Distribution Learning Meets Graph Structure Sampling

Arnab Bhattacharyya
University of Warwick
arnab.bhattacharyya@warwick.ac.uk

Sutanu Gayen
IIT Kanpur
sutanugayen@gmail.com

Philips George John
CNRS@CREATE & Dept of Computer Science
National University of Singapore
philips.george.john@u.nus.edu

Sayantan Sen
Centre for Quantum Technologies
National University of Singapore
sayantan789@gmail.com

N. V. Vinodchandran
University of Nebraska-Lincoln
vinod@cse.unl.edu

Abstract

This work establishes a novel link between the problem of PAC-learning high-dimensional graphical models and the task of (efficient) counting and sampling of graph structures, using an online learning framework. The problem of efficiently counting and sampling graphical structures, such as spanning trees and acyclic orientations, has been a vibrant area of research in algorithms. We show that this rich algorithmic foundation can be leveraged to develop new algorithms for learning high-dimensional graphical models.

We present the first efficient algorithm for (both realizable and agnostic) learning of Bayes nets with a chordal skeleton. In particular, we present an algorithm that, given integers $k, d > 0$, error parameter $\varepsilon > 0$, an undirected chordal graph G on n vertices, and sample access to a distribution P^* on $[k]^n$; (1) returns a Bayes net $\widehat{P}$ with skeleton G and indegree d, whose KL-divergence from P^* is at most ε more than the optimal KL-divergence between P^* and any Bayes net with skeleton G and indegree d, (2) uses $\widetilde{O}(n^3 k^{d+1}/\varepsilon^2)$ samples from P^* and runs in time $\mathrm{poly}(n, k, \varepsilon^{-1})$ for constant d. Prior results in this spirit were for only for trees ($d = 1$, tree skeleton) via Chow-Liu, and in the realizable setting for polytrees (arbitrary d but tree skeleton). Thus, our result significantly extends the state-of-the-art in learning Bayes net distributions. We also establish new results for learning tree and polytree distributions.

1 Introduction

High-dimensional distributions are pivotal in contemporary machine learning, with widespread applications across various domains such as gene regulation networks [46, 16, 17, 40], protein signaling networks [39, 72, 75], brain connectivity networks [53, 78], and psychiatric symptom networks [13, 67, 82]. Probabilistic graphical models provide succinct representations of high-dimensional distributions over an exponentially large sample space such as $\mathbb{R}^n$ or $\{0,1\}^n$. These models leverage the limited dependence between component variables, encoded by a dependency graph, to describe joint probability distributions over a large set of variables in a succinct and interpretable manner. Probabilistic graphical models such as Bayesian networks, Ising models, and

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Gaussian graphical models are extensively utilized to model a wide range of data generation processes in practice (refer to [63, 79, 58] and the references therein). Learning distributions represented by these graphical models is a central challenge with significant theoretical and practical implications.

The present work focuses on learning an unknown Bayesian network from sample data. A Bayesian network (Bayes net) with n variables and alphabet size k is a probability distribution over $[k]^n$ defined by a directed acyclic graph (DAG) G on $[n]$. Each node represents a random variable, which is conditionally independent of non-descendants given its parents. By Bayes rule, the distribution factorizes into n conditional probabilities. If G has in-degree at most d, it requires at most nk^{d+1} parameters to describe the distribution, significantly reducing the descriptional complexity from k^n parameters required for an arbitrary distribution.

Learning Bayesian network distributions involves two steps: structure learning (identifying the dependency graph) and parameter learning (estimating conditional probability tables). Structure learning methods fall into two categories: constraint-based, which iteratively removes edges by testing for conditional independence, and score-based, which assigns scores to DAGs and frames structure recovery as an optimization problem, often solved using heuristics like greedy hill climbing. The current work broadly fits in the framework of score-based approach. However, instead of optimizing the score directly, we use the framework of *online learning* to reduce the problem to *sampling* from a family of high-dimensional structures.

In the online learning framework, the goal is to design a forecaster that observes a sequence of examples $x^{(1)}, x^{(2)}, \ldots, x^{(T)}$, and at each time (or round) t, outputs a prediction $\widehat{p}_t$ based only on $x^{(1)}, \ldots, x^{(t-1)}$. After predicting $\widehat{p}_t$, it observes $x^{(t)}$, and it incurs a loss $\ell(\widehat{p}_t, x^{(t)})$ for a *loss function* ℓ. The cumulative loss of the forecaster is benchmarked against that of a fixed and known set of *experts*. The goal of the algorithm is to minimize the *regret*, defined as the difference between the cumulative loss over all rounds and the loss it would incur if it were to follow the best expert. Online learning is a well-established field with a wide range of applications in theoretical computer science, including game theory, approximation algorithms, and complexity theory (see [42, 44, 8, 31, 69, 9, 60] and the references therein).

In distribution learning, the experts are all the possible candidate Bayesian networks (up to a sufficient discretization). The observations are random samples from the unknown distribution, and we set the loss function to be the negative log-likelihood $\ell(\widehat{p}, x) = -\log \widehat{p}(x)$. The primary obstacle in applying the online approach to distribution learning lies in ensuring computational efficiency. All the standard forecasting algorithms have running time at least linear in the number of experts. In our case, the experts are all the discretization of all candidate Bayes nets, which is exponentially many. A key insight of our work is the discovery of the close relationship between this computational challenge and the task of efficient counting and sampling of DAGs from a class of DAGs. This link allows us to transfer techniques and algorithms from the counting and sampling literature to the realm of distribution learning, leading to significant new results in learning Bayes net distributions.

2 Our Results

We first set up the framework of PAC-learning [76] for distributions; formal definitions appear in Appendix A. We use KL-divergence (denoted as D_{KL}) as the notion of similarity between probability distributions, and we will work with distributions on $[k]^n = \{1, \ldots, k\}^n$. Let $\mathcal{C}$ be a class of such distributions; in our applications, $\mathcal{C}$ will correspond to some family of Bayes nets.

For $\varepsilon > 0$, $A \geq 1$ and two distributions P and $\widehat{P}$ over $[k]^n$, we say $\widehat{P}$ is an (ε, A)-*approximation for* P *with respect to* $\mathcal{C}$ if $\mathsf{D}_{\mathsf{KL}}(P\|\widehat{P}) \leq A \cdot \min_{Q \in \mathcal{C}} \mathsf{D}_{\mathsf{KL}}(P\|Q) + \varepsilon$. When $A = 1$, we simply say $\widehat{P}$ is an ε-*approximation* of P. An algorithm is said to be an *agnostic PAC-learner for* $\mathcal{C}$ if for any $\varepsilon, \delta > 0$ and access to i.i.d. samples from an input distribution P^*, it outputs a distribution $\widehat{P}$ which is an ε-approximation for P^* with probability at least $1 - \delta$. If $\widehat{P}$ is not necessarily in $\mathcal{C}$, the algorithm is called *improper*; otherwise, it is called *proper*. Also, the *realizable setting* corresponds to the case when the input P^* is guaranteed to be in $\mathcal{C}$.

It is well-known (e.g., [12]) that given a DAG G, the minimum KL divergence between P^* and a Bayes net over G can be written as $J_{P^*} - \sum_{i \in V(G)} I(X_i; X_{\mathsf{pa}_G(i)})$, where $X \sim P^*$, I is the mutual

[1]There is a $\mathrm{polylog}(1/\delta)$ dependency here (as opposed to $1/\delta^2$ for the proper learner) hidden in $\tilde{O}(\cdot)$.

