mathbf{x} \setminus \mathbf{S}'}(\mathbf{S}')$ defined above.

An illustration of how the different steps and the different recursive calls to the algorithm lead to the bigger picture of obtaining the probability $P_{\mathbf{x}}(\mathbf{y})$ is presented with two examples in Appendix D.

3 LEARNING INTERVENTIONAL DISTRIBUTION EFFICIENTLY

In this section, we provide an effective PAC version of the ID Algorithm. The algorithm takes an ADMG G, the intervention $\mathbf{X} = \mathbf{x}$, a parameter ε, and random samples from the observational distribution $P(\mathbf{V})$ and outputs a representation of a distribution $\widehat{P}_{\mathbf{x}}(\mathbf{Y})$ where $\mathbf{Y} = \mathbf{V} \setminus \mathbf{X}$ which is ε-close to $P_{\mathbf{x}}(\mathbf{Y})$ in total variation distance. The representation of $\widehat{P}_{\mathbf{x}}(\mathbf{Y})$ returned is a collection of conditional probability tables and hence can be used to generate samples from $\widehat{P}_{\mathbf{x}}(\mathbf{Y})$ as well as evaluate the probability at any $\mathbf{Y} = \mathbf{y}$.

Throughout this section, we will assume $\mathbf{Y} = \mathbf{V} \setminus \mathbf{X}$ and $P_{\mathbf{x}}(\mathbf{Y})$ is identifiable. For simplicity we also use $P_{\mathbf{x}}$ to denote the distribution $P_{\mathbf{x}}(\mathbf{Y})$. Note that we can run the ID Algorithm to efficiently decide whether a particular intervention is identifiable or not. Thus without loss of generality we can assume that step 4a in the ID Algorithm never gets invoked.

Let $\{\mathbf{C}_i\}_i$ denote the c-component partition of G, and let k be the size of the largest c-component. Let $\mathbf{X} = \cup_{i=1}^{\ell} \mathbf{X}_i$ be a partition of $\mathbf{X}$ such that $\mathbf{X}_i \subseteq \mathbf{C}_i$, for $i = 1$ to ℓ without loss of generality. We make the following assumption.

Assumption 3.1. $\mathbf{pa}^+(\mathbf{C}_i)$ *is α-strongly positive for every $1 \leq i \leq \ell$.*

We call the learning algorithm FiniteID. We state the guarantee of the algorithm FiniteID as a theorem below.

Theorem 3.2. *There are three algorithms* FiniteID, FiniteIDEvaluator, *and* FiniteIDSampler *with the following properties. Given an ADMG G with k and ℓ as defined above, and observations from a causal Bayes net P on G,* FiniteID *takes $m = \widetilde{O}\left(\left(\frac{n|\Sigma|^{2k\ell+kd+k}}{\alpha^{k\ell}\varepsilon^2} + \frac{(3k)^{2(k+3)}|\Sigma|^{2k\ell}}{\alpha^2\varepsilon^2}\right)\log\frac{1}{\delta}\right)$ samples from P and in $O(m(n + |\Sigma|^{kd+k}))$ time returns a description* Eff *of a distribution $\widehat{P}_{\mathbf{x}}$ on $\mathbf{Y} = \mathbf{V} \setminus \mathbf{X}$ so that $d_{\mathrm{TV}}(\widehat{P}_{\mathbf{x}}, P_{\mathbf{x}}) \leq \varepsilon$.* FiniteIDEvaluator, *given the output* Eff *of* FiniteID *and any $\mathbf{y} \in \Sigma^{|Y|}$, returns $\widehat{P}_{\mathbf{x}}(\mathbf{y})$ in $O(n|\Sigma|^{2kl+2kd})$ time.* FiniteIDSampler, *given the output* Eff *of* FiniteID, *generates a sample of $\widehat{P}_{\mathbf{x}}$ in $O(n|\Sigma|^{2kl+2kd})$ time.*

Algorithm 2: FiniteID

```
Input  : Assignments $\mathbf{x}$, error parameter $\varepsilon$,
         Observational dist. $P$, ADMG $G$
Output : $\widehat{P}_{\mathbf{x}}(\mathbf{Y})$
  1. $\widehat{Q} \leftarrow$ Learn $P(\mathbf{C}_{>\ell} \mid \mathbf{C}_{\leq\ell} = \mathbf{c}_{\leq\ell})$ for every
     fixing of $\mathbf{c}_{\leq\ell}$ that agrees with the intervention,
     as a Bayes net factorization of Tian and Pearl
     (2002) using the add-1 estimator (Lemma 3.5).
  2. $\widehat{R} \leftarrow$ Learn $P(\mathbf{C}_{\leq\ell} \mid \mathbf{c}_{\mathbf{pa}})$ for every fixing $\mathbf{c}_{\mathbf{pa}}$
     for parents of $\mathbf{C}_{\leq\ell}$ using the recursive
     procedure ID from Shpitser and Pearl (2006)
     using Lemma 3.4 as follows.
     (a) If the current recursive call uses the
         original distribution $P$ from which we are
         sampling, then learn the required joint
         distribution of the subsequent recursive call
         by learning each of its factor from samples.
         Example: to learn a factor $P(\mathbf{B} \mid \mathbf{A})$, learn
         $P(\mathbf{B} \mid \mathbf{A} = \mathbf{a})$ for every $\mathbf{a}$. This step uses
         strong positivity to learn each factor up to
         a point-wise $(1 \pm \varepsilon)$-factor
     (b) If the current recursive call uses some
         distribution $D \neq P$ constructed before,
         then learn each factor of the required
         distribution of the subsequent call by
         taking appropriate ratios from Bayes' rule,
         without using any samples. Example,
         $D(\mathbf{B} \mid \mathbf{A}) = D(\mathbf{AB})/D(\mathbf{A})$.
  3. Return $\mathsf{Eff} \leftarrow \{\widehat{Q}_{i,z}, \widehat{S}_{i,z}\}_{i,z}$
```

Algorithm 3: FiniteIDSampler

```
Input  : $\mathsf{Eff} = \{\widehat{Q}_{i,z}, \widehat{S}_{i,z}\}_{i,z}$ from Algorithm 2
Output : Sampler for $\widehat{P}_{\mathbf{x}}(\mathbf{Y})$
  1. Sample the source nodes of $\widehat{P}_{\mathbf{x}}(\mathbf{y}) =$
     $\prod_{V_i \in \mathbf{C}_{>\ell}} \widehat{Q}(V_i \mid \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq\ell}} \widehat{S}_i(V_i \mid \mathbf{Z}_i)$,
     according to their marginal distribution defined
     in $\mathsf{Eff}$
  2. Sample the nodes which depend on the sources
     according to $\mathsf{Eff}$ and samples from step 1
  3. Sample nodes from subsequent levels according
     to $\mathsf{Eff}$ until sink nodes are sampled
```

Algorithm 4: FiniteIDEvaluator

```
Input  : $\mathsf{Eff} = \{\widehat{Q}_{i,z}, \widehat{S}_{i,z}\}_{i,z}$ from Algorithm 2
Output : Evaluator for $\widehat{P}_{\mathbf{x}}(\mathbf{Y})$
  1. Evaluate $\widehat{P}_{\mathbf{x}}(\mathbf{y})$ using the factorization $\widehat{P}_{\mathbf{x}}(\mathbf{y}) =$
     $\prod_{V_i \in \mathbf{C}_{>\ell}} \widehat{Q}(V_i \mid \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq\ell}} \widehat{S}_i(V_i \mid \mathbf{Z}_i)$,
     according to $\mathsf{Eff}$
```

We give an overview of the proof in the main body of the paper and move details to the supplementary material (Appendix B). Firstly we define,

$$\mathbf{C}_{>\ell} := \cup_{i>\ell}\mathbf{C}_i \text{ and } \mathbf{C}_{\leq\ell} := \cup_{i\leq\ell}\mathbf{C}_i.$$

Let $\mathbf{C}_i \setminus \mathbf{X}_i = \cup_j \mathbf{C}_{ij}$ be the c-component partitions of $G[\mathbf{C}_i \setminus \mathbf{X}_i]$. Then any identifiable joint interventional probability factorizes based on the following claim.

