Learning and Testing Latent-Tree Ising Models Efficiently

Yuval Dagan DAGAN@CSAIL.MIT.EDU
Constantinos Daskalakis COSTIS@CSAIL.MIT.EDU
Anthimos-Vardis Kandiros KANDIROS@MIT.EDU
Department of Electrical Engineering and Computer Science, MIT

Davin Choo DAVIN@U.NUS.EDU
School of Computing, National University of Singapore

Editors: Gergely Neu and Lorenzo Rosasco

Abstract

We provide time- and sample-efficient algorithms for learning and testing latent-tree Ising models, i.e. Ising models that may only be observed at their leaf nodes. On the learning side, we obtain efficient algorithms for learning a tree-structured Ising model whose leaf node distribution is close in total variation distance, improving on the results of Cryan et al. (2001). On the testing side, we provide an efficient algorithm with fewer samples for testing whether two latent-tree Ising models have leaf-node distributions that are close or far in total variation distance. We obtain our algorithms by showing novel localization results for the total variation distance between the leaf-node distributions of tree-structured Ising models, in terms of their marginals on pairs of leaves.

Keywords: Probabilistic graphical models, distribution learning/testing, learning from complex or structured data (e.g. networks, time series), learning with algebraic or combinatorial structure

1. Introduction

Statistical estimation and hypothesis testing challenges involving high-dimensional distributions are central in Statistics, Machine Learning and various other theoretical and applied fields. Core to this challenge is the fact that even the most basic of those challenges, such as uniformity testing, require exponential sample sizes in the dimension to solve if no structural or parametric assumptions are placed on the underlying distributions; see e.g. Daskalakis and Pan (2017); Canonne et al. (2017); Acharya et al. (2018); Daskalakis et al. (2019) for discussions of this point and further references.

The aforementioned exponential sample-size barriers motivate the study of models that sidestep those requirements, e.g. models encapsulating conditional independence structure in the distribution, such as Markov Random Fields (MRFs) and Bayesian networks (Bayes nets). In turn, a vast line of research has studied statistical inference questions involving MRFs and Bayes nets and their applications; see e.g. Pearl (1988); Lauritzen (1996); Jordan (2004); Koller and Friedman (2009) for an introduction to graphical models, their uses, and associated inference algorithms, and see e.g. Chow and Liu (1968); Chow and Wagner (1973); Narasimhan and Bilmes (2004); Ravikumar et al. (2010); Tan et al. (2011); Jalali et al. (2011); Santhanam and Wainwright (2012); Bresler (2015); Vuffray et al. (2016); Klivans and Meka (2017); Hamilton et al. (2017); Dagan et al. (2021); Kandiros et al. (2021); Daskalakis and Pan (2021); Bhattacharyya et al. (2021); Vuffray et al. (2022) and the references in the previous paragraph for some classical work and some recent theoretical progress on learning and testing graphical models as well as other types of statistical inference with them.

Despite the vast study of graphical models, and a recent burst of activity towards computationally and statistically efficient algorithms for inference with them, a broad outstanding challenge in this

space lies in computationally efficient inference with graphical models that have *latent variables*, variables whose realized values we do not have direct observations of. Those are widely motivated in practice (see e.g. Aigner et al. (1984); Bishop (1998); Everett (2013); Bartholomew et al. (2011); Felsenstein (1973)) but inference with them is known to be computationally intractable in general. For example, learning graphical models with latent variables in total variation distance is known to be intractable, even when the underlying graph is a tree (Mossel and Roch, 2005), while in the absence of latent nodes the same problem is computationally tractable, owing to classical work of Chow and Liu (1968) and its recent analysis (Daskalakis and Pan, 2021; Bhattacharyya et al., 2021). Similarly, computing the likelihood of a tree-structured graphical model is tractable in the absence of latent nodes, but becomes intractable in the presence of latent nodes (Chor and Tuller, 2005; Roch, 2006). These computational challenges become more daunting when the underlying graph gets cyclic, and the overall difficulty of handling latent variables has motivated the development of an array of widely-used approximate inference methods, such as the expectation-maximization algorithm of Dempster et al. (1977) and variational inference (see e.g. Blei et al. (2017) for a survey).

A main goal of this work is to advance the frontier of computationally efficient learning and testing of graphical models with latent variables. As learning general tree MRFs over general alphabets is hard (Mossel and Roch, 2005), we focus on the binary-alphabet tree-structured Ising models, which have found extensive use in phylogenetics (Felsenstein, 1973).

We focus on the two following inference goals in this work:

1. **(Proper Learning):** Given sample access to the distribution at the leaves of a tree-structured Ising model $\mathcal{P}$, we want to learn a tree-structured Ising model $\mathcal{Q}$ whose leaf-node distribution is ε-close it total variation distance to that of $\mathcal{P}$.

2. **(Identity Testing):** Given sample access to the distribution at the leaves of two tree-structured Ising models $\mathcal{P}$ and $\mathcal{Q}$ with the same leaf set, we want to distinguish whether the leaf-node distributions of $\mathcal{P}$ and $\mathcal{Q}$ are equal or at least ε-far in total variation distance.

We provide computationally and statistically efficient algorithms for both Goals 1 and 2. Our contribution to Goal 1 is an algorithm whose time- and sample- complexity provide substantial improvements compared to the algorithm by Cryan et al. (2001) as well as the algorithm by Mossel and Roch (2005), which requires restricting the correlations across the edges of the Ising model. We also improve upon work in the phylogenetic literature (see Section 2) which has focused on identifying the latent tree-structure of the model but also requires restrictions on the Ising model to achieve this. Finally, we remark that the computational intractability results of Mossel and Roch (2005) for learning tree-structured graphical models with latent variables do not apply to Goal 1 because we are working with binary-alphabet models.

On the technical front, a fruitful approach towards statistical inference with graphical models uses the paradigm of *localization*, whose goal is to localize the difference between two graphical models to differences of their marginals involving a small number of variables. Such localization properties can be used to distinguish between models for the purposes of hypothesis testing, or be exploited to learn graphical models or perform hypothesis selection. Localization of the KL divergence between two Bayesian networks with the same DAG follows directly from their shared factorization, which implies that the KL divergence between the two Bayes nets is upper bounded by the sum of the KL divergences of their marginals on different neighborhoods of the graph, involving a node and its parents. Similar subadditivity results have been established for total variation and

squared Hellinger distance (Daskalakis and Pan, 2017) as well as for other distances, for MRFs, and for causal models (Acharya et al., 2018; Daskalakis et al., 2019; Ding et al., 2021). Localization results can also be used for comparing graphical models on *different* graphs, as long as the underlying graphs are trees (Daskalakis and Pan, 2017). In turn, the aforementioned localization results have been exploited to show that the celebrated Chow and Liu (1968) algorithm learns tree-structured Ising models with optimal sample complexity (Daskalakis and Pan, 2021) and to obtain optimal algorithms for testing Bayesian networks (Daskalakis and Pan, 2017). Additionally, localization properties of graphical models are implicit in much of the recent burst of activity on learning graphical models referenced earlier in this section and often have applications beyond the actual problem studied by the papers introducing them; for example, a recent work of Bhattacharyya et al. (2022) on independence testing of bounded degree Bayes nets in total variation distance crucially relies on the subadditivity localization result about Hellinger distance by Daskalakis and Pan (2017).

In the presence of latent variables, Bresler and Karzand (2020) gave localization bounds for Ising models with zero external fields, the model which we are studying in this paper. However, as their bound is exponential in the number of variables, we cannot directly apply it to obtain efficient algorithms. As such, another important goal of our work is the following:

3. **(Localization of TV in latent-tree Ising models):** Given two tree-structured Ising models, which have the same leaf set but potentially different underlying trees, upper bound the total variation distance between the leaf-node distributions in terms of the marginals of these models on pairs of leaves. Furthermore, the bound should be *polynomial* in the number of vertices.

In the fully observable case, the localization of distance results that are known for tree-structured graphical models exploit factorization properties of distributions defined on trees and combinatorial results that allow writing two tree-structured graphical models under a common factorization (Daskalakis and Pan, 2017). Meanwhile, a key challenge arising from the presence of latent variables is that the leaf-node distributions result from marginalizing out all non-leaf vertices in the tree-structured models and thus cease to have any tree-structured factorization that we may exploit.

1.1. Results

Let us first formally define tree-structured Ising models.

Tree-structured Ising models. One is given some undirected tree $T = (V, E)$ whose leaves are labelled from 1 to n and whose internal nodes are labelled from $n+1$ to $n+n'$. Without loss of generality, we will assume that all non-leaf nodes have degree 3[1]. For each edge $\{i, j\} \in E$, there is an associated weight $\theta_{ij} \in [-1, 1]$; as T is undirected, we have $\theta_{ij} = \theta_{ji}$. Each node i is assigned a *spin* $x_i \in \{-1, 1\}$, and the probability of a spin-configuration is defined as

$$\Pr\left[x_1, \ldots, x_{n+n'}\right] \propto \prod_{\{i,j\} \in E} \frac{1 + \theta_{ij} x_i x_j}{2} \ .$$

1. Every tree can be converted into one with all non-leaf nodes having degree 3, without affecting the leaf distribution. We just contract every path that consists of nodes of degree 2 into a single edge and split nodes with degree larger than 3 by introducing edges with $\theta = 1$. For more details, see Appendix A.

Note that this definition allows for any tree-structured Ising model with zero external fields[2]. A sample can be obtained by rooting the tree at an arbitrary internal node, drawing a uniform random value for the spin of the root, and randomly propagate the values of the spins along the tree as follows: for any directed edge $i \to j$ such that the spin x_i was already set, we set the spin x_j such that $\Pr[x_j = x_i] = (1 + \theta_{ij})/2$ and $\Pr[x_j = -x_i] = (1 - \theta_{ij})/2$. This process has been used as a model in a variety of applications.

Our first result provides upper bounds on the total variation (TV) distance between any two tree-structures Ising models with latents. Note that in the case of a fully observable tree, these properties can be proven by using the product factorization of the probability distribution over the edges of the tree but such an approach fails in the presence of latents. Instead, we rely on the pairwise-marginals between the leaves, which are readily accessible. In the specific case of an Ising model, for any two nodes i, j of the tree, the marginal distribution of (x_i, x_j) is characterized by

$$\alpha_{ij} := \mathbb{E}[x_i x_j] = \prod_{\{k,l\} \in P_{ij}} \theta_{kl} \ , \tag{1}$$

where P_{ij} is the unique path that connects i and j on the tree. Equation (1), which we call the *multiplication over paths property*, states that to calculate the correlation between two leaves, it suffices to multiply the correlations along all the edges on the path that connects them.

It is well known (see e.g. Section 6.1 in Steel (2016)) that the correlations α_{ij} between all pairs of *leaves* uniquely identifies the underlying tree and edge weights of the distribution. We now provide a simple example to illustrate this fact and refer to Chapter 6 of Steel (2016) for more details. Without loss of generality, for the following example we will assume that all edge correlations θ_{ij} are non-zero. Indeed, if we allow some θ_{ij} to be 0, then the leaves can be uniquely partitioned into a maximal-size collection of subsets that are independent from each other and the below argument can be applied to each subset separately.

Now, consider a model with four leaves, which is also called a *quartet*. There are three possible topologies for a tree with 4 leaves, which are depicted in Figure 2(a). Suppose the true tree has the topology of the first graph in Figure 2(a), where leaves 1,2 and 3,4 are separated by an edge with length θ. Now, suppose we know all the covariances α_{ij} exactly. Let $\theta \in [-1, 1]$ be the weight of the middle edge. Since covariances multiply along paths we have

$$|\alpha_{13}\alpha_{24}| = |\alpha_{14}\alpha_{23}| = \theta^2 |\alpha_{12}\alpha_{34}| \leq |\alpha_{12}\alpha_{34}|$$

Thus, by comparing the three products $|\alpha_{13}\alpha_{24}|, |\alpha_{14}\alpha_{23}|, |\alpha_{12}\alpha_{34}|$ and picking the highest, we can distinguish which of the three topologies of Figure 2(a) is the correct one. If all products are the same then the middle edge is contracted and the topology is a star. For general trees, for two leaves i, j, consider all supersets of four leaves i, j, k, l and suppose we conduct for each one this quartet test. If i, j are never on different sides of the quartet test, then they form a *cherry* (children of the same parent). We can continue inductively to recover the topology of the tree. Regarding the lengths of the edges, notice that in our example, by the above calculation we have

$$\theta^2 = \frac{|\alpha_{13}\alpha_{24}|}{|\alpha_{12}\alpha_{34}|}.$$

2. A more common expression is $\Pr[x_1, \ldots, x_{n+n'}] \propto \exp(\sum_{(i,j)\in E} \beta_{i,j} x_i x_j)/Z$, which can be translated to our setting by substituting $\theta_{ij} = \mathbb{E}[x_i x_j] = (e^{\beta_{ij}} - e^{-\beta_{ij}})/(e^{\beta_{ij}} + e^{-\beta_{ij}})$.

Using this observation, we can identify the lengths of all the edges of the topology. The identifiability of the topology and its edge lengths using covariances is also discussed after Equation 6.1 in Steel (2016). To summarize, if we know the pairwise distances exactly, we could recover the correct topology and weights.

Given m samples from the leaves, these pairwise-marginals can be easily approximately estimated from data samples. It is then clear by the previous discussion that if we know the pairwise correlations up to some good accuracy, we should be able to estimate the distribution of leaves accurately as well, in some metric. Hence, a natural analogue of the marginal distribution of edges in the fully observed tree would be the marginal distribution of all pairs of leaves in the latent-tree. From this perspective, we provide a bound on the total variation of two leaf distributions based solely on their pairwise correlations which can be viewed as an approximate tensorization property for total variation in latent tree Ising models. Our results come in two settings: when both distributions share the same underlying tree structure, and when they do not.

Theorem 1 *Let μ and μ^* be distributions over the leaves of two tree-structured Ising models with n leaves. Suppose $|\mathbb{E}_{\mu}[x_i x_j] - \mathbb{E}_{\mu^*}[x_i x_j]| \leq \varepsilon$ for all pairs of leaves i, j.*

- ***Same topology:*** *If μ and μ^* are defined on the same graph, then $\mathrm{TV}(\mu, \mu^*) \leq 2n^2\varepsilon$.*
- ***Different topologies:*** *If μ and μ^* are defined on two different trees T and T^* such that the minimum diameter of T and T^* is D, then $\mathrm{TV}(\mu, \mu^*) \leq O(Dn^5\varepsilon)$, where O hides absolute constants.*

The previous bound by Bresler and Karzand (2020), which holds for both same and different topologies, was $\mathrm{TV}(\mu, \mu^*) \leq n2^n\varepsilon$; though their setting was slightly more general, as it applied to arbitrary subsets of the nodes of an Ising model, rather than only sets of leaves. We conjecture that the techniques presented here could be useful for obtaining a more general theorem, similar to Proposition 1 in Appendix H of Bresler and Karzand (2020). Such a general result would improve the bounds for learning in k-local-TV (Bresler and Karzand, 2020; Boix-Adsera et al., 2022) from being exponential in k to polynomial (see Section 2 for more details on this line of work). We believe this is a very fruitful direction for future work.

We make the following observations regarding the tightness of Theorem 1. For a fixed topology, the following example shows that the bound of Theorem 1 is off by a factor of at most n. Consider two tree models (I) and (II) with the same topology of a star, with a single latent node u that is connected to n leaves $1, \ldots, n$. In (I), we have $\theta_{ui} = 1$ and in (II) we have $\theta_{ui} = 1 - \varepsilon < 1$, for all $i \in [n]$. Clearly, in (I) it holds that $\alpha_{ij} = 1$ and in (II) that $\alpha_{ij} = (1-\varepsilon)^2$, for all $i, j \in [n]$. We observe that the left hand side of the bound is

$$\frac{1}{2}\left(1 - \left(1 - \frac{\varepsilon}{2}\right)^n - \left(\frac{\varepsilon}{2}\right)^n\right) = \Theta(n\varepsilon)$$

if ε is sufficiently small, while the right hand side is $\Theta(n^2\varepsilon)$. For unknown topology, we do not have a better lower bound.

We believe that Theorem 1 is a result of independent interest beyond the scope of this paper as it can be applied in a variety of applications. For instance, one can obtain a polynomial time algorithm for identity testing of latent tree Ising models by directly combining Theorem 1 with standard ideas from testing. Note that we have diameter $D \in O(\log n)$ in many applications of interest. For example, the phylogenetic trees produced by the recent gene editing technologies are balanced (Jones et al., 2020).

Corollary 2 *Let P, Q be leaf distributions of two potentially different tree Ising models. Suppose we are given access to samples from P and we wish to distinguish whether $P = Q$ or $TV(P, Q) > \varepsilon$. Assume also that the minimum diameter of the two trees is D. Then, there exists a polynomial time algorithm that answers correctly with probability at least $1 - \delta$, with sample size $O\left(\frac{n^{10} D^2 \log \frac{n}{\delta}}{\varepsilon^2}\right)$.*

To see why this is true, let α^P, α^Q denote the correlations for P, Q respectively. Notice that by the Chernoff bound and a union bound, if the number of samples is $O(n^{10} D^2 \log(n/\delta)/\varepsilon^2)$, then with probability $1 - \delta$ the empirical correlations $\hat{\alpha}$ satisfy $|\alpha^P_{ij} - \hat{\alpha_{ij}}| \leq \varepsilon/(2Dn^5)$ for all i, j. Now, suppose $TV(P, Q) > \varepsilon$. Applying Theorem 1 then gives that there should exist a pair i, j with $|\alpha^P_{ij} - \alpha^Q_{ij}| > \varepsilon/(Dn^5)$ This implies that $|\hat{\alpha_{ij}} - \alpha^Q_{ij}| > \varepsilon/(2Dn^5)$, which the tester can detect. If $P = Q$, then $|\hat{\alpha_{ij}} - \alpha^P_{ij}| = |\hat{\alpha_{ij}} - \alpha^Q_{ij}| \leq \varepsilon/(2Dn^5)$ for all i, j.