		Chordal graph with indegree $\leq d$ and known skeleton	Tree with unknown skeleton
Realizable	Proper	$\widetilde{O}\left(\max\left\{\frac{n^3}{\varepsilon^2\delta^2}, \frac{nk^{d+1}}{\varepsilon}\right\}\right)$	$\widetilde{O}\left(\max\left\{\frac{n^3}{\varepsilon^2\delta^2}, \frac{nk^2}{\varepsilon}\right\}\right)$
	Improper	$\widetilde{O}\left(\frac{nk^{d+1}}{\varepsilon\delta}\right)$	$\widetilde{O}\left(\frac{nk^2}{\varepsilon\delta}\right)$
Agnostic	Proper	$\widetilde{O}\left(\max\left\{\frac{n^3}{\varepsilon^2\delta^2}, \frac{nk^{d+1}}{\varepsilon}\right\}\right)$	$\widetilde{O}\left(\max\left\{\frac{n^3}{\varepsilon^2\delta^2}, \frac{nk^2}{\varepsilon}\right\}\right)$
	Improper	$\widetilde{O}\left(\max\left\{\frac{n^4}{\varepsilon^4}, \frac{nk^{d+1}}{\varepsilon}\right\}\right)$[1]	$\widetilde{O}\left(\max\left\{\frac{n^4}{\varepsilon^4}, \frac{nk^2}{\varepsilon}\right\}\right)$

Table 1: Our results: Sample complexities for (ε, δ)-PAC learning (the $\tilde{O}(\cdot)$ notation hides polylog factors)

information, and J_{P^*} is a constant independent of G. Hence, if $\mathcal{C}$ is the class of Bayes nets over DAGs of in-degree d, a natural strategy for designing agnostic learning for $\mathcal{C}$ is the following: First approximate the mutual information between any node and any set of d other nodes up to a suitable additive error. Next, maximize the sum of mutual informations between a node and its d parents, over all possible DAGs with in-degree d. Iterating over all possible DAG structures would then lead to an algorithm with sample complexity $\widetilde{O}(n^2k^{d+1}\varepsilon^{-2})$. However, this algorithm has exponential time complexity and the sample complexity is also suboptimal compared to known lower bounds.

In this work, we give an improper agnostic learning algorithm for Bayes nets with indegree d with sample complexity $\widetilde{O}(nk^{d+1}\varepsilon^{-1})$, which is sample-optimal upto polylogarithmic factors. The algorithm is computationally inefficient. Our main contribution is the design of sample and computational-efficient algorithms for new natural classes of Bayes nets, extending the state of the art. In particular, modifying our algorithm for general bounded-indegree Bayes nets, we give computational and time efficient algorithms for learning *chordal-structured Bayes nets with a known skeleton*. Efficient algorithms are currently known only for learning tree-structured distributions ([12, 30, 25]) and for learning polytree-structured distributions with a given skeleton ([24]).

An undirected graph is chordal if every cycle of length four or more has a chord (an edge connecting two non-adjacent vertices in the cycle). Chordal graphs form a significantly broader class than trees and encompass several well-studied graph families, including interval graphs and k-trees. Consequently, our results represent a major advancement in the state of the art for learning Bayesian network distributions. Beyond their structural significance, chordal graphs play a crucial role in the study of Bayesian networks particularly in causal Bayesian networks [4, 58]. We describe our results next. The sample complexities of our results are summarized in Table 1.

Learning with Known Chordal Skeleton The *skeleton* of a DAG refers to its underlying undirected graph. We consider Bayes nets having a known *chordal* skeleton with bounded indegree and present an efficient algorithm for learning such distributions.

Theorem 2.1. *Let G be an undirected chordal graph on n nodes, and suppose d is a fixed constant. Consider the problem of agnostically learning a distribution w.r.t the class of Bayes nets having skeleton G with indegree $\leq d$. There exist (i) an agnostic improper PAC-learner for this problem using $\widetilde{O}\left(\frac{n^4}{\varepsilon^4} + \frac{nk^{d+1}}{\varepsilon}\right)$ samples that returns an efficiently-samplable mixture of such Bayes nets, and (ii) an agnostic proper PAC-learner using $\widetilde{O}\left(\frac{n^3}{\varepsilon^2\delta^2} + \frac{nk^{d+1}}{\varepsilon}\right)$ samples that returns a single Bayes net. Both algorithms have $\mathrm{poly}(n, k, 1/\varepsilon, 1/\delta)$ running time.*

This is the first result yielding efficient algorithms for agnostic learning Bayes nets on *non-tree* skeletons without further distributional assumptions; see Section 4 for discussion of previous work.

For efficiently learning chordal and polytree distributions, we need to know the correct skeleton (underlying undirected graph). To the best of our knowledge, there is currently no computational hardness result regarding this. Additionally, there have been several works with the known skeleton assumption, even in the context beyond PAC distribution learning. [71] designed an FPT algorithm (in terms of total degree and treewidth) for counting the Markov Equivalence Classes with a given skeleton. On the practical side, several works for Bayes net structure learning first learn a skeleton from the data and then fix the orientations (e.g., see the survey [27], section 4.9.1). However, the

approach of first learning the skeleton and then performing the distribution learning does not have sound theoretical guarantees, since the distance measures in these two contexts are different.

A well-investigated class of Bayes nets is the class of *polytree* distributions: whose DAGs have tree (acyclic) skeletons. Polytrees are especially interesting because they admit fast exact inference [66]. [29] investigated the problem of learning polytree distributions in terms of the negative log-likelihood cost, and showed that it is NP-hard to get a 2-polytree (where indegree is ≤ 2) whose cost is at most c times that of the optimal 2-polytree for some constant $c > 1$, even if we have oracle access to the true entropies (equivalently, infinite samples). Our distribution learning algorithms in contrast achieve an *additive* approximation in the reverse-KL cost. As a direct corollary of the above theorem, we have the following result for bounded indegree polytrees with known skeleton.

Corollary 2.2. *Let $d > 0$ be a fixed constant and G be a given undirected tree. Consider the problem of agnostically learning a distribution w.r.t the class of Bayes nets having skeleton G with indegree $\leq d$, i.e. d-polytrees with skeleton G. There exist (i) an agnostic improper PAC-learner for this problem using $\widetilde{O}\left(\frac{n^4}{\varepsilon^4} + \frac{nk^{d+1}}{\varepsilon}\right)$ samples that returns an efficiently-samplable mixture of such polytrees, and (ii) an agnostic proper PAC-learner using $\widetilde{O}\left(\frac{n^3}{\varepsilon^2\delta^2} + \frac{nk^{d+1}}{\varepsilon}\right)$ samples that returns a single polytree. Both algorithms have $\mathrm{poly}(n, k, 1/\varepsilon, 1/\delta)$ running time.*

The closest related result is that of [24] who designed a PAC-learner in the realizable setting for polytrees with optimal[2] sample complexity $\widetilde{O}(nk^{d+1}\varepsilon^{-1}\log\delta^{-1})$. However, their analysis crucially uses the realizability assumption, and it was left as an open question in that work to find an efficient agnostic learner for polytrees. The above corollary answers this question.