Claim 3.3. *Let $\mathbf{Y} = \mathbf{V} \setminus \mathbf{X}$, $\mathbf{An}^+(\mathbf{Y}) = \mathbf{V}$, and suppose $P_{\mathbf{x}}(\mathbf{y})$ is identifiable. Then,*

$$P_{\mathbf{x}}(\mathbf{y}) = P_{\mathbf{c}_1,\ldots,\mathbf{c}_\ell}(\mathbf{c}_{>\ell}) \prod_{i,j} P_{\mathbf{v}\setminus\mathbf{c}_{ij}}(\mathbf{c}_{ij}).$$

The proof of the claim directly follows from step 3 of the ID algorithm. Hence it suffices for us to learn

$$R := \prod_{i,j} P_{\mathbf{v}\setminus\mathbf{c}_{ij}}(\mathbf{c}_{ij}) = P_{\mathbf{x},\mathbf{c}_{>\ell}}(\mathbf{C}_{\leq\ell} \setminus \mathbf{X}) \qquad \text{and}$$

$$Q := P_{\mathbf{c}_{\leq\ell}}(\mathbf{c}_{>\ell}) = P_{\mathbf{c}_1,\ldots,\mathbf{c}_\ell}(\mathbf{c}_{>\ell}).$$

For every fixing of $\mathbf{pa}^-(\mathbf{C}_{\leq\ell} \setminus \mathbf{X})$, R factorizes as a product of at most $k\ell$ conditional probabilities S_i, each of which has at most $(kd + k)$ conditioning variables. For every fixing of $\mathbf{c}_{\leq\ell}$, Q is Bayes net over at most n variables of in-degree at most $(kd+k)$. Our approach would be to learn R as a set of conditional probability tables and Q as a Bayes net, both up to a total variation error at most ε. Then our learnt distribution will have the following factorization.

$$\widehat{P}_{\mathbf{x}}(\mathbf{y}) = \prod_{V_i \in \mathbf{C}_{>\ell}} \widehat{Q}(V_i \mid \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq\ell}} \widehat{S}_i(V_i \mid \mathbf{Z}_i) \qquad (1)$$

where the effective parents (Definition 2.4) of V_i is denoted by $\mathbf{Z}_i$.

3.1 LEARNING R

Now we focus on learning the distribution

$$R = P_{\mathbf{x},\mathbf{c}_{>\ell}}(\mathbf{C}_{\leq\ell} \setminus \mathbf{X})$$

for every fixing of $\mathbf{pa}^-(\mathbf{C}_{\leq\ell} \setminus \mathbf{X})$.

Lemma 3.4. *Let $\mathbf{Pa}^+(\mathbf{C_i})$ be α-strongly positive. Then for every fixing of $\mathbf{C}_{>\ell}$ (in fact, every fixing of $\mathbf{pa}^-(\mathbf{C}_{\leq\ell} \setminus \mathbf{X})$ without loss of generality), R can be learnt as a product of conditional probabilities $\widehat{R}$ of* *in-degree at most $(kd+k)$ using*

$$m = \widetilde{\Theta}\left(k^3 \ell d \alpha^{-2} \varepsilon^{-2} \log\left(\frac{|\Sigma| k^2 \ell}{\delta}\right)\right)$$

samples and $O(m|\Sigma|^{kd+k})$ time such that with probability at least $(1-\delta)$, $\widehat{R}$ is $(1 \pm (3k)^{k+1}\ell\varepsilon)$-approximate p.m.f. for R.

$R = \prod_{i,j} R_{ij}$, where each distribution $R_{ij} = P_{\mathbf{v}\setminus \mathbf{c}_{ij}}(\mathbf{C}_{ij})$ starts a recursive call of the ID algorithm. Each such recursive call forms a recursion tree whose non-leaves correspond to recursive calls from 2, 3, or 4c, and the leaves correspond to recursive calls from 1 or 4b. Our strategy would be to give a $(1 \pm \varepsilon_2)$-approximate p.m.f. access to the joint distribution for every subsequent recursive call (compared to the true joint distribution), assuming a $(1 \pm \varepsilon_1)$-approximate p.m.f. access to the joint distribution of the current call. Inductively, we'll get a $(1 \pm \varepsilon_3)$-approximate access to the leaf distributions: outputs by the leaf calls. Each distribution R_{ij} is a multiplication of the leaf distributions, possibly with some marginalizations. Marginalization preserves the approximation ratio of the p.m.f. and there are at most k multiplicands, so that our final approximation ratio would be $(1 \pm \varepsilon_3)^k$. This approach gives us an algorithm for learning R approximately.

3.2 LEARNING Q

We next focus on learning the distribution Q, for every fixing of $\mathbf{C}_{\leq \ell}$. Q is a Bayesian network on at most n variables of indegree at most $(kd+k)$ (Tian, 2002). We use conditional sampling to learn this distribution using the add-1 estimator for the conditional probabilities, which was recently analyzed in the context of learning Bayesian networks (Bhattacharyya et al., 2020c). The difference with that work and our work is that we are trying to learn an interventional distribution from observational samples which is different from learning the observational distribution.

Lemma 3.5. *Let $\mathbf{Pa}^{+}(\mathbf{C}_i)$ be α-strongly positive. Then for every fixing of $\mathbf{C}_{\leq \ell}$, Q can be learnt as a Bayes net $\widehat{Q}$ of indegree at most $(kd+k)$ using $m = \widetilde{\Theta}\left(\frac{n|\Sigma|^{kd+k}}{\alpha^{|\mathbf{C}_{\leq \ell}|}\varepsilon} \log\left(\frac{n|\Sigma|^{kd+k+1}}{\delta}\right)\right)$ samples and $O(mn)$ time such that $d_{\mathrm{KL}}(Q, \widehat{Q}) \leq \varepsilon$ with probability at least $(1-\delta)$.*

3.3 COMBINING Q AND R

We shall now discuss the proof of Theorem 3.2.

Proof of Theorem 3.2. It follows from Lemma 3.5 and Pinsker's inequality that for any fixed $\mathbf{c}_{\leq \ell}$, by using $m = \widetilde{\Theta}\left(\frac{n|\Sigma|^{kd+k}}{\alpha^{|\mathbf{C}_{\leq \ell}|}\varepsilon^2} \log\left(\frac{n|\Sigma|^{kd+k+1}}{\delta}\right)\right)$ samples and $O(mn)$ time, we get $d_{\mathrm{TV}}\left(\prod_{V_i \in \mathbf{C}_{>\ell}} Q(V_i \mid \mathbf{Z}_i), \prod_{V_i \in \mathbf{C}_{>\ell}} \widehat{Q}(V_i \mid \mathbf{Z}_i)\right) \leq \varepsilon$ with high probability.

We scale down ε by a $(3k)^{k+1} \cdot \ell$ factor to achieve an algorithm that for any fixed $\mathbf{c}_{>\ell}$, uses $m = O\left((3k)^{2(k+3)}\ell^3 d \alpha^{-2}\varepsilon^{-2} \log\left(\frac{|\Sigma| k \ell}{\delta}\right)\right)$ samples and $O(m|\Sigma|^{kd+k})$ time, to obtain $\widehat{R}$ which is $(1 \pm \varepsilon)$-approximate p.m.f. for R, with high probability, from Lemma 3.4.

Combining the above two pieces, we get at most $\varepsilon(|\Sigma|^{k\ell}+1)$ error as follows. We get the theorem by an appropriate scaling.

$$\begin{aligned}
&d_{\mathrm{TV}}(P_{\mathbf{x}}, \widehat{P}_{\mathbf{x}}) \\
&= \sum_{\mathbf{c}_{\leq \ell}, \mathbf{c}_{>\ell}} \Bigg| \prod_{V_i \in \mathbf{C}_{>\ell}} Q(V_i \mid \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} S_i(V_i \mid \mathbf{Z}_i) - \\
&\qquad \prod_{V_i \in \mathbf{C}_{>\ell}} \widehat{Q}(V_i \mid \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} \widehat{S}_i(V_i \mid \mathbf{Z}_i) \Bigg| \\
&= \sum_{\mathbf{c}_{\leq \ell}, \mathbf{c}_{>\ell}} \Bigg| \prod_{V_i \in \mathbf{C}_{>\ell}} Q(V_i \mid \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} S_i(V_i \mid \mathbf{Z}_i) \\
&\qquad - \prod_{V_i \in \mathbf{C}_{>\ell}} Q(V_i \mid \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} \widehat{S}_i(V_i \mid \mathbf{Z}_i) \\
&\qquad + \prod_{V_i \in \mathbf{C}_{>\ell}} Q(V_i \mid \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} \widehat{S}_i(V_i \mid \mathbf{Z}_i) \\
&\qquad - \prod_{V_i \in \mathbf{C}_{>\ell}} \widehat{Q}(V_i \mid \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} \widehat{S}_i(V_i \mid \mathbf{Z}_i) \Bigg| \\
&\leq \sum_{\mathbf{c}_{>\ell}} \sum_{\mathbf{c}_{\leq \ell}} \Bigg| \prod_{V_i \in \mathbf{C}_{>\ell}} Q(V_i \mid \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} S_i(V_i \mid \mathbf{Z}_i) \\
&\qquad -(1 \pm \varepsilon) \prod_{V_i \in \mathbf{C}_{>\ell}} Q(V_i \mid \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} S_i(V_i \mid \mathbf{Z}_i) \Bigg| \\
&\quad + \sum_{\mathbf{c}_{\leq \ell}} \sum_{\mathbf{c}_{>\ell}} \Bigg| \prod_{V_i \in \mathbf{C}_{>\ell}} Q_i(V_i \mid \mathbf{Z}_i) - \prod_{V_i \in \mathbf{C}_{>\ell}} \widehat{Q}_i(V_i \mid \mathbf{Z}_i) \Bigg| \\
&\leq \varepsilon \sum_{\mathbf{c}_{>\ell}} \sum_{\mathbf{c}_{\leq \ell}} \prod_{V_i \in \mathbf{C}_{>\ell}} Q(V_i \mid \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} S_i(V_i \mid \mathbf{Z}_i) + \sum_{\mathbf{c}_{\leq \ell}} \varepsilon \\
&\leq \varepsilon(1 + |\Sigma|^{k\ell}).
\end{aligned}$$

