Journal of Machine Learning Research 23 (2022) 1-38 Submitted 3/21; Revised 2/22; Published 6/22

Active Structure Learning of Bayesian Networks in an Observational Setting

Noa Ben-David BENDANOA@POST.BGU.AC.IL
Sivan Sabato SABATOS@CS.BGU.AC.IL
Department of Computer Science
Ben-Gurion Univesity of the Negev
Beer Sheva 8410501, Israel.

Editor: Andreas Krause

Abstract

We study active structure learning of Bayesian networks in an observational setting, in which there are external limitations on the number of variable values that can be observed from the same sample. Random samples are drawn from the joint distribution of the network variables, and the algorithm iteratively selects which variables to observe in the next sample. We propose a new active learning algorithm for this setting, that finds with a high probability a structure with a score that is ϵ-close to the optimal score. We show that for a class of distributions that we term *stable*, a sample complexity reduction of up to a factor of $\widetilde{\Omega}(d^3)$ can be obtained, where d is the number of network variables. We further show that in the worst case, the sample complexity of the active algorithm is guaranteed to be almost the same as that of a naive baseline algorithm. To supplement the theoretical results, we report experiments that compare the performance of the new active algorithm to the naive baseline and demonstrate the sample complexity improvements. Code for the algorithm and for the experiments is provided at `https://github.com/noabdavid/activeBNSL`.

Keywords: Active learning, sample complexity, Bayesian networks, graphical models, combinatorial optimization.

1. Introduction

In this work, we study active structure learning of Bayesian networks in an observational (that is, a non-interventional) setting, in which there are external limitations on the number of variable values that can be observed from the same sample. Bayesian networks are a popular modeling tool used in various applications, such as medical analysis (Haddawy et al., 2018; Xu et al., 2018), human activity models (Liu et al., 2018) and engineering (Rovinelli et al., 2018; Cai et al., 2019), among others. A Bayesian network is a graphical model that encodes probabilistic relationships among random variables. The structure of a Bayesian network is represented by a Directed Acyclic Graph (DAG) whose nodes are the random variables, where the edge structure represents statistical dependency relations between the variables. Structure learning of a Bayesian network (see, e.g., Koller and Friedman, 2009) aims to find the structure (DAG) that best matches the joint distribution of given random variables, using i.i.d. samples of the corresponding random vector. Structure learning is used, for instance, for uncovering gene interaction (Friedman et al., 2000), cancer prediction (Witteveen et al., 2018), and fault diagnosis (Cai et al., 2017).

©2022 Noa Ben-David and Sivan Sabato.

License: CC-BY 4.0, see `https://creativecommons.org/licenses/by/4.0/`. Attribution requirements are provided at `http://jmlr.org/papers/v23/21-0296.html`.

In the setting that we study, a limited number of variable measurements can be observed from each random sample. The algorithm iteratively selects which of the variable values to observe from the next random sample. This is of interest, for instance, when the random vectors represent patients participating in a study, and each measured variable corresponds to the results of some medical test. Patients would not agree to undergo many different tests, thus the number of tests taken from each patient is restricted. Other examples include obtaining a synchronized measurement from all sensors in a sensor network with a limited bandwidth (Dasarathy et al., 2016), and recording neural activity with a limited recording capacity (Soudry et al., 2013; Turaga et al., 2013). The goal is to find a good structure based on the the restricted observations, using a small number of random samples. Learning with limited observations from each sample was first studied in Ben-David and Dichterman (1998) for binary classification, and has thereafter been studied for other settings, such as online learning (Cesa-Bianchi et al., 2011; Hazan and Koren, 2012) and linear regression (Kukliansky and Shamir, 2015). This work is the first to study limited observations in the context of structure learning of Bayesian networks.

We study the active structure-learning problem in a score-based framework (Cooper and Herskovits, 1992; Heckerman et al., 1995), in which each possible DAG has a score that depends on the unknown distribution, and the goal is to find a DAG with a near-optimal score. We propose a new active learning algorithm, ActiveBNSL, that finds with a high probability a structure with a score that is ϵ-close to the optimal structure. We compare the sample complexity of our algorithm to that of a naive algorithm, which observes every possible subset of the variables the same number of times. We show that the improvement in the sample complexity can be as large as a factor of $\widetilde{\Omega}(d^3)$, where d is the dimension of the random vector. This improvement is obtained for a class of distributions that we define, which we term *stable* distributions. We further show that in the worst case, the sample complexity of the active algorithm is guaranteed to be almost the same as that of the naive algorithm.

A main challenge in reducing the number of samples in this setting is the fact that each structure usually has multiple equivalent structures with the same quality score. ActiveBNSL overcomes this issue by taking equivalence classes into account directly. An additional challenge that we address is characterizing distributions in which a significant reduction in the sample complexity is possible. Our definition of stable distributions captures such a family of distributions, in which there is a large number of variables with relationships that are easy to identify, and a small number of variables whose relationships with other variables are harder to identify.

Finding the optimal structure for a Bayesian network, even in the fully-observed setting, is computationally hard (Chickering, 1996). Nonetheless, algorithms based on relaxed linear programming are successfully used in practice (Cussens, 2011; Bartlett and Cussens, 2013, 2017). ActiveBNSL uses existing structure-search procedures as black boxes. This makes it easy to implement a practical version of ActiveBNSL using existing software packages. We implement ActiveBNSL and the naive baseline, and report experimental results that demonstrate the sample complexity reduction on stable distributions. The code for the algorithms and for the experiments is provided in `https://github.com/noabdavid/activeBNSL`.

Paper structure We discuss related work in Section 2. The setting and notation are defined in Section 3. In Section 4, a naive baseline algorithm is presented and analyzed. The proposed active algorithm, ActiveBNSL, is given in Section 5. Its correctness is proved in Section 6, and sample complexity analysis is given in Section 7, where we show that significant savings can be obtained for the class of stable distributions that we define. In Section 8, we describe an explicit class of stable distributions. In Section 9, we report experiments that compare the performance of our implementation of ActiveBNSL to that of the naive baseline, on a family of stable distributions. We conclude with a discussion in Section 10. Some technical proofs are deferred to the appendices.

2. Related work

Several general approaches exist for structure learning of Bayesian networks. In the constraint-based approach (Meek, 1995; Spirtes et al., 2000), pairwise statistical tests are performed to reveal conditional independence of pairs of random variables. This approach is asymptotically optimal, but less successful on finite samples (Heckerman et al., 2006), unless additional realizability assumptions are made. When restricting the in-degree of the DAG, the constraint-based approach is computationally efficient. However, the computational complexity and the sample complexity required for ensuring an accurate solution depend on the difficulty of identifying the conditional independence constraints (Canonne et al., 2018) and cannot be bounded independently of that difficulty. The *score-based* approach assigns a data-dependent score to every possible structure, and searches for the structure that maximizes this score. One of the most popular score functions for discrete distributions is the Bayesian Dirichlet (BD) score (Cooper and Herskovits, 1992). Several flavors of this score have been proposed (Heckerman et al., 1995; Buntine, 1991). The BD scores are well-studied (Chickering, 2002; Tsamardinos et al., 2006; Campos and Ji, 2011; Cussens, 2011; Bartlett and Cussens, 2013, 2017) and widely used in applications (see, e.g., Marini et al., 2015; Li et al., 2016; Hu and Kerschberg, 2018).

Finding a Bayesian network with a maximal BD score is computationally hard under standard assumptions (Chickering, 1996). However, successful heuristic algorithms have been suggested. Earlier algorithms such as Chickering (2002); Tsamardinos et al. (2006) integrate greedy hill-climbing with other methods to try and escape local maxima. Other approaches include dynamic programming (Silander and Myllymäki, 2012) and integer linear programming (Cussens, 2011; Bartlett and Cussens, 2013, 2017).

Previous works on active structure learning of Bayesian networks (Tong and Koller, 2001; He and Geng, 2008; Li and Leong, 2009; Squires et al., 2020) assume a causal structure between the random variables, and consider an interventional (or experimental) environment, in which the active learner can set the values of some of the variables and then measure the values of the others. This is crucially different from our model, in which there is no causality assumption and the interaction is only by observing variables, not setting their values.

Active learning for undirected graphical models has been studied both in a full observational setting (Vats et al., 2014; Dasarathy et al., 2016), where all variable values are available in each sample, and in a setting of limited observations (Scarlett and Cevher, 2017; Dasarathy, 2019; Vinci et al., 2019).

More generally, interactive unsupervised learning has been studied in a variety of contexts, including clustering (Awasthi et al., 2017) and compressive sensing (Haupt et al., 2009; Malloy and Nowak, 2014).

3. Setting and Notation

For an integer i, denote $[i] := \{1, \ldots, i\}$. A Bayesian network models the probabilistic relationships between random variables $\mathbf{X} = (X_1, \ldots, X_d)$. It is represented by a DAG $G = (V, E)$ and a set of parameters Θ. $V = [d]$ is the set of nodes, where node i represents the random variable X_i. E is the set of directed edges. We denote the set of parents of a variable $X_i \in \mathbf{X}$ under G by $\Pi_i(G) := \{X_j \mid (j, i) \in E\}$. We omit the argument G when it is clear from context. G and Θ together define a joint distribution over $\mathbf{X}$ which we denote by $P_{G,\Theta}(\mathbf{X})$. In the case where the distributions of all X_i are discrete, $\mathbf{X} := (X_1, \ldots, X_d)$ is a multivariate multinomial random vector and Θ specifies the multinomial distributions $P(X_i \mid \Pi_i)$. In this case, the joint distribution defined by G and Θ is given by

$$P_{G,\Theta}(\mathbf{X}) = \prod_{i \in [d]} P(X_i \mid \Pi_i(G), \Theta).$$

We call any possible pair of a child variable and parent set over $[d]$ a *family*, and denote it by $\langle X_i, \Pi \rangle$, where X_i is the child random variable and $\Pi \subseteq \{X_1, \ldots, X_d\} \setminus \{X_i\}$ defines the set of parents assigned to X_i. We refer to a family with a child variable i as *a family of i*. For convenience, we will use G to denote also the set of families that the structure G induces, so that $f \in G$ if and only if $f = \langle X_i, \Pi_i(G) \rangle$ for some $i \in [d]$.

Structure learning is the task of finding a graph G with a high-quality fit to the true distribution of $\mathbf{X}$. This is equivalent to finding the family of each of the variables in $[d]$. The quality of the fit of a specific structure G can be measured using the following information-criterion score (Bouckaert, 1995):

$$\mathcal{S}(G) := -\sum_{i \in [d]} H(X_i \mid \Pi_i(G)) = -\sum_{f \in G} H(f),$$

where H stands for entropy, and $H(f)$, for a family $f = \langle X_i, \Pi \rangle$, is defined as $H(f) := H(X_i \mid \Pi)$. As shown in Bouckaert (1995), for discrete distributions, maximizing the plug-in estimate of $\mathcal{S}(G)$ from data, or small variants of it, is essentially equivalent to maximizing some flavors of the BD score.

For both statistical and computational reasons, in structure learning one commonly searches for a graph in which each variable has at most k parents, for some small integer constant k (see, e.g., Heckerman, 1998). Denote by $\mathcal{G}_{d,k}$ the set of DAGs over d random variables such that each variable has at most k parents. Denote the set of all possible families over $[d]$ with at most k parents by $\mathcal{F}_{d,k}$. Denote by $\mathcal{S}^* := \max_{G \in \mathcal{G}_{d,k}} \mathcal{S}(G)$ the optimal score of a graph in $\mathcal{G}_{d,k}$ for the true distribution of $\mathbf{X}$. Note that $\mathbf{X}$ can have any joint distribution, not necessarily one of the form $P_{G,\Theta}(\mathbf{X})$ for some $G \in \mathcal{G}_{d,k}$ and Θ. In cases where $P(\mathbf{X})$ is in fact of the latter form, it has been shown for Gaussian and multinomial random variables (Chickering, 2002), that for any graph G that maximizes $\mathcal{S}(G)$, there is some Θ such that the distribution of $\mathbf{X}$ is equal to $P_{G,\Theta}(\mathbf{X})$. In other words, whenever there exists some graph which exactly describes $P(\mathbf{X})$, score maximization will find such a graph. Our goal is to find a structure $\hat{G} \in \mathcal{G}_{d,k}$ such that with a probability of at least $1-\delta$, $\mathcal{S}(\hat{G}) \geq \mathcal{S}^* - \epsilon$. Denote by $\mathcal{G}^*$ the set of all optimal structures, $\mathcal{G}^* := \{G \in \mathcal{G}_{d,k} \mid \mathcal{S}(G) = \mathcal{S}^*\}$.

We study an active setting in which the algorithm iteratively selects a subset of variables to observe, and then draws an independent random copy of the random vector $\mathbf{X}$. In the sampled vector, only the values of variables in the selected subset are revealed. The goal is to find a good structure using a small number of such random vector draws. We study the case where the maximal number of variables which can be revealed in any vector sample is $k+1$. This is the smallest number that allows observing the values of a variable and a set of potential parents in the same random sample. We leave for future work the generalization of our approach to other observation sizes. For a set A, denote by $[\![A]\!]^{k+1}$ the set of subsets of A of size $k+1$. For an integer i, $[\![i]\!]^{k+1}$ is used as a shorthand for $[\![[i]]\!]^{k+1}$.

4. A Naive Algorithm

We first discuss a naive approach, in which all variable subsets of size $k+1$ are observed the same number of times, and these observations are used to estimate the score of each candidate structure. As an estimator for $H(f)$, one can use various options, possibly depending on properties of the distribution of $\mathbf{X}$ (see, e.g., Paninski, 2003). Denoting this estimator by $\hat{H}$, the empirical score of graph G is defined as

$$\hat{\mathcal{S}}(G) := -\sum_{f \in G} \hat{H}(f) = -\sum_{i \in [d]} \hat{H}(X_i \mid \Pi_i(G)). \tag{1}$$

For concreteness, assume that $\{X_i\}_{i \in [d]}$ are discrete random variables with a bounded support size. In this case, $\hat{H}$ can be set to the plug-in estimator, obtained by plugging in the empirical joint distribution of $\{X_i\} \cup \Pi$ into the definition of conditional entropy, where the empirical joint distribution is based on samples in which all of these variables were observed together.

For $\epsilon > 0, \delta \in (0,1)$, let $N(\epsilon, \delta)$ be a *sample-complexity upper bound* for $\hat{H}$. N is a function such that for any fixed $f = \langle i, \Pi \rangle$, if $\hat{H}(f)$ is calculated using at least $N(\epsilon, \delta)$ i.i.d. copies of the vector $\mathbf{X}$ in which the values of X_i and Π are all observed, then with a probability of at least $1-\delta$, $|\hat{H}(f) - H(f)| \leq \epsilon$. For instance, in the case of discrete random variables with the plug-in estimator, the following lemma, proved in Appendix A, shows that one can set $N(\epsilon, \delta) = \widetilde{\Theta}(\log(1/\delta)/\epsilon^2)$. Interestingly, this bound is the same order as the bound for the unconditional entropy (Antos and Kontoyiannis, 2001).

Lemma 1 *Let $\delta \in (0,1)$ and $\epsilon > 0$. Let A, B be discrete random variables with support cardinalities M_a and M_b, respectively. Let $\hat{H}(A \mid B)$ be the plug-in estimator of $H(A \mid B)$, based on N i.i.d. copies of (A, B), where*

$$N \geq \max\left\{\frac{2}{\epsilon^2}\log(\frac{2}{\delta}) \cdot \log^2(2\log(2/\delta)/\epsilon^2), e^2, M_a, (M_a - 1)M_b/\epsilon\right\}.$$

Then $P(|\hat{H}(A \mid B) - H(A \mid B)| > \epsilon) \leq \delta$.

Now, consider a naive algorithm which operates as follows:

1. Observe each variable subset in $[\![d]\!]^{k+1}$ in $N(\epsilon/(2d), \delta/|\mathcal{F}_{d,k}|)$ of the random samples;

2. Output a structure $G \in \mathcal{G}_{d,k}$ that maximizes the empirical score defined in Eq. (1), based on the plug-in estimates of $H(f)$ for each family.