Remark 2.3. We can bound the running time of our learning algorithms for chordal-structured Bayes nets (with known skeleton) as follows: At the outset, for every node $i \in [n]$ and for every choice of the $\leq d$ parents $\mathsf{pa}(i)$, we learn the conditional distribution associated with node i given that choice of parents $\mathsf{pa}(i)$. These are add-one conditional distributions computed from a sufficiently-large ($\widetilde{O}(nk^d/\varepsilon)$) set of samples. Subsequently, the learning algorithm focuses only on the combinatorial problem of learning an acyclic orientation. The running time contribution from the node-distribution learning part is $\widetilde{O}((\Delta k)^d dn^2/\varepsilon)$, where Δ is the maximum (undirected) degree of the skeleton. Here, $n\binom{\Delta}{d} \leq n\Delta^d$ (for $d \ll n$) bounds the number of all possible (node, parent-set) pairings and $\widetilde{O}(dnk^d/\varepsilon)$ is the time for computing a "good" add-one conditional distribution for a given node and parent-set. Note that the runtime is polynomial even if both Δ and d are $O(\log n)$. If d is unbounded, then the runtime can grow at an exponential $2^{d\log(\Delta k)}$ rate. Note that, for unbounded d, an exponential dependence on the runtime and sample complexity is inevitable since chordal-structured indegree-$(n-1)$ Bayes nets with a fixed skeleton (the complete graph on $[n]$) can capture arbitrary n-dimensional distributions (we do not use faithfulness or similar assumptions for distribution learning).

Learning Tree-structured Distributions By *tree-structured distributions* (or simply, trees when the meaning is clear), we mean Bayes nets whose underlying DAG has in-degree 1. They can be equivalently defined as undirected Markov models over (undirected) trees. The celebrated work of [25] developed a polynomial time algorithm for learning tree-structured distributions, if the algorithm is provided the exact mutual information between each pair of variables. PAC-learning guarantees with sample complexity bounds came later [30, 12], In particular, the highlight of these works was establishing that in the realizable setting, i.e., when the samples are being generated by a tree-structured distribution on $[k]^n$, the Chow-Liu algorithm is a PAC-learner with sample complexity $\widetilde{O}(nk^3/\varepsilon)$. While the dependence on n and ε is tight, it was left as an open question whether a better dependence on k is possible.

Our work answers this in the affirmative:

Theorem 2.4. *Let $\mathcal{C}$ be the family of tree-structured distributions over $[k]^n$. There exists an algorithm that for any $\varepsilon > 0$, given sample access to a distribution $P^* \in \mathcal{C}$, returns an ε-approximation $\widehat{P}$ of P^* with probability at least $2/3$ and uses $m = \widetilde{O}(nk^2\varepsilon^{-1})$ samples and $\mathrm{poly}(m)$ running time. The distribution $\widehat{P}$ is a mixture of distributions from $\mathcal{C}$ and is samplable in polynomial time.*

[2] Although not stated in the corollary above, in the realizable setting, our techniques also yield sample complexity with the optimal dependence on n, k, and ε.

Note that in contrast to Theorem 2.1, here, the algorithm does not know the true skeleton a priori. The output distribution $\widehat{P}$ is a mixture of exponentially many trees but can nevertheless be sampled in polynomial time by using the matrix-tree theorem, as we explain later. We note that the dependence of k^2 on the sample complexity is tight. This follows from [14, Theorem 13] (see also [33]), which proves that learning a Bayes net with in-degree d requires $\Omega(nk^{d+1})$ samples, and for tree Bayes nets, the in-degree being $d = 1$, requiring $\Omega(nk^2)$ samples. Learning with mixtures of trees has been studied before ([64, 61, 3, 70]) but in other contexts.

We also note that going beyond trees, the same approach allows us to develop polynomial sample and time algorithms for learning Bayes nets on an unknown DAG whose moralization is promised to have constant vertex cover size. Here, instead of sampling using the matrix-tree theorem, we can utilize a recent result of [50] to sample such DAGs. Details appear in Appendix F.

Why KL divergence? We briefly discuss why learning in KL divergence is relevant. The study of agnostic learning of distributions in KL divergence goes back to at least three decades ago by the works of [2] and [28, 29]. The authors argued that given random samples from an unknown distribution P, minimizing the KL divergence is the same as maximizing the log-likelihood in expectation, due to the following equation: $\mathsf{D}_{\mathsf{KL}}(P||Q) = -H(P) - \mathbb{E}_{x\sim P}[\log Q(x)]$, where $H(P)$ is the entropy of P. KL divergence also appears in the study of density estimation such as Yang-Barron's construction and covering complexity ([84, 81]). [56] also studied the complexity of distribution learning in terms of KL divergence and gave a coding-theoretic interpretation to choosing KL divergence as the distance function. Finally, several recent works have investigated the problem of learning high-dimensional distributions in this stronger KL divergence guarantee, such as [12, 32, 24, 11, 80].

3 Our Techniques

Online Learning Framework to Learning in reverse KL Given i.i.d. samples from a distribution P^*, we are trying to learn it. Roughly, for a random $x \sim P^*$, a good approximate Bayes net P should maximize the probability $P(x)$, or equivalently, minimize the expected log-likelihood $\mathbb{E}_{x\sim P^*}\left[\log \frac{1}{P(x)}\right]$. Keeping this in mind, we define the loss of any Bayes net $\widehat{P}$ predicted by the algorithm to be $\log \frac{1}{\widehat{P}(x)}$ for a sample x.

We follow the online learning framework. Here the algorithm $\mathcal{A}$ observes a set of samples $x^{(1)}, \ldots, x^{(T)} \sim P^*$ over T rounds from an unknown Bayes net P^*. The goal of the algorithm is to learn a distribution $\widehat{P}$ which is close to P^*. After observing each sample $x^{(t)}$, $\mathcal{A}$ predicts a Bayes net $\widehat{P}_t$ and incurs a loss of $\log \frac{1}{\widehat{P}_t(x^{(t)})}$ for this round. However, there is a set of experts $\mathcal{E}$ to help $\mathcal{A}$. For simplicity, we can assume each expert $E \in \mathcal{E}$ is simply one Bayes net among all possible Bayes nets. Had $\mathcal{A}$ stuck to any particular expert $E \in \mathcal{E}$, it would incur a total loss $\sum_{t=1}^{T} \log \frac{1}{E(x^{(t)})}$ over all the T rounds. The algorithm can change the experts in between or do some randomized strategy for choosing the expert among $\mathcal{E}$. Let $\widehat{P}_t$ be its prediction after round t. The regret is defined to be the difference between the loss of the algorithm and that of the best expert: $\sum_{t=1}^{T} \log \frac{1}{E(x^{(t)})} - \min_{E\in\mathcal{E}} \sum_{t=1}^{T} \log \frac{1}{E(x^{(t)})}$.

In our setting, the expert set consists of one Bayes net per DAG from the class of DAGs under consideration (e.g., acyclic orientations of a given skeleton). To associate a Bayes net with a DAG, we approximately learn the conditional distributions at each node using the *add-one* or *Laplace* estimator on a separate set of samples. We have the guarantee that these finitely many Bayes nets form a "cover" for the class of Bayes nets we wish to learn.

We relate the regret mentioned above to the expected average of the KL divergence over the rounds: $\mathbb{E}[\frac{1}{T}\mathsf{D}_{\mathsf{KL}}(P^*||\widehat{P}_t)]$. Once we control the average KL divergence, using the convexity of KL, we can show that the mixture distribution $\frac{1}{T}\sum_{t=1}^{T} \widehat{P}_t$ is also close to P^*. Finally, we translate the above bounds from expectation to high probability using McDiarmid's bounded difference inequality.

We use known bounds on the regret for two classic online learning algorithms: the Exponential Weighted Average (EWA) algorithm and the Randomized Weighted Majority (RWM) algorithm. EWA returns us the mixture $\frac{1}{T}\sum_{t=1}^{T} \widehat{P}_t$ which improperly learns P^* in (reverse) KL. RWM returns a

random Bayes net $\widehat{P}$ which properly learns P^* in expected KL. The pseudocode for these algorithms is given in Algorithm 1 and Algorithm 2.

Algorithm 1: EWA-based learning for Bayes nets

Input : $\mathcal{N} = \{P_1, \ldots, P_N\}, T$, hyperparameter $\eta > 0$.