To further show the utility of Theorem 1, we provide polynomial-time and polynomial-sample algorithms for *learning* tree-structured Ising models. We provide two algorithms: the first requires to know the structure of the tree in advance while the second is assumption-free. Unsurprisingly, the latter requires more samples.

Theorem 3 *Fix error and confidence parameters $\varepsilon > 0$ and $\delta > 0$. Given m samples from the joint distribution over the leaves of an underlying tree-structured Ising model, there exist polynomial-time algorithms for learning a tree-structured Ising model (T, θ) whose marginal over the leaves is ε-close to the true marginal in total variation distance, with success probability $1 - \delta$.*

- ***Known topology:*** *If T is given, then $m \in O(n^4 \log(n/\delta)/\varepsilon^2)$ samples suffice.*
- ***Unknown topology:*** *In general, $m \in O(n^{14} \log(n/\delta)/\varepsilon^6)$ samples suffice.*

The algorithms for both settings consist of two steps: first, they empirically estimate the pairwise correlations between every pair of leaves. Then, they utilize an algorithm that, given these approximate correlations, finds *some* tree Ising model whose pairwise correlations are close to the estimated ones. The guarantees of those algorithms then follow directly from Theorem 1. Notice that for the case of known topology, it is possible to implement this algorithm using a simple linear programming, as outlined in Section 4. Thus, we only need to argue that the pairwise correlations are estimated with accuracy $O(\varepsilon/n^2)$, which gives $O(n^4/\varepsilon^2)$ sample complexity. In contrast, for unknown topology, the algorithm is slightly more complicated, which is why we lose some additional polynomial factors when trying to construct a topology with correlations close to the estimated ones. This is outlined in Appendix D.

To the best of our knowledge, the only prior work that provided a polynomial time algorithm for learning in total variation without any restrictions on the weights of the model was Cryan et al. (2001). While they do not explicitly state the sample complexity required, it can be inferred from their proof that at least $\Omega(n^{89}/\varepsilon^{18})$ samples are necessary, though we note that their result is slightly more general and holds for general trees with a binary alphabet; see Appendix D for a technical comparison between our results. Meanwhile, information theoretically, it is well-known that $\tilde{\Theta}(n/\varepsilon^2)$ samples are sufficient and necessary for learning the joint distribution, both in the known and unknown topology settings; for example, one can show this by modifying the arguments in Devroye et al. (2020); Brustle et al. (2020); Koehler (2020). For completeness, we provide a sketch of these arguments in Appendix E. While our work presents the best known efficient algorithm (runs in polynomial time), it uses $poly(n)/\varepsilon^2$ samples. It remains an open problem whether there exists an efficient algorithm that

only uses $O(n/\varepsilon^2)$ samples or whether there exists a *statistical-computational gap* for the problem at hand.

Remark 4 *The seminar work of Berthet and Rigollet (2013) conjectured that there is a fundamental gap between what is statistically required and what is achievable by computationally efficient algorithms, in the context of the sparse PCA problem. Multiple follow-up works suggest that such a gap is inherent to the problem (e.g. see d'Orsi et al. (2020); Choo and d'Orsi (2021); Ding et al. (2023)) justify the computational hardness via average-case reductions (Brennan and Bresler, 2020), sum-of-squares lower bounds (Raghavendra et al., 2018), or low-degree polynomial arguments (Schramm and Wein, 2022). Many other well-studied problems involving the recovery of planted signals also exhibit such a statistical-computational gap, e.g. tensor PCA, planted clique, etc.; see (Hopkins et al., 2017; Barak et al., 2019; Kunisky et al., 2019; Brennan and Bresler, 2020) for more.*

2. Related Work

A popular method for latent-tree estimation are *tree-metric* approaches, which rely on estimating the pairwise correlations between any two leaves. For these algorithms, there is a vast theoretical analysis which is largely focused on estimating the structure of the tree, namely, finding the set of edges E (Felsenstein, 1973; Chang, 1996; Erdős et al., 1999; Huson et al., 1999; Csurös, 2002; Felsenstein, 2004; King et al., 2003; Daskalakis et al., 2006; Roch, 2006; Mossel, 2007; Gronau et al., 2008; Roch, 2010; Roch and Sly, 2017). The results typically require some upper and lower bounds on the edge-weights a_{ij}. Such bounds guarantee that the structure of the tree can be completely identified from polynomially many samples. In contrast, Daskalakis et al. (2009) design an algorithm that reconstructs as much of the true topology as possible, without assuming bounds on the edge-weights. However, they do not provide any guarantee on the closeness of the learned distribution to the true distribution. Another popular family of algorithms are *likelihood-based* methods (Felsenstein, 1981; Yang, 1997; Stamatakis, 2006; Lee et al., 2006; Wang and Zhang, 2006; Truell et al., 2021), but their convergence guarantees are barely understood (Zwiernik et al., 2017; Daskalakis et al., 2018, 2022).

Beyond trees, the general problem of latent graphical model estimation has received some attention (Bresler et al., 2019; Bresler and Karzand, 2020; Moitra et al., 2021; Goel, 2019; Goel et al., 2020). However, all these algorithms have time- and sample- complexity that is exponential in the maximum degree of the graph. Also, Acharya et al. (2018) study testing of Bayesnets with latent variables, but under the assumption that the c-components have constant size.

Another related line of work is that of estimating a tree from fully observable data, while guaranteeing that the error is bounded in k-*local-TV*: this means that the output model is ε-close in total variation to the true model in any marginal of k nodes (where k is considered small). While the complexity of learning the full tree to ε total variation distance is $\Theta(n \log n/\varepsilon^2)$ (Daskalakis and Pan, 2021; Koehler, 2020), the algorithm of Boix-Adsera et al. (2022) has a sample complexity of $O(\log n \cdot k^2 2^{2k}/\varepsilon^2)$ for learning in k-local TV. The preceding paper of Bresler and Karzand (2020) obtained the same guarantee, however, they assume some upper bounds on the edge correlations θ_{ij}.

3. Technical Contributions – Proof Sketch

We describe the main tools for the proofs of Theorem 1.

3.1. Preliminaries

For the discussion, fix some tree $T = (V, E)$. For any leaves i, j, let P_{ij} denote the path connecting them. Denote by θ any vector in $[-1, 1]^E$ whose entry θ_e denotes the correlation across the edge $e \in E$. When we write $\alpha, \hat{\alpha}$ etc., this corresponds to a vector in $[-1, 1]^{\binom{n}{2}}$, whose entries, α_{ij} are indexed by two distinct leaves $i \neq j$. In general α can be an arbitrary vector, yet, we say that α is *induced* by some probability distribution on a tree T if it represents the pairwise correlations of the leaves in some Ising model that is defined over the tree (i.e. if (1) holds for some edge-correlations $\{\theta_e\}_{e\in E}$). Given a tree T, edge-correlations θ and $x \in \{-1, 1\}^n$, denote by $\Pr_{T,\theta}[x]$ the probability that the leaves equal x under the Ising model defined by T and θ. We say that $\Pr_{T,\theta}$ is the *leaf distribution* over $\{-1, 1\}^n$.

First, we define a pair of leaves i, j to be a *cherry* if they share their common neighbor (recall that a leaf has only one neighbor). In other words, if one directs the edges from some internal node to the leaves, i and j would share their parent.

3.2. An expression for the probability distribution on the leaves from Bresler and Karzand (2020)

We describe a convenient closed-form expression for the probability distribution over the leaves of the tree. To describe it, we begin with some definitions. Let S be a subset of the leaves of even cardinality. Then, there is a *unique*[3] way to partition S into $|S|/2$ pairs $(x_1, y_1), \ldots, (x_{|S|/2}, y_{|S|/2})$ such that the path connecting x_i and y_i is edge-disjoint from the path connecting x_j and y_j, for all $i \neq j$. This partitioning can be obtained by matching leaves that are closest to being a cherry (i.e. siblings have highest precedence) repeatedly. For example, if we have $S = \{i, j, k, \ell, m, p\}$ in the tree shown in Fig. 1, then we partition S into $(i, \ell), (j, m)$ and (k, p). The leaf distribution can be described as the following multilinear function of x, whose coefficients, that are indexed by sets S of even cardinality, rely on the aforementioned partitioning into pairs:

$$f_x^T(\alpha) := 2^{-n} \cdot \sum_{\substack{S\subseteq[n]\\ |S| \text{ even}}} \alpha_S^T \prod_{i\in S} x_i \,, \quad \text{where } \alpha_S^T := \prod_{i=1}^{|S|/2} \alpha_{x_i y_i} \tag{2}$$

The following lemma, which is a special case of Theorem H.1 in Bresler and Karzand (2020), argues that $f_x^T(\alpha)$ is the leaf distribution, as a function of x (proof in Appendix A for completeness).

Lemma 5 (Bresler and Karzand (2020)) *For any latent tree distribution with tree-topology T whose pairwise correlations over the leaves equal $\alpha = (\alpha_{ij})_{i,j \text{ leaves}}$. Then, the probability of any configuration $x = (x_1, \ldots, x_n) \in \{-1, 1\}^n$ on the leaves equals $f_x^T(\alpha)$.*

3.3. Technical tools for the tensorization of Theorem 1 (same topology)

We now utilize Theorem 5 to prove Theorem 1 (same topology). The total variation distance between two distributions on the same topology T with induced weight-vectors α and $\hat{\alpha}$ is $\sum_x |\Pr_{T,\alpha}[x] - \Pr_{T,\hat{\alpha}}[x]|/2 = \sum_x |f_x(\alpha) - f_x(\hat{\alpha})|/2$, where T is omitted from f_x^T for brevity. We would like to bound the above expression, assuming that α and $\hat{\alpha}$ are close and this corresponds to showing some

3. The uniqueness holds if each internal node has degree 3 and this assumption is without loss of generality, as we explain in Appendix A.

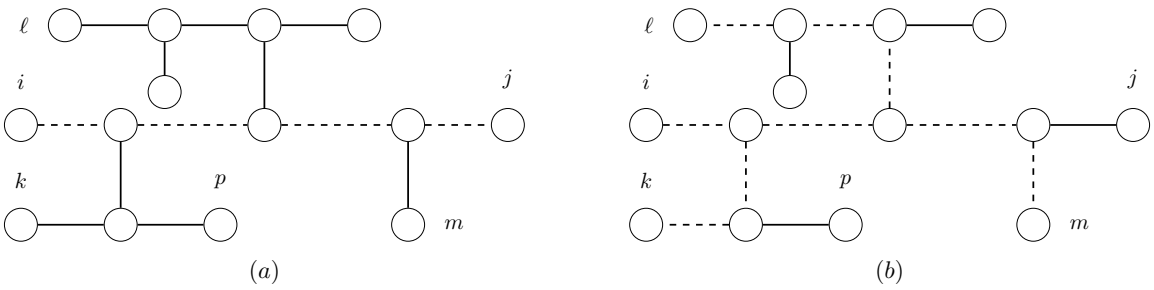


Figure 1: Path removal (dashed edges are removed): In (a), we depict the graph obtained from T by removing the path from i to j. In (b), we depict a removal of the quartet $\{i, k, \ell, m\}$.

Lipschitzness properties on $f_x(\alpha)$. For a vector $\alpha \in [-1,1]^{\binom{n}{2}}$, we say that α is *induced* by some tree distribution, or that a tree distribution *induces* α, if there exists a topology T with leaves in $[n]$ and weights θ_{ij} on the edges of T, such that for all pairs of leaves i, j (1) holds. We show that such a Lipschitzness property holds in the neighborhood of a probability distribution. Namely, that if α is induced by a tree distribution and we change one entry of α, then $f_x(\alpha)$ does not change much:

Lemma 6 (Formal statement in Theorem 11) *Suppose $\alpha \in [-1,1]^{\binom{n}{2}}$ is induced by some probability distribution on a tree T. Denote by $\alpha^{(ij)} \in [-1,1]^{\binom{n}{2}}$ the vector that agrees with α everywhere, except for pair of leaves ij, where $\alpha^{(ij)}_{ij} \neq \alpha_{ij}$. Then,*

$$\sum_{x \in \{-1,1\}^n} |f_x(\alpha) - f_x(\alpha^{(ij)})| \leq |\alpha_{ij} - \alpha^{(ij)}_{ij}|$$

Proof [Proof sketch] Let θ denote the weight vector on the edges of T that induces the correlation-vector α in accordance to (1), and let θ' be the weight vector that is obtained from θ by replacing any weight along the path from i to j with 0. Then, it can be shown that for all $x \in \{-1,1\}^n$, we have

$$\frac{|f_x(\alpha) - f_x(\alpha^{(ij)})|}{|\alpha_{ij} - \alpha^{(ij)}_{ij}|} = \Pr_{T,\theta'}[x]. \tag{3}$$

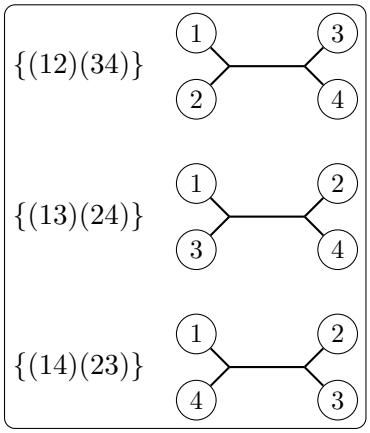
In other words, the ratio above equals the probability of x in the distribution that is obtained from $\Pr_{T,\theta}$ by removing the path from i to j in T, as depicted in Fig. 1(a). Since the right hand side represents a distribution over x, if we sum over x the result equals 1. ■

To bound the total variation between two weight vectors α and $\hat{\alpha}$, one would attempt to directly apply Theorem 6 multiple times, each time substituting one entry of α with its corresponding entry of $\hat{\alpha}$. However, in the process of transforming the vector one coordinate at a time, we may stumble upon an intermediate weight vector α' that is *not* induced by a probability distribution and so Theorem 6 does not apply. Hence, one has to prove an analogue of Theorem 6 for the case that α is close to being induced by a distribution. Interestingly, this can be proved by an inductive application of Theorem 6.

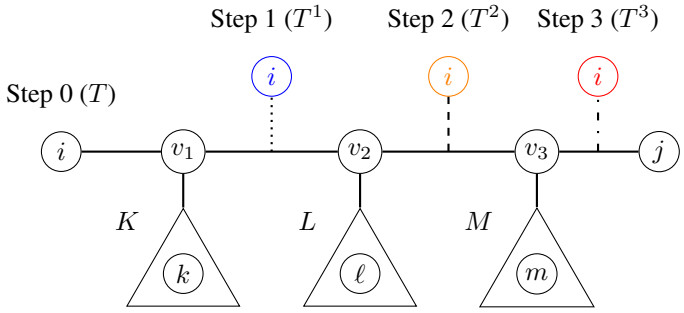
3.4. Technical Tools for the tensorization of Theorem 1 (different topologies)

In this section, we aim to bound the total variation distance between a probability distribution defined on a tree T with weights α and another defined on $\hat{T}$ with weight $\hat{\alpha}$, under the assumptions that the weights are ε-close: $|\alpha_{ij} - \hat{\alpha}_{ij}| \leq \varepsilon$. To compare between two different topologies, we will use the known fact that the topology on the nodes of a tree is completely determined by the set of all subgraphs that are induced by four leaves (*quartets*), if all latent nodes have degree 3. The idea is essentially the same as the one presented in Section 1.1, where it is argued that the pairwise correlations of all leaves determine the distribution. The pairwise distances are used to find the correct topology for each quartet. Then, using this information, we can identify which leaves are cherries (since they will fall on the same side for each quartet that contains both of them) and then proceed inductively to recover the entire topology. For more details, we refer the reader to Section 6.1 in Steel (2016).

Hence, in order to analyze the difference between two graphs, we can analyze the difference between these subgraphs of 4 leaves. This is significantly easier to analyze since the subgraphs contain only four nodes each. For this purpose, we introduce below some useful concepts, inspired by the phylogenetics literature.



(a) The 3 quartet possibilities



(b)

Figure 2: (a) The three possible topologies for a quartet. In the first topology, $\{(12)(34)\}$, the path from 1 to 2 does not intersect the path from 3 to 4. Further, $\alpha_{12}\alpha_{34} \geq \alpha_{13}\alpha_{24} = \alpha_{14}\alpha_{23}$; (b) The different positions of i in its movement across the tree towards j. Each position corresponds to a different tree T^i.

Definitions of a quartet. In the discussion below, we focus without loss of generality in the case where $\alpha_{ij} \geq 0$ for all i, j. Similar claims can be made for arbitrary signs. We will use the notion of a *quartet* of leaves: this is a collection of four leaves, $\{i, j, k, \ell\}$. It is well known in the phylogenetics literature that for every quartet there are 3 topologically distinct ways for these 4 leaves to connect with each other, if we contract all the paths leading to other leaves in the tree. Depending on which of the 3 ways we have, we say that the tree *induces* a topology for a specific quartet. We denote the

three induced topologies by $\{(ij)(k\ell)\}, \{(ik)(j\ell)\}$ and $\{(i\ell)(jk)\}$, where $\{(ij)(k\ell)\}$ means that the path P_{ij} is edge disjoint from the path $P_{k\ell}$, and similarly for the other topologies, as depicted in Fig. 2 (a). For a fixed quartet $\{i, j, k, l\}$, there are three quantities that determine the topology, out of the three possibilities, and those are $\alpha_{ij}\alpha_{k\ell}$, $\alpha_{ik}\alpha_{j\ell}$ and $\alpha_{i\ell}\alpha_{jk}$. It is known that two of these quantities are always equal and always smaller than the third, which determines the true topology: if $\alpha_{ij}\alpha_{k\ell}$ is the largest then $\{(ij)(k\ell)\}$ is the topology. If two trees induce a different topology for a quartet, we say that the trees *disagree* on that quartet, otherwise we say that they *agree* on it. It is known (Steel (2016)) that if two trees agree on all the quartets, then they should have the same topology. Lastly, note that in the special case where leaves $\{i, j\}$ form a cherry, the path P_{ij} does not share edges with the path between any two other leaves. This implies that the topology of $\{ijk\ell\}$ would necessarily be $\{(ij), (j\ell)\}$ for any two other leaves k and ℓ.