Applying a union bound over all the families in $\mathcal{F}_{d,k}$, this guarantees that with a probability at least $1 - \delta$, we have that for each family $f \in \mathcal{F}_{d,k}$, $|\hat{H}(f) - H(f)| \leq \epsilon/(2d)$. Therefore, for any graph $G \in \mathcal{G}_{d,k}$,

$$|\mathcal{S}(G) - \hat{\mathcal{S}}(G)| = |-\sum_{f\in G} H(f) + \sum_{f \in G} \hat{H}(f)| \leq \sum_{f\in G} |\hat{H}(f) - H(f)| \leq \epsilon/2.$$

Letting $\hat{G} \in \operatorname{argmax}_{G\in\mathcal{G}_{d,k}} \hat{\mathcal{S}}(G)$ and $G^* \in \mathcal{G}^*$, with a probability at least $1 - \delta$,

$$\mathcal{S}(\hat{G}) \geq \hat{\mathcal{S}}(\hat{G}) - \epsilon/2 \geq \hat{\mathcal{S}}(G^*) - \epsilon/2 \geq \mathcal{S}(G^*) - \epsilon = \mathcal{S}^* - \epsilon,$$

as required. Note that $|\mathcal{F}_{d,k}| = d\sum_{i\in[k]}\binom{d-1}{i} \leq d(e(d-1)/k)^k$. Therefore, the sample complexity of the naive algorithm is at most

$$|[\![d]\!]^{k+1}| \cdot N(\epsilon/(2d), \delta/|\mathcal{F}_{d,k}|) = \widetilde{\Theta}\left(\frac{kd^{k+3}}{\epsilon^2} \cdot \log(1/\delta)\right), \tag{2}$$

where the right-hand side is given for the case of a discrete distribution with the plug-in entropy estimator discussed above. Below, we compare the sample complexity of the naive algorithm to the sample complexity of the interactive algorithm that we propose. Therefore, it is necessary to discuss the tightness of the upper bound in Eq. (2). Since the naive algorithm samples each subset of size $k + 1$ separately, the factor of $|[\![d]\!]^{k+1}| = \Theta(d^{k+1})$ cannot be avoided. The dependence on $\epsilon/(2d)$ in N is necessary, as can be observed via simple symmetry arguments. Thus, the only possible sources of looseness in this upper bound are the convergence analysis leading to the function $N(\cdot, \cdot)$, and the use of a union bound which leads to the factor of $\log(|\mathcal{F}_{d,k}|) = O(k \log(d))$. Since we use the same convergence function $N(\cdot, \cdot)$ also for analyzing the interactive algorithm, and we treat k as a constant, we conclude that comparing the sample complexity of the interactive algorithm to the bound in Eq. (2) is accurate up to possible logarithmic factors in d.

5. An Interactive Algorithm: ActiveBNSL

The naive algorithm observes all variable subsets the same number of times. Thus, it does not make use of possible differences in difficulty in identifying different parts of the structure. We propose a new interactive algorithm, ActiveBNSL, which uses previous observations to identify such differences to reduce the sample complexity. Our approach is based on incrementally adding families to the output structure, until it is fully defined. This reduces the sample complexity in cases where the early acceptance of families allows removing some variables from observation.

A challenging property of Bayesian networks in this respect is *Markov equivalence.* Two graphs are considered Markov equivalent if they induce the same conditional independence constraints (Andersson et al., 1997). It has been shown that for any dataset, the likelihood of the data given the graph is the same for all Markov equivalent graphs (Heckerman et al., 1995). Since the score defined in Section 3 is the asymptotic log-likelihood of an infinite sample drawn from the distribution, the scores of two Markov equivalent graphs are identical. Almost all DAGs in $\mathcal{G}_{d,k}$ have other Markov-equivalent DAGs in $\mathcal{G}_{d,k}$, and usually there is not even a single family that is shared by all Markov-equivalent graphs. ActiveBNSL addresses this issue by directly handling Markov equivalence classes, which are the equivalence classes induced on $\mathcal{G}_{d,k}$ by the Markov-equivalence relation. Denote the set of equivalence classes (ECs) in $\mathcal{G}_{d,k}$ by $\mathcal{E}_{d,k}$, and note that each equivalence class $E \in \mathcal{E}_{d,k}$ is a set of structures, $E \subseteq \mathcal{G}_{d,k}$. Since all the structures in a given EC have the same score, we can discuss the score of an EC without confusion. For an EC $E \in \mathcal{E}_{d,k}$, denote by $\mathcal{S}(E)$ the (identical) score of the graphs in E. An *optimal EC* is an EC that maximizes the score over all ECs in $\mathcal{E}_{d,k}$. Note that this EC might not be unique.

ActiveBNSL, listed in Alg. 1, gets d (the number of variables) and k (the maximal number of parents) as inputs, in addition to a confidence parameter δ, the required accuracy level ϵ. ActiveBNSL maintains a set $\mathcal{V} \subseteq [d]$ of the variables whose families have been accepted so far (that is, their parent sets in the output structure have been fixed). The set of accepted families is denoted Acc_j, where j denotes an iteration within the algorithm. Acc_j includes exactly one family for each $v \in \mathcal{V}$. The set of candidate families is denoted $\mathrm{Cand}(\mathcal{V})$. This is the set of families that have not been precluded so far from participating in the output structure. For simplicity, we set $\mathrm{Cand}(\mathcal{V})$ to the set of all the families with a child variable not in $\mathcal{V}$. Note that this does not preclude, for instance, families that would create a cycle when combined with Acc_j. As will be evident below, including redundant families in $\mathrm{Cand}(\mathcal{V})$ does not harm the correctness of ActiveBNSL.

ActiveBNSL works in rounds. At each round t, the algorithm observes all the subsets in $[\![d]\!]^{k+1}$ that include at least one variable not in $\mathcal{V}$. Each such subset is observed a sufficient number of times to obtain a required accuracy. Let $\hat{H}_t(f)$ be the estimator for $H(f)$ based on the samples observed until round t, and denote the empirical score at round t by $\hat{\mathcal{S}}_t(G) := -\sum_{f \in G} \hat{H}_t(f)$. Note that unlike the true score, the empirical score of the graphs in an EC might not be the same for all the graphs, due to the use of partial observations. Therefore, we define the empirical score of a set of graphs A as $\hat{\mathcal{S}}(A) := \max_{G \in A} \hat{\mathcal{S}}(G)$. Iteratively within the round, ActiveBNSL finds some equivalence class $\hat{E}$ that maximizes the empirical score $\hat{\mathcal{S}}(\hat{E})$ and is consistent with the families accepted so far. Here, $\mathcal{G}_j := \{G \in \mathcal{G}_{d,k} \mid \mathrm{Acc}_j \subseteq G\}$ denotes the set of graphs that are consistent with these families in iteration j. We also denote $\mathbf{U}(A) := \bigcup_{G \in A} G$, the set of all families in any of the graphs in A.

For $E \in \mathcal{E}_{d,k}$, denote $E^{\cap j} := E \cap \mathcal{G}_j$. ActiveBNSL calculates a threshold θ_j, and searches for a family such that in each of the ECs whose empirical score is θ_j-close to the empirical score of $\hat{E}$, there exists at least one graph which is consistent with Acc_j and includes this family. If such a family exists, it can be guaranteed that it is a part of some optimal structure along with previously accepted families. Therefore, it is accepted, and its child variable is added to $\mathcal{V}$. At the end of every round, the required accuracy level ϵ_t is halved. Iterations continue until $\epsilon_t \leq \epsilon/(d - |\mathcal{V}|)$. It is easy to verify that the total number of rounds is at most $T - 1 = \lceil \log_2(d) \rceil$, where T is defined in ActiveBNSL. In the last round, which occurs outside of the main loop, if any families are still active, the remaining variable subsets are observed based on the required accuracy ϵ. ActiveBNSL then returns a structure which maximizes the empirical score, subject to the constraints set by the families accepted so far.

```
Algorithm 1: ActiveBNSL
   Input: Integers $d, k$; $\delta \in (0,1)$; $\epsilon > 0$.
   Output: A graph $\hat{G}$
1  Intialize: $\mathrm{Acc}_1 \leftarrow \emptyset$, $\mathcal{V} \leftarrow \emptyset$, $N_0 \leftarrow 0$, $t \leftarrow 1$, $j \leftarrow 1$, $T \leftarrow \lceil \log_2(2d) \rceil$, $\epsilon_1 \leftarrow \epsilon$.
2  while $\epsilon_t > \epsilon/(d - |\mathcal{V}|)$ do
3      $N_t \leftarrow N(\epsilon_t/2, \delta/(T|\mathcal{F}_{d,k}|))$
4      For each subset in $[\![d]\!]^{k+1} \setminus [\![\mathcal{V}]\!]^{k+1}$, observe it in $N_t - N_{t-1}$ random samples.
5      repeat
6          $\theta_j \leftarrow (d - |\mathcal{V}|) \cdot \epsilon_t$
7          $\hat{G}_j \leftarrow \mathrm{argmax}\{\hat{\mathcal{S}}_t(G) \mid G \in \mathcal{G}_j\}$        # Recall: $\mathcal{G}_j \equiv \{G \in \mathcal{G}_{d,k} \mid \mathrm{Acc}_j \subseteq G\}$
8          $\hat{E}_j \leftarrow$ the EC in $\mathcal{E}_{d,k}$ that includes $\hat{G}_j$
9          $L_j \leftarrow \{E \in \mathcal{E}_{d,k} \mid (E^{\cap j} \neq \emptyset) \wedge (\hat{\mathcal{S}}_t(\hat{E}_j^{\cap j}) - \hat{\mathcal{S}}_t(E^{\cap j}) \leq \theta_j)\}$
10         if $\exists f \in \mathbf{U}(\hat{E}_j^{\cap j}) \cap \mathrm{Cand}(\mathcal{V})$ such that $\forall E \in L_j,\ f \in \mathbf{U}(E^{\cap j})$ then
11             Set $f \leftarrow \langle X_v, \Pi \rangle$ such that $f$ satisfies the condition above and $v \in [d]$
12             $\mathcal{V} \leftarrow \mathcal{V} \cup \{v\}$, $\mathrm{Acc}_{j+1} \leftarrow \mathrm{Acc}_j \cup \{f\}$            # accept family $f$
           else
13             $\mathrm{Acc}_{j+1} \leftarrow \mathrm{Acc}_j$
14         $j \leftarrow j + 1$
       until $\mathrm{Acc}_j = \mathrm{Acc}_{j-1}$
15     If $|\mathcal{V}| = d$ then set $\hat{G} \leftarrow \mathrm{Acc}_j$ and return $\hat{G}$.
16     $t \leftarrow t + 1$
17     $\epsilon_t \leftarrow \epsilon_{t-1}/2$
18 $\epsilon_{\mathrm{last}} \leftarrow \epsilon/(d - |\mathcal{V}|)$, $N_T \leftarrow N(\epsilon_{\mathrm{last}}/2, \delta/(T|\mathrm{Cand}(\mathcal{V})|))$
19 For each subset in $[\![d]\!]^{k+1} \setminus [\![\mathcal{V}]\!]^{k+1}$, observe it in $N_T - N_{T-1}$ random samples.
20 Return $\hat{G} \leftarrow \mathrm{argmax}\{\hat{\mathcal{S}}_T(G) \mid G \in \mathcal{G}_j\}$.
```

An execution example. To demonstrate how ActiveBNSL works, we now give an example trace for ActiveBNSL with $d = 4, k = 1$. A summary of this trace is provided in Table 1. $\mathcal{F}_{4,1}$ includes four families with an empty parent set, and twelve families with a parent set of size one. Table 2 lists three of the ECs in $\mathcal{E}_{4,1}$ that we will refer to in this execution example. Note that $T = \log_2(8) = 3$.

- After the initialization stage (line 1), ActiveBNSL draws N_1 samples from each pair of variables in $\{1, 2, 3, 4\}$ (line 4).

- Suppose that the graph that maximizes the empirical score in line 7 is $\hat{G}_1 = G_3$, where G_3 is given in Table 2. Thus, in line 8, $\hat{E}_1 = E_3$.

- Since Acc_1 is an empty set, $\mathcal{G}_j = \mathcal{G}_{4,1}$, and so L_1 (line 9) simply includes all the ECs that are empirically θ_1-close to E_3. Suppose that $L_1 = \{E_1, E_2, E_3\}$.

- The condition in line 10 is satisfied by the families $\langle 1, \emptyset \rangle$ and $\langle 2, \{1\} \rangle$ (see G_1, G_2, G_3 in Table 2), and by the family $\langle 1, \{2\} \rangle$ (see G_4, G_5, G_6 in Table 2). Suppose that

j	Acc_j	$\mathcal{V}$	$\hat{G}_j$	$\hat{E}_j$	L_j
1	$\emptyset$	$\emptyset$	G_3	E_3	$\{E_1, E_2, E_3\}$
2	$\{\langle 1, \emptyset \rangle\}$	$\{1\}$	G_3	E_3	$\{E_1, E_2, E_3\}$
3	$\{\langle 1, \emptyset \rangle, \langle 2, \{1\} \rangle\}$	$\{1, 2\}$	G_3	E_3	$\{E_1, E_2, E_3\}$
4	$\{\langle 1, \emptyset \rangle, \langle 2, \{1\} \rangle\}$	$\{1, 2\}$	G_2	E_2	$\{E_1, E_2\}$

Table 1: A summary of the example trace

E_1	E_2	E_3
$1 \rightarrow 2 \rightarrow 3 \rightarrow 4$ G_1	$1 \rightarrow 2 \rightarrow 4 \rightarrow 3$ G_2	$2 \leftarrow 1 \rightarrow 3 \rightarrow 4$ G_3
$1 \leftarrow 2 \rightarrow 3 \rightarrow 4$ G_4	$1 \leftarrow 2 \rightarrow 4 \rightarrow 3$ G_5	$2 \rightarrow 1 \rightarrow 3 \rightarrow 4$ G_6
$1 \leftarrow 2 \leftarrow 3 \rightarrow 4$ G_7	$1 \leftarrow 2 \leftarrow 4 \rightarrow 3$ G_8	$2 \leftarrow 1 \leftarrow 3 \rightarrow 4$ G_9
$1 \leftarrow 2 \leftarrow 3 \leftarrow 4$ G_{10}	$1 \leftarrow 2 \leftarrow 4 \leftarrow 3$ G_{11}	$2 \leftarrow 1 \leftarrow 3 \leftarrow 4$ G_{12}

Table 2: Three ECs in $\mathcal{E}_{4,1}$. Each of these ECs includes four structures.

$f := \langle 1, \emptyset \rangle$ is selected in line 11 and so f is accepted (line 12). Thus, $\text{Acc}_2 = \{\langle 1, \emptyset \rangle\}$, and $\mathcal{V} = \{1\}$.

- In the next iteration, $j = 2$, the algorithm hasn't drawn any new samples. Therefore, since G_3 includes the accepted family, we have $\hat{G}_2 = G_3, \hat{E}_2 = E_3$ as in the previous iteration. Suppose that $L_2 = L_1$.

- Note that only one graph in each EC in Table 2 includes the family $\langle 1, \emptyset \rangle$. Therefore, $E_1^{\cap 2} = \{G_1\}$, $E_2^{\cap 2} = \{G_2\}$, and $E_3^{\cap 2} = \{G_3\}$. The accepted family in this iteration can only be $\langle 2, \{1\} \rangle$, which is shared by G_1, G_2, and G_3. Therefore, we now have $\text{Acc}_3 = \{\langle 1, \emptyset \rangle, \langle 2, \{1\} \rangle\}$, and $\mathcal{V} = \{1, 2\}$.

- In iteration $j = 3$, suppose that $L_3 = L_2 = \{E_1, E_2, E_3\}$. $\hat{G}_3, \hat{E}_3$ and $E_i^{\cap 3} = \{G_i\}$ for $i \in \{1, 2, 3\}$ are the same as in the previous iterations. There is no family for $\{3, 4\}$ that is shared by G_1, G_2, and G_3. Thus, in this iteration the condition on line 10 does not hold, and no family is accepted on iteration 3. The condition ending the repeat-until loop now holds, and ActiveBNSL proceeds to the next round.

- On the next round, $t = 2$, $N_2 - N_1$ additional samples are drawn, such that the total number of samples taken from each subset of size 2 of variables (except for the subset $\{1, 2\}$) is equal to N_2. Now $j = 4$.

- Suppose that $\hat{G}_4 = G_2$ in line 7. Thus, in line 8, $\hat{E}_4 = E_2$. Suppose further that $L_4 = \{E_1, E_2\}$, and note that $E_2^{\cap 4} = \{G_1\}$ and $E_1^{\cap 4} = \{G_2\}$. Again, no family for

$\{3, 4\}$ is shared by G_1 and G_2, and no family is accepted at iteration 4. This causes the second round of the algorithm to terminate.

- The main loop now ends, since $T - 1$ rounds have been completed. The final sampling round, round $T = 3$, occurs outside of the main loop. Again all the subsets of size 2 except for $\{1, 2\}$ are sampled, so the total number of samples is N_3. ActiveBNSL returns the highest-scoring graph that includes the accepted families $\langle 1, \emptyset\rangle, \langle 2, \{1\}\rangle$.

The computational complexity of ActiveBNSL**.** As mentioned in Section 2 above, score-based structure learning is NP-hard in the general case (Chickering, 1996). Thus, ActiveBNSL is not an efficient algorithm in the general case. Nonetheless, ActiveBNSL is structured so that the computationally hard operations are encapsulated in two specific black boxes:

- The score maximization procedure, used in lines 7 and 20. A similar procedure is required by any score-based structure learning algorithm, including the naive algorithm proposed in Section 4, as well as structure learning algorithms for the full information setting (see, e.g., Chickering, 2002; Silander and Myllymäki, 2012; Bartlett and Cussens, 2013). Thus, established and successful efficient heuristics exist for this procedure. In ActiveBNSL, the maximization is constrained to include specific families. This does not add a significant computational burden to these heuristics.