We used $\widehat{S}_i(V_i \mid \mathbf{Z}_i) \leq 1$ in the fourth line.

Now, we show how to generate samples approximately according to the distribution $P_{\mathbf{x}}(\mathbf{y})$. Our algorithm generates samples according to the factorization $\widehat{P}_{\mathbf{x}}(\mathbf{y})$ given in (1). From (1) we have

$$P_{\mathbf{x}}(\mathbf{y}) = \prod_{V_i \in \mathbf{C}_{>\ell}} \widehat{Q}(V_i \mid \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} \widehat{S}_i(V_i \mid \mathbf{Z}_i)$$

where $\hat{Q}$ comes from Lemma 3.5 and $\widehat{S}$ comes from R of Claim 3.4. Note that either in Lemma 2 or Claim 8, the effective graph in any step of the recursion is always a subgraph of G. Hence, the topological order between V_i and $\mathbf{Z}_i$ is never violated. In particular, we can sample each random variable of $\mathbf{Y}$ in the topological order of $\mathbf{V} \setminus \mathbf{X}$. Then at any step of this sampling, whenever we try to sample some $v_i \sim \widehat{Q}(V_i \mid \mathbf{Z}_i)$ or $\sim \widehat{S}_i(V_i \mid \mathbf{Z}_i)$, $\mathbf{Z}_i$ would always be sampled, and hence gets assigned before it. □

Remark 3.6. It follows that we can also sample from any $\mathbf{T} \subseteq \mathbf{V} \setminus \mathbf{X}$, ignoring the unnecessary variables. However, evaluating the marginal on $\mathbf{T}$ in general remains computationally expensive.

4 HARDNESS OF PAC LEARNING MARGINALS

In this section we show that designing an algorithm that outputs a representation of a approximate succinct evaluator for the *marginal* of an interventional distribution in general is computationally hard. This should be contrasted to our learning result in Section 3 for the case when $\mathbf{X}$ is of bounded size and $\mathbf{Y} = \mathbf{V} \setminus \mathbf{X}$. In Section 3 we designed a learning algorithm for $P_{\mathbf{x}}(\mathbf{Y})$ that outputs a representation R which is a generator as well as an evaluator for a distribution $\widehat{P}_{\mathbf{x}}(\mathbf{Y})$ that is ε close to $P_{\mathbf{x}}(\mathbf{Y})$ in total variation distance. The result we establish in this section implies that it is computationally hard to design such a PAC style learning algorithm for the general case of arbitrary $\mathbf{X}$ and $\mathbf{Y}$. In fact we show the hardness for the case when $\mathbf{X}$ is empty intervention and Y is an arbitrary subset of observable variables for a Bayesian network of in-degree 2. While there are hardness results known for computing marginals of Bayes net distributions (Cooper, 1990; Roth, 1996), the hardness of learning in the PAC framework does not directly follow from these works. Due to space limitations, the proofs of the main hardness result and other hardness results are moved to the supplementary material.

Definition 4.1 (Efficient ε-approximate evaluator circuit)**.** *A circuit $\mathcal{E}_P : \{0,1\}^n \to \mathbb{R}$ is called an efficient ε-additive evaluator for a distribution P over $\{0,1\}^n$, if there exists a distribution $\widehat{P}$ over $\{0,1\}^n$ such that: (1) [Additive approximation] $d_{\mathrm{TV}}(P, \widehat{P}) \leq \varepsilon$, (2) [Evaluation query] For any $x \in \{0,1\}^n$, $\mathcal{E}_P$ on input x outputs a number $p \in [1-\varepsilon, 1+\varepsilon]\widehat{P}(x)$ in* poly(n) *time.*

Definition 4.2. *Let* BayesMarginal *be the following learning problem: given a parameter ε, samples from an unknown Bayes net distribution P over binary variables $\mathbf{V}$, and a $\mathbf{Y} \subseteq \mathbf{V}$, output an efficient ε-approximate evaluator circuit $\mathcal{E} : \{0,1\}^{|\mathbf{Y}|} \to \mathbb{R}$ for the marginal distribution of P over $\mathbf{Y}$.*

Now we state the main hardness result.

Theorem 4.3. *If* BayesMarginal *has a randomized polynomial-time learning algorithm even restricted to Bayes nets of in-degree at most 2, then the complexity class* SZK $\subseteq$ BPP.

The class SZK contains several hard computational problems including the Graph Isomorphism problem and is believed to be computationally harder than BPP; the class of problems that admit efficient randomized algorithms. It is widely believed that the class SZK does not admit efficient randomized algorithms.

5 CONCLUSION

Identification of interventional distributions in causal models is a significant problem with many practical applications. There are two distinct bodies of work on this topic. The first focuses on deriving exact graph-theoretic characterizations of identifiability, accompanied by identification algorithms that require infinitely many samples. The second proposes various estimators for estimating causal effects (commonly, population mean of the interventional distribution) under restrictive assumptions on the causal model.

Our work joins a recent thread of works in combining these two perspectives. Additionally, we establish rigorous finite-sample guarantees for the algorithm's output. We also showed hardness of designing similar PAC learning algorithms for identifying the effect of intervention on a subset $\mathbf{X}$ to another arbitrary subset $\mathbf{Y}$ of the observable variables. Some future directions of work include: (i) considering probability densities on continuous-valued variables, (ii) improving the sample complexity bounds or proving improved hardness results, and (iii) prove finite sample complexity bounds for estimators that can be used in practice.

Acknowledgements

A.B. was partially supported by an NRF Fellowship for AI (R-252-000-B13-281), a startup grant (R-252-000-A33-133), and an Amazon Research Award (R-252-000-A61-720). S.G. was supported by A.B.'s NRF Fellowship for AI (R-252-000-B13-281). S.K. was partially supported by NSF 1846300 CAREER and NSF 1815893 grants. N.V.V. was supported in part by National Science Foundation HDR:TRIPODS-1934884.

References

J. Acharya, A. Bhattacharyya, C. Daskalakis, and S. Kandasamy. Learning and testing causal models with interventions. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 9469–9481, 2018.

R. Bhattacharya, R. Nabi, and I. Shpitser. Semiparametric inference for causal effects in graphical models with hidden variables. *arXiv preprint arXiv:2003.12659*, 2020.

A. Bhattacharyya, S. Gayen, S. Kandasamy, A. Maran, and N. V. Vinodchandran. Learning and sampling of atomic interventions from observations. In *International Conference on Machine Learning*, pages 842–853. PMLR, 2020a.

A. Bhattacharyya, S. Gayen, K. S. Meel, and N. V. Vinodchandran. Efficient distance approximation for structured high-dimensional distributions via learning. *Advances in Neural Information Processing Systems*, 33, 2020b.

A. Bhattacharyya, S. Gayen, E. Price, and N. V. Vinodchandran. Near-optimal learning of tree-structured distributions by chow-liu. *CoRR*, abs/2011.04144, 2020c. URL `https://arxiv.org/abs/2011.04144`. ACM STOC 2021.

C. L. Canonne, I. Diakonikolas, D. M. Kane, and A. Stewart. Testing bayesian networks. *IEEE Trans. Inf. Theory*, 66(5):3132–3170, 2020.

V. Chernozhukov, D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey, and J. Robins. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68, 2018.