For the analysis, we would like to quantify how sensitive is the topology of a quartet to changing the weights. For any quartet $\{i, j, k, \ell\}$ and weight-vector α, define

$$\Delta_{ijk\ell}(\alpha) := \max\{\alpha_{ij}\alpha_{k\ell}, \alpha_{ik}\alpha_{j\ell}, \alpha_{i\ell}\alpha_{jk}\} - \min\{\alpha_{ij}\alpha_{k\ell}, \alpha_{ik}\alpha_{j\ell}, \alpha_{i\ell}\alpha_{jk}\}$$

(where the dependence on α is omitted for brevity). Recall from Section 1.1 that the topology of this quartet is determined by the largest of these three products, while the other two smaller products are identical. Notice that if $\|\alpha - \hat{\alpha}\|_\infty \le \varepsilon$, then

$$|\Delta_{ijk\ell}(\alpha) - \Delta_{ijk\ell}(\hat{\alpha})| \le \varepsilon$$

Hence, it is easy to see that two trees T and $\hat{T}$, with ε-close weights α and $\hat{\alpha}$, may disagree only on quartets where $\Delta_{ijk\ell}(\alpha) \le 2\varepsilon$. On the other hand, it is clear that if $\Delta_{ijk\ell}(\alpha)$ is small, then it is impossible for any algorithm to find the right topology for $\{i, j, k, l\}$. As we argue in the sequel, making a mistake for the topology of these quartets only results in a small loss in TV.

General approach Recall from Section 3.3 that we transformed α into $\hat{\alpha}$ by replacing its values one coordinate at a time for the case where $\hat{T} = T$. Here, we first fix α and transform T into $\hat{T}$. In other words, we find a sequence of topologies $T^1 = T, T^2, \ldots, T^k = \hat{T}$ that interpolates between T and $\hat{T}$ in a way that T^i, T^{i+1} only differ in a small part of the graph. After transforming T into $\hat{T}$, we then substitute α with $\hat{\alpha}$.

Transforming one tree into the other. We now describe in more detail the sequence of local moves from T to $\hat{T}$. Intuitively, the quartets are the analog of the pairwise distances in the fixed topology setting, and so we will measure dissimilarity between trees by the number of quartets that they disagree on. Thus, the goal is to produce a sequence of local topological changes that reduces quartet disagreements between T and $\hat{T}$ while ensuring that each consecutive pair of terms $f_x^{T^i}$ and $f_x^{T^{i-1}}$ is close.

The sequence of moves starts by identifying two leaves i, j that are a cherry in $\hat{T}$ but are not a cherry in T (if one exists). Since i, j are not a cherry in T, there is a path connecting them, which involves at least 2 other nodes, by definition of a cherry. Denote the path by $P_{ij} = v_1 - v_2 - \cdots - v_\ell$ where $v_1 = i, v_l = j$ and $l \ge 4$. In the process of transforming T into $\hat{T}$, we select one of the two nodes (according to some criterion), say it is i, and move it along the path P_{ij} in T until it becomes a cherry with j. The sequence involved with making i form a cherry with j is called an *epoch*. To be more specific, the t-th *step* in this epoch involves cutting i from its current place, attaching a node in the middle of edge (v_t, v_{t+1}) and connecting i to that node. After we do that, we also contract any

potential paths of degree 2 nodes that were formed into a single edge. Thus, there is a total of $l-1$ steps in the epoch to move i to j. The different steps of this epoch are shown in Fig. 2 (b), where we can see the different positions of i.

When we are done moving i towards j, we will find a new pair to make a cherry and so on. If T and $\hat{T}$ agree on all cherries, we look for disagreements due to parents of cherries. Specifically, suppose u, v are two parents of cherries that are siblings in $\hat{T}$ but not in T. Then, we will move u to become a sibling with v, which corresponds to an epoch. The idea for the movement is the same as the one described for leaves, except now instead of cutting and pasting u, we cut an paste u together with the subtree that hangs below u and has already been fixed by the algorithm. We then continue this process for grandparents of cherries, and so on. When this process ends, T and $\hat{T}$ are guaranteed to be the same. For more details, see Algorithm 3.

Analyzing one step. Let us focus on the first epoch in the above process. We first notice that the first step of the first epoch does not change the topology over the leaves. This is because after we cut and paste i, we contract all paths of nodes of degree 2. This is clear from Fig. 2(b), where we can see that in T^1, node v_1 has degree 2 and will be contracted, yielding essentially the same topology as $T^0 = T$.

Thus, we will analyze the second step, transforming T^1 into T^2 as shown in Fig. 2 (b). This involves cutting i from the middle of edge (v_1, v_2) and pasting it in the middle of edge (v_2, v_3). We will bound $|f_x^{T^1}(\alpha) - f_x^{T^2}(\alpha)|$ in terms of the parameters $\Delta_{ijk\ell}$ of the quartets that T^1 and T^2 disagree on:

Lemma 7 (Formal statement in Theorem 18) *Let i, j be a cherry in $\hat{T}$ but not in T and let T^1 and T^2 be the topologies defined by the procedure above (depicted in Fig. 2 (b)). Also, denote by U the set of quartets where T^1 and T^2 disagree on. Then,*

$$\sum_{x\in\{-1,1\}^n} |f_x^{T^1}(\alpha) - f_x^{T^2}(\alpha)| \leq \sum_{\{i,k,\ell,m\}\in U} \Delta_{ik\ell m} \tag{4}$$

Proof [Proof sketch] Let θ denote the weight vector on the edges of T that induces the correlation-vector α in accordance to (1). For a quartet of leaves $\{i,k,l,m\}$, let θ_{iklm} be the weight vector that is obtained from θ by replacing the weight of any edge along the paths $P_{ik}, P_{il}, P_{im}, P_{kl}, P_{km}, P_{lm}$ with 0. Then, it can be shown using the expression (2) that for all $x \in \{-1,1\}^n$, we have

$$|f_x^{T^1}(\alpha) - f_x^{T^2}(\alpha)| = \sum_{\{i,k,l,m\}\in U} \Delta_{iklm}(\alpha) \Pr_{T,\theta_{i,k,l,m}}[x]. \tag{5}$$

In other words, the difference above equals a sum of terms, each one corresponding to the probability of x in the distribution that is obtained from $\Pr_{T,\theta}$ by removing all the paths of the quartet $\{i,k,l,m\}$ from T, as depicted in Fig. 1(b). Summing over x completes the proof. ■

Theorem 7 can be seen as an analogue of Theorem 6 but we have quartets instead of pairwise distances. It shows a type of Lipschitzness of $f_x^{T^1}$ when we change the topology of some of the quartets.

Next, we will bound the right hand side of (4). Essentially, since we know that T and $\hat{T}$ disagree on the quartets in U, then it should be the case that $\Delta_{iklm}(\alpha)$ is small, otherwise the algorithm would be able to find the correct topology for these quartets. The next lemma formalizes this idea.

Lemma 8 (Formal statement in Theorem 19) *Each term $\Delta_{ik\ell m}$ in the right hand side of (4) is bounded by 2ε.*

Proof [Proof sketch] Using a simple case analysis, it can be shown that the only quartets that can change topology are those that contain i, some $k \in K$, some $\ell \in L$ and some $m \in M \cup \{j\}$ (see Fig. 2 (b)). The quartet topology is $\{(ik)(\ell m)\}$ in T^1 and $\{(im)(k\ell)\}$ in T^2 and we would like to argue that $\Delta_{ik\ell m} \leq 2\varepsilon$. First, let us assume that $m = j$. Then, the topology of $\{i, k, l, m\}$ in T^1 equals that of T, since T and T^1 share the same leaf topology, while the quartet topology of T^2 equals that of $\hat{T}$: indeed, the topology of $\hat{T}$ is $\{(ij)(k\ell)\}$ since (i, j) is a cherry in $\hat{T}$, as assumed in the algorithm and this is also the topology in T^2. In particular, since the topology in T^1 is different than that in T^2, then the topology in T is different than that in $\hat{T}$. As explained after the definition of $\Delta_{ik\ell m}$ above, this implies that $\Delta_{ik\ell m}(\alpha) \leq 2\varepsilon$. In particular, this provides a bound on $\Delta_{ik\ell m}$ as required. The case that $j \neq m$ is more complicated and it relies on the fact that i can be selected such that $\Delta_{ik\ell m} \leq \Delta_{jk\ell m}$ and considering the quartet (j, k, ℓ, m). ■

Completing the proof After ensuring that the move described in Theorem 7 incurs a small loss, the natural next step is to repeatedly apply a variant of Theorem 7 for each other step of the sequence and obtain a bound for $|f_x^T(\alpha) - f_x^{\hat{T}}(\alpha)|$, using the triangle inequality. We would like to analyze the total loss incurred in all the steps. We divide these steps into rounds, where in the first round we move leaves, in the second round we move parents of leaves etc. The number of rounds is bounded by the diameter D of $\hat{T}$. We can show that in each round, every quartet changes topology at most 4 times. Furthermore, in the general variant of Theorem 8 that corresponds to a movement of a subtree (rather than a leaf), ε is replaced with $n\varepsilon$. Since there are at most $\binom{n}{4}$ quartets, the total loss incurred in TV during all these steps is $O(Dn^5\varepsilon)$.

4. Algorithm

Algorithm 1 Learn a tree-structured distribution with a *known* topology

Input : An unweighted tree $T = (V, E)$ with leaf labels $\{1, \ldots, n\}$, tolerance parameter $\eta > 0$, and $\{\hat{\alpha}_{ij}\}_{i,j\in\{1,\ldots,n\},\, i\neq j}$ such that $|\alpha^*_{ij} - \hat{\alpha}_{ij}| \leq \eta$

Output : Weight $\theta_{k\ell}$ for each edge $\{k, \ell\} \in E$

Let $\{w_{k\ell}\}_{\{k,\ell\}\in E}$ be any solution satisfying the following linear constraints:

$$\begin{aligned} \log(\hat{\alpha}_{ij} - \eta) \leq \textstyle\sum_{\{k,\ell\}\in \mathrm{path}(i,j)} w_{k\ell} \leq \log(\hat{\alpha}_{ij} + \eta) \quad & \text{for all leaves } i \neq j \\ w_{k\ell} \leq 0 \quad & \text{for all edges } \{k, \ell\} \end{aligned} \tag{6}$$

For each edge $\{k, \ell\} \in E$, set $\theta_{k\ell} \leftarrow e^{w_{k\ell}}$.

return $\{\theta_{k\ell}\}_{\{k,\ell\}\in E}$

We describe the algorithms of Theorem 3 both for the case that the tree topology is known and when it is unknown, given m samples $(x_1^1, \ldots x_n^1), \ldots, (x_1^m, \ldots, x_n^m) \in \{-1, 1\}^n$. Both algorithms first estimate the covariance between any two leaves from samples, setting $\hat{\alpha}_{ij} = \frac{1}{m}\sum_{\ell=1}^m x_i^\ell x_j^\ell$. We note that by Chernoff-Hoeffding and a union bound, with probability at least $1 - \delta$ $\forall i \neq j$: $|\hat{\alpha}_{ij} - \alpha^*_{ij}| \leq \eta := \sqrt{2\log(n^2/\delta)/m}$. Given such estimates on the covariance, our algorithms

will find some weighted tree whose correlations α_{ij} between the leaves are close to the estimated correlations $\hat{\alpha}_{ij}$. By the triangle inequality, the correlations of the estimated tree are close to the true correlations and the result will follow by applying Theorem 1.

Known tree topology. We describe an algorithm that learns the weights of a fixed tree, given the estimated correlations $\hat{\alpha}_{ij}$. From the previous discussion, we can assume that all these estimations are accurate up to an additive error of $\eta = \sqrt{2\log(n^2/\delta)/m}$. For simplicity, we will assume that the edge weights θ^*_{kl} are non-negative, which implies that $\alpha^*_{ij} \geq 0$ for all i, j. In Appendix B.2, we show how this technique can be modified to handle arbitrary signs.

We will construct a linear program that finds weights $\theta_{k\ell} \geq 0$ on the edges $(k, \ell) \in E$. The variables of the linear program are $(w_{k\ell})_{(k,\ell)\in E}$ and they signify $w_{k\ell} = \log \theta_{k\ell}$. We would like our output to satisfy the following constraints: (1) $\theta_{k\ell} \in [0, 1]$, which can be rewritten as $w_{k\ell} \leq 0$; and (2) For any leaves i, j, $\hat{\alpha}_{ij} - \eta \leq \alpha_{ij} \leq \hat{\alpha}_{ij} + \eta$. If we take log and substitute $\alpha_{ij} = \prod_{(k,l)\in \mathrm{path}(i,j)} \theta_{k\ell}$ according to (1), we get the linear constraints described in (6) (while using the convention $\log x = -\infty$ for $x \leq 0$). This yields a linear program for finding the logarithms of the weights of the tree, and we can obtain weights for the tree by exponentiation of these log-values.

Algorithm 2 Learn a tree-structured distribution with an *unknown* topology

Input : Leaf correlation estimates $\{\hat{\alpha}_{ij}\}_{i,j\in\{1,\ldots,n\},\ i\neq j}$ and parameters $\eta', \xi, \delta > 0$.
Output : A weighted forest F

1 Let F be the forest output by the algorithm of Daskalakis et al. (2009) when given weights $\{\hat{\alpha}_{ij}\}_{i\neq j}$ and parameters $\xi, \delta > 0$ as input.
2 For each $T = (V, E) \in F$, run Algorithm 1 with T, η', and $\hat{\alpha}_{i,j}$ to obtain weights $\{\theta_{k\ell}\}_{\{k,\ell\}\in E}$.
3 For each $T = (V, E) \in F$, set tree edge weights to $\{\theta_{k\ell}\}_{\{k,\ell\}\in E}$.
4 **return** F

Unknown tree topology. If we do not know the tree structure, we use the algorithm of Daskalakis et al. (2011) that, given approximations $\hat{\alpha}_{ij}$ of the correlations between the leaves, finds a forest that shares multiple properties with the original tree. Then, for any tree in this forest, we compute weights on the edges, using Algorithm 1 and return the weighted forest, as summarized in Algorithm 2. To analyze this algorithm, we perform a series of careful contractions and deletions of edges, that transform this forest into one where each subtree has exactly the same topology as the one induced by the true tree on that particular subset of leaves. Then, crucially, we use the analysis for the known topology setting, to bound the difference in total variation between learned marginal distribution over the leaves of each subtree and the true marginal distribution. For details, see Appendix D.

Acknowledgments

This research/project is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-PhD/2021-08-013). Constantinos Daskalakis, Yuval Dagan and Vardis Kandiros was supported by NSF Awards CCF-1901292, DMS-2022448 and DMS2134108, a Simons Investigator Award, the Simons Collaboration on the Theory of Algorithmic Fairness, a DSTA grant, and the DOE PhILMs project (DE-AC05-76RL01830). Vardis Kandiros was also supported by a Fellowship of the Eric and Wendy Schmidt Center at the Broad Institute of MIT

and Harvard and by the Onassis Foundation-Scholarship ID: F ZP 016-1/2019-2020. Part of this work was done while the authors were visiting the Simons Institute for the Theory of Computing.

References

Jayadev Acharya, Arnab Bhattacharyya, Constantinos Daskalakis, and Saravanan Kandasamy. Learning and testing causal models with interventions. *Advances in Neural Information Processing Systems*, 31, 2018.

Dennis J Aigner, Cheng Hsiao, Arie Kapteyn, and Tom Wansbeek. Latent variable models in econometrics. *Handbook of econometrics*, 2:1321–1393, 1984.

Boaz Barak, Samuel Hopkins, Jonathan Kelner, Pravesh K Kothari, Ankur Moitra, and Aaron Potechin. A nearly tight sum-of-squares lower bound for the planted clique problem. *SIAM Journal on Computing*, 48(2):687–735, 2019.

David J Bartholomew, Martin Knott, and Irini Moustaki. *Latent variable models and factor analysis: A unified approach*, volume 904. John Wiley & Sons, 2011.

Quentin Berthet and Philippe Rigollet. Complexity Theoretic Lower Bounds for Sparse Principal Component Detection. In *Conference on Learning Theory*, pages 1046–1066, 2013.

Arnab Bhattacharyya, Sutanu Gayen, Eric Price, and NV Vinodchandran. Near-optimal learning of tree-structured distributions by chow-liu. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, 2021.

Arnab Bhattacharyya, Clément L Canonne, and Qiping Yang. Independence testing for bounded degree bayesian networks. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 15027–15038. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/611252d40f23c8b57a8bc9ffb577419b-Paper-Conference.pdf.

Christopher M Bishop. Latent variable models. In *Learning in graphical models*, pages 371–403. Springer, 1998.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Enric Boix-Adsera, Guy Bresler, and Frederic Koehler. Chow-liu++: Optimal prediction-centric learning of tree ising models. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 417–426. IEEE, 2022.

Matthew Brennan and Guy Bresler. Reducibility and statistical-computational gaps from secret leakage. In *Conference on Learning Theory*, pages 648–847. PMLR, 2020.

Guy Bresler. Efficiently learning ising models on arbitrary graphs. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 771–782, 2015.

Guy Bresler and Mina Karzand. Learning a tree-structured ising model in order to make predictions. *The Annals of Statistics*, 48(2):713–737, 2020.

Guy Bresler, Frederic Koehler, and Ankur Moitra. Learning restricted boltzmann machines via influence maximization. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 828–839, 2019.