- The assignment of ECs that satisfy the required constraints in line 9. This procedure can be divided into two steps:

    1. Compute all the high-scored structures that are consistent with the accepted families;
    2. Aggregate equivalent structures into ECs.

    Step 1 can be replaced with an efficient heuristic proposed in Liao et al. (2019), which is derived from score-maximization heuristics. Step 2 requires computing the EC of each of these structures, which can be done efficiently.

This encapsulation thus allows using heuristics for these black boxes to obtain a practical algorithm, as we show in Section 9 below, whereas an exact implementation of these black boxes would be exponential in d in the worst case (see, e.g., Chickering, 2002; Liao et al., 2019).

In the next section, we show that ActiveBNSL indeed finds an ϵ-optimal structure with a probability at least $1 - \delta$.

6. Correctness of ActiveBNSL

The following theorem states the correctness of ActiveBNSL.

Theorem 2 *With a probability at least $1 - \delta$, the graph $\hat{G}$ returned by* ActiveBNSL *satisfies $\mathcal{S}(\hat{G}) \geq \mathcal{S}^* - \epsilon$.*

To prove Theorem 2, we first give some preliminaries. Let $\mathcal{V}_t$ be the set $\mathcal{V}$ at the beginning of round t. Define the following event:

$$\eta := \{\forall t \in [T], \forall f \in \mathrm{Cand}(\mathcal{V}_t), |\hat{H}_t(f) - H(f)| \leq \epsilon_t/2\}.$$

Noting that for any family in $\mathrm{Cand}(\mathcal{V}_t)$, $\hat{H}_t(f)$ is estimated based on $N_t = N(\epsilon_t/2, \delta/(T|\mathcal{F}_{d,k}|))$ samples, it is easy to see that by a union bound over at most T rounds and at most $|\mathcal{F}_{d,k}|$ families in each round, η holds with a probability at least $1 - \delta$.

Let an *iteration* of ActiveBNSL be a single pass of the inner loop starting at line 6. Let J be the total number of iterations during the entire run of the algorithm. Note that a single round t can include several iterations j. Denote by $\mathcal{V}_{(j)}$ the value of $\mathcal{V}$ at the start of iteration j. Under η, it follows from above that at any iteration j in round t, for any graph $G \in \mathcal{G}_j$ (recall that we treat G as the set of families that the graph consists of),

$$|\mathcal{S}(G \setminus \mathrm{Acc}_j) - \hat{\mathcal{S}}_t(G \setminus \mathrm{Acc}_j)| \leq \sum_{f \in G \setminus \mathrm{Acc}_j} |H(f) - \hat{H}_t(f)| \leq (d - |\mathcal{V}_{(j)}|) \cdot \epsilon_t/2 = \theta_j/2, \quad (3)$$

where θ_j is defined in line 6 of Alg. 1. We now give a bound on the empirical score difference between equivalence classes. For $i \in 1, 2$, let $E_i \in \mathcal{E}_{d,k}$ be two equivalence classes such that $E_i^{\cap j} \neq \emptyset$, and $G_i \in \mathrm{argmax}_{G \in E_i^{\cap j}} \hat{\mathcal{S}}_t(G)$. We have $G_i \in \mathcal{G}_j$. Hence, $|\mathcal{S}(G_i \setminus \mathrm{Acc}_j) - \hat{\mathcal{S}}_t(G_i \setminus \mathrm{Acc}_j)| \leq \theta_j/2$. In addition, $\mathcal{S}(G_i \setminus \mathrm{Acc}_j) = \mathcal{S}(E_i) - \mathcal{S}(\mathrm{Acc}_j)$, and by the definition of the empirical score of a set of structures, we have that $\hat{\mathcal{S}}_t(E_i^{\cap j}) = \hat{\mathcal{S}}_t(G_i) = \hat{\mathcal{S}}_t(G_i \setminus \mathrm{Acc}_j) + \hat{\mathcal{S}}_t(\mathrm{Acc}_j)$. Therefore, under η,

$$\begin{aligned}
&\hat{\mathcal{S}}_t(E_1^{\cap j}) - \hat{\mathcal{S}}_t(E_2^{\cap j}) = \hat{\mathcal{S}}_t(G_1 \setminus \mathrm{Acc}_j) - \hat{\mathcal{S}}_t(G_2 \setminus \mathrm{Acc}_j) \\
&= (\hat{\mathcal{S}}_t(G_1 \setminus \mathrm{Acc}_j) - \mathcal{S}(G_1 \setminus \mathrm{Acc}_j)) + (\mathcal{S}(G_1 \setminus \mathrm{Acc}_j) - \mathcal{S}(G_2 \setminus \mathrm{Acc}_j)) \\
&\quad + (\mathcal{S}(G_2 \setminus \mathrm{Acc}_j) - \hat{\mathcal{S}}_t(G_2 \setminus \mathrm{Acc}_j)) \\
&\leq \mathcal{S}(G_1 \setminus \mathrm{Acc}_j) - \mathcal{S}(G_2 \setminus \mathrm{Acc}_j) + \theta_j \\
&= \mathcal{S}(E_1) - \mathcal{S}(E_2) + \theta_j. \qquad (4)
\end{aligned}$$

The following lemma shows that if a family is accepted in the main loop of Alg. 1, then it is in some optimal structure which includes all the families accepted so far. This shows that in ActiveBNSL, there is always some optimal structure that is consistent with the families accepted so far.

Denote the set of optimal graphs in $\mathcal{G}_j$ by $\mathcal{G}_j^* := \mathcal{G}^* \cap \mathcal{G}_j$.

Lemma 3 *Assume that η occurs. Then for all $j \in [J]$, $\mathcal{G}_j^* \neq \emptyset$.*

Proof The proof is by induction on the iteration j. For $j = 1$, we have $\mathcal{G}_j = \mathcal{G}_{d,k}$, hence $\mathcal{G}_1^* = \mathcal{G}^* \neq \emptyset$. Now, suppose that the claim holds for iteration j. Then $\mathcal{G}_j^* \neq \emptyset$. We show that if any family f is accepted in this iteration, it satisfies $f \in \mathbf{U}(\mathcal{G}_j^*)$. This suffices to prove the claim, since $\mathcal{G}_j^* \subseteq \mathcal{G}_j$ implies that all the graphs in $\mathcal{G}_j^*$ include Acc_j; Combined with $f \in \mathbf{U}(\mathcal{G}_j^*)$, this implies that there is at least one graph in $\mathcal{G}_j^*$ that includes $\mathrm{Acc}_{j+1} \equiv \mathrm{Acc}_j \cup \{f\}$, thus $\mathcal{G}_{j+1}^* \neq \emptyset$, as needed.

Suppose that f is accepted at iteration j. Let $E_* \subseteq \mathcal{G}^*$ be an optimal EC such that $E_*^{\cap j} \neq \emptyset$; one exists due to the induction hypothesis. We show that $f \in \mathbf{U}(E_*^{\cap j})$, thus proving that

$f \in \mathbf{U}(\mathcal{G}^*_j)$. By event η and Eq. (4), we have $\hat{\mathcal{S}}_t(\hat{E}^{\cap j}_j) - \hat{\mathcal{S}}_t(E^{\cap j}_*) \leq \mathcal{S}(\hat{E}_j) - \mathcal{S}(E_*) + \theta_j \leq \theta_j$. The last inequality follows since $\mathcal{S}(\hat{E}_t) \leq \mathcal{S}(E_*)$. Therefore, for L_j as defined in line 9 of Alg. 1, we have $E_* \in L_j$. Since f is accepted, it satisfies the condition in line 10, therefore $f \in \mathbf{U}(E^{\cap j}_*)$. This proves the claim. ■

Using Lemma 3, the correctness of ActiveBNSL can be shown.

Proof [Proof of Theorem 2] Assume that η holds. As shown above, this occurs with a probability at least $1 - \delta$. Let $\mathrm{Acc} := \mathrm{Acc}_J$ be the set of accepted families at the end of the main loop of Alg. 1. For any $G \in \mathcal{G}_J$, we have $\mathrm{Acc} \subseteq G$, hence $\mathcal{S}(G) = \mathcal{S}(\mathrm{Acc}) + \mathcal{S}(G \setminus \mathrm{Acc})$, and similarly for $\hat{\mathcal{S}}_T$. In addition, for each family $f \in G \setminus \mathrm{Acc}$, we have $f \in \mathrm{Cand}(\mathcal{V}_T)$, since $\mathrm{Cand}(\mathcal{V}_T)$ includes all the families of a child node whose family was not yet accepted. So, by event η we have $|\hat{H}_T(f) - H(f)| \leq \epsilon_{\mathrm{last}}/2$, where ϵ_{last} is defined in line 18 of Alg. 1. Therefore, as in Eq. (3), for any $G \in \mathcal{G}_J$,

$$|\mathcal{S}(G \setminus \mathrm{Acc}) - \hat{\mathcal{S}}_T(G \setminus \mathrm{Acc})| \leq (d - |\mathcal{V}_T|) \cdot \epsilon_{\mathrm{last}}/2 = \epsilon/2.$$

By Lemma 3, we have that $\mathcal{G}^*_J \neq \emptyset$. Let G^* be some graph in $\mathcal{G}^*_J \subseteq \mathcal{G}_J$. Recall that also $\hat{G} \in \mathcal{G}_J$. By the definition of $\hat{G}$, $\hat{\mathcal{S}}_T(\hat{G}) \geq \hat{\mathcal{S}}_T(G^*)$. Hence, $\hat{\mathcal{S}}_T(\hat{G} \setminus \mathrm{Acc}) \geq \hat{\mathcal{S}}_T(G^* \setminus \mathrm{Acc})$. Combining this with the inequality above, it follows that

$$\mathcal{S}(\hat{G} \setminus \mathrm{Acc}) \geq \hat{\mathcal{S}}_T(\hat{G} \setminus \mathrm{Acc}) - \epsilon/2 \geq \hat{\mathcal{S}}_T(G^* \setminus \mathrm{Acc}) - \epsilon/2 \geq \mathcal{S}(G^* \setminus \mathrm{Acc}) - \epsilon.$$

Therefore, $\mathcal{S}(\hat{G}) \geq \mathcal{S}(G^*) - \epsilon$, which proves the claim. ■

7. The Sample Complexity of ActiveBNSL

We now analyze the number of samples drawn by ActiveBNSL. We show that it is never much larger than that of the naive algorithm, while it can be significantly smaller for some classes of distributions. For concreteness, we show sample complexity calculations using the value of $N(\epsilon, \delta)$ given in Section 4 for discrete distributions with the plug-in entropy estimator. We assume a regime with a small constant k and a possibly large d.

ActiveBNSL requires the largest number of samples if no family is accepted until round $T \equiv \lceil \log_2(2d) \rceil$. In this case, the algorithm pulls all subsets in $[\![d]\!]^{k+1}$, each for $N(\epsilon/(2d), \delta/(T|\mathcal{F}_{d,k}|))$ times. Thus, the worst-case sample complexity of ActiveBNSL is

$$|[\![d]\!]^{k+1}| \cdot N(\epsilon/(2d), \delta/(T|\mathcal{F}_{d,k}|)) = \widetilde{\Theta}\left(\frac{kd^{k+3}}{\epsilon^2}\log(1/\delta)\right).$$

In the general case, the sample complexity of ActiveBNSL can be upper-bounded based on the round in which each variable was added to $\mathcal{V}$. Let $t_1 < \ldots < t_n$ be the rounds in which at least one variable was added to $\mathcal{V}$, and denote $t_{n+1} := T$. Let V_i for $i \in [n]$ be the total number of variables added until the end of round t_i, and set $V_0 = 0, V_{n+1} = d$. Note that V_n is the size of $\mathcal{V}$ at the end of ActiveBNSL. In round t_i, $V_i - V_{i-1}$ variables are added to $\mathcal{V}$. Denoting by Q_i the number of subsets that are observed by ActiveBNSL for the last time in round t_i, we have

$$Q_i := |[\![V_i]\!]^{k+1} \setminus [\![V_{i-1}]\!]^{k+1}| = \binom{V_i}{k+1} - \binom{V_{i-1}}{k+1} \leq (V_i - V_{i-1})V_i^k.$$

The sample complexity of ActiveBNSL is at most $\sum_{i\in[n+1]} Q_i N_{t_i}$, where

$$N_{t_i} \equiv N(\epsilon_{t_i}/2, \delta/(T|\mathcal{F}_{d,k}|)) = \widetilde{O}\left(\frac{1}{\epsilon_{t_i}^2} k \log(d/\delta)\right).$$

The sample complexity is thus upper-bounded by

$$\widetilde{O}\left(\Big(\sum_{i\in[n]} (V_i - V_{i-1})V_i^k \cdot \frac{1}{\epsilon_{t_i}^2} + kd^k(d-V_n)^3 \cdot \frac{1}{\epsilon^2}\Big) \log(d/\delta)\right). \tag{5}$$

Since ϵ_t is decreasing in t, if most variables are added to $\mathcal{V}$ early in the run of the algorithm, considerable savings in the sample complexity are possible. We next describe a family of distributions for which ActiveBNSL can have a significantly smaller sample complexity compared to the naive algorithm. In the next section, we describe a general construction that satisfies these requirements.

Definition 4 *Let $\gamma > 0$, and let $V \subseteq \mathbf{X}$. A (γ, V)-*stable distribution *over $\mathbf{X}$ is a distribution which satisfies the following conditions.*

1. *In every $G \in \mathcal{G}_{d,k}$ such that $\mathcal{S}(G) \geq \mathcal{S}^* - \gamma$, the parents of all the variables in V are also in V.*
2. *There is a unique optimal EC for the marginal distribution on V, and the difference in scores between the best EC and the second-best EC on V is more than γ.*

Theorem 5, stated below, states that the sample complexity improvement for a discrete stable distribution can be as large as a factor of $\widetilde{\Omega}(d^3)$. In Section 8, we give a specific construction of a class of distributions that satisfies the necessary conditions of Theorem 5. This class is characterized by the existence of two similar variables, one a noisy version of the other.

Theorem 5 *Let $\gamma > 0$. Let $v := |V|$. Let $\mathcal{D}$ be a discrete (γ, V)-stable distribution, and assume that* ActiveBNSL *samples from $\mathcal{D}$. Then, the sample complexity of* ActiveBNSL *is at most*

$$\widetilde{O}\left(\left(\frac{d^2 v^{k+1}}{\gamma^2} + \frac{d^k(d-v)^3}{\epsilon^2}\right) \cdot k \log(1/\delta)\right). \tag{6}$$

Furthermore, if $v = d - O(1)$ and $\epsilon = O(\gamma d^{-3/2})$, then the sample complexity improvement factor of ActiveBNSL *compared to the naive algorithm is $\widetilde{\Omega}(d^3)$.*

To prove Theorem 5, we first prove several lemmas. The following lemma gives a sufficient condition for a family to be accepted by ActiveBNSL at a given iteration.

Lemma 6 *Assume that η occurs. Let j be an iteration in the run of* ActiveBNSL. *For a family f, define*

$$S_j^{\neg}(f) := \max\{\mathcal{S}(E) \mid E \in \mathcal{E}_{d,k}, E^{\cap j} \neq \emptyset, f \notin \mathbf{U}(E^{\cap j})\}.$$

This is the maximal score of an equivalence class which is consistent with families accepted so far and also inconsistent with f.