G. F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, 42 (2):393–405, 1990. ISSN 0004-3702. doi: https://doi.org/10.1016/0004-3702(90)90060-D. URL `https://www.sciencedirect.com/science/article/pii/000437029090060D`.

S. Dasgupta. The sample complexity of learning fixed-structure bayesian networks. *Mach. Learn.*, 29(2-3):165–180, 1997.

L. Devroye and L. Györfi. *Nonparametric density estimation.* Wiley series in probability and mathematical statistics. Wiley, New York, 1985. ISBN 0471816469.

L. Henckel, E. Perković, and M. H. Maathuis. Graphical criteria for efficient total effect estimation via adjustment in causal linear models. *arXiv preprint arXiv:1907.02435*, 2019.

Y. Huang and M. Valtorta. Identifiability in causal bayesian networks: A sound and complete algorithm. In *AAAI*, pages 1149–1154, 2006.

Y. Jung, J. Tian, and E. Bareinboim. Learning causal effects via weighted empirical risk minimization. *Advances in neural information processing systems*, 33, 2020a.

Y. Jung, J. Tian, and E. Bareinboim. Estimating causal effects using weighting-based estimators. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 10186–10193. AAAI Press, 2020b.

Y. Jung, J. Tian, and E. Bareinboim. Estimating identifiable causal effects through double machine learning. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, 2021.

M. Kearns, Y. Mansour, D. Ron, R. Rubinfeld, R. E. Schapire, and L. Sellie. On the learnability of discrete distributions. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 273–282, 1994.

E. H. Kennedy, S. Balakrishnan, and L. Wasserman. Semiparametric counterfactual density estimation. *arXiv preprint arXiv:2102.12034*, 2021.

L. Malka. How to achieve perfect simulation and a complete problem for non-interactive perfect zero-knowledge. *Journal of Cryptology*, 28(3):533–550, 2015.

J. Pearl. *Probabilistic reasoning in intelligent systems - networks of plausible inference.* Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann, 1989.

J. Pearl. *Causality.* Cambridge university press, 2009.

D. Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1): 273–302, 1996. ISSN 0004-3702. doi: https://doi.org/10.1016/0004-3702(94)00092-1. URL `https://www.sciencedirect.com/science/article/pii/0004370294000921`.

A. Rotnitzky and E. Smucler. Efficient adjustment sets for population average causal treatment effect estimation in graphical models. *J. Mach. Learn. Res.*, 21:188–1, 2020.

A. Sahai and S. P. Vadhan. A complete problem for statistical zero knowledge. *J. ACM*, 50(2): 196–249, 2003. doi: 10.1145/636865.636868. URL `https://doi.org/10.1145/636865.636868`.

D. W. Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

I. Shpitser and J. Pearl. Identification of joint interventional distributions in recursive semi-markovian causal models. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, AAAI*, 2006.

B. W. Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

J. Tian. *Studies in causal reasoning and learning*. University of California, Los Angeles, 2002.

J. Tian and J. Pearl. A general identification condition for causal effects. In *AAAI/iaai*, pages 567–573, 2002.

T. Verma and J. Pearl. Causal networks: Semantics and expressiveness. In *Proceedings of the 4th Conference on Uncertainty in Artificial Intelligence*, pages 69–78, 1990. ISBN 0-444-88650-8.

T. Watson. The complexity of estimating min-entropy. *computational complexity*, 25(1):153–175, 2016.

Supplementary Material: Efficient Interventional Distribution Learning in the PAC Framework

A PRELIMINARIES ON CAUSAL BAYESIAN NETWORKS

We follow Pearl's formalism of Causal Bayesian Networks (Pearl, 2009).

Definition A.1 (Causal Bayes Net)**.** *A* causal Bayes net $\mathcal{P}$ *is a collection of interventional distributions that can be defined in terms of a tuple* $\langle \mathbf{V}, \mathbf{U}, G, \{\Pr(V_i \mid \boldsymbol{\pi}(V_i)) : V_i \in \mathbf{V}, \boldsymbol{\pi}(V_i) \in \Sigma^{|\mathbf{\Pi}(V_i)|}\}, \{\Pr[\mathbf{U}]\}\rangle$*, where (i)* $\mathbf{V} = (V_1, \ldots, V_n)$ *and* $\mathbf{U}$ *are the set of observable and hidden variables respectively, (ii)* G *is a directed acyclic graph on* $\mathbf{V} \cup \mathbf{U}$*, (iii)* $\Pr[V_i \mid \boldsymbol{\pi}(V_i)]$ *is the conditional probability distributions of* $V_i \in \mathbf{V}$ *given that its parents in* $\mathbf{V} \cup \mathbf{U}$*,* $\mathbf{\Pi}(V_i)$*, take the values* $\boldsymbol{\pi}(V_i)$*, and (iv)* $\Pr[\mathbf{U}]$ *is the distribution of the hidden variables* $\mathbf{U}$*.* G *is said to be the* causal graph *corresponding to* $\mathcal{P}$*.*

Such a causal Bayes net $\mathcal{P}$ *defines a unique interventional distribution* $P_{\mathbf{x}}(\mathbf{V})$ *for every subset* $\mathbf{X} \subseteq \mathbf{V}$ *(including* $\mathbf{X} = \emptyset$*) and assignment* $\mathbf{x} \in \Sigma^{|\mathbf{X}|}$*, as follows. For all* $\mathbf{v} \in \Sigma^{|\mathbf{V}|}$*:*

$$P_{\mathbf{x}}(\mathbf{v}) = \begin{cases} \sum_{\mathbf{u}} \prod_{V_i \in \mathbf{V} \setminus \mathbf{X}} \Pr[v_i \mid (\mathbf{v}, \mathbf{u})_{\mathbf{\Pi}(V_i)}] \cdot \Pr[\mathbf{u}] & \textit{if } \mathbf{v} \textit{ is consistent with } \mathbf{x} \\ 0 & \textit{otherwise.} \end{cases}$$

Graphically, for a given $\mathbf{X} \subseteq \mathbf{V}$ and an assignment $\mathbf{x}$ to $\mathbf{X}$, the interventional distribution $P_{\mathbf{x}}(\mathbf{V})$ is the Bayes net distribution defined on the DAG where the set of incoming edges to $\mathbf{X}$ are removed and $\mathbf{X}$ is fixed to $\mathbf{x}$ with probability 1. All other variables follow the usual parent-child relation and factorization of P.

It is standard in the literature to assume that the unobservable variables in $\mathbf{U}$ have exactly two observable children. In this case, Causal Bayes nets can be represented using an *Acyclic Directed Mixed Graph* (ADMG in short) representation over only the observable vertices $\mathbf{V}$. An ADMG consists of a set of variables and two kinds of edges: directed edges $E^{\rightarrow}$ and bi-directed edges $E^{\leftrightarrow}$. The directed edges $(X \rightarrow Y)$ denote parent-child relationship as in a DAG. The bi-directed edges $(X \longleftrightarrow Y)$ denote an indirect correlation between X and Y due to a hidden parent U. Thus, the edge set of an ADMG is the union of the directed edges $E^{\rightarrow}$ and the bidirected edges $E^{\leftrightarrow}$. For ADMG we need to redefine some of the notation. For any given set $\mathbf{S} \subseteq \mathbf{V}$, $G_{\overline{\mathbf{S}}}$ denotes the graph obtained from G by removing the *incoming edges to* $\mathbf{S}$. Note that the bi-directed edges indicate edges from unobservables to observables and hence the bi-directed edges incident to $\mathbf{S}$ in G also gets removed in $G_{\overline{\mathbf{S}}}$. Also it is not useful to explicitly represent unobservables with a single child, hence those bi-directed edges incident to $\mathbf{S}$ in G will not be present in $G_{\overline{\mathbf{S}}}$. For any subset $\mathbf{S} \subseteq \mathbf{V}$, $\mathbf{An}^{+}(\mathbf{S})$ and $\mathbf{Pa}^{+}(\mathbf{S})$ denotes the set of all observable ancestors and parents of $\mathbf{S}$ (including $\mathbf{S}$) respectively. We use $\mathbf{Pa}^{-}(\mathbf{S}) = \mathbf{Pa}^{+}(\mathbf{S}) \setminus \mathbf{S}$. We assume the indices of the observable vertices of ADMG $\mathbf{V} = \{V_1, V_2, V_3, \ldots, V_i, V_{i+1}, \ldots\}$ are arranged in a topological ordering. Finally, for any subset $\mathbf{S} \subseteq \mathbf{V}$, $G[\mathbf{S}]$ denotes the induced subgraph of G over $\mathbf{S}$.