Output : Sampler for $\widehat{P}$.

$w_{i,0} \leftarrow 1$ for each $i \in [N]$.
for $t \leftarrow 1$ *to* T **do**
  Observe sample $x^{(t)} \sim P^*$.
  Update $w_{i,t} \leftarrow w_{i,t-1} \cdot P_i(x^{(t)})^\eta$ for each $i \in [N]$.
function EWA-SAMPLER()
  Sample $t \leftarrow [T]$ uniformly, then sample $i \sim [N]$ with probability $\frac{w_{i,t-1}}{\sum_{j\in[N]} w_{j,t-1}}$.
  return $x \sim P_i$.
return EWA-SAMPLER `/* This is a sampler for` $\widehat{P}$. `*/`

Algorithm 2: RWM-based learning for Bayes nets

Input : $\mathcal{N} = \{P_1, \ldots, P_N\}, T$, hyperparameter $\eta > 0$.

Output : $\widehat{P} \in \mathcal{N}$.

$w_{i,0} \leftarrow 1$ for each $i \in [N]$.
for $t \leftarrow 1$ *to* T **do**
  Sample i_t from $[N]$ with $\Pr(i_t = i) = \frac{w_{i,t-1}}{\sum_{j\in[N]} w_{j,t-1}}$.
  Observe sample $x^{(t)} \sim P^*$.
  for $i \in [N]$ **do**
    $w_{i,t} \leftarrow w_{i,t-1} \cdot P_i(x^{(t)})^\eta$.
Sample t uniformly from $[T]$.
return $\widehat{P} \leftarrow P_{i_t}$.

Efficient Learning of Restricted Classes of Bayes Nets Our learning algorithm for Bayes nets mentioned above is sample-optimal but not time-efficient in general since the number of experts to be maintained is of exponential size. However, we observe that for special cases of Bayes nets, we can efficiently sample from the experts according to the randomized strategy of the algorithm. As a remark, the idea that the computational barrier of RWM or EWA may be side-stepped by developing efficient sampling schemes was also used in a recent work on fast equilibrium computation in structured games [8] and partly motivated our work.

To see the simplest example of this idea, suppose $\mathcal{P} = \{P_1, \ldots, P_N\}$ is a set of distributions over $[k]$, and let $\mathcal{P}^{\otimes n} = \mathcal{P} \times \mathcal{P} \times \cdots \times \mathcal{P}$ be a set of product distributions over $[k]^n$. Each element of $\mathcal{P}^{\otimes n}$ is indexed as $P_{\mathbf{i}}$ for $\mathbf{i} = (i_1, \ldots, i_n)$, so that $P_{\mathbf{i}}(x) = \prod_{j=1}^n P_{i_j}(x_j)$. The size of $\mathcal{P}^{\otimes n}$ is clearly N^n, so it is infeasible to work with it directly. The RWM algorithm maintains a distribution over $\mathcal{P}^{\otimes n}$, so that the probability that RWM picks $P_{\mathbf{i}}$ for its prediction $\widehat{P}_t$ at time t is proportional to $\prod_{s=1}^{t-1} P_{\mathbf{i}}(x^{(s)})^\eta$, where $x^{(s)}$ is the observed sample at time s and $\eta > 0$ is a parameter. Therefore:

$$\Pr_{\mathsf{RWM}}[\widehat{P}_t = P_{\mathbf{i}}] = \frac{\prod_{s=1}^{t-1} P_{\mathbf{i}}(x^{(s)})^\eta}{\sum_{\mathbf{i}'} \prod_{s=1}^{t-1} P_{\mathbf{i}'}(x^{(s)})^\eta} = \prod_{j=1}^n \frac{\prod_{s=1}^{t-1} P_{i_j}(x_j^{(s)})^\eta}{\sum_{i'_j} \prod_{s=1}^{t-1} P_{i'_j}(x_j^{(s)})^\eta}.$$

The crucial observation is that RWM maintains a product distribution over product distributions, and so we can sample each P_{i_j} from $\mathcal{P}$ independently.

When the underlying Bayes net is a tree, i.e. of indegree 1, we show that RWM samples a random rooted spanning arborescence from a weighted complete graph. The probability to output a particular arborescence A is proportional to $\prod_{e\in A} w_e$ where each w_e is a weight that can be explicitly computed in terms of the observed samples and the parameters of the algorithm. It is well-known that the *matrix-tree theorem* (more precisely, *Tutte's theorem*) for counting weighted arborescences can be used for this purpose, and hence, we obtain an alternative to the Chow-Liu algorithm for approximately learning a tree Bayes net efficiently and sample-optimally.

Next, we generalize our algorithm to polytree-structured Bayes nets where the underlying skeleton is acyclic. Here, we are assuming that the skeleton is given, so that the goal of the algorithm is to learn an acyclic orientation of the skeleton. For simplicity, suppose the skeleton is known to be the path. Given a particular orientation of the edges, we obtain a particular Bayes net structure. Once the structure is fixed, the conditional probability distribution corresponding to each edge parent→child is set according to the empirical statistics in a separate batch of samples. This will completely specify a Bayes net P, which can assign probability $P(x^{(t)})$ for the sample $x^{(t)}$. Therefore, we can also compute the total loss $\ell_P = \sum_{t=1}^T \log P(x^{(t)})^{-1}$. Then, each structure P will be chosen proportional

to $e^{-\eta \ell_P}$ in the RWM algorithm. In order to sample a particular Bayes net among the entire class using RWM, we need to first compute the normalization constant of the RWM sampler's distribution: $Z := \sum_{P \in \mathcal{P}} e^{-\eta \ell_P}$ over the class $\mathcal{P}$ of all (discretized) path Bayes nets. A particular path Bayes net P will be chosen with probability $e^{-\eta \ell_P}/Z$ by the RWM's sampler. We first show how to compute Z efficiently using dynamic programming.

We now show how to compute Z by induction on the set of vertices of the path. Suppose Z_j is the normalization constant obtained by only restricting to the first $j+1$ nodes in the path. That is, if $\mathcal{P}_j$ is the class of Bayes nets corresponding to all orientations of the undirected path on $j+1$ nodes, then $Z_j = \sum_{P \in \mathcal{P}_j} e^{-\eta \ell_P}$, where ℓ_P only computes the loss based on the first $j+1$ variables. For the induction, we maintain more refined information for each j. Let $\mathcal{P}_{j,\leftarrow}$ and $\mathcal{P}_{j,\rightarrow}$ be the class of all discretized Bayes nets on $j+1$ variables with a path skeleton and the last edge pointing left and right, respectively. Correspondingly, define $Z_{j,\leftarrow}$ and $Z_{j,\rightarrow}$; clearly, $Z_j = Z_{j,\leftarrow} + Z_{j,\rightarrow}$. Inductively, assume that $Z_{j,\leftarrow}$ and $Z_{j,\rightarrow}$ are already computed. We then need to compute $Z_{j+1,\leftarrow}$ and $Z_{j+1,\rightarrow}$.

If the $(j+1)$-th edge orients rightward, then the parents of nodes $1, \ldots, j+1$ do not change, while the new node $j+2$ has parent $j+1$. We can accommodate this new edge by simply adding the negative log of the conditional probability due to this new edge to the loss restricted to the first $j+1$ variables. We can compute $Z_{j+1,\rightarrow} = (Z_{j,\leftarrow} + Z_{j,\rightarrow}) e^{-\eta \Delta}$, by computing $\Delta = \sum_{t=1}^{T} \log P(x_{j+2}^{(t)} \mid x_{j+1}^{(t)})^{-1}$.