Suppose that for some $f \in \mathbf{U}(\mathcal{G}_j^*) \cap \mathrm{Cand}(\mathcal{V}_{(j)})$, we have $S_j^{\neg}(f) < \mathcal{S}^* - 2\theta_j$. Then, some family is accepted by ActiveBNSL *at iteration j.*

Proof Let $E_* \subseteq \mathcal{G}^*$ be an optimal EC such that $E_*^{\cap j} \neq \emptyset$. By Lemma 3, such an EC exists. Suppose that the assumption of the lemma holds. Then, for some $f \in \mathbf{U}(\mathcal{G}_j^*) \cap \mathrm{Cand}(\mathcal{V}_{(j)})$, we have $S_j^{\neg}(f) < \mathcal{S}^* - 2\theta_j$. It follows that for any $E \in \mathcal{E}_{d,k}$ such that $E^{\cap j} \neq \emptyset$ and $f \notin \mathbf{U}(E^{\cap j})$, $2\theta_j < \mathcal{S}(E_*) - \mathcal{S}(E)$. By the definition of $\hat{E}_j$ (line 8 in Alg. 1), we have $\hat{\mathcal{S}}_t(\hat{E}_j^{\cap j}) \geq \hat{\mathcal{S}}_t(E_*^{\cap j})$. In addition, since event η occurs, Eq. (4) holds. Therefore,

$$\hat{\mathcal{S}}_t(\hat{E}_j^{\cap j}) - \hat{\mathcal{S}}_t(E^{\cap j}) \geq \hat{\mathcal{S}}_t(E_*^{\cap j}) - \hat{\mathcal{S}}_t(E^{\cap j}) \geq \mathcal{S}(E_*) - \mathcal{S}(E) - \theta_j > \theta_j.$$

It follows that $E \notin L_j$, where L_j is as defined in line 9 of Alg. 1. Therefore, $\forall E \in L_j, f \in \mathbf{U}(E^{\cap j})$. In particular, $f \in \mathbf{U}(\hat{E}_j^{\cap j})$. Since $f \in \mathrm{Cand}(\mathcal{V}_{(j)})$, it follows that $f \in \mathbf{U}(\hat{E}_j^{\cap j}) \cap \mathrm{Cand}(\mathcal{V}_{(j)})$. Therefore, the condition in line 10 is satisfied, which implies that some family is accepted during iteration j. ■

Next, we give a characterization of the optimal structures for $\mathcal{D}$. Let $\mathcal{G}_V$ be the set of DAGs with an in-degree at most k over V. Denote $\bar{V} := \mathbf{X} \setminus V$. We term a set of families $F \subseteq \mathcal{F}_{d,k}$ a *legal family set for $\bar{V}$* if it includes exactly one family for each variable in $\bar{V}$ and no other families, and has no cycles. First, we provide an auxiliary lemma.

Lemma 7 *If $G \in \mathcal{G}_V$ and F is a legal family set for $\bar{V}$, then $G \cup F \in \mathcal{G}_{d,k}$.*

Proof Denote $G' := G \cup F$. First, we show that G' has an indegree of at most k. Any $G \in \mathcal{G}_V$ has exactly one family for every variable in V with at most k parents. Thus, in G', there is exactly one family for each variable in $V \cup \bar{V}$, and each family includes at most k parents. Thus, the degree constraint is not violated in G'.

Next, we address the acyclicity constraint. Assume in contradiction that G' contains a cycle, and let (A, B) be an edge in the cycle. This means that there is a directed path from B to A in G'. Let $(B, v_1, \ldots, v_m, A)$ be the nodes in the path. Note that G includes only edges from V to V, and F includes edges from $V \cup \bar{V}$ to $\bar{V}$. Consider the possible cases:

1. $A \in \bar{V}$, $B \in V$: Neither G nor F include edges from $\bar{V}$ to V. Thus, this case is impossible.

2. $A \in V$, $B \in V$: In this case, (A, B) must be an edge in G. In addition, since there are no edges from $\bar{V}$ to V in G', all the nodes $v_1, \ldots, v_m$ on the path from B to A are also in V. This means that this path is entirely in G, which contradicts the acyclicity of G.

3. $A \in V$, $B \in \bar{V}$: In this case, the path from B to A includes at least one edge from $\bar{V}$ to V. However, neither G nor F include edges from $\bar{V}$ to V. Therefore, this case is impossible.

4. $A \in \bar{V}$, $B \in \bar{V}$: In this case, (A, B) must be in F. In addition, since there are no edges from $\bar{V}$ to V in G', all the nodes $v_1, \ldots, v_m$ on the path form B to A are also in $\bar{V}$. This means that this path is entirely in F, which contradicts the acyclicity of F.

Since all cases are impossible, the existence of a cycle in G' is impossible. This completes the proof. ■

Next, we provide the following characterization of the optimal structures for $\mathcal{D}$.

Lemma 8 *Let $\mathcal{D}$ be a (γ, V)-stable distribution. Let $G^* \in E_*$, and let $F \subseteq G^*$ be set of the families of $\bar{V}$ in G^*. Then the following hold:*

1. *$G^* \setminus F \in \mathcal{G}_V$;*

2. *$\mathcal{S}(G^* \setminus F) = \max_{G \in \mathcal{G}_V} \mathcal{S}(G)$;*

3. *The score of F is maximal among the legal family sets for $\bar{V}$.*

Proof By the assumptions on the distribution, since G^* is an optimal structure, $\bar{V}$ has no outgoing edges into V. Therefore, $G^* \setminus F \in \mathcal{G}_V$. This proves the first part of the lemma.

To prove the second part of the lemma, let $G_1^* \in \mathcal{G}_V$ be some DAG over V with a maximal score. Note that F is a legal family set for $\bar{V}$. Let $G_1 := G_1^* \cup F$. By Lemma 7, $G_1 \in \mathcal{G}_{d,k}$. Then,

$$\mathcal{S}(G^* \setminus F) + \mathcal{S}(F) = \mathcal{S}(G^*) \geq \mathcal{S}(G_1) = \mathcal{S}(G_1^*) + \mathcal{S}(F).$$

Therefore, $\mathcal{S}(G^* \setminus F) \geq \mathcal{S}(G_1^*)$. Hence, $G^* \setminus F$ is also optimal over V. This proves the second part of the lemma.

To prove the third part of the lemma, let $F' \neq F$ be some legal family set for $\bar{V}$. Then, by Lemma 7, $G_2 := (G^* \setminus F) \cup F' \in \mathcal{G}_{d,k}$, and $\mathcal{S}(G_2) = \mathcal{S}(G^*) - \mathcal{S}(F) + \mathcal{S}(F')$. Since G^* is an optimal structure, $\mathcal{S}(G_2) \leq \mathcal{S}(G^*)$. Therefore, $\mathcal{S}(F) \geq \mathcal{S}(F')$. This holds for all legal family sets of $\bar{V}$. Therefore, the score of F is maximal among such sets, which proves the third part of the lemma. ■

The next lemma shows that equivalence classes with a near-optimal score include graphs with common families.

Lemma 9 *Let $\mathcal{D}$ be a (γ, V)-stable distribution. For a set of families $A \in \mathcal{F}_{d,k}$ and an equivalence class E, denote $E^{\cap A} := \{G \in E \mid A \subseteq G\}$. Let $G^* \in E_*^{\cap A}$, and suppose that there is some $\tilde{E} \in \mathcal{E}_{d,k}$ such that $\tilde{E} \neq E_*$, $\mathcal{S}(\tilde{E}) \geq \mathcal{S}^* - \gamma$ and $\tilde{E}^{\cap A} \neq \emptyset$.*

Then, there exists a graph $\tilde{G} \in \tilde{E}^{\cap A}$ such that the families of all the variables in V are the same in both G^* and $\tilde{G}$.

To prove this lemma, we use a known characterization of a Markov equivalence class, which relies on the notion of a *v-structure*.[1] In a DAG G, a v-structure is an ordered triplet of nodes (X, Y, Z) such that G contains the edges (X, Y) and (Z, Y), and there is no edge in either direction between X and Z in G.

Theorem 10 (Verma and Pearl 1991) *Two DAGs are equivalent if and only if they have the same skeletons (underlying undirected graphs) and the same set of v-structures.*

Proof [of Lemma 9] Let G be some graph in $\tilde{E}^{\cap A}$. Let F, F^* be the families of $\bar{V}$ in G and G^* respectively. Note that both F and F^* are legal family sets for $\bar{V}$, since they are subsets of DAGs in $\mathcal{G}_{d,k}$. Define the graph $G_1 := G^* \setminus F^*$, and set $\tilde{G} := G_1 \cup F$. By Lemma 8, $G_1 \in \mathcal{G}_V$. Thus, by Lemma 7, we also have $\tilde{G} \in \mathcal{G}_{d,k}$. Furthermore, all of the variables in V have the same families in both G^* and $\tilde{G}$.

To show that $\tilde{G}$ satisfies the conditions of the lemma, it is left to show that $\tilde{G} \in \tilde{E}^{\cap A}$. First, note that $A \subseteq \tilde{G}$, since any family in A is either in G_1 or in F. We now prove that $\tilde{G} \in \tilde{E}$, by showing that G and $\tilde{G}$ are Markov equivalent. Define $G_2 := G \setminus F$. Since $\mathcal{S}(G) = \mathcal{S}(\tilde{E}) \geq \mathcal{S}^* - \gamma$, by assumption 1 in Def. 4, it holds that in G all the variables in V have parents in V. Thus, $G_2 \in \mathcal{G}_v$.

We now prove that G_1 and G_2 are Markov equivalent, and conclude that the same holds for G and $\tilde{G}$. First, we show that $\mathcal{S}(G_2) = \mathcal{S}(G_1)$. We have $G_1, G_2 \in \mathcal{G}_v$, and by Lemma 8, G_1 is an optimal graph on V. Therefore, $\mathcal{S}(G_2) \leq \mathcal{S}(G_1)$. Now, suppose for contradiction that $\mathcal{S}(G_2) < \mathcal{S}(G_1)$. Then, by assumption 2 in Def. 4, we have $\mathcal{S}(G_2) < \mathcal{S}(G_1) - \gamma$. We also have, by Lemma 8, that $\mathcal{S}(F) \leq \mathcal{S}(F^*)$. It follows that

$$\mathcal{S}(G) = \mathcal{S}(G_2) + \mathcal{S}(F) \leq \mathcal{S}(G_2) + \mathcal{S}(F^*) < \mathcal{S}(G_1) - \gamma + \mathcal{S}(F^*) = \mathcal{S}(G^*) - \gamma = \mathcal{S}^* - \gamma.$$

Therefore, $\mathcal{S}(G) < \mathcal{S}^* - \gamma$. But $G \in \tilde{E}$, and so $\mathcal{S}(G) \geq \mathcal{S}^* - \gamma$, leading to a contradiction. It follows that $\mathcal{S}(G_2) = \mathcal{S}(G_1)$, implying that G_2 is also an optimal structure on V. Combining this with the uniqueness of the optimal EC on V, as given in assumption 2 of Def. 4, we conclude that G_1 and G_2 are Markov equivalent.

To show that G and $\tilde{G}$ are Markov equivalent, we first observe that since G_1 and G_2 are equivalent, then by Theorem 10, they have the same skeleton and the same set of v-structures. Therefore, $G_1 \cup F$ and $G_2 \cup F$ also have the same skeleton. In addition, they have the same set of v-structures, as follows: The v-structures with a child in V are in G_1 and G_2 and so they are shared; The v-structures with a child in $\bar{V}$ are those with parents in F and no edges between the parents. Since both graphs share the same skeleton, these must be the same v-structures. This proves that G and $\tilde{G}$ are Markov equivalent, thus $\tilde{G} \in \tilde{E}$, and so, as observed above, also $\tilde{G} \in \tilde{E}^{\cap A}$, which completes the proof. ■

The last lemma required for proving Theorem 5 shows that for a stable distribution, ActiveBNSL is guaranteed to accept families early.

Lemma 11 *Let $\mathcal{D}$ be a (γ, V)-stable distribution. Consider a run of* ActiveBNSL *in which η holds. Then,* ActiveBNSL *accepts at least $|V|$ families by the end of the first round t which satisfies $\epsilon_t \leq \gamma/(2d)$.*

[1] Also known as an "immorality" (see, e.g., Verma and Pearl, 1990)

Proof Suppose that ActiveBNSL has accepted less than $|V|$ families until some iteration j. Then $V \setminus \mathcal{V}_{(j)} \neq \emptyset$. Let $f \in \mathbf{U}(\mathcal{G}_j^*) \cap \mathrm{Cand}(\mathcal{V}_{(j)})$ be a family of some variable in $V \setminus \mathcal{V}_{(j)}$ that belongs to an optimal structure in $\mathcal{G}_j^*$. Note that such a family exists, since by Lemma 3, $\mathcal{G}_j^* \neq \emptyset$. Let $\tilde{E} \in \mathcal{E}_{d,k}$ such that $\tilde{E}^{\cap j} \neq \emptyset$ and $\mathcal{S}(\tilde{E}) \geq \mathcal{S}^* - \gamma$. Let $G^* \in \mathcal{G}_j^*$. By Lemma 9 with $A := \mathrm{Acc}_j$, there exists a graph $\tilde{G} \in \tilde{E}^{\cap j}$ such that $f \in \tilde{G}$. Therefore, $f \in \mathbf{U}(\tilde{E}^{\cap j})$. Since this holds for any $\tilde{E} \in \mathcal{E}_{d,k}$ that satisfies the conditions above, it follows that for $S_j^{\neg}(f)$ as defined in Lemma 6, we have that $S_j^{\neg}(f) < \mathcal{S}^* - \gamma$. The conditions of Lemma 6 thus hold if $2\theta_j \leq \gamma$. In this case, some family will be accepted at iteration j. In round t, $\theta_j \leq d\epsilon_t$, and the round only ends when no additional families are accepted. Therefore, at least $|V|$ families will be accepted until the end of the first round t with $\epsilon_t \leq \gamma/(2d)$. ■

We are now ready to prove Theorem 5.

Proof [of Theorem 5] Since $\mathcal{D}$ is a discrete distribution, the sample complexity upper bounds in Eq. (5) and Eq. (2) can be used. To upper bound the sample complexity of ActiveBNSL, observe that by Lemma 11, on the first round t with $\epsilon_t \leq \gamma/(2d)$, at least $|V| = v$ families have been accepted. Following Eq. (5) and the notation therein, the sample complexity of ActiveBNSL can be upper bounded by setting $n = 1$, $V_1 = v$, $t_1 = t$, and getting an upper bound of

$$\widetilde{O}\left(\left(v^{k+1}\cdot\frac{1}{\epsilon_t^2} + d^k(d-v)^3\cdot\frac{1}{\epsilon^2}\right)\cdot\log(d^{k+1}/\delta)\right) \quad = \widetilde{O}\left(\left(\frac{d^2v^{k+1}}{\gamma^2}+\frac{d^k(d-v)^3}{\epsilon^2}\right)\cdot k\log(1/\delta)\right),$$

where we used the fact that $\epsilon_t \in (\gamma/(4d), \gamma/(2d)]$. This proves Eq. (6).

By Eq. (2), the sample complexity of the naive algorithm is $\widetilde{\Omega}(\frac{d^{k+3}}{\epsilon^2}\cdot k\log(1/\delta))$. Substituting $v = d - O(1)$ in Eq. (6), we get that ActiveBNSL uses $\widetilde{O}\left((\frac{d^{k+3}}{\gamma^2}+\frac{d^k}{\epsilon^2})\cdot k\log(1/\delta)\right)$ samples. Dividing this by the sample complexity of the naive algorithm, we get a sample complexity ratio of $\widetilde{O}((\epsilon/\gamma)^2 + 1/d^3)$. For $\epsilon = O(\gamma d^{-3/2})$, this ratio is $\widetilde{O}(1/d^3)$. The sample complexity of ActiveBNSL is thus a factor of $\widetilde{\Omega}(d^3)$ smaller than that of the naive algorithm in this case. This completes the proof. ■

Theorem 5 gives a set of conditions that, if satisfied by a distribution, lead to a significant sample complexity reduction when using ActiveBNSL. In the next section, we give a construction which satisfies this set of conditions.

8. A Class of Stable Distributions

In this section, we give an explicit construction of a family of distributions that are stable according to Def. 4. The construction starts with a distribution with a unique optimal EC, and augments it with a noisy version of one of its variables. This leads to a structure learning problem in which most variable dependencies are easy to identify, but it is difficult to discern between the two versions of the noisy variable. In this situation, the advantage of ActiveBNSL is manifested, since it can request more samples from the families that are more difficult to choose from.

Let $\mathcal{D}_1$ be a distribution over a finite set of at least k variables $\mathbf{X}_1$, one of which is denoted X_a, which satisfies the following properties for some values $\beta, \alpha > 0$.

(I) $\mathcal{D}_1$ has a unique optimal EC, and the difference in scores between this EC and the next-best EC is at least β.

(II) $X_a \in \mathbf{X}_1$ does not have children in any of the structures in the optimal EC

(III) $H(X_a \mid \mathbf{X}_1 \setminus \{X_a\}) = \alpha$.

Property (II) holds, for instance, in the case that the optimal EC includes a graph in which X_a has no children and no edges between any of its parents (see Figure 1 for illustration). By Theorem 10, in this case, X_a is the child in a v-structure with each two of its parents, and since v-structures are preserved within the optimal EC, X_a has no children in any of the equivalent structures.

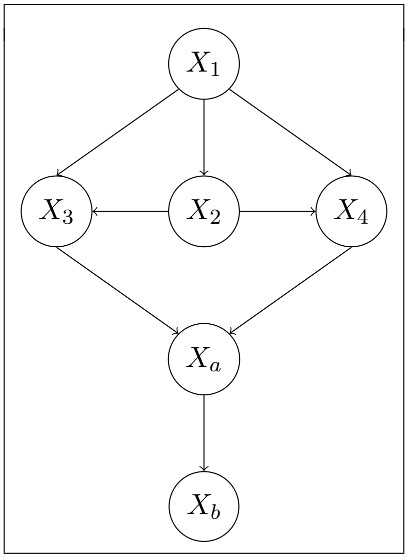

Figure 1: An illustration of an EC that satisfies property (II).