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

Clément L Canonne, Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Testing bayesian networks. In *Conference on Learning Theory*, pages 370–448. PMLR, 2017.

Joseph T Chang. Full reconstruction of markov models on evolutionary trees: identifiability and consistency. *Mathematical biosciences*, 137(1):51–73, 1996.

Davin Choo and Tommaso d'Orsi. The complexity of sparse tensor pca. *Advances in Neural Information Processing Systems*, 34:7993–8005, 2021.

Benny Chor and Tamir Tuller. Maximum likelihood of evolutionary trees is hard. In *Annual International Conference on Research in Computational Molecular Biology*, pages 296–310. Springer, 2005.

C Chow and T Wagner. Consistency of an estimate of tree-dependent probability distributions (corresp.). *IEEE Transactions on Information Theory*, 19(3):369–371, 1973.

CKCN Chow and Cong Liu. Approximating discrete probability distributions with dependence trees. *IEEE transactions on Information Theory*, 14(3):462–467, 1968.

Mary Cryan, Leslie Ann Goldberg, and Paul W Goldberg. Evolutionary trees can be learned in polynomial time in the two-state general markov model. *SIAM Journal on Computing*, 31(2): 375–397, 2001.

Miklós Csurös. Fast recovery of evolutionary trees with thousands of nodes. *Journal of Computational Biology*, 9(2):277–297, 2002.

Yuval Dagan, Constantinos Daskalakis, Nishanth Dikkala, and Anthimos Vardis Kandiros. Learning ising models from one or multiple samples. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 161–168, 2021.

Constantinos Daskalakis and Qinxuan Pan. Square hellinger subadditivity for bayesian networks and its applications to identity testing. In *Conference on Learning Theory*, pages 697–703. PMLR, 2017.

Constantinos Daskalakis and Qinxuan Pan. Sample-optimal and efficient learning of tree ising models. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, 2021.

Constantinos Daskalakis, Elchanan Mossel, and Sébastien Roch. Optimal phylogenetic reconstruction. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 159–168, 2006.

Constantinos Daskalakis, Elchanan Mossel, and Sébastien Roch. Phylogenies without branch bounds: Contracting the short, pruning the deep. In *Annual International Conference on Research in Computational Molecular Biology*, pages 451–465. Springer, 2009.

Constantinos Daskalakis, Elchanan Mossel, and Sébastien Roch. Evolutionary trees and the ising model on the bethe lattice: a proof of steel's conjecture. *Probability Theory and Related Fields*, 149(1):149–189, 2011.

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

Constantinos Daskalakis, Yuval Dagan, and Anthimos-Vardis Kandiros. Where does em converge in gaussian latent tree models? In *Conference on Learning Theory (COLT)*, 2022.

Costis Daskalakis, Christos Tzamos, and Manolis Zampetakis. Bootstrapping em via power em and convergence in the naive bayes model. In *International Conference on Artificial Intelligence and Statistics*, pages 2056–2064. PMLR, 2018.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1): 1–22, 1977.

Luc Devroye, Abbas Mehrabian, and Tommy Reddad. The minimax learning rates of normal and ising undirected graphical models. *Electronic Journal of Statistics*, 14(1):2338–2361, 2020.

Mucong Ding, Constantinos Daskalakis, and Soheil Feizi. Gans with conditional independence graphs: On subadditivity of probability divergences. In *International Conference on Artificial Intelligence and Statistics*, pages 3709–3717. PMLR, 2021.

Yunzi Ding, Dmitriy Kunisky, Alexander S Wein, and Afonso S Bandeira. Subexponential-time algorithms for sparse pca. *Foundations of Computational Mathematics*, pages 1–50, 2023.

Tommaso d'Orsi, Pravesh K Kothari, Gleb Novikov, and David Steurer. Sparse pca: algorithms, adversarial perturbations and certificates. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 553–564. IEEE, 2020.

Péter L Erdős, Michael A Steel, László A Székely, and Tandy J Warnow. A few logs suffice to build (almost) all trees (i). *Random Structures & Algorithms*, 14(2):153–184, 1999.

B Everett. *An introduction to latent variable models*. Springer Science & Business Media, 2013.

Joseph Felsenstein. Maximum-likelihood estimation of evolutionary trees from continuous characters. *American journal of human genetics*, 25(5):471, 1973.

Joseph Felsenstein. Evolutionary trees from gene frequencies and quantitative characters: finding maximum likelihood estimates. *Evolution*, pages 1229–1242, 1981.

Joseph Felsenstein. *Inferring phylogenies*, volume 2. Sinauer associates Sunderland, MA, 2004.

Surbhi Goel. Learning restricted boltzmann machines with arbitrary external fields. *arXiv preprint arXiv:1906.06595*, 2019.

Surbhi Goel, Adam Klivans, and Frederic Koehler. From boltzmann machines to neural networks and back again. *Advances in Neural Information Processing Systems*, 33:6354–6365, 2020.

Ilan Gronau, Shlomo Moran, and Sagi Snir. Fast and reliable reconstruction of phylogenetic trees with very short edges. In *SODA*, volume 8, pages 379–388, 2008.

Linus Hamilton, Frederic Koehler, and Ankur Moitra. Information theoretic properties of markov random fields, and their algorithmic applications. *Advances in Neural Information Processing Systems*, 30, 2017.

Samuel B Hopkins, Pravesh K Kothari, Aaron Potechin, Prasad Raghavendra, Tselil Schramm, and David Steurer. The power of sum-of-squares for detecting hidden structures. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 720–731. IEEE, 2017.

Daniel H Huson, Scott M Nettles, and Tandy J Warnow. Disk-covering, a fast-converging method for phylogenetic tree reconstruction. *Journal of computational biology*, 6(3-4):369–386, 1999.

Ali Jalali, Pradeep Ravikumar, Vishvas Vasuki, and Sujay Sanghavi. On learning discrete graphical models using group-sparse regularization. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 378–387. JMLR Workshop and Conference Proceedings, 2011.

Matthew G Jones, Alex Khodaverdian, Jeffrey J Quinn, Michelle M Chan, Jeffrey A Hussmann, Robert Wang, Chenling Xu, Jonathan S Weissman, and Nir Yosef. Inference of single-cell phylogenies from lineage tracing data using cassiopeia. *Genome biology*, 21(1):1–27, 2020.

Michael I Jordan. Graphical models. *Statistical science*, 19(1):140–155, 2004.

Vardis Kandiros, Yuval Dagan, Nishanth Dikkala, Surbhi Goel, and Constantinos Daskalakis. Statistical estimation from dependent data. In *International Conference on Machine Learning*, pages 5269–5278. PMLR, 2021.

Valerie King, Li Zhang, and Yunhong Zhou. On the complexity of distance-based evolutionary tree. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, page 444. SIAM, 2003.

Adam Klivans and Raghu Meka. Learning graphical models using multiplicative weights. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 343–354. IEEE, 2017.

Frederic Koehler. A note on minimax learning of tree models. 2020.

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

Dmitriy Kunisky, Alexander S Wein, and Afonso S Bandeira. Notes on computational hardness of hypothesis testing: Predictions using the low-degree likelihood ratio. In *ISAAC Congress (International Society for Analysis, its Applications and Computation)*, pages 1–50. Springer, 2019.

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

Chunghau Lee, Sigal Blay, Arne Ø Mooers, Ambuj Singh, and Todd H Oakley. Comet: A mesquite package for comparing models of continuous character evolution on phylogenies. *Evolutionary Bioinformatics*, 2:117693430600200022, 2006.

Ankur Moitra, Elchanan Mossel, and Colin P Sandon. Learning to sample from censored markov random fields. In *Conference on Learning Theory*, pages 3419–3451. PMLR, 2021.

Elchanan Mossel. Distorted metrics on trees and phylogenetic forests. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(1):108–116, 2007.

Elchanan Mossel and Sébastien Roch. Learning nonsingular phylogenies and hidden markov models. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 366–375, 2005.

Mukund Narasimhan and Jeff A Bilmes. Pac-learning bounded tree-width graphical models. In *Proc. 20th Ann. Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2004.

Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.

Prasad Raghavendra, Tselil Schramm, and David Steurer. High dimensional estimation via sum-of-squares proofs. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pages 3389–3423. World Scientific, 2018.

Pradeep Ravikumar, Martin J Wainwright, and John D Lafferty. High-dimensional ising model selection using l 1-regularized logistic regression. *The Annals of Statistics*, 38(3):1287–1319, 2010.

Sebastien Roch. A short proof that phylogenetic tree reconstruction by maximum likelihood is hard. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(1):92–94, 2006.

Sebastien Roch. Toward extracting all phylogenetic information from matrices of evolutionary distances. *Science*, 327(5971):1376–1379, 2010.

Sebastien Roch and Allan Sly. Phase transition in the sample complexity of likelihood-based phylogeny inference. *Probability Theory and Related Fields*, 169(1):3–62, 2017.

Narayana P Santhanam and Martin J Wainwright. Information-theoretic limits of selecting binary graphical models in high dimensions. *IEEE Transactions on Information Theory*, 58(7):4117–4134, 2012.

Tselil Schramm and Alexander S Wein. Computational barriers to estimation from low-degree polynomials. *The Annals of Statistics*, 50(3):1833–1858, 2022.

Alexandros Stamatakis. Raxml-vi-hpc: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics*, 22(21):2688–2690, 2006.

Mike Steel. *Phylogeny: discrete and random processes in evolution*. SIAM, 2016.

Vincent YF Tan, Animashree Anandkumar, Lang Tong, and Alan S Willsky. A large-deviation analysis of the maximum-likelihood learning of markov tree structures. *IEEE Transactions on Information Theory*, 57(3):1714–1735, 2011.

Michael Truell, Jan-Christian Hütter, Chandler Squires, Piotr Zwiernik, and Caroline Uhler. Maximum likelihood estimation for brownian motion tree models based on one sample. *arXiv preprint arXiv:2112.00816*, 2021.

Marc Vuffray, Sidhant Misra, Andrey Lokhov, and Michael Chertkov. Interaction screening: Efficient and sample-optimal learning of ising models. *Advances in neural information processing systems*, 29, 2016.

Marc Vuffray, Sidhant Misra, and Andrey Y Lokhov. Efficient learning of discrete graphical models. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124017, 2022.

Yi Wang and Nevin Lianwen Zhang. Severity of local maxima for the em algorithm: Experiences with hierarchical latent class models. In *Probabilistic Graphical Models*, pages 301–308. Citeseer, 2006.

Ziheng Yang. Paml: a program package for phylogenetic analysis by maximum likelihood. *Computer applications in the biosciences*, 13(5):555–556, 1997.

Yannis G Yatracos. Rates of convergence of minimum distance estimators and kolmogorov's entropy. *The Annals of Statistics*, 13(2):768–774, 1985.

Piotr Zwiernik, Caroline Uhler, and Donald Richards. Maximum likelihood estimation for linear gaussian covariance models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(4):1269–1292, 2017.

Appendix A. A formula for the leaf distribution from Bresler and Karzand (2020)

Suppose we have a tree Ising model T. We show how we can convert it into another Ising model T', such that T' has latent nodes of degree exactly 3 and the leaf distributions of T and T' are the same. Indeed, suppose i is a latent node of degree 2. Let u, v be the neighbors of i in T. Then, we can delete i in T' and replace the edges $(i, u), (i, v)$ with the edge (u, v) with $\theta_{uv} = \theta_{iu}\theta_{iv}$. If x_u, x_v and x'_u, x'_v denote the random variables of leaves u, v in T and T', respectively, then it is clear that

$$\mathbb{E}[x_u x_v] = \mathbb{E}[x_u x_i^2 x_v] = \mathbb{E}[x_u x_i x_v x_i] = \mathbb{E}[x_u x_i]\mathbb{E}[x_v x_i] = \theta_{ui}\theta_{vi} = \mathbb{E}[x'_u x'_v]$$

Thus, the distribution of the pair x_u, x_v doesn't change from this local change. By the Markov property, this implies that the distribution of leaves also doesn't change. We can apply this procedure succesively to eliminate all nodes with degree 2.

Now, suppose there is a latent node r in T with degree $k > 3$. Let $u_1, \dots, u_k$ be the neighbors of i in T. We delete r and introduce two nodes s, t which are connected by an edge (s, t) with $\theta_{st} = 1$. We connect u_1, u_2 to s with the same edge weights as u_1, u_2 had with r. Lastly, we connect $u_3, \dots, u_k$ to t with same edge weights as before. Clearly, the topology is still a tree, and $x_s = x_t$ always, since $\theta_{st} = 1$. Thus, x_s, x_t have the same value that x_r had in T, and so the distribution of the leaves does not change. Furthermore, x_s has degree 3 and x_t has degree $k - 1$. Thus, we can apply this procedure successively until all nodes have degree 3. The final tree that we obtain is T', and we just argued that it has the same leaf distribution and all latent nodes have degree 3. Thus, from now on we will make the implicit assumption that all internal nodes have degree 3.

Next, we introduce some convenient notation. Let S be a subset of the leaves of even cardinality. Then, there is a natural way to partition the leaves in S in $|S|/2$ pairs using the following criterion: each leaf i in S is matched with its closest relative in S in the tree. Yet, to be more exact, we say that a matching of S is a *closest relative matching* if for any two distinct pairs (i, j) and (k, ℓ) in the matching, the path from i to j does not intersect the path from k to ℓ. An example of such a matching is given in Figure 4. In the following proposition, we prove that there is a unique such matching, and use this matching to find an expression to the leaf distribution of an Ising model:

Lemma 9 *Let $x_1, \dots, x_n$ denote the values over the n leaves of a tree T with pairwise correlations $\alpha \in [-1, 1]^{\binom{n}{2}}$. Then, the following holds:*

- *Any subset $S \subseteq [n]$ of even cardinality has a unique closest relative matching.*
- *Define for any subset $S \subseteq [n]$ of even cardinality*

$$\alpha_S := \prod_{k=1}^{|S|/2} \alpha_{i_k j_k}$$

where $(i_1, j_1), \dots, (i_{|S|/2-1}, j_{|S|/2-1})$ are the pairs in the closest relative matching. Then, we have

$$\Pr[x_1, \dots, x_n] = \frac{\sum_{\text{even subsets } S \subseteq [n]} \alpha_S \prod_{i \in S} x_i}{2^n} \tag{7}$$

Proof Recall that we can assume that there are no nodes of degree 2 (otherwise, we can contract maximal paths of degree-2 nodes and replace them by a single edge whose weight is the product

of all edges in the path. This does not change the leaf distribution.) We will prove both statements together by induction. We will focus on the proof for the probability expression, and the uniqueness of the matching will come as a by-product. The base case considers either 0, 1 or 2 leaves, and follows trivially. For the induction step, suppose the claim is true for all trees having at most $n-1$ leaves. Let T be a tree with n leaves. We are interested in the probability $\Pr[x_1,\dots,x_n]$ of the leaves taking some specific values. First of all, since T is assumed to contain no nodes of degree 2, we know that there exists at least one cherry, i.e. a pair of leaves that share their parent. Also, without loss of generality, suppose that leaves $n-1$ and n form a cherry and denote by p their common parent in the tree. Lastly, denote by $\theta_{(p,n-1)}$ and $\theta_{(p,n)}$ the weights of the edges $(p,n-1)$ and (p,n) respectively. Then, we know that x_n, x_{n-1} are conditionally independent from the rest of the tree conditioned on y_p, where y_p denotes the value of node p. Thus, we can write

$$\begin{aligned}\Pr[x_1,\dots,x_n] = &\Pr[x_1,\dots,x_{n-2},y_p=1]\Pr[x_{n-1},x_n \mid y_p=1]\\ &+\Pr[x_1,\dots,x_{n-2},y_p=-1]\Pr[x_{n-1},x_n \mid y_p=-1]\end{aligned}$$

Now, notice that we can view the nodes $1,\dots,n-2,p$ as the leaves of a tree T' which is simply T after deleting leaves $n-1$ and n and the edges $(n-1,p)$ and (n,p). Hence, we can apply the induction hypothesis on the distribution of $x_1,\dots,x_{n-2},y_p$. Recall that the expression for the probability distribution is a function of the expressions α_S of all the even subsets S of the leaves. Hence, we would like to compare the coefficients α_S between the distribution over $x_1,\dots,x_n$ and the distribution over $x_1,\dots,x_{n-2},y_p$ that is used for the induction hypothesis. In order for such a comparison to be possible, we divide the collections of even subsets of the leaves of T and T' into categories. We start with the leaves of tree T. Denote by $\mathcal{S}$ the set of all even subsets of the set of leaves $\{1,\dots,n\}$. Clearly, we can partition $\mathcal{S}$ into 4 disjoint subsets:

$$\begin{aligned}\mathcal{S}_{--} &:= \{\text{even subsets of } [n] \text{ not containing neither } n-1 \text{ nor } n\}\\ \mathcal{S}_{+-} &:= \{\text{even subsets of } [n] \text{ containing } n-1 \text{ but not containing } n\}\\ \mathcal{S}_{-+} &:= \{\text{even subsets of } [n] \text{ not containing } n-1 \text{ but containing } n\}\\ \mathcal{S}_{++} &:= \{\text{even subsets of } [n] \text{ containing both } n-1 \text{ and } n\}\end{aligned}$$

Similarly, we define analogues to be applied on the distribution that is used in the induction hypothesis. In particular, define by $\mathcal{R}$ the collection of all even subsets of $[n-2]\cup\{p\}$. We can also partition $\mathcal{R}$ into the following subsets:

$$\begin{aligned}\mathcal{R}_{-} &:= \{\text{even subsets of } [n-2]\cup\{p\} \text{ not containing } p\}\\ \mathcal{R}_{+} &:= \{\text{even subsets of } [n-2]\cup\{p\} \text{ containing } p\}\end{aligned}$$

While applying the induction hypothesis, once computing $\Pr[x_1,\dots,x_n]$ we will split the sum in (7) into four sums over the different subsets of $\mathcal{S}$ that were defined above. Similarly, while computing $\Pr[x_1,\dots,x_{n-2},y_p]$ we will split the sum into terms corresponding to $\mathcal{R}_-$ and $\mathcal{R}_+$. In order to be able to compare between these two sums, we will map each of the four subsets of $\mathcal{S}$ to a subset of $\mathcal{R}$.