If the $(j+1)$-th edge orients leftward, the adjustment is slightly trickier as node $j+1$ will get a new parent $j+2$, while the new node $j+2$ has no parent. In that case, we need to first subtract out the previous sum of negative log conditional probabilities at $j+1$. Let us define:

$$\Delta_1 = \sum_{t=1}^{T} \log P(x_{j+1}^{(t)} \mid x_j^{(t)})^{-1}, \ \Delta_2 = \sum_{t=1}^{T} \log P(x_{j+1}^{(t)})^{-1}, \ \Delta_3 = \sum_{t=1}^{T} \log P(x_{j+1}^{(t)} \mid x_j^{(t)}, x_{j+2}^{(t)})^{-1},$$

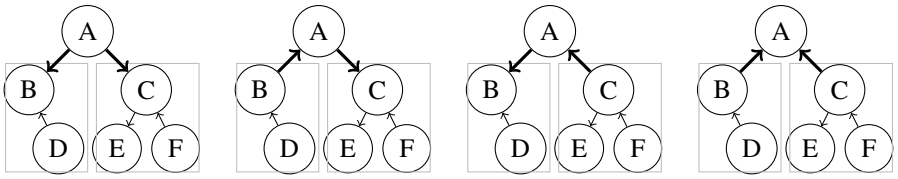
$$\Delta_4 = \sum_{t=1}^{T} \log P(x_{j+1}^{(t)} \mid x_{j+2}^{(t)})^{-1}, \Delta_5 = \sum_{t=1}^{T} \log P(x_{j+2}^{(t)})^{-1}.$$

If node j is not a parent of node $j+1$, then node $j+1$ contributed Δ_2 loss to $Z_{j,\leftarrow}$ while now it contributes Δ_4 loss to $Z_{j+1,\leftarrow}$. Otherwise, it contributed Δ_1 loss to $Z_{j,\rightarrow}$ while now it contributes Δ_3 loss to $Z_{j+1,\leftarrow}$. The new node $j+2$ contributes Δ_5 loss to $Z_{j+1,\leftarrow}$ independent of what happens to the other variables. Summarizing:

$$Z_{j+1,\leftarrow} = Z_{j,\leftarrow} \cdot e^{-\eta(\Delta_4 - \Delta_2 + \Delta_5)} + Z_{j,\rightarrow} \cdot e^{-\eta(\Delta_3 - \Delta_1 + \Delta_5)}.$$

It is easy to see that these updates can be performed efficiently using an appropriate dynamic programming table. Once we have computed the total sum $Z = Z_{n,\leftarrow} + Z_{n,\rightarrow}$, sampling a structure according to the sampler's distribution can simply be done by suitably unrolling the DP table.

Figure 1: Given a rooted polytree skeleton, for each node v, and for each fixed orientation of edges incident to v, we maintain the total weight of all consistent orientations of the subtree rooted at v. Above, the orientations of edges incident to B and C are fixed. This is needed when computing the weight for the subtree rooted at A, since in the first two panels, the in-degree of C change from 1 to 2, while in the second two panels, C's in-degree does not change.



The argument described above extends to learning bounded indegree polytrees and bounded indegree chordal graphs. For polytrees, the idea is illustrated in Figure 1. For chordal graphs, the algorithm first builds a clique tree decomposition and uses this structure for dynamic programming. The obvious issue with chordal graphs is that some orientations may lead to cycles, unlike the case for polytrees. However, chordal graphs enjoy certain nice property (see Lemma C.6) that allows us to independently perform weighted counting/sampling of acyclic orientations in each subtree of the clique tree.

Agnostic Learning via Maximum Likelihood Estimation An arguably more natural approach to PAC learning in KL divergence is to maximize the empirical log-likelihood (MLE) over a suitably-discretized class of distributions (e.g., see [41], Theorem 17). The Chow-Liu algorithm for tree distributions can also be viewed through this lens. Note however that, despite a long history of study, Chow-Liu is not known to attain the sample complexity bound in theorem 2.4 for learning trees in the realizable setting.

For the problem of learning polytrees and chordal-structured distributions, we can in fact adapt our algorithm to maximize likelihood and thus, get a sample complexity bound which is comparable to our Theorem B.12 (up to log factors) for proper learning in KL. But it does not yield the near-optimal bounds (for constant failure probability) that we get for improper learning (Theorem B.11) in the realizable case. The challenge of implementing the Maximum Likelihood (ML) algorithm over an exponential-sized class of distributions is *efficiency* — a naive approach would take exponential time. The dynamic programming algorithms that we develop for efficient weighted counting and sampling of DAG structures (which we use to implement EWA / RWM) can also be used to implement MLE efficiently for polytree/chordal-structured distributions given the skeleton. We give an outline of this in Appendix G.

4 Related Works

[51] gave the first sample complexity bounds for agnostic learning of a Bayes net with known structure from samples in KL divergence. This work also gave an efficient algorithm for special cases such as trees using the classical Chow-Liu algorithm. Subsequently, [28] gave an efficient algorithm for learning an unknown Bayes net (discrete and Gaussian) on a fixed structure. This result was improved to a sample-optimal learning of fixed-structure Bayes nets in [10, 12].

The general problem of distribution learning of Bayes networks with unknown DAG structure has remained elusive so far. It has not been shown to be NP-hard, although some related problems and specific approaches are NP-hard [21, 23, 26, 55]. Many of the early approaches required *faithfulness*, a condition which permits learning of the Markov equivalence class, e.g. [73, 22, 47]. Finite sample complexity of such algorithms has also been studied, e.g. [45]. Specifically for polytrees, [68, 49] studied recovery of the DAG for polytrees under the infinite sample regime and in the realizable setting, while [24] gave finite sample complexity for this problem. [48] studied the more general problem of learning Bayes nets, and their sufficient conditions simplified in the setting of polytrees.

A notable prior work in the context of the current paper is the work by [1], which also explores the improper learning of Bayesian networks with polynomial sample and time complexities. However, our research diverges from theirs in three critical ways: firstly, their study does not offer any proper learning algorithms; secondly, it lacks agnostic learning guarantees; and thirdly, their approach does not achieve optimal sample complexity in the realizable setting. While they do demonstrate a "graceful degradation" as inputs deviate from the hypothesis class, this does not equate to a definitive agnostic learning guarantee as provided in our work. On a positive note, their research does attain polynomial sample and time complexities for learning any Bayesian network with a bounded total degree in the realizable setting. It is worth noting that our results for distributions with chordal skeleton are applicable even when the total degree is unbounded, provided that the indegree remains bounded, a scenario where the findings of Abbeel, Koller, and Ng would not be applicable.

Online Learning of Structured Distributions The approach of using the online learning framework for distribution learning has been considered in the literature [18, 83, 77]. These works use EWA algorithm and output the mixture distribution. However, they primarily focus on minimizing the number of samples, and are not computationally efficient in general. Since we are interested in computationally efficient learning of high-dimensional distributions, their approaches do not directly translate to our context. The closest we get is the *Sparsitron* algorithm by Klivans and Meka ([57]) which learns an unknown Ising model from samples. However, Sparsitron is typical to Ising models where the conditional distribution at any component follows a logistic regression model which the Sparsitron algorithm learns.

Although not for distribution learning, a similar use of the multiplicative weights update method appears in Freund and Schapire's well-known AdaBoost algorithm ([43]) where the algorithm

[3]This work studies a more general notion of factor graphs.