We now define a distribution which is similar to $\mathcal{D}_1$, except that X_a has a slightly noisy copy, denoted X_b. Let the set of variables be $\mathbf{X} := \mathbf{X}_1 \cup \{X_b\}$, and set $d := |\mathbf{X}|$. For any $\lambda \in \Big(0, \min(\alpha, \frac{\beta}{3d})\Big)$, we define the distribution $\mathcal{D}_2(\lambda)$ of $\mathbf{X}$ in which the marginal of $\mathcal{D}_2(\lambda)$ over $\mathbf{X}_1$ is $\mathcal{D}_1$, and X_b is defined as follows:

(IV) X_b is an independently noisy version of X_a. Formally, there is a hidden Bernoulli variable C which is independent of $\mathbf{X}_1$, such that $P(X_b = X_a \mid C = 1) = 1$, and $(X_b \perp \mathbf{X}_1) \mid C = 0$.

(V) The probability $P(C = 1)$ is such that $\max(H(X_b \mid X_a), H(X_a \mid X_b)) = \lambda$.

The following theorem shows that $\mathcal{D}_2$ is stable where the set V includes all but one variable. Thus, Theorem 5 holds with $q = 1$, leading to a sample complexity improvement factor as large as $\widetilde{\Omega}(d^3)$ for this construction.

Theorem 12 *$\mathcal{D}_2(\lambda)$ is a (γ, V)-stable distribution for $V := \mathbf{X}_1 \setminus X_a$ and for all $\gamma \leq \beta - 3d\lambda$.*

Before turning to the proof of Theorem 12, we note that Theorem 5 assumes a given required accuracy ϵ. If the score of the second-best EC is much smaller than $\mathcal{S}^* - \epsilon$, and this is known in advance, then one could set ϵ to a larger value, thus potentially voiding the conclusion of Theorem 5. The following theorem shows that this is not the case for the construction of $\mathcal{D}_2$. Its proof is provided in Appendix B.

Theorem 13 *For $\epsilon \in \Big(0, 2 \cdot \min(\alpha, \frac{\beta}{3d})\Big)$, there exists a value of λ such that in $\mathcal{D}_2(\lambda)$ there exists a non-optimal EC with a score of at least $S^* - \epsilon$.*

We now prove Theorem 12 by proving several lemmas. The first lemma gives useful technical inequalities. Its proof is provided in Appendix B.

Lemma 14 *Assume the distribution $\mathcal{D}_2(\lambda)$ defined above. For any $\Pi, \Pi_b, \Pi_a \subseteq \mathbf{X}_1$ and any $Y \in \mathbf{X}_1$, the following holds:*

$$|H(X_b \mid \Pi) - H(X_a \mid \Pi)| \leq \lambda, \tag{7}$$

$$|H(Y \mid \Pi, X_b) - H(Y \mid \Pi, X_a)| \leq 3\lambda. \tag{8}$$

The next lemma shows that any DAG for $\mathcal{D}_2$ can be transformed to a DAG with a similar score in which X_b has no children. The proof of the lemma is provided in Appendix B.

Lemma 15 *Let $G \in \mathcal{G}_{d,k}$. There exists a graph $\tilde{G} \in \mathcal{G}_{d,k}$ that satisfies the following properties:*

1. *$|\mathcal{S}(\tilde{G}) - \mathcal{S}(G)| < 3d\lambda$;*
2. *In $\tilde{G}$, X_b has no children;*
3. *The children of X_a in $\tilde{G}$ are exactly the union of the children of X_a and the children of X_b in G (except for X_a, if it is a child of X_b in G).*

Next, we use Lemma 15 to prove that property 1 in Def. 4 holds for $\mathcal{D}_2$.

Lemma 16 *Assume the distribution $\mathcal{D}_2(\lambda)$ as defined above. Let G be a structure such that at least one of X_a, X_b has children other than X_a, X_b. Then, $\mathcal{S}(G) < \mathcal{S}^* - \beta + 3d\lambda$.*

Proof In this proof, all scores are calculated for the joint distribution defined by $\mathcal{D}_2(\lambda)$. Let Π_b be a parent set of size k for X_b that maximizes the family score. Formally,

$$\Pi_b \in \operatorname*{argmax}_{\Pi \subseteq \mathbf{X}_1 : |\Pi| = k} \mathcal{S}(\{\langle X_b, \Pi \rangle\}).$$

Let $f_b^* = \langle X_b, \Pi_b \rangle$. Denote by $\mathcal{G}_{d-1,k}$ the set of DAGs with an in-degree of at most k over $\mathbf{X}_1$. Let $G_1 \in \mathcal{G}_{d-1,k}$ be such a DAG with a maximal score, and denote this score $\bar{\mathcal{S}}$. Let $G_1' = G_1 \cup \{f_b^*\}$. This is a DAG in $\mathcal{G}_{d,k}$ (over $\mathbf{X} \equiv \mathbf{X}_1 \cup \{X_b\}$), with a score $\mathcal{S}(G_1') = \bar{\mathcal{S}} + \mathcal{S}(\{f_b^*\})$. Therefore, $\mathcal{S}^* \geq \bar{\mathcal{S}} + \mathcal{S}(\{f_b^*\})$.

Now, let G be a structure such that at least one of X_a, X_b has children other than X_a, X_b. We consider two cases:

1. X_b has no children in G.

2. X_b has children in G.

In case 1, X_a has children in $\mathbf{X}_1$. Let f_b be the family of X_b in G. The graph $G \setminus \{f_b\}$ is a DAG over $\mathbf{X}_1$ in which X_a has children. Therefore, by assumption (II) in the definition of $\mathcal{D}_1$, $G \setminus \{f_b\}$ is not an optimal DAG, and so by assumption (I), it holds that $\mathcal{S}(G \setminus \{f_b\}) \leq \bar{\mathcal{S}} - \beta$. Therefore, we have

$$\mathcal{S}(G) = \mathcal{S}(G \setminus \{f_b\}) + \mathcal{S}(\{f_b\}) \leq \mathcal{S}(G \setminus \{f_b\}) + \mathcal{S}(\{f_b^*\}) \leq \bar{\mathcal{S}} - \beta + \mathcal{S}(\{f_b^*\}) \leq \mathcal{S}^* - \beta.$$

The last inequality follows from the fact proved above that $\mathcal{S}^* \geq \bar{\mathcal{S}} + \mathcal{S}(\{f_b^*\})$. This proves the statement of the lemma for case 1.

In case 2, X_b has children in G. Denote this set of children by W. By Lemma 15, there exists a graph $\tilde{G} \in \mathcal{G}_{d,k}$ such that $|\mathcal{S}(G) - \mathcal{S}(\tilde{G})| < 3d\lambda$, X_b has no children, and the children of X_a are $W \setminus \{X_a\}$ as well as the children of X_a in G. Now, consider two cases.

- $W \neq \{X_a\}$. In this case, $W \setminus \{X_a\} \neq \emptyset$ but cannot include X_b, and so in $\tilde{G}$, X_a has at least one child which is different from X_b.
- $W = \{X_a\}$. In this case, from the definition of G, it follows that X_a has children other than X_b in G, and so also in $\tilde{G}$.

In both cases, we get that in $\tilde{G}$, X_b has no children and X_a has at least one child other than X_b. Thus, $\tilde{G}$ satisfies the conditions of case 1 discussed above. It follows that $\mathcal{S}(\tilde{G}) \leq \mathcal{S}^* - \beta$. Therefore, $\mathcal{S}(G) < \mathcal{S}^* - \beta + 3d\lambda$. This proves the statement of the lemma for case 2, thus completing the proof. ■

The next lemma shows that property 2 in Def. 4 of a stable distribution holds for $\mathcal{D}_2$, for any $\gamma < \beta$

Lemma 17 *Consider the marginal of $\mathcal{D}_2(\lambda)$ on $\mathbf{X}_1 \setminus \{X_a\}$. There is a unique optimal EC for this distribution, and the difference in scores between the optimal EC and the second-best EC is at least β.*

Proof Let E_1 be an optimal EC for the considered marginal over $\mathbf{X}_1 \setminus \{X_a\}$. Let $G_1 \in E_1$. Let Π_a be a parent set of size k for X_a that maximizes the family score. Formally,

$$\Pi_a \in \operatorname*{argmax}_{\Pi \subseteq \mathbf{X}_1 \setminus X_a : |\Pi| = k} \mathcal{S}(\{\langle X_a, \Pi \rangle\}).$$

Denote $f_a^* := \langle X_a, \Pi_a \rangle$. Let G be an optimal graph over $\mathbf{X}_1$, and let f_a be the family of X_a in G. By assumption (II) on $\mathcal{D}_1$, X_a has no children in G. Therefore, $G \setminus \{f_a\}$ is a DAG with an in-degree at most k over $\mathbf{X}_1 \setminus \{X_a\}$. Thus, $\mathcal{S}(G \setminus \{f_a\}) \leq \mathcal{S}(G_1)$. It follows that

$$\mathcal{S}(G) = \mathcal{S}(G \setminus \{f_a\}) + \mathcal{S}(\{f_a\}) \leq \mathcal{S}(G \setminus \{f_a\}) + \mathcal{S}(\{f_a^*\}) \leq \mathcal{S}(G_1) + \mathcal{S}(\{f_a^*\}) = \mathcal{S}(G_1 \cup \{f_a^*\}).$$

Therefore, $G_1 \cup \{f_a^*\}$ is also optimal on $\mathbf{X}_1$. Now, assume for contradiction that there is another optimal EC over $\mathbf{X}_1 \setminus \{X_a\}$, denoted E_2, and let $G_2 \in E_2$. By the same analysis as above, $G_2 \cup \{f_a^*\}$ is also optimal on $\mathbf{X}_1$. However, by Theorem 10, since G_1 and G_2 are not equivalent, then $G_1 \cup \{f_a^*\}$ and $G_2 \cup \{f_a^*\}$ are also not equivalent, contradicting assumption (I) on $\mathcal{D}_1$. Therefore, E_1 is the only optimal EC over $\mathbf{X}_1 \setminus \{X_a\}$.

Now, let G_3 be non-optimal DAG with an in-degree at most k over $\mathbf{X}_1 \setminus \{X_a\}$, so $\mathcal{S}(G_3) < \mathcal{S}(G_1)$. By assumption (I) on $\mathcal{D}_1$, $\mathcal{S}(G_3 \cup \{f_a^*\}) \leq \mathcal{S}(G_1 \cup \{f_a^*\}) - \beta$. Therefore, $\mathcal{S}(G_3) \leq \mathcal{S}(G_1) - \beta$. This proves the claim. ■

Theorem 12 is now an immediate consequence of the lemmas above, since both properties of Def. 4 hold for all positive $\gamma \leq \beta - 3d\lambda$, by Lemma 16 and Lemma 17.

9. Experiments

In this section, we report an empirical comparison between the naive algorithm, given in Section 4, and ActiveBNSL, given in Section 5. The code for both algorithms and for the experiments below is available at `https://github.com/noabdavid/activeBNSL`.

We implemented the algorithms for the case of discrete distributions using the plug-in estimator for conditional entropy to calculate the empirical score $\hat{\mathcal{S}}$ (see Eq. (1)), as discussed in Section 4. To avoid spurious score ties, we augmented the empirical score with an additional tie-breaking term. Since the empirical entropy never increases when adding a potential parent to a family, any family with a maximal score that has fewer than k parents can be augmented with additional parents without decreasing the score. This creates spurious optimal families, thus significantly reducing the number of cases where an optimal family can be identified and accepted by ActiveBNSL. To break the score ties, we added to the entropy of each family a small penalty term that prefers smaller families. Formally, we used the following modified empirical score:

$$\hat{\mathcal{S}}(G) := -\sum_{i \in [d]} \left(\hat{H}(X_i \mid \Pi_i(G)) + \beta |\Pi_i(G)| \cdot \hat{H}(X_i) \right), \tag{9}$$

where $\beta := 0.001$.

We now describe how we implemented the non-trivial steps in each algorithm. The naive algorithm aims to output a network structure in $\mathcal{G}_{d,k}$ that maximizes the empirical score, as defined above. Finding a structure with the maximal score is computationally hard in the general case, as discussed in Section 1. We use the well-established algorithm GOBNILP (Cussens, 2011), as implemented in the eBNSL package[2] (Liao et al., 2019), which attempts to find such a structure using a linear programming approach.

We now turn to the implementation of ActiveBNSL. To compute N_t, the number of required samples (see line 3), we used the formula given in the bound in Lemma 1. To compute L_j, the set of possible equivalence classes (see line 9), we used the eBNSL algorithm, also implemented in the eBNSL package. The input to eBNSL includes the score of each family and a positive real number ϵ. The algorithm heuristically attempts to output all network structures in $\mathcal{G}_{d,k}$ with a score gap of at most ϵ from the maximal achievable score. To compute L_j, ActiveBNSL sets the ϵ input value of eBNSL to θ_j. The list of structures provided as output by eBNSL is then divided into equivalence classes, using the characterization given in Theorem 10. To impose the constraint that L_j should only

[2] `https://github.com/alisterl/eBNSL`

d	**r**	$\epsilon \equiv$ **d/r**	**N naive**	**N active**	**sample ratio**	**% accepted families**
6	2^{13}	0.00073	**43**$\times10^{12}$	51$\times10^{12}$	118%	0%
	2^{15}	0.00018	**88**$\times10^{13}$	104$\times10^{13}$	118%	0%
	2^{17}	0.00005	**17**$\times10^{15}$	20$\times10^{15}$	118%	0%
	2^{19}	0.00001	**33**$\times10^{16}$	39$\times10^{16}$	117%	0%
7	2^{13}	0.00085	81$\times10^{12}$	**73**$\pm10\times10^{12}$	90%	11% ±5.71%
	2^{15}	0.00021	16$\times10^{14}$	**13**$\times10^{14}$	84%	14%
	2^{17}	0.00005	32$\times10^{15}$	**25**$\pm4\times10^{15}$	79%	17% ±8.57%
	2^{19}	0.00001	62$\times10^{16}$	**21**$\times10^{16}$	34%	42%
8	2^{13}	0.00098	**13**$\times10^{13}$	15$\times10^{13}$	116%	0%
	2^{15}	0.00024	27$\times10^{14}$	**24**$\times10^{14}$	87%	12%
	2^{17}	0.00006	54$\times10^{15}$	**47**$\times10^{15}$	87%	12%
	2^{19}	0.00002	105$\times10^{16}$	**26**$\times10^{16}$	24%	50%
9	2^{13}	0.00110	21$\times10^{13}$	**19**$\times10^{13}$	91%	11%
	2^{15}	0.00027	43$\times10^{14}$	**21**$\times10^{14}$	48%	33%
	2^{17}	0.00007	85$\times10^{15}$	**41**$\times10^{15}$	48%	33%
	2^{19}	0.00002	165$\times10^{16}$	**30**$\times10^{16}$	18%	55%
10	2^{13}	0.00122	31$\times10^{13}$	**24**$\pm2\times10^{13}$	76%	18% ±4.00%
	2^{15}	0.00031	64$\times10^{14}$	**34**$\times10^{14}$	54%	30%
	2^{17}	0.00008	126$\times10^{15}$	**68**$\times10^{15}$	54%	30%
	2^{19}	0.00002	244$\times10^{16}$	**69**$\pm43\times10^{16}$	28%	49% ±13.00%
11	2^{13}	0.00134	45$\times10^{13}$	**34**$\times10^{13}$	75%	18%
	2^{15}	0.00034	90$\times10^{14}$	**53**$\times10^{14}$	58%	27%
	2^{17}	0.00008	17$\times10^{16}$	**10**$\times10^{16}$	58%	27%
	2^{19}	0.00002	34$\times10^{17}$	**20**$\times10^{17}$	59%	27%
12	2^{13}	0.00146	**61**$\times10^{13}$	72$\times10^{13}$	116%	0%
	2^{15}	0.00037	124$\times10^{14}$	**34**$\pm6\times10^{14}$	27%	47% ±3.82%
	2^{17}	0.00009	245$\times10^{15}$	**67**$\pm16\times10^{15}$	27%	51% ±3.33%
	2^{19}	0.00002	475$\times10^{16}$	**74**$\pm22\times10^{16}$	15%	58% ±5.03%

Table 3: Experiment results for the synthetic networks. N stands for the number of samples used by each algorithm. "sample ratio" is the sample size savings rate by ActiveBNSL compared to the naive algorithm (N active / N naive). “% accepted families” is the fraction of the families that ActiveBNSL accepted before the last stage.

include structures consistent with the accepted families, as required by the definition of L_j, ActiveBNSL provides eBNSL with structural constraints as additional input. These constraints specify sets of edges that must or must not be included in the output structures.