First, note that $\mathcal{S}_{--}=\mathcal{R}_-$ since both equal the collection of even subsets of $[n-2]$. Hence, it follows by induction hypothesis that the sets $S\in\mathcal{S}_{--}$ have a unique closest relative matching. Further,

$$\sum_{S\in\mathcal{S}_{--}}\prod_{i\in S}x_i\alpha_S = \sum_{S\in\mathcal{R}_-}\prod_{i\in S}x_i\alpha_S\ .$$

Next, notice that there is a bijection between $\mathcal{S}_{++}$ and $\mathcal{R}_{-}$, which takes any $S \in \mathcal{S}_{++}$ to $S \setminus \{n-1, n\}$. We use this to prove that there exists a closest-relative matching for any $S \in \mathcal{S}_{++}$. Indeed, we can match $n-1$ with n and then match $S \setminus \{n-1, n\}$. This is possible by induction hypothesis. Further, this matching is unique. This is true because any such matching must match $n-1$ with n, and the matching in $S \setminus \{n-1, n\}$ is unique by induction hypothesis. Using this bijection between the matchings of $\mathcal{S}_{++}$ and $\mathcal{R}_{-}$, we derive that

$$\sum_{S \in \mathcal{S}_{++}} \prod_{i \in S} x_i \alpha_S = x_{n-1} x_n \alpha_{n-1,n} \sum_{S \in \mathcal{R}_{-}} \prod_{i \in S} x_i \alpha_S \ .$$

Further, notice that there is a bijection between $\mathcal{S}_{+-}$ and $\mathcal{R}_{+}$, which takes $S \in \mathcal{S}_{+-}$ to $S \setminus \{n-1\} \cup \{p\}$. Further, to argue that each $S \in \mathcal{S}_{+-}$ contains a unique closest-relative matching, it is easy to see that there is a one-to-one correspondence between the closest-relative matchings of $S \setminus \{n-1\} \cup \{p\}$ and that of S. Indeed, for any closest-relative matching of $S \setminus \{n-1\} \cup \{p\}$, we can obtain a closest relative matching of S by replacing p with $n-1$, and vice versa. Notice that the path from $n-1$ to any other vertex can be obtained from the path from p to that vertex by adding the edge $(n-1, p)$ in the beginning. Hence, for any leaf i we have $\alpha_{n-1,i} = \theta_{n-1,p} \alpha_{p,i}$. In particular, this implies that $\alpha_S = \theta_{n-1,p} \alpha_{S \setminus \{n-1\} \cup \{p\}}$. Summing over $S \in \mathcal{S}_{+-}$, we get

$$\sum_{S \in \mathcal{S}_{+-}} \prod_{i \in S} x_i \alpha_S = x_{n-1} \theta_{n-1,p} \sum_{S \in \mathcal{R}_{+}} \prod_{i \in S \setminus \{p\}} x_i \alpha_S \ .$$

Similarly, the sets $S \in \mathcal{S}_{-+}$ also have a unique closest-relative matching, and

$$\sum_{S \in \mathcal{S}_{-+}} \prod_{i \in S} x_i \alpha_S = x_n \theta_{n,p} \sum_{S \in \mathcal{R}_{+}} \prod_{i \in S \setminus \{p\}} x_i \alpha_S \ .$$

Using the above expressions, we can complete the proof, using the induction hypothesis:

$$\begin{aligned}
\Pr[x_1, \ldots, x_n] &= \Pr[x_1, \ldots, x_{n-2}, y_p = 1] \Pr[x_{n-1}, x_n | y_p = 1] \\
&\quad + \Pr[x_1, \ldots, x_{n-2}, y_p = -1] \Pr[x_{n-1}, x_n | y_p = -1] \\
&= \frac{\sum_{S \in \mathcal{R}_{-}} \prod_{i \in S} x_i \alpha_S + \sum_{S \in \mathcal{R}_{+}} \prod_{i \in S \setminus \{p\}} x_i \alpha_S}{2^{n-1}} \frac{1 + x_n \theta_{n,p}}{2} \frac{1 + x_{n-1} \theta_{n-1,p}}{2} \\
&\quad + \frac{\sum_{S \in \mathcal{R}_{-}} \prod_{i \in S} x_i \alpha_S - \sum_{S \in \mathcal{R}_{+}} \prod_{i \in S \setminus \{p\}} x_i \alpha_S}{2^{n-1}} \frac{1 - x_n \theta_{n,p}}{2} \frac{1 - x_{n-1} \theta_{n-1,p}}{2} \\
&= \frac{\sum_{S \in \mathcal{R}_{-}} \prod_{i \in S} x_i \alpha_S + x_n x_{n-1} \theta_{n,p} \theta_{n-1,p} \sum_{S \in \mathcal{R}_{-}} \prod_{i \in S} x_i \alpha_S}{2^n} \\
&\quad + \frac{x_n \theta_{n,p} \sum_{S \in \mathcal{R}_{+}} \prod_{i \in S \setminus \{p\}} x_i \alpha_S + x_{n-1} \theta_{n-1,p} \sum_{S \in \mathcal{R}_{+}} \prod_{i \in S \setminus \{p\}} x_i \alpha_S}{2^n} \\
&= \frac{\sum_{S \in \mathcal{S}_{--}} \prod_{i \in S} x_i \alpha_S + \sum_{S \in \mathcal{S}_{++}} \prod_{i \in S} x_i \alpha_S + \sum_{S \in \mathcal{S}_{-+}} \prod_{i \in S} x_i \alpha_S + \sum_{S \in \mathcal{S}_{+-}} \prod_{i \in S} x_i \alpha_S}{2^n} \\
&= \frac{\sum_{S \in \mathcal{S}} \prod_{i \in S} x_i \alpha_S}{2^n}
\end{aligned}$$

■

Appendix B. Proof of Theorem 1 (Same topology) and Theorem 3 (Known topology)

B.1. Proof of Theorem 1 (Same Topology)

In this Section, we provide the proof for bounding the TV distance between two models with the same topology (Theorem 1). This argument immediately implies an algorithm for TV-learning using $O(n^4/\varepsilon^2)$ samples from the leaves.

We first restate Theorem 1 for the same topology with a bit more detail.

Theorem 10 *Let T be a tree and $\alpha, \hat{\alpha} \in [-1,1]^{\binom{n}{2}}$ be two tree metrics on T. Suppose $\|\alpha - \hat{\alpha}\|_\infty \leq \varepsilon$, for some $\varepsilon > 0$. Let $\mu, \hat{\mu}$ be the corresponding distribution on the leaves of T with metric $\alpha, \hat{\alpha}$ respectively. Then,*

$$TV(\mu, \hat{\mu}) \leq 2n^2\varepsilon$$

To start, let T be a tree with n leaves. We will refer to the leaf set as $[n]$, so each number corresponds to one leaf. We define for each $x \in \{-1,1\}^n$ and for each tree topology T a function $f_x^T : [-1,1]^{\binom{n}{2}} \mapsto \mathbb{R}$ as

$$f_x^T(\alpha) := \frac{\sum_{\text{even subsets } S \subseteq [n]} \alpha_S^T \prod_{i \in S} x_i}{2^n} \tag{8}$$

Notice the similarity of this expression with the probability distribution of the leaves. However, this is a multilinear function that is defined for any vector $\alpha \in [-1,1]^{\binom{n}{2}}$, which might not necessarily arise from a tree metric on the leaves. This motivates the following definition. For a vector $\alpha \in [-1,1]^{\binom{n}{2}}$, we say that α is *induced* by a metric in T if there exists an assignment θ_e of weights for each edge e of T, such that for all leaves i, j

$$\alpha_{ij} = \prod_{e \in P_{ij}} \theta_e$$

In that case, we will refer to α as a *tree metric* on T. Now, bounding $TV(\mu, \hat{\mu})$ essentially amounts to bounding

$$\sum_{x \in \{-1,1\}^n} |\mu(x) - \hat{\mu}(x)| = \sum_{x \in \{-1,1\}^n} |f_x(\alpha) - f_x(\hat{\alpha})|$$

Thus, the problem amounts to bounding the Lipschitzness of f_x. We will bound this quantity by substituting one by one the coordinates of α with $\hat{\alpha}$. We first introduce some relevant definitions. We will need a total ordering of the pairs (i,j) of leaves. The precise ordering doesn't matter, but for simplicity let's say we pick the lexicographic order. This means that $(i,j) < (k,l)$ if and only if $i < k$ or $i = k$ and $j < l$. We use the notation $(i,j) \leq (k,l)$ as a substitute for $(i,j) < (k,l)$ or $(i,j) = (k,l)$. Suppose we order all pairs of leaves in lexicographic order. Then, we denote the t-th pair in this order as (i_t, j_t). For each $0 \leq t \leq \binom{n}{2}$, we define the vector $\alpha^t \in [-1,1]^{\binom{n}{2}}$ as

$$\alpha_{kl}^t = \begin{cases} \hat{\alpha}_{kl} \text{ , if } (k,l) \leq (i_t, j_t) \\ \alpha_{kl} \text{ , otherwise} \end{cases}$$

For $t = 0$, the convention is that $\alpha^t = \alpha$. Notice that $\alpha^{\binom{n}{2}} = \hat{\alpha}$. Also, denote $T \setminus \{i,j\}$ the topology that is obtained from T if we remove all edges on the path P_{ij} from T.

We first prove a Lemma about what happens to the expression of $f_x^T(\alpha)$ if we change exactly one coordinate of α. This is a purely combinatorial statement that relies on the structure of the coefficients α_S.

Lemma 11 *Let T be a tree and i, j two leaves of T. Also, let $\alpha, \beta \in [-1,1]^{\binom{n}{2}}$ such that $\alpha_{kl} = \beta_{kl}$ if $(k,l) \neq (i,j)$. Let $\gamma \in [-1,1]^{\binom{n}{2}}$ be defined as follows*

$$\gamma_{kl} = \begin{cases} 0 \text{ , if } P_{ij} \text{ and } P_{kl} \text{ have common edges} \\ \alpha_{kl} \text{ , otherwise} \end{cases}$$

Then,

$$f_x^T(\alpha) - f_x^T(\beta) = x_i x_j (\alpha_{ij} - \beta_{ij}) f_x^{T \setminus \{i,j\}}(\gamma) \tag{9}$$

Proof We have

$$f_x^T(\alpha) - f_x^T(\beta) = \frac{\sum_{\text{even subsets } S \subseteq [n]} \left(\alpha_S^T - \beta_S^T\right) \prod_{i \in S} x_i}{2^n}$$

We first notice that if $\{i,j\}$ is not a subset of S, then α_{ij}, β_{ij} will not appear in α_S^T, β_S^T respectively. This means that $\alpha_S^T = \beta_S^T$, since α, β agree on the rest of the coordinates. Hence, we focus on the collection of subsets

$$\mathcal{S}_1 := \{S \subseteq [n] : \{i,j\} \subseteq S \text{ and } |S| \text{ even}\}$$

Suppose $v_0 = i, v_1, \ldots, v_k = j$ is the path connecting i, j in T. Since each non-leaf node has degree 3, each one of the nodes $v_1, \ldots, v_{k-1}$ has exactly one other neighbor outside of the path. We can view this as each v_i being the root of some subtree T_i that starts from the neighbor of v_i that is outside of the path. Let A_i be the set of leaves on subtree T_i. Notice that the sets A_i partition $[n] \setminus \{i,j\}$. Figure 3 depicts these sets of leaves along the path.

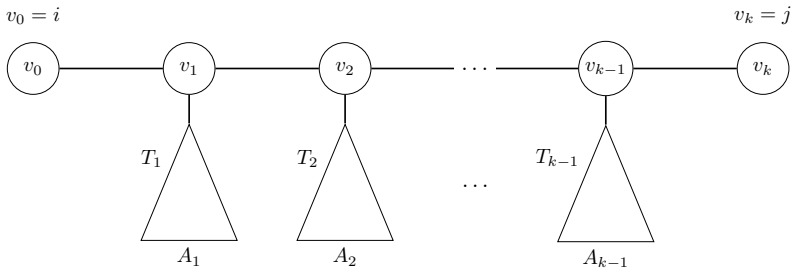


Figure 3: The figure shows the path connecting i to j and the subtrees that will become connected components if we remove this path from the graph.

We now want to determine which elements of the family $\mathcal{S}_1$ of subsets gives different coefficients for α, β. We first show that if for some $S \in \mathcal{S}_1$ we have $|S \cap A_r|$ being odd for some $0 < r < k$, then $\alpha_S^T = \beta_S^T$. The reason is the following: suppose there exists r_0 with $|S \cap A_{r_0}|$ being odd. Suppose also without loss of generality that this is the smallest r for which this property holds. This means that for $r < r_0$ we have $|S \cap A_r|$ is even. Thus, by the matching process described in Section A, it is clear that for each $r < r_0$, the leaves in $S \cap A_r$ will be matched in pairs inside the tree T_r and not with some leaves outside of the tree. Also, since $|S \cap A_{r_0}|$ is odd, the leaves in $S \cap A_{r_0}$ will be matched with each other, except one leaf, call it w, which will be left unmatched. Then, the matching process dictates that w should be matched with i. Hence, i will not be matched with j for this subset S. An example of this situation can be seen in Figure 4.

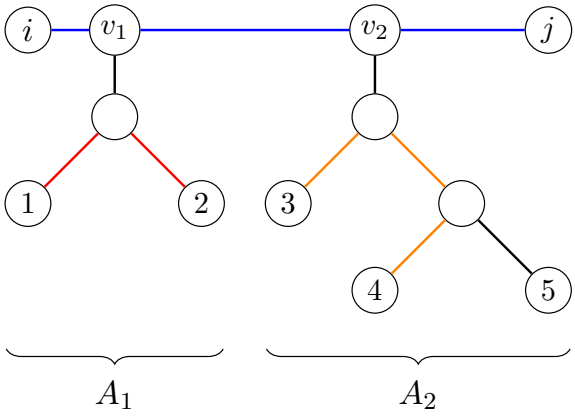


Figure 4: In this example, we have the subset $S = \{i, j, 1, 2, 3, 4\}$. Notice that $|S \cap A_1|, |S \cap A_2|$ are even. Clearly, the closest relative matching is $(i, j), (1, 2), (3, 4)$. If we had the set $S' = \{i, 1, 3, j\}$, then the matching would be $(i, 1), (3, j)$.

Hence, in α_S^T we will have the factor α_{iw} instead of α_{ij}. This means that α_{ij} does not appear in α_S^T and similarly β_{ij} does not appear in β_S^T. But this implies that $\alpha_S^T = \beta_S^T$, since α, β agree on all the other coordinates. This proves our claim.

Hence, if we define the set

$$\mathcal{S}_2 := \{\{i, j\} \cup \left(\cup_{r=1}^{k-1} S_r\right) : S_r \subseteq A_r, |S_r| \text{ even, for all } r\}$$

then

$$f_x^T(\alpha) - f_x^T(\beta) = \frac{1}{2^n} \sum_{S \in \mathcal{S}_2} \left(\alpha_S^T - \beta_S^T\right) \prod_{i \in S} x_i$$

Since S_r has an even number of leaves, they will be matched inside tree T_r, regardless of the topology of the rest of the tree. This leaves i, j, which will be matched together. This enables us to write

$$\alpha_S = \alpha_{ij} \prod_{r=1}^{k-1} \alpha_{S_r}^{T_r} \quad , \quad \beta_S = \beta_{ij} \prod_{r=1}^{k-1} \alpha_{S_r}^{T_r}$$

The reason we wrote α_{S_r} in the expression of β_S is that α, β agree on all coordinates other than ij. Hence,

$$f_x^T(\alpha) - f_x^T(\beta) = (\alpha_{ij} - \beta_{ij}) x_i x_j \left(\frac{1}{4} \prod_{r=1}^{k-1} \underbrace{\frac{1}{2^{|A_r|}} \sum_{S_r \subseteq A_r, |S_r| \text{even}} \alpha_{S_r}^{T_r} \prod_{u \in S_r} x_u}_{f_{x_{A_r}}^{T_r}(\alpha)} \right)$$

$$= (\alpha_{ij} - \beta_{ij}) x_i x_j f_x^{T \setminus \{i,j\}}(\gamma)$$

Let us explain the last equality. The r-th element in this product corresponds to the expression of the probability distribution on the leaves A_r of the subtree T_r, with pairwise correlations that are given by α (we are slightly abusing notation when we pass α as an argument in $f_x^{T_r}$, since it is the vector of pairwise correlations for the whole tree). Hence, the product of these terms is the expression of a product probability distribution over the subsets A_r and i, j being independent from everyone else (this is the $1/4$ term). This is exactly the expression of the distribution on $T \setminus \{i, j\}$ with correlations given by α, except for pairs of leaves that belong to different subtrees, which have correlation 0. This is exactly how we defined γ, hence the result follows.

■

Lemma 11 tells us how much f_x changes when we change one coordinate, corresponding to some pair (i, j). Hence, our efforts will now be focused on bounding the expression on the RHS of (9). If this expression corresponds to a distribution on $T \setminus \{i, j\}$, then this term can easily be bounded. However, as we will see, that will not always be the case and we need to be more careful. The following Lemma contains the Lipschitzness property that we would like to prove.