	Structure	Efficient?	Agnostic?	Additional assumptions
[12]	Tree	Yes	Yes	None
[24]	Polytree	Yes	No	Known skeleton
[1]	Bounded total degree [3]	Yes	No	None
[14]	Bounded in-degree	No	No	None
Our results	Tree	Yes	Yes	None
	Chordal skeleton, bounded in-degree	Yes	Yes	Known skeleton

Table 2: Comparison with existing works.

implicitly creates a sequence of probability measures. Later work on the hard-core lemma, such as [7], explicitly focus on efficient sampling from the iterates of multiplicative weights update.

Robust Learning In the field of distribution learning, it is commonly assumed that all samples are consistently coming from an unknown distribution. However, real-world conditions often challenge this assumption, as samples may become corrupted—either removed or substituted by samples from an alternate distribution. Under such circumstances, the theoretical assurances of traditional algorithms may no longer apply. This discrepancy has spurred interest in developing robust learning algorithms capable of tolerating sample corruption. Recent years have seen notable advancements in this area, including the development of algorithms for robustly learning Bernoulli product distributions [37], and enhancing the robustness of learning Bayes nets [20]. See [62, 34, 35, 6, 52, 59, 35, 36, 19, 15] and the references therein for a sample of current works in this area. These works primarily focus on guarantees with respect to the total variation distance.

Of particular relevance is the TV-*contamination model*. Here, if the distribution to be learnt is P, one gets samples from a 'contaminated' distribution Q with $\mathsf{d}_{\mathsf{TV}}(P, Q) \leq \eta$. Note that this is a stronger model than *Huber contamination* ([54]), where the noise is restricted to be additive, meaning that an adversary adds a limited number of noisy points to a set of uncontaminated samples from P.

One can interpret our results using a KL-*contamination model*. If the distribution to be learnt is an unknown P promised to belong to a class $\mathcal{C}$, the contaminated distribution Q is some distribution satisfying $\mathsf{D}_{\mathsf{KL}}(Q\|P) \leq \eta$. The noise is again non-additive, but the model is weaker than TV-contamination. Any (η, A) approximation for Q with respect to $\mathcal{C}$ yields a distribution $\widehat{P}$ such that $\mathsf{D}_{\mathsf{KL}}(Q\|\widehat{P}) \leq (A+1)\eta$. Therefore, we get that for Hellinger distance:

$$\mathsf{H}(P, \widehat{P}) \leq \mathsf{H}(P, Q) + \mathsf{H}(\widehat{P}, Q) \leq \sqrt{\eta} + \sqrt{(A+1)\eta} \leq \sqrt{(2A+3)\eta}.$$

Similarly, one can also bound $\mathsf{d}_{\mathsf{TV}}(P, \widehat{P}) = O(\sqrt{\eta})$ for constant A. To the best of our knowledge, the KL-contamination model has not been explicitly considered before, but if one were to directly apply the results of [20] with the assumption that $\mathsf{D}_{\mathsf{KL}}(Q\|P) \leq \eta$, one would obtain a distribution $\widehat{P}$ such that $\mathsf{d}_{\mathsf{TV}}(P, \widehat{P}) = O(\sqrt{\eta \log 1/\eta})$, worse than ours by a $\sqrt{\log 1/\eta}$ factor which seems unavoidable using their approach [38]. Moreover, their results require that $\mathcal{C}$ be a class of *balanced* Bayes nets, a technical condition which is not needed for our analysis [4]. However, we would like to note that $\mathsf{D}_{\mathsf{KL}}(Q||P)$ can be large as compared to $\mathsf{d}_{\mathsf{TV}}(Q, P)$, so this holds when $\mathsf{D}_{\mathsf{KL}}(Q||P)$ is small.

5 Discussion

Conclusion In this work, we established a novel connection between distribution learning and graphical structure sampling algorithms via the framework of online learning. Leveraging this connection, we designed efficient algorithms for agnostically learning bounded indegree chordal-structured distributions, with polynomial sample complexity. These algorithms only require knowledge of the

[4] A Bayes net is said to be *c-balanced* for some $c > 0$ if all conditional probability table values $\in [c, 1-c]$.

distribution's skeleton, without needing information on the edge directions. Since polytree-structured distributions are a subset of chordal-structured distributions, our result also gives new results on the well-studied problem of learning polytree-structured distributions. Interestingly, our method also leads to a new algorithm for learning tree-structured distributions, which is significantly different from the extremely well studied Chow-Liu algorithm. Finally, we also give an improper learning algorithm that, with probability $2/3$, gives an $(\varepsilon, 3)$-approximation with respect to tree-structured distributions, which has a quadratic sample complexity advantage over Chow-Liu.

Organization of the supplementary material Due to shortage of space, the rest of the paper is presented in the supplementary material. It is organized as follows. In Appendix A, we present the preliminaries required for this work. Appendix B establishes the connection between regret in online learning to KL divergence in the scenario of agnostic learning of distributions. It also presents several necessary techniques from online learning along with the EWA and RWM algorithms that will be used later in our work. In Appendix C, we present our results on learning chordal-structured distributions. In Appendix D, we discuss our results on learning tree-structured distributions and present our alternative proper learning algorithm. In Appendix E, we give the lower bound of learning tree-structured distributions. In Appendix F, we design efficient learning algorithms for Bayes nets over graphs with bounded vertex cover. Finally, in Appendix G, we outline how our algorithms can be adapted to efficiently compute maximum likelihood.

6 Open Problems

Our work opens up several interesting research avenues.

- An intriguing question is whether we can extend our result for chordal graphs of bounded indegree to general graphs of bounded treewidth and bounded indegree. Interestingly, [74] showed that counting the number of acyclic orientations reduces to the evaluation of the Tutte polynomial at the point (2,0), and the Tutte polynomial can be evaluated efficiently for bounded treewidth graphs [65, 5]. This is relevant because the weights that EWA/RWM maintain are in some sense a weighted count of the number of acyclic orientations of the skeleton. However, we did not find a deletion-contraction recurrence for these weights, and so their connection to the Tutte polynomial is unclear.
- Another important follow-up direction for learning Bayes nets would be to search over *Markov equivalence classes* rather than DAG's. A Markov equivalence class corresponds to the set of DAGs that represent the same class of Bayes nets, and they can be represented as partially directed graphs (*essential graphs*) that satisfy some special graphical properties. It would be interesting to explore if the structure of essential graphs can be used to speed up weighted counting and sampling; indeed, a very recent work by [71] gives a polynomial time algorithm for uniformly sampling an essential graph that is consistent with a given skeleton.
- What is the role of *approximate sampling* in the context of distribution learning? So far, in this work, we have only used exact sampling algorithms for spanning arborescences and acyclic orientations. Can Markov chain techniques be brought to good use here? Our work further motivates settling the complexity status of approximately counting the number of acyclic orientations of an undirected graph; this question is a long-standing open problem in the counting/sampling literature.
- Finally, while we have restricted ourselves to learning Bayes nets here, our framework is quite general and also applies to learning other classes of distributions, such as Ising models and factor models. We leave these questions for future work.

Acknowledgments and Disclosure of Funding

The authors would like to thank the anonymous reviewers for their comments which improved the presentation of the paper. AB and PGJ's research were supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. SS's research is supported by the NRF Investigatorship award (NRF-NRFI10-2024-0006) and CQT Young Researcher Career Development Grant (25-YRCDG-SS).

AB and SS were also supported by National Research Foundation Singapore under its NRF Fellowship Programme (NRF-NRFFAI1-2019-0002). AB was additionally supported by an Amazon Research Award and a Google South/Southeast Asia Research Award. SG's work is partially supported by the SERB CRG Award CRG/2022/007985. NVV's work was supported in part by NSF CCF grants 2130608 and 2342244 and a UNL Grand Challenges Catalyst Competition Grant.