The notion of a *c-component* introduced by Tian and Pearl (2002) plays a central role in the identification of causal effects.

Definition A.2 (c-component)**.** *Let* G *be an ADMG. Then a set of vertices* $\mathbf{C}$ *of* G *is a* c-component *of* G *if every pair of vertices in* $\mathbf{C}$ *is connected by a path of only bi-directed edges.*

Definition A.3 (c-component factorization)**.** *For any ADMG* G *over observables* $\mathbf{V}$*, the* c-component factorization $C(G)$ *is the partition of vertices* $\mathbf{V}$ *into* $\{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_m\}$ *such that the induced subgraphs* $G[\mathbf{C}_i]$*s are (maximal) c-components. Then the observational distribution of any causal Bayes net on* G *gets factorized as:* $P[\mathbf{V}] = \prod_{i=1}^{m} P_{\mathbf{V} \setminus \mathbf{C}_i}[\mathbf{C}_i]$*.*

Definition A.4 (Effective parents of V_i in G)**.** *For any vertex V_i, let $\mathbf{C}$ be the c-component of G that contains V_i. Then the* effective parents *of V_i is $\mathbf{Pa}^+(\mathbf{C}) \cap \{V_1, V_2, \ldots, V_{i-1}\}$.*

Definition A.5 (α-strong positivity for c-components)**.** *A c-component $\mathbf{C}$ is said to be α-strongly positive if for every assignment $\mathbf{z}$, $P(\mathbf{pa}^+(\mathbf{C}) = \mathbf{z}) \geq \alpha$.*

B OMITTED DETAILS FROM SECTION 3

Recursive calls are taken from steps 2, 3, or 4c. In 2 and 3, the subsequent call just uses the current distribution (or a marginal of it). In 4c, the subsequent call uses a very different distribution. Therefore, in a path to the leaf in the recursion tree, we need to give a new joint distribution whenever 4c is taken. We use the following result to give this distribution, based on whether 4c was taken for the first time in this root-to-leaf path or not.

Claim B.1. *Let P_{bef} and P_{aft} respectively be the two distributions just before and during the recursive call at step 4c of ID.*

- *If $P_{\text{bef}} = P$, we can give a $(1 \pm 3k\varepsilon)$-approximate p.m.f. of $\widehat{P}_{\text{aft}}$ for P_{aft} using $O((kd+d)\alpha^{-2}\varepsilon^{-2}\log\frac{|\Sigma|}{\delta})$ samples and $O((kd+d)|\Sigma|^{kd+d}\alpha^{-2}\varepsilon^{-2}\log\frac{|\Sigma|}{\delta})$ time.*
- *If $P_{\text{bef}} \neq P$ and we have a $(1 \pm \varepsilon)$-approximate $\widehat{P}_{\text{bef}}$ for P_{bef}, we can give a $(1 \pm 3k\varepsilon)$-approximate p.m.f. $\widehat{P}_{\text{aft}}$ for P_{aft} using $O(k|\Sigma|^{2k})$ time and no samples.*

Proof. Note that each factor of the joint distribution in step 4c is of the form $S(V_i \mid \mathbf{Z}_i)$, where $\mathbf{Z}_i$ consists of the effective parents of V_i.

In case 1, we take enough samples to empirically learn $P_{\text{bef}}(V_i \mid \mathbf{z}_i)$ for every fixing of $\mathbf{z}_i$. Due to α-strong positivity, $\min(P_{\text{bef}}(v_i \mid \mathbf{z}_i), P_{\text{bef}}(\mathbf{z}_i)) \geq \alpha$ for any $v_i, \mathbf{z}_i$. Hence from Chernoff's bound, using $O\left(\alpha^{-2}\varepsilon^{-2}\log\left(\frac{1}{\delta}\right)\right)$ samples the learnt distribution would be point-wise $(1 \pm \varepsilon)$-close with high probability. Since $|\mathbf{Z}_i| \leq (kd+d)$ our final sample complexity is $m = O((kd+d)\alpha^{-2}\varepsilon^{-2}\log\frac{|\Sigma|}{\delta})$ and the time complexity is $O(m|\Sigma|^{kd+d})$. Since each factor is $(1 \pm \varepsilon)$-approximate, the approximation for the joint distribution is at most $(1 \pm \varepsilon)^k$.

In case 2, we compute $P_{\text{bef}}(v_i \mid \mathbf{z}_i) = P_{\text{bef}}(v_i, \mathbf{z}_i)/P_{\text{bef}}(\mathbf{z}_i)$ by appropriate marginalizations of S, which preserves the approximation. Since 4c is taken at least once before, the graph-size is at most k. Due to the ratio, along with a maximum of k multiplications, the final approximation becomes $\left(\frac{1+\varepsilon}{1-\varepsilon}\right)^k$-factor. This does not involve sampling and can be done in $O(k|\Sigma|^{2k})$ time, since in this case P_{aft} must be over at most k variables.

In either case, we return $\widehat{P}_{\text{aft}}$ as a p.m.f. table of size $|\Sigma|^k$. □

Similarly for the leaf calls from steps 1 or 4b, we approximate their (terminal) output distributions (henceforth referred to as leaf distributions) by marginalization or conditional sampling, depending on whether the current distribution is P or not. Inductively, we would get an approximation guarantee between the true and estimated leaf distributions. Our output just consists of p.m.f. tables for each joint distribution $\widehat{P}_{\text{bef}}$ just before a leaf call. In that case, each leaf distribution (denoted by S and $\widehat{S}$) is simply a marginalization (step 1) or product of conditional probabilities (step 4b) of $\widehat{P}_{\text{bef}}$, which can be efficiently computed.

We now analyze the leaf calls from 1 or 4b. Again, we split into two cases, depending on whether the leaf call was taken using the original distribution P or not. Let P_{bef} be the distribution during the leaf call.

Claim B.2. *Let S be a true leaf distribution in the recursion tree output from step 1 or 4b. Then our corresponding learnt distribution $\widehat{S}$ as mentioned above is point-wise $(1 \pm (3k)^k\varepsilon)$-approximate for S.*

Proof. If $P_{\text{bef}} = P$ we get a sampling access to P_{bef}. If $P_{\text{bef}} \neq P$, we get P_{bef} as a p.m.f. table.

Step 1 returns the joint distribution over $\mathbf{Y}$, over at most $|\Sigma|^k$ items. If $P_{\text{bef}} = P$, we learn $S(\mathbf{Y})$ up to point-wise $(1 \pm \varepsilon)$-factor using $\widetilde{O}(k\alpha^{-1}\varepsilon^{-2})$ samples with high probability. If $P_{\text{bef}} \neq P$, we output P_{bef} itself as $\widehat{S}$. In either case, we return a p.m.f. table of size at most $|\Sigma|^k$.

If 4b is taken with $P_{\text{bef}} = P$, we need to learn each $P(V_i \mid \mathbf{z}_i)$, where $\mathbf{Z}_i$ is the effective parent set of V_i, and is of size at most $(kd + d)$. So, we iterate through all possible $\mathbf{z}_i$ and learn each term $\widehat{S}(V_i \mid \mathbf{z}_i) = P(V_i \mid \mathbf{z}_i)$ point-wise up to $(1 \pm \varepsilon)$-error by rejection sampling with high probability. There are at most k such terms, so that the sample complexity would be $m = \widetilde{O}((kd+d)\alpha^{-2}\varepsilon^{-2})$ and the time complexity would be $O(m|\Sigma|^{kd+d})$.

If 4b is taken with $P_{\text{bef}} \neq P$, then it must be preceded by a call from 4b, since only that step changes the distribution by our construction. Then P_{bef} is a distribution over at most k variables. We obtain all required terms $\widehat{S}(v_i \mid \mathbf{z}_i) = P_{\text{bef}}(v_i \mid \mathbf{z}_i) = P_{\text{bef}}(v_i, \mathbf{z}_i)/P_{\text{bef}}(\mathbf{z}_i)$ by appropriate marginalizations of P_{bef} using $\widetilde{O}(|\Sigma|^{2k})$ time and no samples.