Network	**d**	**r**	$\epsilon \equiv$ **d/r**	**N naive**	**N active**	**sample ratio**	**% accepted families**
Cancer	5	2^{11}	0.00244	**96**$\times10^{10}$	116$\times10^{10}$	120%	0%
		2^{13}	0.00061	**20**$\times10^{12}$	24$\times10^{12}$	119%	0%
		2^{15}	0.00015	41$\times10^{13}$	**30**$\times10^{13}$	73%	20%
		2^{17}	0.00004	81$\times10^{14}$	**13**$\times10^{14}$	16%	60%
		2^{19}	0.00001	157$\times10^{15}$	**24**$\times10^{15}$	15%	60%
Earthquake	5	2^{11}	0.00244	**96**$\times10^{10}$	116$\times10^{10}$	120%	0%
		2^{13}	0.00061	**20**$\times10^{12}$	24$\times10^{12}$	119%	0%
		2^{15}	0.00015	**41**$\times10^{13}$	49$\times10^{13}$	119%	0%
		2^{17}	0.00004	81$\times10^{14}$	**21**$\pm9\times10^{14}$	25%	52% ±9.80%
		2^{19}	0.00001	157$\times10^{15}$	**47**$\pm18\times10^{15}$	30%	48% ±9.80%
Survey	6	2^{11}	0.00293	**20**$\times10^{11}$	24$\times10^{11}$	118%	0%
		2^{13}	0.00073	43$\times10^{12}$	**21**$\times10^{12}$	48%	33%
		2^{15}	0.00018	88$\times10^{13}$	**22**$\times10^{13}$	25%	50%
		2^{17}	0.00005	174$\times10^{14}$	**44**$\times10^{14}$	25%	50%
		2^{19}	0.00001	338$\times10^{15}$	**86**$\times10^{15}$	25%	50%
Asia	8	2^{11}	0.00391	**65**$\times10^{11}$	76$\times10^{11}$	116%	0%
		2^{13}	0.00098	**13**$\times10^{13}$	15$\times10^{13}$	116%	0%
		2^{15}	0.00024	27$\times10^{14}$	**24**$\times10^{14}$	87%	12%
		2^{17}	0.00006	545$\times10^{14}$	**64**$\pm28\times10^{14}$	11%	65% ±7.50%
		2^{19}	0.00002	1055$\times10^{15}$	**94**$\pm39\times10^{15}$	8%	67% ±6.12%
Sachs	11	2^{11}	0.00537	**48**$\times10^{12}$	54$\pm6\times10^{12}$	111%	1% ±5.45%
		2^{13}	0.00134	**10**$\times10^{14}$	11$\pm1\times10^{14}$	109%	2% ±5.82%
		2^{15}	0.00034	204$\times10^{14}$	**57**$\pm31\times10^{14}$	28%	59% ±22.73%
		2^{17}	0.00008	402$\times10^{15}$	**25**$\pm15\times10^{15}$	6%	75% ±4.17%
		2^{19}	0.00002	776$\times10^{16}$	**49**$\pm14\times10^{16}$	6%	71% ±2.73%

Table 4: Experiment results for benchmark networks. N stands for the number of samples used by each algorithm. "sample ratio" is the sample size savings rate by ActiveBNSL compared to the naive algorithm (N active / N naive). “% accepted families” is the fraction of the families that ActiveBNSL accepted before the last stage.

To find a a structure that maximizes the empirical score, ActiveBNSL uses GOBNILP, providing it with constraints that impose the inclusion of accepted families.

We ran experiments using data generated from synthetically constructed networks, and from discrete networks from the Bayesian Network Repository,[3] which provides Bayesian networks that are commonly used as benchmarks. We generated synthetic networks with

[3] `https://www.bnlearn.com/bnrepository/`

6-12 nodes, and tested all benchmark networks from the "small networks" category of the repository, which includes networks with up to 11 nodes. In all the networks, all nodes represented Bernoulli random variables taking values in $\{0, 1\}$, except for the benchmark networks Survey and Sachs, which include multinomially distributed variables with 3 possible values.

The synthetic networks describe stable distributions of the type presented in Section 8, in which each of the network variables has at most two parents. The nodes in a network with d nodes, denoted $\mathcal{B}_d$, are denoted: $X_1, \dots, X_{d-2}, X_a, X_b$. Figure 2 illustrates the graph structures for $\mathcal{B}_6$ and $\mathcal{B}_7$. In all networks, X_1 has no parents and is a Bernoulli(1-ρ) random variable, where $\rho = 0.99$. X_2 has only X_1 as a parent and is equal to X_1 with an independent probability of $1 - \rho$ (otherwise, it is equal to $1 - X_1$). Each of the variables X_i for $i \in \{3, ..., d-2\}$ has two parents $X_j, X_{j'}$ for some $j, j' < i$. The value of X_i is $\text{xor}(X_j, X_{j'})$ with an independent probability of ρ. In all networks, X_a is the child of X_3 and X_4, and the value of X_a is $\text{xor}(X_3, X_4)$ with an independent probability of ρ. X_b is a slightly noisy version of X_a, such that $X_a = X_b$ with an independent probability of $1 - 5 \cdot 10^{-6}$. This construction satisfies the conditions in Section 8 with $\mathbf{X}_1 := \{X_1, \dots, X_{d-2}, X_a\}$. Since X_b is almost always equal to X_a, it is hard to distinguish between them using samples. In particular, the structure in which X_3, X_4 are the parents of X_b and the latter is the parent of X_a has a score which is very close to optimal. The advantage of the active algorithm is in being able to focus on observations that distinguish between X_a and X_b.

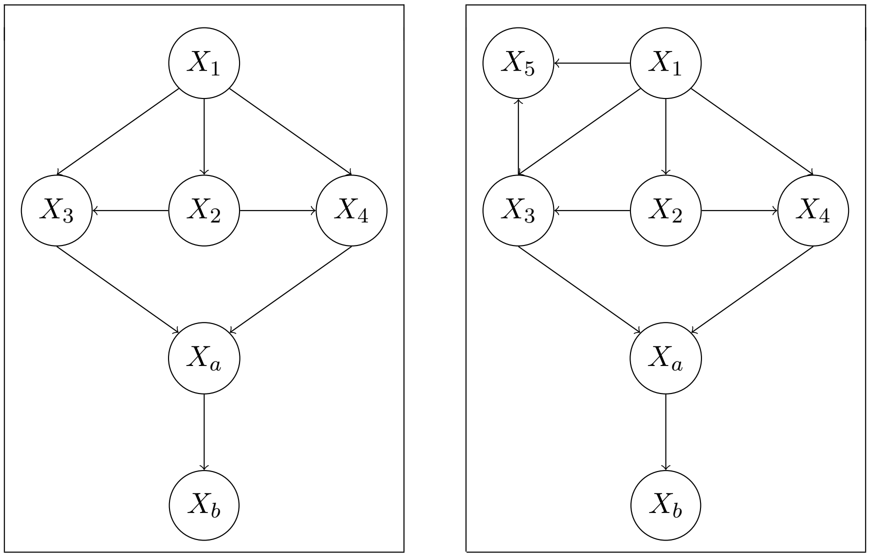

Figure 2: Illustration of the networks $\mathcal{B}_6$ and $\mathcal{B}_7$ used in the experiments.

In all of the synthetic and benchmark networks, the true network could be represented with $k = 2$, except for the benchmark Sachs network, in which the true network requires $k = 3$. Accordingly, the value of k was set to 2 for all but the Sachs network, where it was set to 3. In all of the experiments, we set $\delta = 0.05$.

To set the values of ϵ in each experiment in a comparable way, we adjusted this value by the number of network nodes, denoted by d, by fixing the ratio $r := d/\epsilon$ and calculating

ϵ based on this ratio. We ran experiments with $r := 2^j$ for a range of values of j. Each experiment configuration was ran 10 times, and we report the average and standard deviation of the results. In all of the experiments, both the naive algorithm and ActiveBNSL found an ϵ-optimal structure. Note that for the naive algorithm, all repeated experiments use the same sample size, since this size is pre-calculated.

Detailed results of the synthetic and the benchmark experiments are provided in Table 3 and Table 4, respectively. The "sample ratio" column calculates the average reduction in the number of samples when using ActiveBNSL instead of the naive algorithm, where a number below 100% indicates a reduction in the number of samples. The last column lists the percentage of families that were accepted early in each experiment.

It can be seen in both tables that ActiveBNSL obtains a greater reduction for larger values of r (smaller values of ϵ). This is also depicted visually in Figure 3, which plots the sample sizes obtained by the algorithms for the synthetic networks as a function of d for three values of r. It can be seen that for $r = 2^{13}$, there is almost no difference between the required sample sizes, while some sample saving is observed for $r = 2^{15}$, and a large saving is observed for $r = 2^{19}$. For the two greater values of r, it is apparent that the advantage increases with d, showing that larger values of d lead to a larger sample size reduction. In the most favorable configurations for both the synthetic and the benchmark networks, when $r = 2^{19}$, and the largest tested value of d, ActiveBNSL samples only 15% and 6%, respectively, of the number of samples required by the naive algorithm.

The increase in sample size reduction when increasing r can be explained by observing in both tables that when r is small, only a small fraction of the families is accepted early by ActiveBNSL, leading to both algorithms requiring about the same sample size. In general, ActiveBNSL is more successful when it accepts a larger fraction of the families early.

To summarize, the reported experiments demonstrate that despite the computational hardness of ActiveBNSL in the general case, it can be successfully implemented in practice using established approaches, and obtain a substantial improvement in the number of required samples. We note that the number of samples required for both the naive algorithm and ActiveBNSL is quite large in our experiments. This is a consequence of the specific estimator and concentration bounds that we use, which may be loose in some cases. We expect that this number can be significantly reduced by using tighter concentration bounds tailored to this task. We leave this challenge for future work.

10. Discussion

This is the first work to study active structure learning for Bayesian networks in an observational setting. The proposed algorithm, ActiveBNSL, can have a significantly improved sample complexity in applicable cases, and incurs a negligible sample complexity penalty in other cases.

We considered the case where k, the maximal number of parents, is smaller by exactly one than the number of variable observations that can be made in a single sample. We now briefly discuss the implications of other cases. Denote by l the number of variables that can be simultaneously observed, as determined based on external constraints (for instance, l may indicate the maximal number of tests that each patient in an experiment can undergo). As mentioned in Section 2, if $k > l - 1$ then the score cannot be computed directly. Thus, if

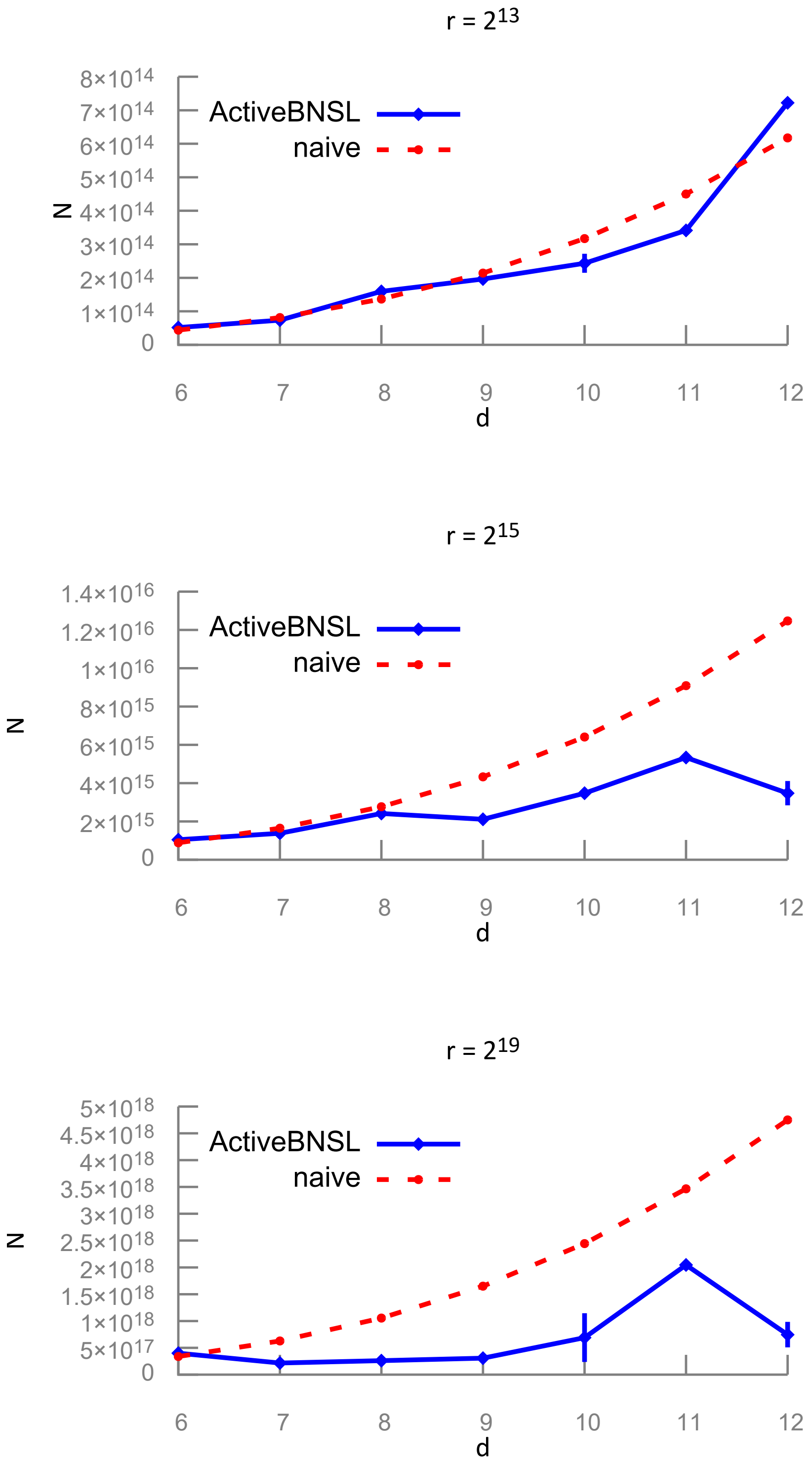

Figure 3: Number of samples taken by ActiveBNSL (active) and the naive algorithm, N, as a function of d, the number of nodes in the network, for three values of r.

there are no other limitations on k, one may set $k := l - 1$ to obtain the least constrained problem. Nonetheless, in some cases this value of k may be too large, especially if l is large, for instance due to computational reasons. This raises an interesting question: How do the results above generalize for the case of $k < l - 1$? In this case, a non-interactive naive algorithm may be defined by fixing a set of subsets of size l that cover all the possible subsets of size $k + 1$, and sampling each of those sufficiently for uniform convergence. A version of ActiveBNSL might then be suggested, which samples the same set of fixed subsets of size l, and stops sampling such a subset only if a family was accepted for each of the variables in the subset. Characterizing the improvement of the active algorithm over the naive one in this case would depend on the initial choice of fixed subsets and their relationship to the underlying distribution. We leave a full analysis of this scenario for future work.

We characterized a family of distributions in which ActiveBNSL can obtain a significant sample complexity reduction, and demonstrated the reduction in experiments. A full characterization of distributions in which a sample complexity reduction is possible is an interesting open problem for future work.

References

Steen A Andersson, David Madigan, and Michael D Perlman. A characterization of Markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25:505–541, 1997.

András Antos and Ioannis Kontoyiannis. Convergence properties of functional estimates for discrete distributions. *Random Structures & Algorithms*, 19:163–193, 2001.

Pranjal Awasthi, Maria Florina Balcan, and Konstantin Voevodski. Local algorithms for interactive clustering. *The Journal of Machine Learning Research*, 18:75–109, 2017.

Mark Bartlett and James Cussens. Advances in Bayesian network learning using integer programming. In *the Twenty Ninth Conference on Uncertainty in Artificial Intelligence*, pages 181–191, 2013.

Mark Bartlett and James Cussens. Integer linear programming for the Bayesian network structure learning problem. *Artificial Intelligence*, 244:258–271, 2017.

Shai Ben-David and Eli Dichterman. Learning with restricted focus of attention. *Journal of Computer and System Sciences*, 56(3):277–298, 1998.

Remco Ronaldus Bouckaert. *Bayesian belief networks: from construction to inference.* PhD thesis, Utrecht University, 1995.

Wray Buntine. Theory refinement on Bayesian networks. In *Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence*, pages 52–60, 1991.

Baoping Cai, Lei Huang, and Min Xie. Bayesian networks in fault diagnosis. *IEEE Transactions on Industrial Informatics*, 13:2227–2240, 2017.

Baoping Cai, Xiangdi Kong, Yonghong Liu, Jing Lin, Xiaobing Yuan, Hongqi Xu, and Renjie Ji. Application of Bayesian networks in reliability evaluation. *IEEE Transactions on Industrial Informatics*, 15:2146–2157, 2019.