We would like to thank Debojyoti Dey, a Ph.D. student at IIT Kanpur, for discussions regarding robust learning algorithms in high dimensions. AB would also like to thank Daniel Beaglehole for a short meeting which seeded the idea for this work.

References

[1] Pieter Abbeel, Daphne Koller, and Andrew Y Ng. Learning factor graphs in polynomial time and sample complexity. *Journal of Machine Learning Research*, 7:1743–1788, 2006.

[2] Naoki Abe, Manfred K Warmuth, and Jun-ichi Takeuchi. Polynomial learnability of probabilistic concepts with respect to the kullback-leibler divergence. In *Proceedings of the Fourth Annual Workshop on Computational Learning Theory*, pages 277–289, 1991.

[3] Animashree Anandkumar, Daniel J. Hsu, Furong Huang, and Sham M. Kakade. Learning mixtures of tree graphical models. In *Advances in Neural Information Processing Systems*, pages 1061–1069, 2012.

[4] Steen A Andersson, David Madigan, and Michael D Perlman. A characterization of markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541, 1997.

[5] Artur Andrzejak. An algorithm for the tutte polynomials of graphs of bounded treewidth. *Discrete mathematics*, 190(1-3):39–54, 1998.

[6] Sivaraman Balakrishnan, Simon S Du, Jerry Li, and Aarti Singh. Computationally efficient robust sparse estimation in high dimensions. In *Conference on Learning Theory*, pages 169–212, 2017.

[7] Boaz Barak, Moritz Hardt, and Satyen Kale. The uniform hardcore lemma via approximate bregman projections. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 1193–1200, 2009.

[8] Daniel Beaglehole, Max Hopkins, Daniel Kane, Sihan Liu, and Shachar Lovett. Sampling equilibria: Fast no-regret learning in structured games. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, (SODA)*, pages 3817–3855. SIAM, 2023.

[9] Soheil Behnezhad, Avrim Blum, Mahsa Derakhshan, MohammadTaghi Hajiaghayi, Christos H Papadimitriou, and Saeed Seddighin. Optimal strategies of blotto games: Beyond convexity. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 597–616, 2019.

[10] Arnab Bhattacharyya, Sutanu Gayen, Kuldeep S Meel, and NV Vinodchandran. Efficient distance approximation for structured high-dimensional distributions via learning. *Advances in Neural Information Processing Systems*, 33:14699–14711, 2020.

[11] Arnab Bhattacharyya, Davin Choo, Rishikesh Gajjala, Sutanu Gayen, and Yuhao Wang. Learning sparse fixed-structure gaussian bayesian networks. In *International Conference on Artificial Intelligence and Statistics*, pages 9400–9429, 2022.

[12] Arnab Bhattacharyya, Sutanu Gayen, Eric Price, Vincent YF Tan, and NV Vinodchandran. Near-optimal learning of tree-structured distributions by chow and liu. *SIAM Journal on Computing*, 52(3):761–793, 2023.

[13] Lynn Boschloo, Claudia D van Borkulo, Mijke Rhemtulla, Katherine M Keyes, Denny Borsboom, and Robert A Schoevers. The network structure of symptoms of the diagnostic and statistical manual of mental disorders. *PloS one*, 10(9):e0137621, 2015.

[14] Johannes Brustle, Yang Cai, and Constantinos Daskalakis. Multi-item mechanisms without item-independence: Learnability via robustness. In *Proceedings of the 21st ACM Conference on Economics and Computation*, pages 715–761, 2020.

[15] Clément Canonne, Samuel B Hopkins, Jerry Li, Allen Liu, and Shyam Narayanan. The full landscape of robust mean testing: Sharp separations between oblivious and adaptive contamination. In *Proceedings of the 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2159–2168, 2023.

[16] Robert Castelo and Alberto Roverato. A robust procedure for gaussian graphical model search from microarray data with p larger than n. *Journal of Machine Learning Research*, 7:2621–2650, 2006.

[17] Robert Castelo and Alberto Roverato. Reverse engineering molecular regulatory networks from microarray data with qp-graphs. *Journal of Computational Biology*, 16(2):213–227, 2009.

[18] Olivier Catoni. The mixture approach to universal model selection. *In Preprints of École Normale Supérieure*, 1997.

[19] Yu Cheng and Honghao Lin. Robust learning of fixed-structure bayesian networks in nearly-linear time. In *9th International Conference on Learning Representations (ICLR)*, 2021.

[20] Yu Cheng, Ilias Diakonikolas, Daniel Kane, and Alistair Stewart. Robust learning of fixed-structure bayesian networks. *Advances in Neural Information Processing Systems*, 31:10304–10316, 2018.

[21] David Maxwell Chickering. Learning bayesian networks is np-complete. In *Learning from Data - Fifth International Workshop on Artificial Intelligence and Statistics (AISTATS)*, pages 121–130, 1995.

[22] David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3:507–554, 2002.

[23] Max Chickering, David Heckerman, and Chris Meek. Large-sample learning of bayesian networks is np-hard. *Journal of Machine Learning Research*, 5:1287–1330, 2004.

[24] Davin Choo, Joy Qiping Yang, Arnab Bhattacharyya, and Clément L. Canonne. Learning bounded-degree polytrees with known skeleton. In *International Conference on Algorithmic Learning Theory*, volume 237, pages 402–443, 2024.

[25] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. Inf. Theory*, 14(3):462–467, 1968.

[26] Paul Dagum and Michael Luby. An optimal approximation algorithm for bayesian inference. *Artificial Intelligence*, 93(1):1–28, 1997.

[27] Ronan Daly, Shen Qiang, and Stuart Aitken. Learning bayesian networks: Approaches and issues. *Knowledge Engineering Review*, 26(2):99–127, June 2011.

[28] Sanjoy Dasgupta. The sample complexity of learning fixed-structure bayesian networks. *Machine Learning*, 29:165–180, 1997.

[29] Sanjoy Dasgupta. Learning polytrees. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 134–141, 1999.

[30] Constantinos Daskalakis and Qinxuan Pan. Sample-optimal and efficient learning of tree ising models. In *In Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 133–146, 2021.

[31] Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a nash equilibrium. *Communications of the ACM*, 52(2):89–97, 2009.

[32] Constantinos Daskalakis, Nishanth Dikkala, and Gautam Kamath. Testing ising models. *IEEE Transactions on Information Theory*, 65(11):6829–6852, 2019.

[33] Luc Devroye and Gábor Lugosi. *Combinatorial Methods in Density Estimation*. Springer New York, NY, 2001.

[34] Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Being robust (in high dimensions) can be practical. In *International Conference on Machine Learning*, pages 999–1008, 2017.

[35] Ilias Diakonikolas, Gautam Kamath, Daniel M Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robustly learning a gaussian: Getting optimal error, efficiently. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2683–2702, 2018.

[36] Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Learning geometric concepts with nasty noise. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1061–1073, 2018.

[37] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high-dimensions without the computational intractability. *SIAM Journal on Computing*, 48(2):742–864, 2019.

[38] Ilias Diakonikolas, Daniel M Kane, and Yuxin Sun. Optimal sq lower bounds for robustly learning discrete product distributions and ising models. In *Conference on Learning Theory*, pages 3936–3978, 2022.

[39] Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[40] David Edwards, Gabriel CG De Abreu, and Rodrigo Labouriau. Selecting high-dimensional mixed graphical models using minimal aic or bic forests. *BMC bioinformatics*, 11:1–13, 2010.

[41] Jon Feldman, Ryan O'Donnell, and Rocco A. Servedio. Learning mixtures of product distributions over discrete domains. *SIAM Journal on Computing*, 37(5):1536–1564, 2008.