In both the above cases of 4b, we return all possible conditional probability tables required for evaluating the output formula of this step. Note that the recursion depth is at most k and we lose a factor of $(1 \pm 3k\varepsilon)$ from $(1 \pm \varepsilon)$ in each depth. Therefore, the approximation ratio for the leaf distributions is at most $(1 \pm (3k)^k\varepsilon)$. □

Proof of Lemma 3.4. Each R_{ij} is a product of at most k leaf distributions. Our final output distribution $\widehat{R}_{ij}$ just uses $\widehat{S}$ in place of each such leaf distribution S. Since $R = \prod_{i,j} R_{ij}$ is a product of at most $k^2\ell$ leaf distributions, the lemma thus follows. □

We will now introduce some of the tools used in proving Lemma 3.5. The following Claim relates the p.m.f.s of Q and P. Here we used the c-component factorization from (Tian, 2002, Lemma 3) to write $Q = \prod_{V_i \in \mathbf{C}_{>\ell}} P(V_i \mid \mathbf{Z}_i)$, where $\mathbf{Z}_i$ is the effective parents of V_i.

Claim B.3. *Let* $\mathbf{v}$ *be an assignment to* $\mathbf{V}$ *and* $\mathbf{w}$ *be its restriction to* $\mathbf{C}_{>\ell}$*. Then* $\alpha^{|\mathbf{C}_{\leq \ell}|} \leq \frac{P(\mathbf{v})}{Q(\mathbf{w})} \leq 1$.

Proof. $\frac{P(\mathbf{v})}{Q(\mathbf{w})} = \frac{\prod_{V_i \in \mathbf{V}} P(v_i|\mathbf{z}_i)}{\prod_{V_i \in \mathbf{C}_{>\ell}} P(v_i|\mathbf{z}_i)} = \prod_{V_i \in \mathbf{C}_{\leq \ell}} P(v_i \mid \mathbf{z}_i) \geq \prod_{V_i \in \mathbf{C}_{\leq \ell}} P(v_i \circ \mathbf{z}_i) \geq \alpha^{|\mathbf{C}_{\leq \ell}|}$. □

We closely follow the d_{KL}-learning result for Bayes nets given in Bhattacharyya et al. (2020c). Let $\mathbf{Z}'_i = \mathbf{Z}_i \backslash \mathbf{C}_{\leq \ell}$. Our algorithm just learns the add-1 empirical distribution $\widehat{P}(v_i \mid \mathbf{Z}'_i = \mathbf{a}))$ on the conditional samples from $P(v_i \mid \mathbf{Z}'_i = \mathbf{a})$. Our learnt distribution $\widehat{Q}$ consists of the $\widehat{P}(v_i \mid \mathbf{Z}'_i = \mathbf{a})$'s, in place of every $P(v_i \mid \mathbf{Z}'_i = \mathbf{a})$ in the Bayes net factorization for Q.

Fact B.4 (Dasgupta (1997); Canonne et al. (2020); Bhattacharyya et al. (2020c))**.** $d_{\text{KL}}(Q, \widehat{Q}) = \sum_{i \in \mathbf{C}_{>\ell}} \sum_{\mathbf{a}} Q(\mathbf{Z}'_i = \mathbf{a}) \cdot d_{\text{KL}}(P(v_i \mid \mathbf{Z}'_i = \mathbf{a}), \widehat{P}(v_i \mid \mathbf{Z}'_i = \mathbf{a}))$.

We also have the following guarantee about the add-1 estimator.

Fact B.5 (Bhattacharyya et al. (2020c))**.** *Let D be an unknown distribution over k items and $\widehat{D}$ be its add-1 empirical distribution of m samples. Then if* $m \gtrsim \frac{k}{\varepsilon} \log\left(\frac{k}{\delta}\right)\left(\log\left(\frac{k}{\varepsilon}\right) + \log\log\left(\frac{k}{\delta}\right)\right)$ *then* $d_{\text{KL}}(D, \widehat{D}) \leq \varepsilon$ *with probability at least* $(1 - \delta)$.

We are now ready to prove Lemma 3.5.

Proof of Lemma 3.5. We analyze the two cases: $Q(\mathbf{Z}'_i = \mathbf{a}) \geq \frac{\varepsilon}{n|\Sigma|^{kd+k}\log(m+|\Sigma|)}$ and otherwise.

In the former case, $P(\mathbf{Z}'_i = \mathbf{a}) \geq \frac{\alpha^{|\mathbf{C}_{\leq \ell}|}\varepsilon}{n|\Sigma|^{kd+k+1}}$ from Claim B.3 and $m = \widetilde{\Theta}\left(\frac{n|\Sigma|^{kd+k}}{\alpha^{|\mathbf{C}_{\leq \ell}|}\varepsilon} \log\left(\frac{n|\Sigma|^{kd+k+1}}{\delta}\right)\right)$ samples would ensure at least $\widetilde{\Theta}(\frac{n|\Sigma|^{kd+k+1} \cdot Q(\mathbf{Z}'_i = \mathbf{a})}{\varepsilon} \log\left(\frac{n|\Sigma|^{kd+k+1}}{\delta}\right))$ conditional samples are seen from $P(v_i \mid \mathbf{Z}'_i = \mathbf{a})$ with high probability from Chernoff's bound. Hence, $d_{\text{KL}}(P(v_i \mid \mathbf{Z}'_i = \mathbf{a}), \widehat{P}(v_i \mid \mathbf{Z}'_i = \mathbf{a})) \leq \frac{\varepsilon}{n|\Sigma|^{kd+k}Q(\mathbf{Z}'_\mathbf{i} = \mathbf{a})}$ from Fact B.5 except with probability at most $\frac{\varepsilon}{n|\Sigma|^{kd+k}}$.

In the later case, $Q(\mathbf{Z}'_i = \mathbf{a}) \leq \frac{\varepsilon}{n|\Sigma|^{kd+k}\log(m+|\Sigma|)}$ and $d_{\mathrm{KL}}(P(v_i \mid \mathbf{Z}'_i = \mathbf{a}), \widehat{P}(v_i \mid \mathbf{Z}'_i = \mathbf{a})) \leq \log(m + |\Sigma|)$, since the least add-1 probability of at most m conditional samples is $\frac{1}{m+|\Sigma|}$.

Combining all the cases, the summation of the RHS of Fact B.4 evaluates to at most $O(\varepsilon)$. □

Now we describe the overall construction of the evaluator for $P_{\mathbf{x}}(\mathbf{Y}) = Q \cdot R$, where $Q = \prod_{V_i \in \mathbf{C}_{>\ell}} Q(V_i \mid \mathbf{Z}_i)$, $R = \prod_{V_i \in \mathbf{C}_{\leq \ell}} S_i(V_i \mid \mathbf{Z}_i)$, $\mathbf{Z}_i$ is the conditioning set for V_i, Q is the Bayes net and S_i's are the leaf distributions defined before. This is very similar to a Bayes net factorization except that the probability distribution for the factors need not be the same. Our evaluator is

$$\widehat{P}_{\mathbf{x}}(\mathbf{y}) = \prod_{V_i \in \mathbf{C}_{>\ell}} \widehat{Q}(V_i \mid \mathbf{Z}_i) \prod_{V_i \in \mathbf{C}_{\leq \ell}} \widehat{S}_i(V_i \mid \mathbf{Z}_i), \tag{2}$$

where $\widehat{Q}$ comes from Lemma 3.5 and $\widehat{S}_i$ comes from R in Lemma 3.4. Let $\widehat{Q}_{i,z} = \widehat{Q}(V_i \mid Z_i = z)$ and $\widehat{S}_{i,z} = \widehat{S}(V_i \mid Z_i = z)$. Our FiniteID algorithm returns a collection of tables $\mathsf{Eff} = \{\widehat{Q}_{i,z}, \widehat{S}_{i,z}\}_{i,z}$.

C HARDNESS OF PAC LEARNING MARGINALS

In this section we show that designing an algorithm that outputs a representation of a approximate succinct evaluator for the *marginal* of an interventional distribution in general is computationally hard. This should be contrasted to our learning result in Section 3 for the case when $\mathbf{X}$ is of bounded size and $\mathbf{Y} = \mathbf{V} \setminus \mathbf{X}$. In Section 3 we designed a learning algorithm for $P_{\mathbf{x}}(\mathbf{Y})$ that outputs a representation R which is a generator as well as an evaluator for a distribution $\widehat{P}_{\mathbf{x}}(\mathbf{Y})$ that is ε close to $P_{\mathbf{x}}(\mathbf{Y})$ in total variation distance. The result we establish in this section implies that it is computationally hard to design such a PAC style learning algorithm for the general case of arbitrary $\mathbf{X}$ and $\mathbf{Y}$. In fact we show the hardness for the case when $\mathbf{X}$ is empty intervention and Y is an arbitrary subset of observable variables for a Bayesian network of in-degree 2.

We need the following definitions.