Lemma 12 *Let T be a tree and $\alpha, \hat{\alpha} \in [-1, 1]^{\binom{n}{2}}$, where α is a metric on T. Suppose that $\|\alpha - \hat{\alpha}\|_\infty \leq \varepsilon/(2n^2)$, where $\varepsilon \in (0, 1)$. Then, for any $t \geq 1$*

$$\sum_{x \in \{-1,1\}^n} \left| f_x^T(\alpha^t) - f_x^T(\alpha^{t-1}) \right| \leq \frac{\varepsilon}{n^2} \tag{10}$$

Proof We will prove (10) via induction on t. First, we define for all $s, t \leq \binom{n}{2}$ the vector

$$\gamma_{kl}^{t,s} = \begin{cases} 0 \text{ , if } P_{i_t j_t} \text{ and } P_{kl} \text{ have common edges} \\ \alpha_{kl}^s \text{ , otherwise} \end{cases}$$

The base case $t = 1$ corresponds to the pair of leaves $(1, 2)$. Since α^1 and $\alpha^0 = \alpha$ differ only in the pair $(1, 2)$, by Lemma 11 we have

$$\sum_{x \in \{-1,1\}^n} \left| f_x^T(\alpha^1) - f_x^T(\alpha) \right| \leq |\alpha_{12} - \hat{\alpha}_{12}| \sum_{x \in \{-1,1\}^n} \left| f_x^{T \setminus \{1,2\}}(\gamma^{1,0}) \right|$$

Now, we notice that $f_x^{T \setminus \{1,2\}}(\gamma^{1,0})$ is actually the probability distribution on T that results from α if we set $\theta_e = 0$ for all $e \in P_{12}$. Hence, we can remove the absolute value and get

$$\sum_{x \in \{-1,1\}^n} \left| f_x^T(\alpha^1) - f_x^T(\alpha) \right| \leq |\alpha_{12} - \hat{\alpha}_{12}| \sum_{x \in \{-1,1\}^n} f_x^{T \setminus \{1,2\}}(\gamma^{1,0}) = |\alpha_{12} - \hat{\alpha}_{12}| \leq \frac{\varepsilon}{2n^2}$$

Hence, the base case of the induction is proven.

Now, suppose we have proved the claim for all $t' < t$.

By applying Lemma 11, we again obtain

$$\sum_{x \in \{-1,1\}^n} \left| f_x^T(\alpha^t) - f_x^T(\alpha^{t-1}) \right| \leq |\alpha_{i_t j_t} - \hat{\alpha}_{i_t j_t}| \sum_{x \in \{-1,1\}^n} \left| f_x^{T \setminus \{i_t, j_t\}}(\gamma^{t,t-1}) \right| \tag{11}$$

Now, the problem is that $\gamma^{t,s}$ contains some coordinates of α and some coordinates of $\hat{\alpha}$. As a result, the expression $f_x^T(\gamma^t)$ is not necessarily a probability distribution anymore. We will try to relate this quantity to a true distribution.

Essentially, $\gamma^{t,s}$ is the same as α^s, except for pairs of leaves that belong to different components of $T \setminus \{i_t, j_t\}$. Now, we can write

$$\sum_{x\in\{-1,1\}^n} \left| f_x^{T\setminus\{i_t,j_t\}}(\gamma^{t,s}) \right|$$
$$\leq \sum_{x\in\{-1,1\}^n} \left| f_x^{T\setminus\{i_t,j_t\}}(\gamma^{t,0}) \right| + \sum_{s=1}^{t-1} \sum_{x\in\{-1,1\}^n} \left| f_x^{T\setminus\{i_t,j_t\}}(\gamma^{t,s}) - f_x^{T\setminus\{i_t,j_t\}}(\gamma^{t,s-1}) \right|$$

First of all, we notice that the expression $f_x^{T\setminus\{i_t,j_t\}}(\gamma^{t,0})$ is the expression of a probability distribution on T, which is obtained from α by setting $\theta_e = 0$ for all edges $e \in P_{i_t j_t}$. Hence, the first term of the RHS sums up to 1. As for the second sum, each term of the outer sum has exactly the form of (10), where the starting α is $\gamma^{t,0}$ and we have substituted at most $t-1$ with $\hat{\alpha}$, since $s \leq t-1$. Hence, we can apply the inductive assumption to get

$$\sum_{s=1}^{t-1} \sum_{x\in\{-1,1\}^n} \left| f_x^{T\setminus\{i_t,j_t\}}(\gamma^{t,s}) - f_x^{T\setminus\{i_t,j_t\}}(\gamma^{t,s-1}) \right| \leq (t-1)\frac{\varepsilon}{n^2} \leq \varepsilon$$

since $t \leq \binom{n}{2} \leq n^2$. Overall, this gives

$$\sum_{x\in\{-1,1\}^n} \left| f_x^{T\setminus\{i_t,j_t\}}(\gamma^{t,s}) \right| \leq 1 + \varepsilon$$

Now, plugging this in (11) gives us

$$\sum_{x\in\{-1,1\}^n} \left| f_x^T(\alpha^t) - f_x^T(\alpha^{t-1}) \right| \leq |\alpha_{i_t j_t} - \hat{\alpha}_{i_t j_t}|(1+\varepsilon) \leq \frac{\varepsilon}{n^2}$$

since $\varepsilon < 1$. Thus, the inductive step is complete and the claim is proved. ■

We are now ready to prove Theorem 10.

Proof [Proof of Theorem 10] Let $\varepsilon' = 2n^2\varepsilon$. We divide into cases.

Case 1: Suppose $\varepsilon' < 1$. Then, Lemma 12 applies and we get

$$TV(\mu,\hat{\mu}) = \sum_{x\in\{-1,1\}^n} \left| f_x^T(\alpha) - f_x^T(\hat{\alpha}) \right| \leq \sum_{t=1}^{\binom{n}{2}} \sum_{x\in\{-1,1\}^n} \left| f_x^T(\alpha^t) - f_x^T(\alpha^{t-1}) \right|$$
$$\leq \sum_{t=1}^{\binom{n}{2}} \frac{\varepsilon'}{n^2} \leq \varepsilon' = 2n^2\varepsilon$$

Case 2: Suppose $\varepsilon' \geq 1$. This means that

$$TV(\mu, \hat{\mu}) \leq 1 \leq 2n^2\varepsilon$$

so the claim is trivial in that case.

■

B.2. Proof of Theorem 3 (Known Topology)

We can now also conclude the proof of Theorem 3 for known topology, which we sketched in earlier Sections.

Proof [Proof of Theorem 3 (Known topology)] Let $\alpha \in [-1,1]^n$ denote the vector of correlations of leaves in the model we are trying to learn and μ denote the distribution on the leaves for this model. Let us consider the sample mean obtained from m independent samples $x^{(1)}, \ldots, x^{(m)}$.

$$\hat{\alpha}_{ij} = \sum_{k=1}^{m} \frac{x_i^{(k)} x_j^{(k)}}{m}$$

By standard Chernoff bounds, we know that with probability at least $1-\delta$, for all leaves i, j

$$|\alpha_{ij} - \hat{\alpha}_{ij}| \leq \sqrt{\frac{2\log(n^2/\delta)}{m}} := \eta \tag{12}$$

We run Algorithm 1 with this η parameter.

First, we claim that the LP that Algorithm 1 solves has a feasible solution for this choice of η. To show that, we will construct a feasible solution of the program. If $\theta \in [-1,1]^{|E|}$ is the vector of the edge weights of the model we are trying to learn, then for all leaves i, j

$$\alpha_{ij} = \prod_{(k,l)\in P_{ij}} \theta_{kl}$$

This implies that

$$|\alpha_{ij}| = \prod_{(k,l)\in P_{ij}} |\theta_{kl}|$$

Thus, if we set $w_{kl} = \ln|\theta_{kl}|$ for all edges (k,l), we have that

$$\sum_{(k,l)\in P_{i,j}} w_{kl} = \ln\left(\prod_{(k,l)\in P_{ij}} |\theta_{kl}|\right) = \log|\alpha_{ij}|$$

We know that with probability at least $1-\delta$

$$||\alpha_{ij}| - |\hat{\alpha}_{ij}|| \leq |\alpha_{ij} - \hat{\alpha}_{ij}| \leq \eta$$

which implies by the previous observations that

$$\log(|\hat{\alpha}_{ij}| - \eta) \leq \sum_{(k,l)\in P_{i,j}} w_{kl} \leq \log(|\hat{\alpha}_{ij}| + \eta)$$

Hence, the inequality constraint for feasibility is satisfied. Hence, this is a feasible solution for the program.

Let $\tilde{\theta}_{kl}$ be the edge weights that are returned by the LP and $\tilde{\alpha} \in [0,1]^{\binom{n}{2}}$ be the pairwise correlations that are induced by these weights. We need to figure out the correct signs for each θ_{kl}. Let $s_{kl} \in \{-1,1\}$ be a sign variable for each edge (k,l). Also, let $s(\alpha)_{ij} \in \{-1,1\}$ be the sign of α_{ij} and likewise define $s(\alpha_{ij})$. If we find an assignment of the s_{kl} variables such that

$$\prod_{(k,l)\in P_{ij}} s_{kl} = s(\alpha_{ij})$$

for all pairs i,j, then it follows that

$$\left|\prod_{(k,l)\in P_{ij}} s_{kl}\tilde{\theta_{kl}} - \alpha_{ij}\right| = \left|\prod_{(k,l)\in P_{ij}} s_{kl} \prod_{(k,l)\in P_{ij}} \tilde{\theta_{kl}} - s(\alpha_{ij})|\alpha_{ij}|\right| = \left|\prod_{(k,l)\in P_{ij}} \tilde{\theta_{kl}} - |\alpha_{ij}|\right| \leq \eta \tag{13}$$

Thus, we will now focus on finding such s_{kl} and the output of the algorithm will be $\overline{\theta_{kl}} = s_{kl}\tilde{\theta_{kl}}$. First of all, we need a way to figure out $s(\alpha_{ij})$ for all i,j. Let $U = \{(i,j) : |\hat{\alpha}_{ij}| > \eta\}$. By the approximation guarantee $|\alpha_{ij} - \hat{\alpha_{ij}}| < \eta$, we conclude that for all $(i,j) \in U$, $s(\alpha_{ij}) = s(\hat{\alpha_{ij}})$. Hence, we build up the system of equations

$$\prod_{(k,l)\in P_{ij}} s_{kl} = s(\hat{\alpha_{ij}}) \text{ for all } (i,j) \in U \tag{14}$$

This can be viewed as a system of linear equations in $\mathbf{F}_2$, which is the field with 2 elements. Hence, we can use the standard Gaussian elimination algorithm to solve it. Since $s(\hat{\alpha_{ij}}) = s(\alpha_{ij})$ for all $(i,j) \in U$, we know that this system has at least one solution, namely setting s_{kl} to be the sign of θ_{kl} in the true model. Let $\tilde{s}$ be the solution that is returned by the Gaussian elimination algorithm. There might be many solutions, since it's possible that the system is underdetermined, which could be cause by the absence of some equations for $(i,j) \notin U$. But in any case, we know that $\tilde{s}$ satisfies (14).

Now, we set $\overline{\theta_{kl}} = \tilde{s_{kl}}\tilde{\theta_{kl}}$ and let $\overline{\alpha} \in [-1,1]^{\binom{n}{2}}$ be the correlations that are induced by $\overline{\theta}$, namely $\overline{\alpha_{ij}} = \tilde{\alpha_{ij}} \prod_{(k,l)\in P_{ij}} s_{kl}$. If $(i,j) \in U$, then (14) holds, which means that by (13) we have $|\overline{\alpha_{ij}} - \alpha_{ij}| \leq \eta$. If $(i,j) \notin U$, then $|\hat{\alpha}_{ij}| \leq \eta$ implies $|\alpha_{ij}| \leq 2\eta$ and $|\tilde{\alpha_{ij}}| = |\overline{\alpha_{ij}}| \leq 2\eta$. Thus, $|\overline{\alpha_{ij}} - \alpha_{ij}| \leq 4\eta$.

Next, we argue about the TV distance between the distribution $\overline{\mu}$ that is induced on the leaves by the output $\overline{\alpha}$ of the algorithm and μ. We just proved that for all i,j

$$|\alpha_{ij} - \overline{\alpha_{ij}}| \leq 4\eta$$

We can then apply Theorem 10, which gives

$$TV(\mu,\tilde{\mu}) \leq 8n^2\eta = 8n^2\sqrt{\frac{2\log(n^2/\delta)}{m}}$$

To make this quantity smaller than ε, we need

$$m = \Theta\left(\frac{n^4\log(n/\delta)}{\varepsilon^2}\right)$$

samples.

■

Appendix C. Proof of Theorem 1 (Different topologies)

Let us start with some definitions. Let T be a tree with n leaves and where all non-leaf nodes have degree 3. For a vector $\alpha \in [-1,1]^{\binom{n}{2}}$, we say that α is *induced* by a metric in T if there exists an assignment $\theta_e \in [-1,1]$ of weights for each edge e of T, such that for all leaves i, j

$$\alpha_{ij} = \prod_{e \in P_{ij}} \theta_e$$

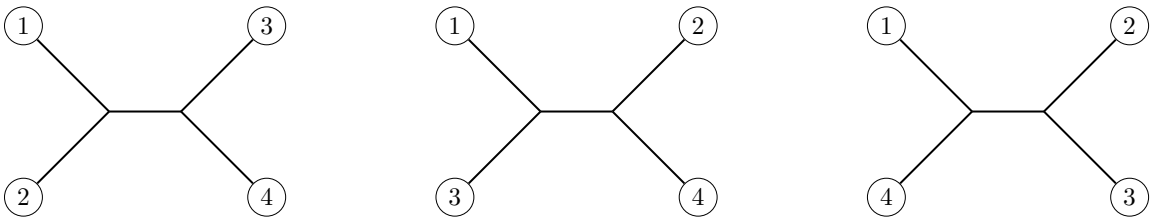
If μ is the distribution of the leaves of T when the vector of pairwise distances between leaves is α, we say that μ is specified by the pair (T, α). For an even subset $S \subseteq [n]$, we denote α_S^T the coefficient of $\prod_{i \in S} x_i$ in the Fourier expansion of the probability distribution on a tree T. In this Section, we prove the following Theorem, which is a restatement of Theorem 1 (different topologies).

Theorem 13 *Let $T, \hat{T}$ denote the topologies of two trees with n leaves, where each non-leaf node has degree 3. Let $\alpha, \hat{\alpha} \in [-1,1]^{\binom{n}{2}}$ denote vectors of pairwise distances that are induced by some tree metric on T and $\hat{T}$, respectively. Let also $\hat{\mu}$ be the distribution that is induced on the leaves by $(\hat{T}, \hat{\alpha})$ and μ the one induced by (T, α). Finally, let D be the minimum diameter of $T, \hat{T}$. Suppose that for all leaves $i \neq j$ we have that $|\alpha_{ij} - \hat{\alpha}_{ij}| \leq \varepsilon$. Then,*

$$TV(\mu, \hat{\mu}) \leq Cn^5 D\varepsilon \tag{15}$$

where C is an absolute constant.

For four leaves i, j, k, l, we will denote a quartet of leaves by $\{i, j, k, l\}$ when we do not wish to specify their relative placement. The following fact is folklore: if we contract all edges that do not belong to some path between two leaves in the quartet, then we might end up with one of three possible topologies. We call this the topology of the quartet $\{i, j, k, l\}$. The 3 topologies are shown in Fig. 5. For example, if we are in the first topology, we write $\{(12)(34)\}$ to denote that fact. We might refer to the quartet as either $\{1, 2, 3, 4\}$ of $\{(12)(34)\}$, depending on whether we want to highlight the topology of the quartet or not.



Possibility 1: $\{(12)(34)\}$ Possibility 2: $\{(13)(24)\}$ Possibility 3: $\{(14)(23)\}$

Figure 5: The three possible topologies for a quartet. In Possibility 1, the path from 1 to 2 does not intersect the path from 3 to 4. Further, $\alpha_{12}\alpha_{34} \geq \alpha_{13}\alpha_{24} = \alpha_{14}\alpha_{23}$ (and similarly for Possibilities 2 and 3).

As explained in Fig. 5, what distinguishes the topology is the relative order between the products $\alpha_{12}\alpha_{34}, \alpha_{13}\alpha_{24}, \alpha_{14}\alpha_{23}$. When α is induced by some tree metric, then two of these products will

always be equal and the third will be larger or equal. Depending on which of the products is larger, we get one of the tree possible topologies (if all products are equal then we will choose the topology arbitrarily). Hence, if for some reason we cannot distinguish which of the three products is larger, then we intuitively expect that it will be hard to find the correct topology for a quartet. We give the relevant definitions below.

Definition 14 *Let i, j, k, l be a quartet of four leaves of T. We define*

$$\Delta_{ijkl}(\alpha) := \max(\alpha_{ij}\alpha_{kl}, \alpha_{ik}\alpha_{jl}, \alpha_{il}\alpha_{jk}) - \min(\alpha_{ij}\alpha_{kl}, \alpha_{ik}\alpha_{jl}, \alpha_{il}\alpha_{jk})$$

Definition 15 *Let i, j, k, l be a quartet of four leaves of T. We say that this is an ε-*good quartet *w.r.t. some vector α if*

$$\Delta_{ijkl}(\alpha) > \varepsilon$$

A quartet of leaves that is not an ε-good quartet is called an *ε-bad quartet*. Intuitively, if a quartet is good, then it is easy for an algorithm to distinguish which is the correct topology of these four leaves out of the three possibilities. If it is bad, then the topology is very close to being a star and so all three possibilities are roughly equivalent. One thing to note is that if $|\hat{\alpha} - \alpha^*| \leq \varepsilon$, then

$$|\Delta_{ijkl}(\hat{\alpha}) - \Delta_{ijkl}(\alpha^*)| \leq 2\varepsilon$$

for all i, j, k, l. Hence, it does not really make a difference whether a quartet is good or bad with respect to $\hat{\alpha}$ or α^*, since there is only an $O(\varepsilon)$-additive error.

We now proceed with the proof of Theorem 13. First, we need to formally define some notions of cutting and pasting nodes in different parts of the tree. This will prove useful in having a unified vocabulary when describing the process of interpolating between two trees.