[42] Yoav Freund and Robert E Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the ninth annual conference on Computational learning theory*, pages 325–332, 1996.

[43] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[44] Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1-2):79–103, 1999.

[45] Nir Friedman and Zohar Yakhini. On the sample complexity of learning bayesian networks. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Articial Intelligence (UAI)*, pages 274–282, 1996.

[46] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using bayesian networks to analyze expression data. In *Proceedings of the fourth annual international conference on Computational molecular biology*, pages 127–135, 2000.

[47] Nir Friedman, Iftach Nachman, and Dana Pe'er. Learning bayesian network structure from massive datasets: The "sparse candidate" algorithm. *arXiv preprint arXiv:1301.6696*, 2013.

[48] Ming Gao and Bryon Aragam. Efficient bayesian network structure learning via local markov boundary search. *Advances in Neural Information Processing Systems*, 34:4301–4313, 2021.

[49] Dan Geiger, Azaria Paz, and Judea Pearl. Learning causal trees from dependence information. In *AAAI*, pages 770–776, 1990.

[50] Juha Harviainen and Mikko Koivisto. Revisiting bayesian network learning with small vertex cover. In *Uncertainty in Artificial Intelligence*, pages 819–828. PMLR, 2023.

[51] Klaus-U Höffgen. Learning and robust learning of product distributions. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 77–83, 1993.

[52] Samuel B Hopkins and Jerry Li. Mixture models, robustness, and sum of squares proofs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1021–1034, 2018.

[53] Shuai Huang, Jing Li, Liang Sun, Jieping Ye, Adam Fleisher, Teresa Wu, Kewei Chen, Eric Reiman, Alzheimer's Disease NeuroImaging Initiative, et al. Learning brain connectivity of alzheimer's disease by sparse inverse covariance estimation. *NeuroImage*, 50(3):935–949, 2010.

[54] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pages 492–518. Springer, 1992.

[55] David R. Karger and Nathan Srebro. Learning markov networks: maximum bounded tree-width graphs. In *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms (SODA)*, pages 392–401, 2001.

[56] Michael J. Kearns, Yishay Mansour, Dana Ron, Ronitt Rubinfeld, Robert E. Schapire, and Linda Sellie. On the learnability of discrete distributions. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, pages 273–282, 1994.

[57] Adam R. Klivans and Raghu Meka. Learning graphical models using multiplicative weights. In *Symposium on Foundations of Computer Science (FOCS)*, pages 343–354, 2017.

[58] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[59] Pravesh K Kothari, Jacob Steinhardt, and David Steurer. Robust moment estimation and improved clustering via sum of squares. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1035–1046, 2018.

[60] Dan Kovenock and Brian Roberson. Coalitional colonel blotto games with application to the economics of alliances. *Journal of Public Economic Theory*, 14(4):653–676, 2012.

[61] M. Pawan Kumar and Daphne Koller. Learning a small mixture of trees. In *Advances in Neural Information Processing Systems*, pages 1051–1059, 2009.

[62] Kevin A Lai, Anup B Rao, and Santosh Vempala. Agnostic estimation of mean and covariance. In *Proceedings of the 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 665–674, 2016.

[63] Steffen L Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.

[64] Marina Meila and Michael I. Jordan. Learning with mixtures of trees. *Journal of Machine Learning Research*, 1:1–48, 2000.

[65] Steven D Noble. Evaluating the tutte polynomial for graphs of bounded tree-width. *Combinatorics, probability and computing*, 7(3):307–321, 1998.

[66] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier, 2014.

[67] Victor Peralta, Gustavo J Gil-Berrozpe, Julián Librero, Ana Sánchez-Torres, and Manuel J Cuesta. The symptom and domain structure of psychotic disorders: a network analysis approach. *Schizophrenia Bulletin Open*, 1(1):sgaa008, 2020.

[68] George Rebane and Judea Pearl. The recovery of causal poly-trees from statistical data. *Int. J. Approx. Reason.*, 2(3):341, 1988.

[69] Aviad Rubinstein. Settling the complexity of computing approximate two-player nash equilibria. *ACM SIGecom Exchanges*, 15(2):45–49, 2017.

[70] Nikil Roashan Selvam, Honghua Zhang, and Guy Van den Broeck. Mixtures of all trees. In *International Conference on Artificial Intelligence and Statistics*, pages 11043–11058. PMLR, 2023.

[71] Vidya Sagar Sharma. A fixed-parameter tractable algorithm for counting markov equivalence classes with the same skeleton. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, (AAAI)*, pages 20532–20539, 2024.

[72] Victor Spirin and Leonid A Mirny. Protein complexes and functional modules in molecular networks. *Proceedings of the national Academy of sciences*, 100(21):12123–12128, 2003.

[73] Peter Spirtes and Clark Glymour. An algorithm for fast recovery of sparse causal graphs. *Social science computer review*, 9(1):62–72, 1991.

[74] Richard P Stanley. Acyclic orientations of graphs. *Discrete Mathematics*, 5(2):171–178, 1973.

[75] John Thomas, Naren Ramakrishnan, and Chris Bailey-Kellogg. Graphical models of residue coupling in protein families. In *Proceedings of the 5th international workshop on Bioinformatics*, pages 12–20, 2005.

[76] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[77] Dirk van der Hoeven, Nikita Zhivotovskiy, and Nicolò Cesa-Bianchi. High-probability risk bounds via sequential predictors. *arXiv preprint arXiv:2308.07588*, 2023.

[78] Gaël Varoquaux, Alexandre Gramfort, Jean-Baptiste Poline, and Bertrand Thirion. Brain covariance selection: better individual functional connectivity models using population prior. In *Advances in Neural Information Processing Systems*, volume 23, pages 2334–2342, 2010.

[79] Martin J Wainwright, Michael I Jordan, et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

[80] Yuhao Wang, Ming Gao, Wai Ming Tai, Bryon Aragam, and Arnab Bhattacharyya. Optimal estimation of gaussian (poly) trees. In *International Conference on Artificial Intelligence and Statistics*, pages 3619–3627. PMLR, 2024.

[81] Yihong Wu. Lecture notes on information-theoretic methods for high-dimensional statistics. *Lecture Notes for ECE598YW (UIUC)*, 16:15, 2017.

[82] Shanghong Xie, Erin McDonnell, and Yuanjia Wang. Conditional gaussian graphical model for estimating personalized disease symptom networks. *Statistics in medicine*, 41(3):543–553, 2022.

[83] Yuhong Yang. Mixing strategies for density estimation. *Annals of Statistics*, pages 75–87, 2000.

[84] Yuhong Yang and Andrew Barron. Information-theoretic determination of minimax rates of convergence. *Annals of Statistics*, pages 1564–1599, 1999.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract states some learning sample complexities (in concise form) which match the theorems, and the introduction section has the formal theorem statements, problem definitions, etc.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: It is a theoretical paper, and we give the problem statements and theorems precisely including all assumptions.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All these things are included, with complete proofs in the supplementary appendices.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [NA]

   Justification: The paper has no experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [NA]

Justification: The paper has no experimental results.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [NA]

   Justification: The paper has no experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: The paper has no experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [NA]

   Justification: The paper has no experimental results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors have reviewed the guidelines. We use no human subjects, no datasets, and the paper is entirely theoretical in nature and only considers abstract problems (no societal impact).

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The work is entirely theoretical on an abstract problem, and there is no societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not propose datasets or models, only algorithms.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: The paper does not use existing assets, except for other papers which have been properly cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets, other than the algorithms documented in the paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not use crowdsourcing/human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper is entirely theoretical and does not require such approvals.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: We have not used LLMs at all in obtaining the algorithms/theorems/other results in the paper.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.