Cassio P de Campos and Qiang Ji. Efficient structure learning of Bayesian networks using constraints. *Journal of Machine Learning Research*, 12:663–689, 2011.

Clément L Canonne, Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Testing conditional independence of discrete distributions. In *Information Theory and Applications Workshop*, pages 1–57, 2018.

Nicolo Cesa-Bianchi, Shai Shalev-Shwartz, and Ohad Shamir. Efficient learning with partially observed attributes. *Journal of Machine Learning Research*, 12(10):2857–2878, 2011.

David Maxwell Chickering. Learning Bayesian networks is NP-complete. In *Learning from data*, pages 121–130. Springer, 1996.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3(11):507–554, 2002.

Gregory F Cooper and Edward Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.

James Cussens. Bayesian network learning with cutting planes. In *the Tweny Seventh Conference on Uncertainty in Artificial Intelligence*, pages 153–160, 2011.

Gautam Dasarathy. Gaussian graphical model selection from size constrained measurements. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 1302–1306. IEEE, 2019.

Gautamd Dasarathy, Aarti Singh, Maria-Florina Balcan, and Jong H Park. Active learning algorithms for graphical model selection. In *Artificial Intelligence and Statistics*, pages 1356–1364, 2016.

Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3-4):601–620, 2000.

Peter Haddawy, AHM Imrul Hasan, Rangwan Kasantikul, Saranath Lawpoolsri, Patiwat Sa-angchai, Jaranit Kaewkungwal, and Pratap Singhasivanon. Spatiotemporal Bayesian networks for malaria prediction. *Artificial Intelligence in Medicine*, 84:127–138, 2018.

Jarvis Haupt, Rui Castro, and Robert Nowak. Distilled sensing: Selective sampling for sparse signal recovery. In *Artificial Intelligence and Statistics*, pages 216–223, 2009.

Elad Hazan and Tomer Koren. Linear regression with limited observation. In *Proceedings of the Twenty Ninth International Conference on Machine Learning*, pages 1865–1872. Omnipress, 2012.

Yang-Bo He and Zhi Geng. Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research*, 9:2523–2547, 2008.

David Heckerman. A tutorial on learning with Bayesian networks. In *Learning in Graphical Models*, pages 301–354. Springer, 1998.

David Heckerman, Dan Geiger, and David M Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

David Heckerman, Christopher Meek, and Gregory Cooper. A Bayesian approach to causal discovery. In *Innovations in Machine Learning*, pages 1–28. Springer, 2006.

Hengyi Hu and Larry Kerschberg. Evolving medical ontologies based on causal inference. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 954–957, 2018.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.

Doron Kukliansky and Ohad Shamir. Attribute efficient linear regression with distribution-dependent sampling. In *International Conference on Machine Learning*, pages 153–161, 2015.

Guoliang Li and Tze-Yun Leong. Active learning for causal Bayesian network structure with non-symmetrical entropy. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 290–301, 2009.

Yifeng Li, Haifen Chen, Jie Zheng, and Alioune Ngom. The max-min high-order dynamic Bayesian network for learning gene regulatory networks with time-delayed regulations. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 13:792–803, 2016.

Zhenyu A Liao, Charupriya Sharma, James Cussens, and Peter van Beek. Finding all Bayesian network structures within a factor of optimal. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7892–7899, 2019.

Li Liu, Shu Wang, Bin Hu, Qingyu Qiong, Junhao Wen, and David S Rosenblum. Learning structures of interval-based Bayesian networks in probabilistic generative model for human complex activity recognition. *Pattern Recognition*, 81:545–561, 2018.

Matthew L Malloy and Robert D Nowak. Near-optimal adaptive compressed sensing. *IEEE Transactions on Information Theory*, 60:4001–4012, 2014.

Simone Marini, Emanuele Trifoglio, Nicola Barbarini, Francesco Sambo, Barbara Di Camillo, Alberto Malovini, Marco Manfrini, Claudio Cobelli, and Riccardo Bellazzi. A dynamic Bayesian network model for long-term simulation of clinical complications in type 1 diabetes. *Journal of Biomedical Informatics*, 57:369–376, 2015.

Colin McDiarmid. On the method of bounded differences. *Surveys in Combinatorics*, 141: 148–188, 1989.

Christopher Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 403–410, 1995.

Liam Paninski. Estimation of entropy and mutual information. *Neural Computation*, 15: 1191–1253, 2003.

Andrea Rovinelli, Michael D Sangid, Henry Proudhon, Yoann Guilhem, Ricardo A Lebensohn, and Wolfgang Ludwig. Predicting the 3d fatigue crack growth rate of small cracks using multimodal data via Bayesian networks: In-situ experiments and crystal plasticity simulations. *Journal of the Mechanics and Physics of Solids*, 115:208–229, 2018.

Jonathan Scarlett and Volkan Cevher. Lower bounds on active learning for graphical model selection. In *Artificial Intelligence and Statistics*, pages 55–64. PMLR, 2017.

Ohad Shamir, Sivan Sabato, and Naftali Tishby. Learning and generalization with the information bottleneck. *Theoretical Computer Science*, 411(29-30):2696–2711, 2010.

Tomi Silander and Petri Myllymäki. A simple approach for finding the globally optimal Bayesian network structure. *arXiv preprint arXiv:1206.6875*, 2012.

Daniel Soudry, Suraj Keshri, Patrick Stinson, Min-hwan Oh, Garud Iyengar, and Liam Paninski. A shotgun sampling solution for the common input problem in neural connectivity inference. *arXiv preprint arXiv:1309.3724*, 2013.

Peter Spirtes, Clark N Glymour, Richard Scheines, David Heckerman, Christopher Meek, Gregory Cooper, and Thomas Richardson. *Causation, Prediction, and Search.* MIT press, 2000.

Chandler Squires, Sara Magliacane, Kristjan Greenewald, Dmitriy Katz, Murat Kocaoglu, and Karthikeyan Shanmugam. Active structure learning of causal DAGs via directed clique trees. *Advances in Neural Information Processing Systems*, 33, 2020.

Simon Tong and Daphne Koller. Active learning for structure in Bayesian networks. In *International Joint Conference on Artificial Intelligence*, volume 17, pages 863–869, 2001.

Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006.

Srini Turaga, Lars Buesing, Adam M Packer, Henry Dalgleish, Noah Pettit, Michael Hausser, and Jakob H Macke. Inferring neural population dynamics from multiple partial recordings of the same neural circuit. In *Advances in Neural Information Processing Systems*, pages 539–547, 2013.

Divyanshu Vats, Robert Nowak, and Richard Baraniuk. Active learning for undirected graphical model selection. In *Artificial Intelligence and Statistics*, pages 958–967, 2014.

T Verma and J Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, pages 220–227, 1990.

Thomas Verma and Judea Pearl. *Equivalence and Synthesis of Causal Models.* UCLA, Computer Science Department, 1991.

Giuseppe Vinci, Gautam Dasarathy, and Genevera I Allen. Graph quilting: graphical model selection from partially observed covariances. *arXiv preprint arXiv:1912.05573*, 2019.

Annemieke Witteveen, Gabriela F Nane, Ingrid MH Vliegen, Sabine Siesling, and Maarten J IJzerman. Comparison of logistic regression and Bayesian networks for risk prediction of breast cancer recurrence. *Medical Decision Making*, 38(7):822–833, 2018.

Selene Xu, Wesley Thompson, Sonia Ancoli-Israel, Lianqi Liu, Barton Palmer, and Loki Natarajan. Cognition, quality-of-life, and symptom clusters in breast cancer: Using Bayesian networks to elucidate complex relationships. *Psycho-oncology*, 27(3):802–809, 2018.

Appendix A. Convergence of the conditional entropy plug-in estimator

In this section, we prove Lemma 1, which gives a bound on the number of random samples required to estimate the conditional entropy of a discrete random variable. We first bound the bias of the conditional-entropy plug-in estimator. Paninski (2003) showed that the bias of the unconditional entropy estimator, for any random variable A with support size M_a, satisfies

$$-\frac{M_a - 1}{N} < \mathbb{E}[\hat{H}(A)] - \mathbb{E}[H(A)] < 0. \tag{10}$$

The following lemma derives an analog result for the conditional entropy.

Lemma 18 *Let A, B be discrete random variables with support cardinalities M_a and M_b, respectively. Let $\hat{H}(A \mid B)$ be the plug-in estimator of $H(A \mid B)$, based on N i.i.d. copies of (A, B). Then*

$$-\frac{(M_a - 1)M_b}{N} \leq \mathbb{E}[\hat{H}(A \mid B)] - H(A \mid B) \leq 0.$$

Proof Let $\mathcal{B}$ be the support of B. For any $b \in \mathcal{B}$, denote $p_b := P(B = b)$ and let N_b be the number of examples in the sample with $B = b$. We have

$$\mathbb{E}[\hat{H}(A \mid B)] = \sum_{b \in \mathcal{B}} \mathbb{E}\left[\frac{N_b}{N} \hat{H}(A \mid B = b)\right]. \tag{11}$$

Bounding a single term in the sum, we have

$$\begin{aligned}
\mathbb{E}\left[\frac{N_b}{N} \cdot \hat{H}(A \mid B = b)\right] &= \sum_{n \in [N]} \frac{n}{N} \cdot P(N_b = n) \cdot \mathbb{E}[\hat{H}(A \mid B = b) \mid N_b = n] \\
&\geq \sum_{n \in [N]} \frac{n}{N} \cdot P(N_b = n) \cdot (H(A \mid B = b) - (M_a - 1)/n), \qquad (12) \\
&= p_b \cdot H(A \mid B = b) - (M_a - 1)/N,
\end{aligned}$$

where the inequality on line (12) follows from Eq. (10), and the last equality follows since

$$\sum_{n \in [N]} \frac{n}{N} P(N_b = n) = \mathbb{E}[N_b/N] = p_b,$$

and

$$\sum_{n \in [N]} \frac{n}{N} P(N_b = n)(M_a - 1)/n = (M_a - 1)/N.$$

Plugging this into Eq. (11), we get

$$\begin{aligned}
\mathbb{E}[\hat{H}(A \mid B)] &= \sum_{b \in \mathcal{B}} \mathbb{E}\left[\frac{N_b}{N} \cdot \hat{H}(A \mid B = b)\right] \\
&\geq \sum_{b \in \mathcal{B}} (p_b \cdot H(A \mid B = b) - (M_a - 1)/N) \\
&= H(A \mid B) - (M_a - 1)M_b/N.
\end{aligned}$$

Proving that the bias is non-positive is done in a similar way. This completes the proof. ■

We now use the lemma above to prove Lemma 1.

Proof [Proof of Lemma 1] By McDiarmid's inequality (McDiarmid, 1989), if the maximal absolute change that a single random sample can induce on the plug-in estimator is θ, then

$$P(|\hat{H}(A \mid B) - \mathbb{E}[\hat{H}(A \mid B)]| > \epsilon) \leq 2\exp(-2\epsilon^2/(N\theta^2)).$$

Let $\mathcal{B}$ be the support of B. For any $b \in \mathcal{B}$, denote $p_b := P(B = b)$ and let $\hat{p}_b$ be the empirical fraction of samples with $B = b$. We have

$$\hat{H}(A \mid B) = \sum_{b \in \mathcal{B}} \hat{p}_b \cdot \hat{H}(A \mid B = b),$$

where $\mathcal{B}$ is the support of B and By Shamir et al. (2010), the maximal change that can be induced on $\hat{H}(A \mid B = b)$ when replacing an example (a, b) in the sample with an example (a', b) is $\log(\hat{p}_b N)/(\hat{p}_b N)$. Therefore, this type of replacement induces a change of at most $\log(\hat{p}_b N)/N \leq \log(N)/N$ on $\hat{H}(A \mid B)$. Now, if an example (a, b) is replaced with an example (a', b'), where $b \neq b'$, then the maximal change it induces on $\hat{H}(A \mid B)$ is $\log(M_a)/N$, since for any b, $\hat{H}(A \mid B = b) \in [0, \log(M_a)]$. Therefore, if $N \geq M_a$, both types of sample replacements induce a change of at most $\log(N)/N$ on $\hat{H}(A \mid B)$. It follows that

$$P(|\hat{H}(A \mid B) - \mathbb{E}[\hat{H}(A \mid B)]| > \epsilon) \leq 2\exp(-\frac{2N\epsilon^2}{\log^2(N)}). \tag{13}$$

If $\epsilon \geq (M_a - 1)M_b/N$, then

$$P(|\hat{H}(A \mid B) - H(A \mid B)| > 2\epsilon) \leq 2\exp(-\frac{2N\epsilon^2}{\log^2(N)}). \tag{14}$$

Lastly, we derive a lower bound for N such that the RHS is at most δ. Let $\alpha \geq 1$ and assume $N \geq e^2$. We start by showing that if $N \geq \alpha \log^2(\alpha)$, then $N/\log^2(N) \geq \alpha/4$. $N/\log^2(N)$ is monotonic increasing for $N \geq e^2$. Therefore, for $N \geq \alpha \log^2(\alpha)$,

$$\frac{N}{\log^2(N)} \geq \frac{\alpha \log^2(\alpha)}{\log^2(\alpha \log^2(\alpha))} = \frac{\alpha \log^2(\alpha)}{(\log(\alpha) + \log(\log^2(\alpha)))^2}$$

Since $\alpha \geq 1$, we have $\log^2(\alpha) \leq \alpha$. It follows that $\log(\log^2(\alpha)) \leq \log(\alpha)$. Therefore,

$$(\log(\alpha) + \log(\log^2(\alpha)))^2 \leq (\log(\alpha) + \log(\alpha))^2 = 4\log^2(\alpha).$$

We conclude that for $N \geq \alpha \log^2(\alpha)$, $N/\log^2(N) \geq \alpha/4$. Setting $\alpha := 2\log(2/\delta)/\epsilon^2$, we obtain that if $N \geq \alpha \log^2 \alpha$, then $N/\log^2(N) \geq \log(2/\delta)/(2\epsilon^2)$, which implies that the RHS of Eq. (14) is at most δ. ■

Appendix B. Proofs for technical lemmas in Section 8

In this section, we provide the proofs of technical lemmas stated in Section 8.
Proof [of Lemma 14] To prove Eq. (7), note that

$$\begin{aligned} H(X_a, X_b \mid \Pi) &= H(X_a, X_b, \Pi) - H(\Pi) = H(X_b \mid X_a, \Pi) + H(X_a, \Pi) - H(\Pi) \\ &= H(X_b \mid X_a) + H(X_a \mid \Pi). \end{aligned}$$

The last equality follows since by assumption (IV), $\Pi \to X_a \to X_b$ is a Markov chain, so that $H(X_b \mid X_a, \Pi) = H(X_b \mid X_a)$. From the above it we have

$$H(X_a \mid \Pi) = H(X_a, X_b \mid \Pi) - H(X_b \mid X_a).$$

It follows that

$$\begin{aligned} |H(X_b \mid \Pi) - H(X_a \mid \Pi)| &= |H(X_b \mid \Pi) - H(X_a, X_b \mid \Pi) + H(X_b \mid X_a)| \\ &= |H(X_b, \Pi) - H(X_a, X_b, \Pi) + H(X_b \mid X_a)| \\ &= |H(X_b \mid X_a) - H(X_a \mid X_b, \Pi)| \\ &\leq \max(H(X_b \mid X_a), H(X_a \mid X_b, \Pi)) \\ &\leq \max(H(X_b \mid X_a), H(X_a \mid X_b)) \leq \lambda. \end{aligned}$$

The last inequality follows from the definition of λ. This proves Eq. (7).

To prove Eq. (8), note that

$$\begin{aligned} H(Y, X_b \mid \Pi, X_a) &= H(Y, X_b, \Pi, X_a) - H(\Pi, X_a) \\ &= H(X_b \mid \Pi, Y, X_a) + H(\Pi, Y, X_a) - H(\Pi, X_a) \\ &= H(X_b \mid X_a) + H(Y \mid \Pi, X_a), \end{aligned}$$

where the last equality follows since by assumption (IV), $(Y, \Pi) \to X_a \to X_b$ is a Markov chain, so that $H(X_b \mid \Pi, Y, X_a) = H(X_b \mid X_a)$. The equality above implies that

$$H(Y \mid \Pi, X_a) = H(Y, X_b \mid \Pi, X_a) - H(X_b \mid X_a).$$

It follows that

$$\begin{aligned} &|H(Y \mid \Pi, X_b) - H(Y \mid \Pi, X_a)| \\ &= |H(Y, \Pi, X_b) - H(\Pi, X_b) - H(Y, X_b \mid \Pi, X_a) + H(X_b \mid X_a)| \\ &= |H(Y, \Pi, X_b) - H(\Pi, X_b) - H(Y, X_b, \Pi, X_a) + H(\Pi, X_a) + H(X_b \mid X_a)| \\ &= | - H(X_a \mid Y, \Pi, X_b) - H(X_b \mid \Pi) + H(X_a \mid \Pi) + H(X_b \mid X_a)| \\ &\leq H(X_a \mid Y, \Pi, X_b) + |H(X_b \mid \Pi) - H(X_a \mid \Pi)| + H(X_b \mid X_a) \\ &\leq H(X_a \mid X_b) + |H(X_b \mid \Pi) - H(X_a \mid \Pi)| + H(X_b \mid X_a) \leq 3\lambda. \end{aligned}$$

The last inequality follows from Eq. (7) and the definition of λ. This proves Eq. (8). ■

Proof [of Lemma 15] If X_b has no children in G, then set $\tilde{G} := G$. It is easy to see that in this case, $\tilde{G}$ satisfies all the required properties.