Definition 16 *Let $T = G(V, E)$ denote the topology of a tree where every node has degree at most 3. We define* BINARY(T) *to be the tree that is obtained from T by contracting all maximal paths of nodes of degree 2 into a single edge. In other words, it is obtained if we succesively find a degree 2 node u with edges $(u, v), (u, w)$ and replace it with edge (v, w), until we cannot find such a node.*

Notice that the output of BINARY is also described in Definition 23. For an example of how BINARY works, see Fig. 6. Clearly, BINARY(T) satisfies the property that every non-leaf node has degree 3.

Definition 17 *Let $T = G(V, E)$ denote the topology of a tree and suppose $(u, v) \in E$. Let $(r, s) \in E$ be some other edge of T, where we might have $\{r, s\} \cap \{u, v\} \neq \emptyset$. The only requirement is that r, s belong to a different component than u when we remove edge (u, v) from T. We define* CUTPASTE$(T, u, v, (r, s))$ *to be the tree that is obtained from T as follows: we delete edges (u, v) and (r, s), we add a node t and we add the edges $(t, u), (t, r), (t, s)$. This produces a tree T'. We set* CUTPASTE$(T, u, v, (r, s))$ = BINARY(T').

An example of applying CUTPASTE can be seen in Fig. 7.

Intuitively, CUTPASTE encodes the following process: we delete edge (u, v), we add a node t in the middle of edge (r, s) and we attach u together with it's connected component to t. This is why the order of u, v as arguments of CUTPASTE is important, while the order of r, s is not.

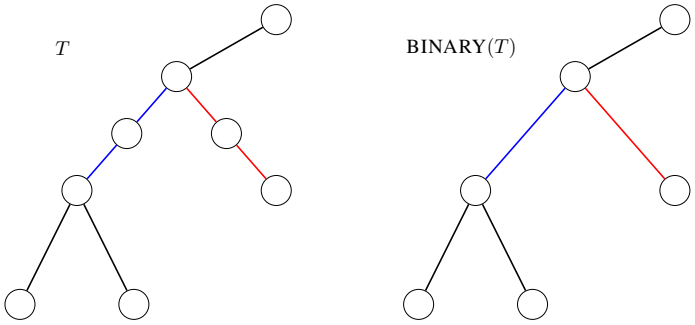


Figure 6: On the left we have the tree before the contraction. On the right, we have the tree after applying BINARY. We have highlighted with similar colors the path that is contracted on the left and the final edge on the right.

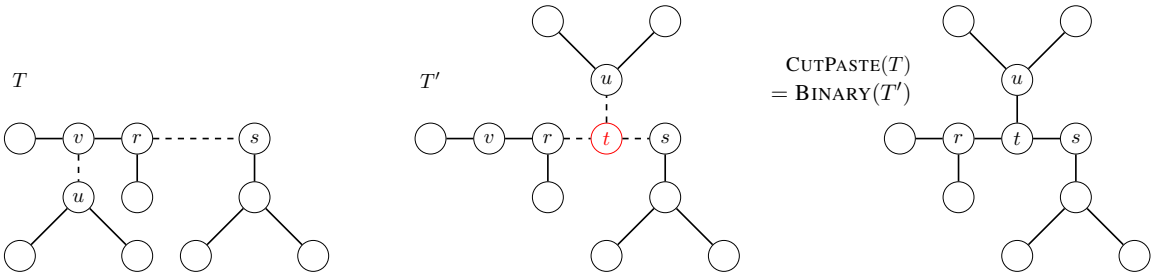


Figure 7: The Figure shows the output of CUTPASTE$(T, u, v, (r, s))$. In the first step we cut u from it's place and paste it in the middle of edge (r, s). In the second step we contract paths of degree 2 nodes.

Before describing the interpolation process between the two trees, we prove a lemma about CUTPASTE. It shows what changes in the distribution if we change the tree according to CUTPASTE. For a tree T and a quartet $\{w, z, y, u\}$, let's denote by $T \setminus \{w, z, y, u\}$ the graph that we get if we remove all paths of the quartet $\{w, z, y, u\}$ from T. An example is given in Fig. 1(b).

Lemma 18 *Let $T = (V, E)$ be a tree and $\alpha \in [-1, 1]^{\binom{n}{2}}$. Let i, j be two nodes in V(leaf or non-leaf) and denote $i = v_0, v_1, \ldots, v_m = j$ to be the nodes in the path that connects them in the tree, with $m \geq 3$. Define $T^k = \text{CUTPASTE}(T, i, v_1, (v_k, v_{k+1}))$ for all $0 < k < m$. Denote by U_k the set of quartets where T^k and T^{k+1} differ. Lastly, let's define the vector $\alpha^{wzyu} \in [-1, 1]^{\binom{n}{2}}$ as*

$$\alpha_{kl}^{wzyu} = \begin{cases} \alpha_{kl} & \text{, if the path } P_{kl} \text{ has no common edges with any paths of the quartet } \{w, z, y, u\} \\ 0 & \text{, otherwise} \end{cases}$$

Then,

$$f_x^{T^k}(\alpha) - f_x^{T^{k+1}}(\alpha) = \sum_{(w,z,y,u) \in U_k} \left(\alpha_{\{w,z,y,u\}}^{T^k} - \alpha_{\{w,z,y,u\}}^{T^{k+1}} \right) x_w x_z x_y x_u \cdot f_x^{T^k \setminus \{w,z,y,u\}}(\alpha^{wzyu})$$

where $f_x^T(\alpha)$ was defined in (2).

Proof Let I_i denote the subset of leaves that lie on the subtree where i belongs to if we delete edge (i, v_1) from the tree, and analogously we define I_j as the set of leaves of the subtree where j belongs to after this removal. By definition, we have

$$\begin{aligned} f_x^{T_k}(\alpha) - f_x^{T^{k+1}}(\alpha) &= \frac{1}{2^n} \sum_{S \subseteq [n], |S| even} (\alpha_S^{T^k} - \alpha_S^{T^{k+1}}) \prod_{u \in S} x_u \\ &= \frac{1}{2^n} \sum_{S \subseteq [n], |S| even, S \cap I_i \neq \emptyset} (\alpha_S^{T^k} - \alpha_S^{T^{k+1}}) \prod_{u \in S} x_u \end{aligned}$$

The last equality follows because the relative topology of the leaves $[n] \setminus I_i$ does not change, which means the coefficients α_S for $S \subseteq [n] \setminus I_i$ also do not change. Now, for $0 < k < m$ let us define S_k to be the subset of leaves on the connected component that v_k belongs to, if we remove all edges of the path P_{ij} from the graph. Let's also define $L_k = \cup_{q=1}^{k} S_q, R_k = \cup_{q=k}^{m-1} S_q \cup I_j$. We will characterize the set of quartets U_k that change from T_k to T_{k+1}. An illustration of all these concepts we just defined is given in Fig. 8.

First of all, notice that $I_i, I_j, \{S_q\}_{q=1}^{m-1}$ partitions the set of leaves. It is straightforward to see that

$$U_k = \{\{w, z, y, u\} : w \in I_i, z \in L_k, y \in S_{k+1}, u \in R_{k+2}\}$$

Now let's fix a quartet $\{w, z, y, u\} \in U_k$. We would like to characterize the even subsets $S \supset \{w, z, y, u\}$ such that

$$\alpha_S^{T^k} = \alpha_{\{w,z,y,u\}}^{T^k} \alpha_{S \setminus \{w,z,y,u\}}^{T^k}$$

Denote by $\mathcal{S}_{w,z,y,u}$ this collection of subsets. Essentially, these are the subsets were the matchings happen so that w, z and y, u are matched together. The reason we are interested in these subsets is that these are exactly the subsets where $\alpha_S^{T^k}$ and $\alpha_S^{T^{k+1}}$ will be different (once we enumerate over all

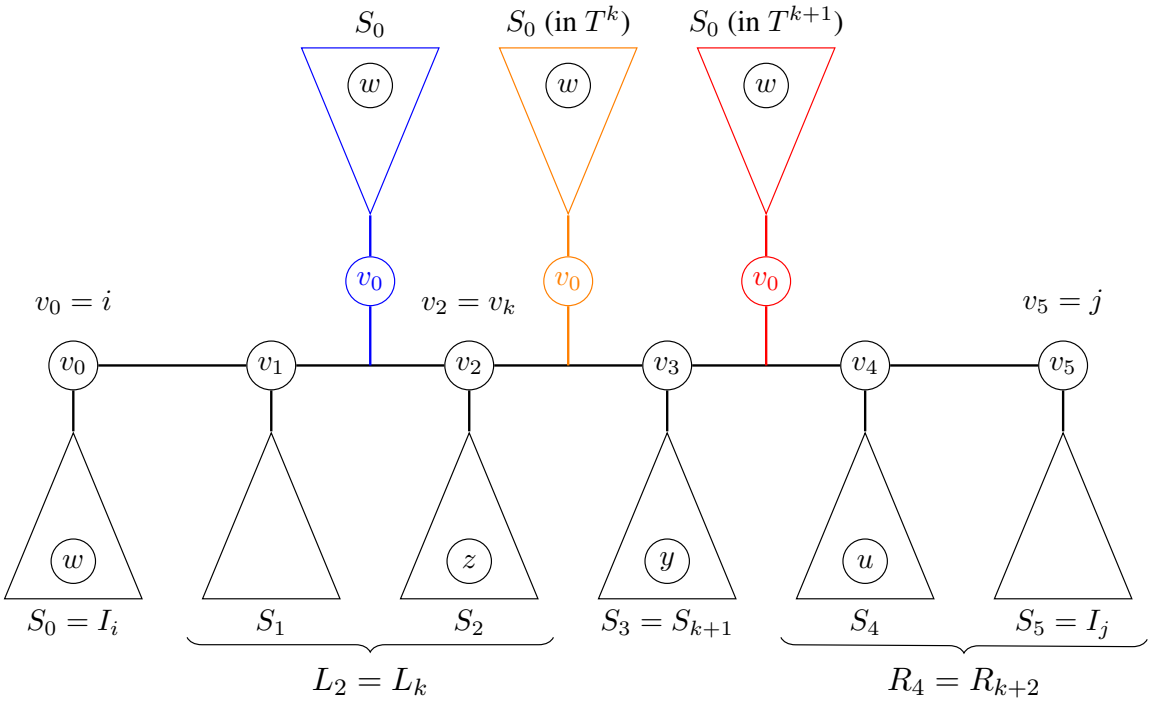


Figure 8: This is an illustration of how CUTPASTE cuts i from a place and moves it along the path to $v_5 = j$, one step at a time. This is also exactly the same movement that is done by Algorithm 3, where one move corresponds to moving v_0 one step to the right. Note that quartet $\{w, z, y, u\}$ is changed when we move v_0 from the left of v_3 to the right of v_3. For illustration we denote $k = 2$ and we depict the movement from tree T^k to T^{k+1}.

quartets $\{w, z, y, u\}$ in U_k) Our strategy to understand how these sets look like will be similar to the one employed in the proof of Theorem 11. In particular, let us consider removing the paths of the quartet $\{w, z, y, u\}$ from T^k. This leaves us with a collection of connected subtrees, each with a leaf set A_r. Here, r ranges from 1 to l where l is the number of these components. The set of leaves can be partitioned as

$$[n] = \{w, z, y, u\} \cup (\cup_{r \leq l} A_r)$$

It should then be clear from the figure that $S \in \mathcal{S}_{w,z,y,u}$ if and only if $|S \cap A_i|$ is even, for all i. To justify that, let's see what happens if for some r $|S \cap A_r|$ was odd. Then, there would be a leaf $b \in A_r$ that would be left unmatched in A_r. As we can see from Fig. 9, there are 5 different possible positions that b can lie in the relative topology of the quartet $\{w, z, y, u\}$. However, from these, only 4 are possible, since b cannot lie in the middle of the quartet. The reason is that by definition of $\{w, z, y, u\}$ there is no node in the middle edge of that quartet, so there is no subtree that is hanging from there.

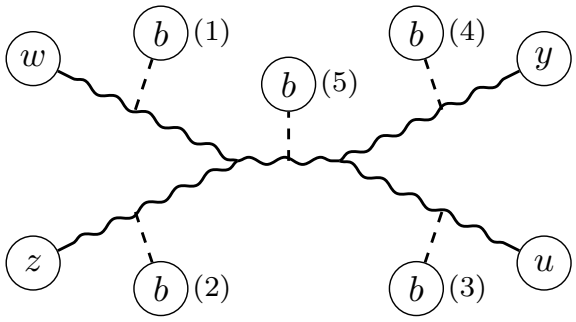


Figure 9: The 5 different placings of b relative to the quartet $\{w, z, y, u\}$. Notice that position (5) is actually not possible, as there is no node in the middle of the quartet (see Fig. 8)

Hence, b should lie closer to one of the 4 leaves. Let's assume w.l.o.g. that it lies closer to w. Then, a similar argument as in Theorem 11 applies. In particular, we can also assume w.l.o.g. that b is the closest leaf in w that is left unmatched by it's subtree A_r (otherwise we consider the closest one instead of b). Then, b has to be matched with w in $\alpha_S^{T^k}$, which means that the term α_{wz} will not appear in that expression. This means that $S \notin \mathcal{S}_{w,z,y,u}$, a contradiction. Thus, we established that

$$\mathcal{S}_{w,z,y,u} = \{\{w, z, y, u\} \cup (\cup_{r \leq l} S_r) : S_r \subseteq A_r, |S_r| \text{ even } \}$$

Note that the sets $\mathcal{S}_{w,z,y,u}$ are disjoint for different quartets $\{w, z, y, u\}$. Also, it is easy to see that $S \in \mathcal{S}_{w,z,y,u}$ if and only if

$$\alpha_S^{T^{k+1}} = \alpha_{\{w,z,y,u\}}^{T^{k+1}} \alpha_{S \setminus \{w,z,y,u\}}^{T^k}$$

Hence, the sets S such that $\alpha_S^{T^k} \neq \alpha_S^{T^{k+1}}$ are precisely the union $\cup_{\{w,z,y,u\}\in U_k}\mathcal{S}_{w,z,y,u}$. Now, notice that

$$\frac{1}{2^n}\sum_{S\in\mathcal{S}_{w,z,y,u}}(\alpha_S^{T^{k+1}}-\alpha_S^{T^k})\prod_{c\in S}x_u$$
$$=\left(\alpha_{\{w,z,y,u\}}^{T^k}-\alpha_{\{w,z,y,u\}}^{T^{k+1}}\right)x_wx_zx_yx_u\underbrace{\frac{1}{2^n}\prod_r\left(\sum_{S\subseteq A_r,|S|even}\alpha_S\prod_{c\in S\cap A_r}x_c\right)}_{f_x^{T\setminus\{w,z,y,u\}}(\alpha^{wzyu})}$$

The last equality is true, since it has the form of a product distribution over the subsets A_i, which is exactly the distribution of the topology $T\setminus\{w,z,y,u\}$. The weights in each subtree remain the same, but across subtrees the correlations are 0, which is why the argument is α^{wzyu} now. Summing over all $\{w,z,y,u\}\in U_k$ gives us the desired claim.

■

We now describe the process of interpolating between T and $\hat{T}$. We first give the pseudocode, which is Algorithm 3. We note that even though we call this process an algorithm, it will only be used as part of the Analysis of the TV distance between two trees. Hence, we are not concerned with its computational complexity.

The interpolation will be carried away in *rounds*. Each round corresponds to a run of the outer `While` loop. In the first round ($q=1$), we make sure that any two leaves that form a cherry in $\hat{T}$ will also form a cherry in T. At the end of the first round, we update the set of leaves by removing leaves that are cherries and adding their parents. Hence, in the second round, we make sure that parents of leaves that are cherries in $\hat{T}$ become also cherries in T and so on.

Let us now describe in a bit more detail what happens in each round. First of all, notice that the L in the for loop condition is evaluated at the start of the loop. This means that if we change it during the run of the loop, the number of iterations will not be affected. In the first round, this set L corresponds to the leaf set $[n]$. We proceed to search for a pair i,j that is a cherry in $\hat{T}$ but not in T. If such a pair i,j is found, the we have to move one of them towards the other to make them a cherry. This sequence of moves is called an *epoch* and corresponds to a run of the first `If` statement inside the `For`. We include an extra `If` statement since we want to choose the *weakest* of i,j to move (we will see why this is important later). To move i towards j, we use the fuction SEQUENCE. This gives us all the intermediate topologies that are needed to move i to j. Each of these topologies corresponds to a *move*. Hence, an epoch consists of moves. The movement is by cutting i from it's current placement and pasting it in all the edges of the path to j consecutively, similarly to what is shown in Fig. 8. After this movement is made, T_3 is updated to store the new topology.

We now explain the significance of the second `If` statement. If i,j was not a cherry in T but was in $\hat{T}$, then the previous `If` fixed that. Now, the second `If` locates all these cherries that are common in T_3 and $\hat{T}$ and removes them from the leaf set L, while adding their parent. This means that the subtree rooted in the parent will not be changed after that point, since it has the same topology in $T_3,\hat{T}$ and instead will be moved around with it's parent in subsequent steps. Hence, in the second round, L will contain some parents of leaves and possibly some leaves that were not matched into cherries in the first round. We give an example run of Algorithm 3 in Fig. 10.