Definition C.1 (Polynomial-time samplable distributions)**.** *Given a Boolean circuit C_n mapping n bits to m bits, the distribution sampled by C_n is obtained by uniformly choosing $x \in \{0,1\}^n$ and evaluating C on x. A distribution is polynomial-time samplable distribution if it is sampled by a circuit C_n of size* $\mathrm{poly}(n)$*. We often use C itself to denote the distribution sampled by the circuit C_n.*

We will use the following hardness result for the testing of two polynomial-time samplable distributions.

Definition C.2 (Testing of polynomial-time samplable distributions)**.** *Let* DistCkt *be the following computational problem: given an encodings of two boolean circuits C_n and D_n, both of which output exactly $m = \mathrm{poly}(n)$ bits, distinguish between the two cases: $d_{\mathrm{TV}}(C_n, D_n) \leq 1/3$ versus $d_{\mathrm{TV}}(C_n, D_n) \geq 2/3$.*

Theorem C.3 (Sahai and Vadhan (2003))**.** DistCkt *is complete for the complexity class Statistical Zero Knowledge (denoted* SZK*).*

The class SZK contains several hard computational problems including the Graph Isomorphism problem and is believed to be computationally harder than BPP; the class of problems that admit efficient randomized algorithms. It is widely believed that the class SZK does not admit efficient randomized algorithms.

In the following, we show that DistCkt reduces to the problem of computing an efficient, approximate evaluator for the marginal distribution of a Bayes net, thereby showing that the later problem is hard. We formally define this problem now.

Definition C.4 (Efficient ε-approximate evaluator circuit)**.** *A circuit $\mathcal{E}_P : \{0,1\}^n \to \mathbb{R}$ is called an efficient ε-additive evaluator for a distribution P over $\{0,1\}^n$, if there exists a distribution $\widehat{P}$ over $\{0,1\}^n$ such that:*

1. *[Additive approximation] $d_{\mathrm{TV}}(P, \widehat{P}) \leq \varepsilon$*
2. *[Evaluation query] For any $x \in \{0,1\}^n$, $\mathcal{E}_P$ on input x outputs a number $p \in [1-\varepsilon, 1+\varepsilon]\widehat{P}(x)$ in* $\mathrm{poly}(n)$ *time.*

Definition C.5. *Let* BayesMarginal *be the following learning problem: given a parameter ε, samples from an unknown Bayes net distribution P over binary variables $\mathbf{V}$, and a $\mathbf{Y} \subseteq \mathbf{V}$, output an efficient ε-approximate evaluator circuit $\mathcal{E} : \{0,1\}^{|\mathbf{Y}|} \to \mathbb{R}$ for the marginal distribution of P over $\mathbf{Y}$.*

We also need the following result for approximating the variation distance additively when there are evaluator circuits for distributions.

Theorem C.6 (Bhattacharyya et al. (2020b))**.** *Let P and Q be two unknown distributions over Ω. Then given access to two efficient ε-approximate evaluator circuits $\mathcal{E}_P$ and $\mathcal{E}_Q$ for P and Q, we can estimate $d_{\mathrm{TV}}(P,Q)$ up to an additive 4ε error with $(1-\delta)$ probability using $O(\varepsilon^{-2} \log\left(\frac{1}{\delta}\right))$ samples from P and $O(\varepsilon^{-2} \log\left(\frac{1}{\delta}\right))$ evaluation queries to $\mathcal{E}_P$ and $\mathcal{E}_Q$.*

Now we will prove the main result of this section.

Theorem C.7. *If* BayesMarginal *has a randomized polynomial-time learning algorithm even restricted to Bayes nets of in-degree at most 2, then* DistCkt $\in$ BPP *and hence* SZK $\subseteq$ BPP.

Proof. Let $\mathcal{A}$ be the hypothetical randomized polynomial-time learning algorithm for BayesMarginal. We will design a randomized polynomial-time algorithm for DistCkt using $\mathcal{A}$ as follows.

Let C_n and D_n are the two Boolean circuits. Firstly, we assume without loss of generality all the AND, OR, NOT gates of C_n and D_n have at most 2 input gates; if not we can convert any gate with k inputs $(k-1)$ gates each with input 2 by stacking such gates. This increasing the circuit size (the number of gates) by at most a polynomial factor. Then, the two circuits can be interpreted as Bayes nets of in-degree at most 2, whose source nodes variables take random (Bernoulli(1/2)) and intermediate nodes follow deterministic functions. We slight abuse of notation and also denote by C_n and D_n the corresponding Bayes nets. Let S_C and S_D denote the sets of variables corresponding output gates of C_n and D_n. Note that $|S_C| = |S_D|$. Let C_{all} and D_{all} denote the joint distributions of all the variables (*all the gates*) of the Bayes nets C_n and D_n, respectively. Hence, the original samplable distributions C_n and D_n are the marginals of C_{all} and D_{all} respectively over the sets S_C and S_D.

We run $\mathcal{A}$ on $(C_{\mathrm{all}}, S_C, 1/40)$ to get an efficient ε-approximate evaluator $\mathcal{E}_C$ for C_n with high probability. Similarly we run $\mathcal{A}$ on $(C_{\mathrm{all}}, S_C, 1/40)$ to get an efficient ε-approximate evaluator $\mathcal{E}_D$ for D_n. Moreover, C_n can be sampled in randomized polynomial time. Hence from Theorem C.6, using $\mathcal{E}_C$, $\mathcal{E}_D$, and samples from C_n, we can approximate $d_{\mathrm{TV}}(C_n, D_n)$ up to an additive error 1/10 in polynomial-time with high probability. The algorithm finally accept if the approximate value of $d_{\mathrm{TV}}(C_n, D_n)$ is $> 1/2$ and reject if it is $\leq 1/2$. This shows that under the assumption on the learnability of BayesMarginal, DistCkt $\in$ BPP. □

We also show a NP-hardness result for getting a *multiplicatively* approximate evaluator for the marginal of a Bayes.

Definition C.8 (Efficient c-multiplicative evaluator circuit)**.** *A circuit $\mathcal{E}_P : \{0,1\}^n \to \mathbb{R}$ is called an efficient c-multiplicative evaluator circuit for a distribution P over $\{0,1\}^n$, if there exists a distribution $\widehat{P}$ over $\{0,1\}^n$ such that:*

1. *[Multiplicative approximation] $\widehat{P}(x)/P(x) \in [1/c, c]$ for some constant $c > 1$ and for any $x \in \{0,1\}^n$.*
2. *[Evaluation] for any $x \in \{0,1\}^n$, $\mathcal{E}_P$ on input x outputs $\widehat{P}(x)$ in* poly(n) *time.*

Definition C.9 (BayesMarginalMult)**.** *Let* BayesMarginalMult *be the following problem: given a Bayesian network P over the binary variables $\mathbf{V}$, and a $\mathbf{S} \subseteq \mathbf{V}$, return an efficient c-multiplicative evaluator circuit for the marginal distribution of P over S.*

We reduce from the well known NP-complete problem circuit evaluation problem.

Definition C.10 (CircEval)**.** *Given the encoding of a Boolean circuit $C : \{0,1\}^n \to \{0,1\}$ as input, decide whether there exists an x such that $C(x) = 1$ or not.*

Theorem C.11. CircEval *is* NP*-complete.*

We give a reduction from CircEval to BayesMarginalMult, showing that the later problem is unlikely to be in randomized polynomial-time.

Theorem C.12. *If* BayesMarginalMult *has a randomized polynomial time algorithm even for Bayes nets of indegree at most 2, then* CircEval $\in$ BPP *and hence* NP $\subseteq$ BPP*.*

Proof. Let $\mathcal{A}$ be a hypothetical randomized polynomial time algorithm for BayesMarginalMult. Let $C : \{0,1\}^n \to \{0,1\}$ be the instance of the CircEval problem. Let b denote the output bit of C. We also denote by C, the joint distribution of all its gates when its input bits are chosen randomly.

As argued in the proof of Theorem C.7, C is a Bayes net of indegree at most 2 over all its gates without loss of generality. We invoke $\mathcal{A}$ with the subset being $\{b\}$ and the mulplicative ratio being any constant $c > 1$. If C has no satisfying input, then $b \sim \text{Bernoulli}(0)$. If C has a satisfying input, $b \sim \text{Bernoulli}(p)$ for some $p \geq 1/2^n$. Therefore, a c-factor approximation of the bias of b would decide CircEval. □

D EXAMPLES

D.1 ID ALGORITHM

The following two examples are used here to illustrate the algorithm in action and how the different steps and the different recursive calls to the algorithm lead to the bigger picture of obtaining the probability $P_{\mathbf{x}}(\mathbf{y})$.