Otherwise, denote the set of children of X_b in G by W. Denote by Π_b, Π_a the parent sets of X_b, X_a in G, respectively, and by Π_w the parents of $X_w \in W$. Denote by $f_v := \langle X_v, \Pi_v \rangle \in G$ the family of X_v in G. We construct $\tilde{G}$ from G by replacing the families f_v, for X_a, X_b and all $X_w \in W$, by new families $\tilde{f}_v := \langle X_v, \tilde{\Pi}_v \rangle$, as follows. First, define $\tilde{\Pi}_a$ and $\tilde{\Pi}_b$:

1. If X_b has no directed path to X_a, define $\tilde{\Pi}_a := \Pi_a$ and $\tilde{\Pi}_b := \Pi_b$.

2. Otherwise (X_b has a directed path to X_a), if X_b is a parent of X_a, make X_a a parent of X_b instead. In any case, switch between the other parents of X_a and X_b. Formally, set $\tilde{\Pi}_a = \Pi_b$. If $X_b \in \Pi_a$, $\tilde{\Pi}_b := \Pi_a \setminus \{X_b\} \cup \{X_a\}$. Otherwise, $\tilde{\Pi}_b := \Pi_a$.

Next, for each $w \in W \setminus \{X_a\}$, make X_a a parent of w instead of X_b. Formally, $\tilde{\Pi}_w := \Pi_w \setminus \{X_b\} \cup \{X_a\}$. This completes the definition of $\tilde{G}$.

It is easy to see that the maximal in-degree of $\tilde{G}$ is at most that of G, which is at most k, as required. We now show that $\tilde{G}$ is a DAG, thus proving that $\tilde{G} \in \mathcal{G}_{k,d}$.

Claim: $\tilde{G}$ is a DAG.
Proof of claim: Consider first the case where X_b does not have a directed path to X_a. Let (A, B) be an edge in $\tilde{G}$ that is not in G. Then $B \in W$ and $A = X_a$. In G, there is no directed path from B to X_a, since B is a child of X_b and this would create a directed path from X_b to X_a. But all paths from B in $\tilde{G}$ are the same as in G. Thus, there is no cycle in $\tilde{G}$.

Now, consider the case where X_b has a directed path to X_a. Let (A, B) be an edge in $\tilde{G}$ that is not in G. We will show that it does not create a cycle in $\tilde{G}$. Divide into cases:

1. If $B = X_b$, then (A, B) is not an edge in a cycle in $\tilde{G}$, since X_b has no children in $\tilde{G}$.

2. If $B \in W$, then $A = X_a$, since X_a is the only parent added to W. Thus, (A, B) is part of a cycle in $\tilde{G}$ only if B is an ancestor of X_a in $\tilde{G}$. Assume in contradiction that this is the case, and let $v_1, \ldots, v_l$ be the nodes in a shortest directed path from B to X_a. v_l is a parent of X_a in $\tilde{G}$, thus it is a parent of X_b in G. The nodes $v_1, \ldots, v_l$ cannot be X_a or X_b (because it has no children in $\tilde{G}$), thus their parents are the same in G and $\tilde{G}$. It follows that the directed path $B, v_1, \ldots, v_l, X_b$ exists in G. However, B is a child of X_b in G, thus this implies a cycle in G, a contradiction. Therefore, in this case (A, B) is not part of a cycle.

3. If $B = X_a$, then $A \in \Pi_b$. Thus, (A, B) is part of a cycle if there is a directed path from X_a to A in $\tilde{G}$. Assume in contradiction that this is the case, and let $v_1, \ldots, v_l$ be the nodes in a shortest directed path from X_a to A in $\tilde{G}$. The nodes $v_2, \ldots, v_l$ and their parents cannot be X_a or X_b, thus their parents are the same in G and $\tilde{G}$. It follows that the directed path $v_1, v_2, \ldots, v_l, A$ exists in G. Now, if v_1 is a child of X_b in G, this means that there is a directed path in G from X_b to its parent A, which means that there is a cycle in G. If v_1 is a child of X_a in G, then, since X_b has a directed path to X_a in G, this again means that there is a path from X_b to A in G, leading to a cycle in G. In both cases, this contradicts the fact that G is a DAG. Thus, in this case as well, (A, B) is not part of a cycle.

Therefore, in this case as well, $\tilde{G}$ is a DAG. This completes the proof of the claim. □

To show that $\tilde{G}$ satisfies the properties stated in the lemma, observe first that properties (2) and (3) are immediate from the construction of $\tilde{G}$. Property (1) requires a bound on the difference in scores of G and $\tilde{G}$. We have

$$|\mathcal{S}(G) - \mathcal{S}(\tilde{G})| \leq \sum_{w \in W \setminus \{X_a\}} |H(f_w) - H(\tilde{f}_w)| + |H(f_a) - H(\tilde{f}_a) + H(f_b) - H(\tilde{f}_b)|.$$

To bound the first term, let $\Pi'_b := \Pi_w \setminus \{X_b\}$. Then

$$|H(f_w) - H(\tilde{f}_w)| = |H(w \mid \Pi'_b, X_b) - H(w \mid \Pi'_b, X_a)| \leq 3\lambda,$$

where the last inequality follows from Eq. (8). Since there are at most $d-2$ such terms, the first term is at most $3(d-2)\lambda$.

To bound the second term, first note that it is equal to zero if X_b has no directed path to X_a in G. If X_b does have a directed path to X_a in G, then let $\Pi'_a := \Pi_a \setminus \{X_b\}$. If X_b is not a direct parent of X_a, then

$$\begin{aligned}&|H(f_b) - H(\tilde{f}_b) + H(f_a) - H(\tilde{f}_a)|\\ &= |H(X_a \mid \Pi'_a) + H(X_b \mid \Pi_b) - H(X_a \mid \Pi_b) - H(X_b \mid \Pi'_a)| =: \Delta.\end{aligned}$$

If X_b is a direct parent of X_a, then

$$\begin{aligned}&|H(f_a) - H(\tilde{f}_a) + H(f_b) - H(\tilde{f}_b)|\\ &= |H(X_a \mid \Pi'_a, X_b) + H(X_b \mid \Pi_b) - H(X_a \mid \Pi_b) - H(X_b \mid \Pi'_a, X_a)|\\ &= |H(X_a \mid \Pi'_a, X_b) + H(X_b \mid \Pi_b) - H(X_a \mid \Pi_b) - H(X_b \mid \Pi'_a, X_a)|\\ &= |H(X_b \mid \Pi_b) - H(X_a \mid \Pi_b) + H(X_a, \Pi'_a) - H(X_b, \Pi'_a)| = \Delta.\end{aligned}$$

Now, Δ can be bounded as follows:

$$\Delta \leq |H(X_b \mid \Pi_b) - H(X_a \mid \Pi_b)| + |H(X_a \mid \Pi'_a) - H(X_b \mid \Pi'_a)| \leq 2\lambda,$$

where the last inequality follows from Eq. (7). Thus, the second term is upper bounded by 2λ in all cases. Property (1) is proved by summing the bounds of both terms.

■

Lastly, we prove Theorem 13.

Proof [of Theorem 13] Set $\lambda < \epsilon/2$. We prove that there exists a non-optimal EC with a score at least $\mathcal{S}^* - 2\lambda$. Let $G^* \in \mathcal{G}_{d,k}$ be an optimal structure for $\mathcal{D}_2(\lambda)$. By Theorem 12, $\mathcal{D}_2(\lambda)$ is $(\beta - 3d\lambda, \mathbf{X}_1 \setminus \{X_a\})$-stable. Therefore, by Lemma 8, the set of families of $\bar{V} \equiv \{X_b, X_a\}$ in G^* is an optimal legal family set. Denote this set $F := \{f_a, f_b\}$. First, we prove that X_a and X_b must be in a parent-child relationship.

Claim: In F, either X_a is a parent of X_b, or vice versa.
Proof of claim: Assume for contradiction that the claim does not hold. We define a new legal family set F' for $\bar{V}$ and show that its score is larger than the score of F, thus contradicting the optimality of F. Denote the parent set of X_a in f_a by $\bar{\Pi}_a$. Define a new

family $f'_a := \langle X_a, \Pi'_a \rangle$, where Π'_a is obtained by adding X_b as a parent to $\bar{\Pi}_a$. In addition, if this increases the number of parents of X_a over k, then one of the other elements in $\bar{\Pi}_a$ is removed in Π'_a. Let $F' = \{f'_a, f_b\}$. According to Property (II) in the definition of $\mathcal{D}_2$, X_a has no children in $\mathbf{X}_1 \setminus \{X_a\}$, therefore, X_a has no children in G^*, and so adding the edge (X_a, X_b) cannot create a cycle. Therefore, F' is a legal family set. We have

$$\begin{aligned}\mathcal{S}(F') - \mathcal{S}(F) &= \mathcal{S}(\{\langle X_a, \Pi'_a \rangle\}) - \mathcal{S}(\{\langle X_a, \Pi_a \rangle\}) = H(X_a \mid \Pi_a) - H(X_a \mid \Pi'_a) \\ &\geq H(X_a \mid \mathbf{X}_1 \setminus \{X_a\}) - H(X_a \mid X_b) \geq \alpha - \lambda,\end{aligned}$$

where the last inequality follows from assumptions (III) and (V). Now, since by definition, $\lambda < \alpha$, it follows $\mathcal{S}(F') > \mathcal{S}(F)$, which contradicts the optimality of F. This completes the proof of the claim. □

Now, define a new graph as follows: Let $\tilde{F}$ a family set which is the same as F, except that the roles of X_b and X_a are swapped. Define $\tilde{G} := G^* \setminus F \cup \tilde{F}$. Notice that by Theorem 12, $\mathcal{D}_2$ is (γ, V)-stable for $V = \mathbf{X}_1 \setminus \{X_a\}$. Then, by Property 1 in Def. 4, both X_a and X_b have no children in V. This means that in G^*, one of X_a and X_b has the other as a child, and the other has no children. Thus, swapping them cannot create a cycle, implying that $\tilde{G} \in \mathcal{G}_{d,k}$ and that $\tilde{F}$ is a legal family set.

We now show that $\mathcal{S}^* - 2\lambda \leq \mathcal{S}(\tilde{G}) < \mathcal{S}^*$, thus proving the theorem. Since we have

$$\mathcal{S}^* - \mathcal{S}(\tilde{G}) = \mathcal{S}(F) - \mathcal{S}(\tilde{F}),$$

it suffices to show that $0 < \mathcal{S}(F) - \mathcal{S}(\tilde{F}) \leq 2\lambda$.

Let $\bar{\Pi}_a$, $\bar{\Pi}_b$ be the parent sets of X_a and X_b in F, respectively. Let $\Pi_a := \bar{\Pi}_a \setminus \{X_b\}$ and $\Pi_b := \bar{\Pi}_b \setminus \{X_a\}$. If X_a is a parent of X_b in F, then

$$\mathcal{S}(F) - \mathcal{S}(\tilde{F}) = H(X_a \mid \Pi_b, X_b) + H(X_b \mid \Pi_a) - H(X_a \mid \Pi_a) - H(X_b \mid \Pi_b, X_a).$$

Since $H(X_b \mid \Pi_b, X_a) = H(X_b, \Pi_b, X_a) - H(X_a \mid \Pi_b) - H(\Pi_b)$, and symmetrically for $H(X_a \mid \Pi_b, X_b)$, it follows that

$$H(X_a \mid \Pi_b, X_b) - H(X_b \mid \Pi_b, X_a) = H(X_a \mid \Pi_b) - H(X_b \mid \Pi_b).$$

Therefore,

$$\mathcal{S}(F) - \mathcal{S}(\tilde{F}) = H(X_a \mid \Pi_b) - H(X_b \mid \Pi_b) + H(X_b \mid \Pi_a) - H(X_a \mid \Pi_a). \tag{15}$$

By a symmetric argument, Eq. (15) holds also if X_b is a parent of X_a in F. Combining Eq. (15) with Eq. (7) in Lemma 14, we have that $|\mathcal{S}(F) - \mathcal{S}(\tilde{F})| \leq 2\lambda$. We have left to show that $\mathcal{S}(F) > \mathcal{S}(\tilde{F})$, which will complete the proof of the theorem.

We have, for any set $\Pi \subseteq \mathbf{X}_1$,

$$H(X_b, X_a \mid \Pi) = H(X_b \mid X_a, \Pi) + H(X_a \mid \Pi) = H(X_b \mid X_a) + H(X_a \mid \Pi),$$

where the last inequality follows since by assumption (IV), $\Pi \to X_a \to X_b$ is a Markov chain. Therefore,

$$H(X_b \mid \Pi) \equiv H(X_b, X_a \mid \Pi) - H(X_a \mid X_b, \Pi) = H(X_b \mid X_a) + H(X_a \mid \Pi) - H(X_a \mid X_b, \Pi).$$

Using the above for Π_a and Π_b, we get

$$H(X_b \mid \Pi_a) - H(X_b \mid \Pi_b) = H(X_a \mid \Pi_a) - H(X_a \mid \Pi_b) + H(X_a \mid X_b, \Pi_b) - H(X_a \mid X_b, \Pi_a).$$

Combined with Eq. (15), it follows that

$$\mathcal{S}(F) - \mathcal{S}(\tilde{F}) = H(X_a \mid X_b, \Pi_b) - H(X_a \mid X_b, \Pi_a). \tag{16}$$

Now, in F, one of X_b, X_a is the parent of the other, and in $\tilde{F}$, the other case holds. We show that in both cases, $\mathcal{S}(F) - \mathcal{S}(\tilde{F}) > 0$.

1. Suppose that X_a is the parent of X_b. In this case, we can assume without loss of generality that $\Pi_b = \emptyset$ in F, since $H(X_b \mid X_a) = H(X_b \mid X_a, \Pi_b)$ for any set $\Pi_b \subseteq \mathbf{X}_1$, due to assumption (IV). Therefore, from Eq. (16),

$$\begin{aligned}\mathcal{S}(F) - \mathcal{S}(\tilde{F}) &= H(X_a \mid X_b) - H(X_a \mid X_b, \Pi_a)\\ &= H(X_a, X_b) - H(X_b) - H(X_a, X_b, \Pi_a) + H(X_b, \Pi_a)\\ &= H(\Pi_a \mid X_b) - H(\Pi_a \mid X_a, X_b) = H(\Pi_a \mid X_b) - H(\Pi_a \mid X_a).\end{aligned}$$

   The last inequality follows since $X_b \to X_a \to \Pi_a$ is a Markov chain by assumption (IV). Now, by assumption (IV), we have

$$\begin{aligned}&H(\Pi_a \mid X_b) \geq H(\Pi_a \mid X_b, C) &(17)\\ &= H(\Pi_a \mid X_a, C = 1)\mathbb{P}[C = 1] + H(\Pi_a \mid C = 0)\mathbb{P}[C = 0]\\ &= H(\Pi_a \mid X_a)\mathbb{P}[C = 1] + H(\Pi_a)\mathbb{P}[C = 0]. &(18)\end{aligned}$$

   In addition, $H(\Pi_a \mid X_a) < H(\Pi_a)$. This is because Π_a is an optimal parent set for X_a from $\mathbf{X}_1$, and by assumption (I) on $\mathcal{D}_1$, it is unique. Therefore, $H(X_a \mid \Pi_a) > H(X_a)$, which implies $H(\Pi_a \mid X_a) < H(\Pi_a)$. Combined with Eq. (18), and since $\mathbb{P}[C = 0] > 0$ by assumption (IV), we have $H(\Pi_a \mid X_b) > H(\Pi_a \mid X_a)$. It follows that $\mathcal{S}(F) - \mathcal{S}(\tilde{F}) > 0$, as required.

2. Suppose that X_b is a parent of X_a in F. Assume for contradiction that $\mathcal{S}(F) = \mathcal{S}(\tilde{F})$. Then $\tilde{F}$ is an optimal legal family set in which X_a is a parent of X_b. Therefore, by the first case above, $\mathcal{S}(\tilde{F}) > \mathcal{S}(F)$, in contradiction to the optimality of F.

It follows that $\mathcal{S}(F) > \mathcal{S}(\tilde{F})$, which proves the claim. Thus, $\tilde{G}$ is not an optimal graph, as required. This completes the proof of the theorem. ■