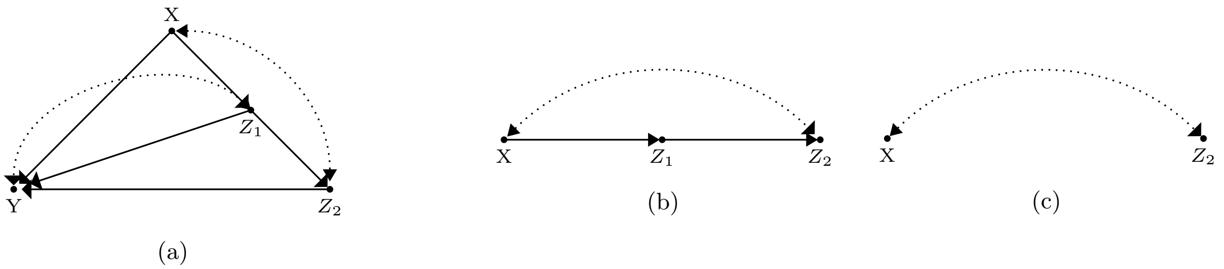
Example 1. Consider the ADMG shown in Figure 1a taken from Tian (2002) and the identification of $P_x(y, z_1, z_2)$ from the observational distribution on this graph. Note that the conditions in steps 1 and 2 of the ID Algorithm do not hold and hence step 3 gets invoked. Since the c-components of $G[\{Z_1, Z_2, Y\}]$ are $\{Y, Z_1\}$ and $\{Z_2\}$, we get the following formula: $P_x(y, z_1, z_2) = P_{x,z_2}(y, z_1) \cdot P_{x,y,z_1}(z_2)$.

For $P_{x,z_2}(y, z_1)$, the recursive procedure invokes step 4b to obtain the following equivalent expression $P(z_1 \mid x) \cdot P(y \mid z_1, z_2, x)$. For the other term $P_{x,y,z_1}(z_2)$, since Y is not an ancestor of Z_2, the intervention on Y is not particularly useful. This simplification is taken care by step 2, which reduces to the following query $P_{x,y,z_1}(z_2) = Q_{x,z_1}(z_2)$ over Graph 1b, where $Q(X, Z_1, Z_2) = \sum_y P(X, Z_1, Z_2, y) = P(X, Z_1, Z_2)$.

In the next recursive step, the algorithm invokes step 4c with $\mathbf{S}' = \{Z_2, X\}$, henceforth reduces to a new question $Q_{x,z_1}(z_2) = R_x(z_2)$ in Graph 1c, where $R(Z_2, X) = Q(X) \cdot Q(Z_2 \mid X, z_1)$. Finally, the algorithm invokes step 2, which obtains $R_x(z_2) = R(z_2) = \sum_{x'} Q(x') \cdot Q(z_2 \mid x', z_1)$.

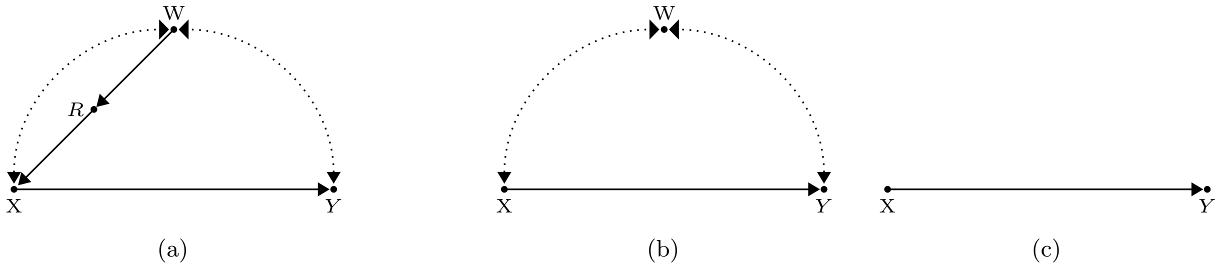
Example 2. Now consider the question of identifying $P_{x,r,w}(y)$ for Figure 2a taken from Jung et al. (2020b). Note that the conditions in steps 1, 2 and 3 do not hold for the required query, hence the algorithm directly invokes step 4c where $\mathbf{S}' = \{X, Y, W'\}$. This invocation boils down to the identification question of determining $Q_{w,x}(y)$ in Graph 2b, where $Q(W, X, Y) := P_r(W, X, Y) = P(W) \cdot P(X \mid W, r) \cdot P(Y \mid W, X, r)$. In the next recursive call, the algorithm will stop at step 2 resulting in the identification of $R_x(y)$ in Graph 2c, where $R(X, Y) = \sum_{w'} Q(w', X, Y) = \sum_{w'} P(w') \cdot P(X \mid w', r) \cdot P(Y \mid X, w', r)$. Finally, step 4b gets invoked in the next call which obtains the following expression:

$$\begin{aligned} R_x(y) &= R(y \mid x) \\ &= \frac{\sum_{w'} P(w') \cdot P(x \mid w', r) \cdot P(y \mid x, w', r)}{\sum_{w',y} P(w') \cdot P(x \mid w', r) \cdot P(y \mid x, w', r)}. \end{aligned}$$



(a) (b) (c)

Figure 1



(a) (b) (c)

Figure 2

D.2 EVALUATOR AND GENERATOR

We now illustrate our evaluator and generator with the help of the two simple examples presented before. In the following formulas, we use a (primed random variable) X' to denote an identical copy of the random variable X, to distinguish it from the intervention assignment $X = x$.

Example 1: Consider the graph in Figure 1a. We would like to learn the intervention $P_x(z_1, z_2, y)$. The following sequence of steps are taken in our algorithm.

1. The starting graph has two c-components (X, Z_2) and (Y, Z_1). Only (X, Z_2) contains an intervening variable. So, we'll assume (X, Z_1, Z_2) is α-strongly positive.
2. Step 3 is taken first and generates: $\mathrm{ID}((y, z_1), (x, z_2), P, G) * \mathrm{ID}(z_2, (x, y, z_1), P, G)$. Since, P is unchanged, we don't do anything.
3. The first ID takes step 4b and generates the formula: $P(y \mid x, z_1, z_2) \cdot P(z_1 \mid x)$. Note that we are learning this distribution jointly as a Bayes net, assuming α-strong positivity of only (X, Z_1, Z_2).
4. Y is marginalized out in step 2 from the second ID. Hence, we don't change the distribution. Next call is $\mathrm{ID}(z_2, (x, z_1), P(X, Z_1, Z_2), H)$, where H is the graph of Figure 1b.
5. Step 4c is taken next: $\mathrm{ID}(z_2, x, S(X, Z_2), T)$, where $S(X, Z_2) = P_{z_1}(X, Z_2) = P(X) \cdot P(Z_2 \mid z_1, X)$. We learn S up to point-wise $(1 \pm 6\varepsilon)$-factor using $\widetilde{O}(\alpha^{-2}\varepsilon^{-2})$ samples and store it as a table. T is the graph shown in Figure 1c.
6. Step 2 next prunes X from T. The distribution is a marginal and hence S is unchanged.
7. Step 1 generates the formula: $S(z_2)$.

At this point, we will be able to evaluate and sample $P(y \mid x, z_1, z_2) \cdot P(z_1 \mid x) \cdot S(z_2)$ using our Bayes net P and the explicit table for S.

Example 2: Consider the graph in Figure 2a. We are interested to learn the intervention $P_{(w,r,x)}(y)$. The following sequence of steps are taken in our algorithm.

1. The starting graph G has two c-components: $\{W, X, Y\}$ and $\{R\}$. Both these components contain an intervening variable and hence assumed α-strongly positive together with their parents.
2. First, step 4c gets invoked: which returns $\mathrm{ID}(y, (w, x), Q(W, X, Y), H)$, where $Q(W, X, Y) = P_r(W, X, Y) = P(W) \cdot P(X \mid W, r) \cdot P(Y \mid X, r, W)$. So, we learn Q as a point-wise $(1 \pm 9\varepsilon)$-approx. p.m.f. table using $\widetilde{O}(\alpha^{-2}\varepsilon^{-2})$ samples from P. The graph H is shown in Figure 2b.
3. Next, step 2 prunes W from H. The new distribution is the marginal of Q on X, Y and so Q's table is passed on unchanged. The new graph T is shown in Figure 2c. So, the next call is $\mathrm{ID}(y, x, Q(X, Y), T)$.
4. Finally, step 4b is taken, which generates the final formula $Q(y \mid x)$.

At this point, we will be able to evaluate and sample $Q(y \mid x)$ using our stored table for Q.