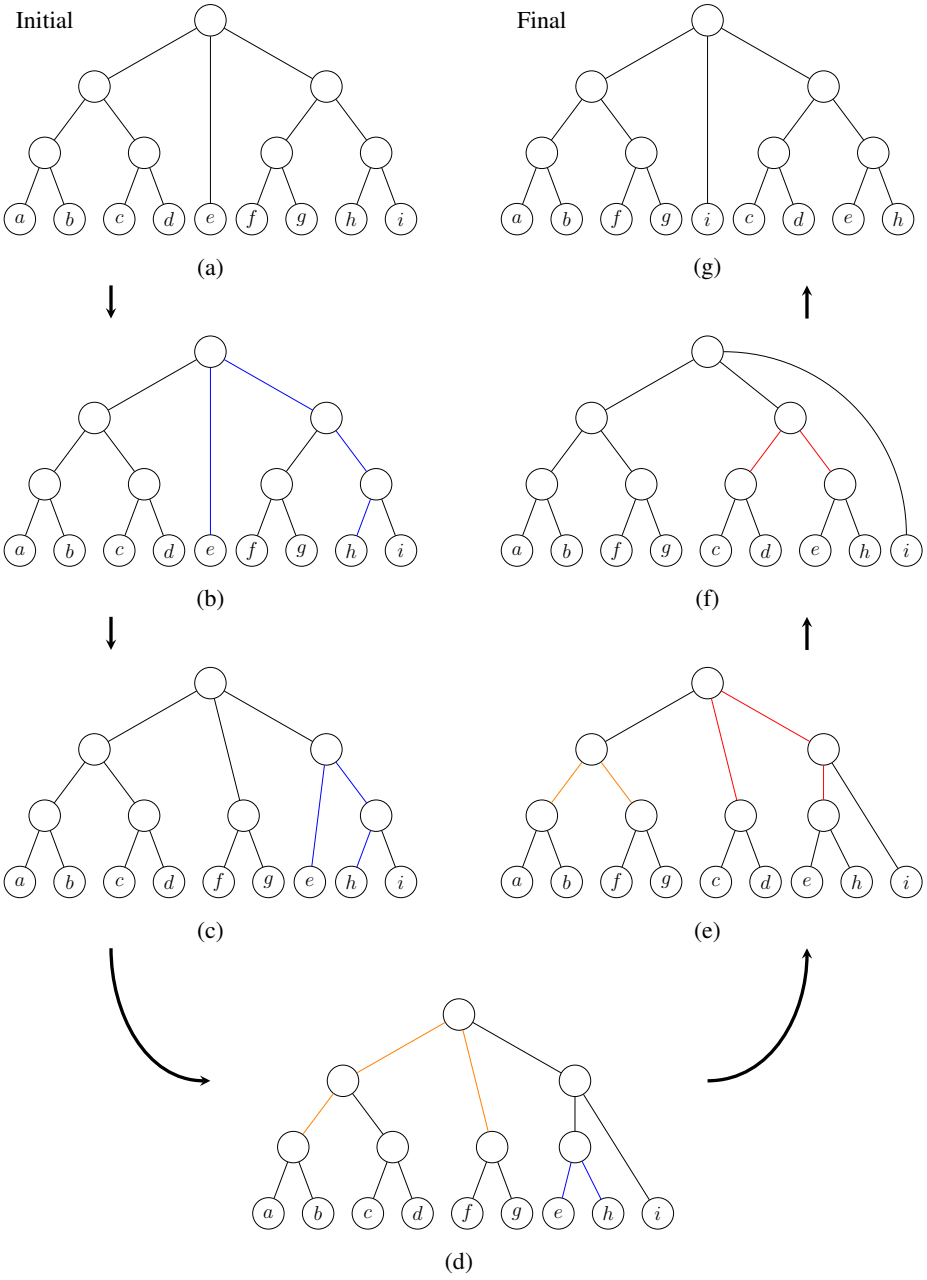


Figure 10: Example run of Algorithm 3. In the first round, we have epoch 1. In epoch 1, e becomes a cherry with h. In the second round, we have epochs number 2 and 3. In epoch 2, the parent of f, g becomes a cherry with the parent of a, b. In epoch 3, the parent of c, d becomes a cherry with the parent of e, h. After that, we have reached the final topology.

Algorithm 3 Interpolation between two tree topologies

Input : Unweighted trees $T = (V, E)$ and $\hat{T} = (\hat{V}, \hat{E})$ with leaf labels $\{1, \dots, n\}$, leaf correlations $\{\alpha_{ij}\}_{i,j\in\{1,\dots,n\},\ i\neq j}$ of tree T, and leaf correlations $\{\hat{\alpha}_{ij}\}_{i,j\in\{1,\dots,n\},\ i\neq j}$ of tree $\hat{T}$

Output : List L of topologies generated during interpolation

$T_1 \leftarrow T$
$S \leftarrow \{T_1\}$
$L \leftarrow \{1, \dots, n\}$
while $|L| \geq 4$ **do** `// a new round starts`
  for $i, j \in L$ **do**
    if CHERRY$(i, j, T_1) ==$ FALSE ***and*** CHERRY$(i, j, \hat{T}) ==$ TRUE **then** `// new epoch`
      $p \leftarrow$ common neighbor of i, j in $\hat{T}$
      $I \leftarrow$ set of leaves in the same component at i, if we remove (i, p) from $\hat{T}$
      $J \leftarrow$ set of leaves in the same component at j, if we remove (j, p) from $\hat{T}$
      $z \leftarrow \arg\max_{u\in I} \hat{\alpha}_{iu}$
      $w \leftarrow \arg\max_{u\in J} \hat{\alpha}_{ju}$
      if $\hat{\alpha}_{zp} > \hat{\alpha}_{wp}$ **then**
        Switch i, j
      end
      $k \leftarrow$neighbor of i on the path P_{ij}
      $l \leftarrow$ neighbor of j on the path P_{ij}
      $S_2 \leftarrow$ SEQUENCE(T_1, i, j); `// sequence of moves`
      $T_1 \leftarrow$ CUTPASTE$(T_1, i, k, (j, l))$; `// make` i, j `a cherry`
      $S \leftarrow S \cup S_2$
    end
    if CHERRY$(i, j, \hat{T}) ==$ TRUE **then** `// remove cherries where` $T_1, \hat{T}$ `agree`
      $p \leftarrow$ common neighbor of i, j in $\hat{T}$
      $L \leftarrow (L \setminus \{i, j\}) \cup \{p\}$
    end
  end
end
return L

Let's introduce a bit of notation about this process. Suppose q is a round, t is some epoch of this round, and m is some move in epoch t. We denote (i_t, j_t) the pair of leaves from L that is selected during epoch t of the algorithm. Suppose the length of the path $P_{i_t j_t}$ is l_t. Then, we denote by $v_0^t = i_t, v_1^t, \dots, v_{l_t}^t = j_t$ be the nodes in the path from i_t to j_t, which has length l_t. We denote by T^m the topology that we get before move m and T^{m+1} the one we get after the move. We also define $T^0 = T$. Formally, if m' is the first move of epoch t, we have,

$$T^{m+1} = \text{CUTPASTE}(T^{m'}, i_t, v_1^t, (v_m^t, v_{m+1}^t))$$

It is implied that in the definition of T^{qrs} we do not delete the leaves that have already been fixed into cherries. Note that i_t, j_t might correspond to some internal nodes. Let I_t, J_t be the set of leaves in the same component as i_t, j_t respectively, if we remove all the edges in the path from i_t to j_t.

Algorithm 4 SEQUENCE

Input : Tree T and two leaf indices i and j
Output : Modified tree S

$S \leftarrow \{T\}$
Let $\{v_0, \ldots, v_m\}$ be path from i to j ; // $v_0 = i, v_m = j$
for $r \leftarrow 1$ **to** $m - 1$ **do**
  $T_2 \leftarrow \text{CUTPASTE}(T, i, v_1, (v_r, v_{r+1}))$; // a move happens here
  $S \leftarrow S \cup \{T_2\}$
end
return S

Algorithm 5 CHERRY

Input : Tree T and two leaf indices i and j
Output : Boolean whether i and j form a cherry in T

if *i, j have a common neighbor in T* **then**
  return TRUE
else
  return FALSE
end

We collect here some observations about Algorithm 3 that will prove useful in the sequel.

Observation 1 *The total number of epochs for a single run of Algorithm 3 is at most n.*

Proof Each time an epoch is complete, we build a subtree of strictly larger size than before (size stands for number of leaves here) which agrees with $\hat{T}$. Since there are at most n leaves, we need at most n steps until we reach $\hat{T}$. Hence, there are at most n epochs in the whole process. ■

Observation 2 *Any quartet $\{i, j, k, l\}$ changes topology at most once per epoch.*

Proof Follows by inspecting the set of quartets that change during a move, which was described in Theorem 18. It is trivial to see that these sets are disjoint for different moves in a single epoch. ■

Observation 3 *For any epoch t and any move m in t, the subtree induced by the leaves in I_t and J_t is identical in T^m and in $\hat{T}$.*

Proof This follows inductively by the construction of the Algorithm. When a node i_t is selected, it is either a leaf or some node that was added in L after it's two children $i_{t'}, j_{t'}$ became cherries in T_1 during a previous epoch $t' < t$. Inductively, the subtrees rooted in $i_{t'}, j_{t'}$ have the same topology in T_1 and $\hat{T}$. Since $i_{t'}, j_{t'}$ are siblings in T_1 and in T' in epoch t, we conclude that the subtrees rooted at i_t are also identical in T_1 and $\hat{T}$ for epoch t. Same reasoning applies for J_t. ■

Observation 4 *At the end of the last M of Algorithm 3 we have $T^{M+1} = \hat{T}$.*

Proof Let us define the graph H_t as follows: it is obtained by running Algorithm 3 until epoch t and each time $t' \leq t$ we make a cherry with $i_{t'}, j_{t'}$, we remove the subtrees $I_{t'}, J_{t'}$, so that $i_{t'}, j_{t'}$ become leaves. By Observation 3 we know that the subtrees we remove in each epoch have the same topology as in $\hat{T}$. Obviously, the number of leaves in H_t shrink with each epoch, until we have 3 leaves u, v, w, for which there is only one possible topology. At that point, topology T^{M+1} is obtained by placing the subtrees for u, v, w back. By Observation 3, we know that these subtrees have the topology of $\hat{T}$, hence T^{M+1} should also have the topology of $\hat{T}$. ■

Observation 5 *If D is the diameter of $\hat{T}$, there are at most $\lceil D/2 \rceil$ rounds when we run Algorithm 3. Furthermore, each leaf is moved in at most one epoch per round.*

Proof Consider the graph H_t that was defined in the proof of Observation 4. We will show that the largest path in H_t shrinks by at least 2 edges in each round. It then follows that there will be at most $\lceil D/2 \rceil$ rounds in total.

Let u, v be two leaves of H_t such that P_{uv} in H_t has length equal to the diameter D of H_t. Let p be the only neighbor of u in H_t. Clearly, u is also part of the path P_{uv}. Let w be the neighbor of p that does not lie on the path P_{uv} (since p has degree 3, such a neighbor should exist). We claim that w should be a leaf, otherwise we could extend the path P_{up} into one with larger length than P_{uv}. Thus, u, w should be siblings, which means they will be selected in the current round to be paired into a cherry, which will remove them from the graph and will leave p as a leaf. Thus, P_{uv} will shring by one edge on the side of u and for the same reason will also shring by one edge on the side of v. This proves our claim.

For the second claim, any leaf u is moved only when some subtree with root i_t is moved and u belongs in this subtree. Suppose that this happens during a round, resulting in i_t, j_t becoming a cherry. Then, we can see that Algorithm 3 then removes i_t, j_t from the list of leaves L, which means that i_t will not move again for the remainder of that round (it's parent p_t is not considered in the `For` loop of the current round). Hence, u remains fixed for the remaining of that round.

■

We first argue that during this interpolation process, only bad quartets change topology. This is crucial, since good quartets should be maintained if we wish to lose only a little in TV.

Lemma 19 *Let $T = G(V, E)$ and $\hat{T} = G(\hat{V}, \hat{E})$ be two trees with tree metrics $\alpha, \hat{\alpha}$ respectively. We assume that $\|\alpha - \hat{\alpha}\|_\infty \leq \varepsilon$. Suppose we run the procedure 3 with input $T, \hat{T}, \alpha, \hat{\alpha}$. Let T^m, T^{m+1} be two arbitrary consecutive steps in this process. Let U_m be the set of quartets where T^m, T^{m+1} disagree. Then, for all $(w, z, y, u) \in U_m$, we have that*

$$\Delta_{w,z,y,u}(\alpha) \leq 20n\varepsilon$$

Proof We will denote by t the epoch where move m belongs to. We will prove the claim inductively over t. We will prove that if a quartet $\{w, z, y, u\}$ is changed during the t-th epoch, then

$$\Delta_{w,z,y,u}(\alpha) \leq 20t\varepsilon$$

This obviously implies the final claim since $t \leq n$ by Observation 1. Since the base case is the same as the inductive step, we give the inductive step proof only.

Suppose we are at epoch t. First of all, we know that node i_t is selected to be moved towards j_t, where the intermediate nodes are $v_0^t = i_t, v_1^t, \dots, v_{m_t}^t = j_t$. Similarly to the proof of Theorem 18, let I_t be the set of leaves on the same component with i_t if we remove edge (v_0^t, v_1^t), S_i^t the set of leaves on the same component as v_i^t, if we remove edges $(v_{i-1}^t, v_i^t), (v_i^t, v_{i+1}^t)$ from the tree, and J_t be the set of leaves on the same component with j_t if we remove edge $(v_{m_t-1}^t, v_{m_t}^t)$. Also, let us define $L_{ts} = \cup_{k \le s} S_k^t, R_{ts} = \cup_{k \ge s} S_k^t \cup J_t$ for all $s \le m_t$. The situation is similar to the one presented in Fig. 8.

Suppose T^m corresponds to node i_t being pasted in the middle of edge (v_s^t, v_{s+1}^t) for some fixed s. As we saw in the proof of Theorem 18, the set U_m of quartets that differ in T^m, T^{m+1} can be written as

$$U_m = \{\{w, z, y, u\} : w \in I_t, z \in L_s^t, y \in S_{s+1}^t, u \in R_{s+2}^t\}$$

Note that all the quartets in U_m are considered to change at epoch t. Suppose there exists a quartet $\{w, z, y, u\} \in U_m$ such that

$$\Delta_{w,z,y,u}(\alpha) > 20t\varepsilon$$

First, we will assume that $u \in \cup_{k \ge s} S_k^t$. Afterwards, we will deal with the case $u \in J_t$, which will actually prove to be easier. The first thing we observe is that we can assume without loss of generality that the topology of $\{w, z, y, u\}$ has not been altered in any previous epoch. The reason is that if it was altered at some epoch $t' < t$, then by the inductive assumption, we already have

$$\Delta_{w,z,y,u}(\alpha) \le 20t'\varepsilon < 20t\varepsilon$$

and we have nothing to prove. Hence, we can assume w.l.o.g. that it is the first time that it is changing topology. Note also that by Observation 2 a quartet changes topology at most once per epoch. Since it has not changed topology before, it follows that it's topology in T is $\{(wz)(yu)\}$. It is straightforward to notice that

$$\Delta_{w,z,y,u}(\alpha) = \Delta_{w,z,y,u}(|\alpha|)$$

where $|\alpha|$ is the vector of absolute values of α. This implies that

$$\Delta_{w,z,y,u}(\alpha) = |\alpha_{wz}||\alpha_{yu}| - |\alpha_{zy}||\alpha_{wu}| > 20t\varepsilon$$

Our assumption about $\alpha, \hat{\alpha}$ implies that

$$|\alpha_{wz}||\alpha_{yu}| - |\hat{\alpha}_{wz}||\hat{\alpha}_{yu}| \le 2\varepsilon \quad , \quad |\alpha_{zy}||\alpha_{wu}| - |\hat{\alpha}_{zy}||\hat{\alpha}_{wu}| \le 2\varepsilon$$

Hence, we have

$$|\hat{\alpha}_{wz}||\hat{\alpha}_{yu}| - |\hat{\alpha}_{zy}||\hat{\alpha}_{wu}| > 20t\varepsilon - 4\varepsilon$$

Let p_t be the common parent of i_t, j_t in the tree $\hat{T}$. Now, by the construction of procedure 3(first `If` statement), we know that

$$\max_{f \in J_t} |\hat{\alpha}_{f,p_t}| \ge \max_{f \in I_t} |\hat{\alpha}_{f,p_t}| \tag{16}$$

Now, we know by Observation 3 that the subtrees rooted at i_t and j_t with leaf sets I_t and J_t respectively have the same topology in T^m and $\hat{T}$. Since $z, u \notin I_t$, we can write (see Fig. 11).

$$|\hat{\alpha}_{wz}||\hat{\alpha}_{yu}| - |\hat{\alpha}_{zy}||\hat{\alpha}_{wu}| = |\hat{\alpha}_{w,p_t}|(|\hat{\alpha}_{z,p_t}||\hat{\alpha}_{yu}| - |\hat{\alpha}_{zy}||\hat{\alpha}_{u,p_t}|)$$

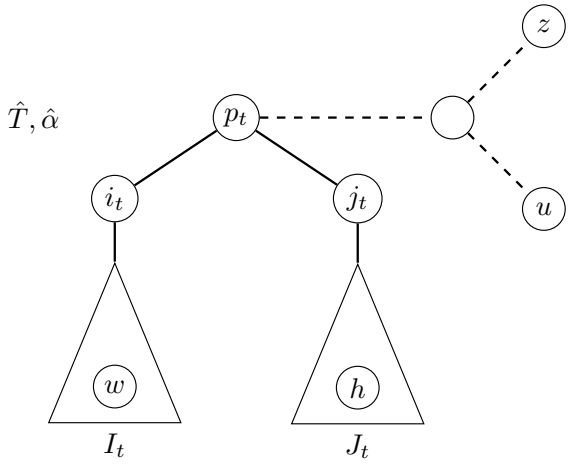


Figure 11: From the picture, it is clear that $\hat{\alpha}_{wz} = \hat{\alpha}_{wp_t}\hat{\alpha}_{p_tz}$

Let $h = \arg\max_{f \in J_t} |\hat{\alpha}_{f,p_t}|$. Then, by(16) we have

$$|\hat{\alpha}_{h,p_t}|(|\hat{\alpha}_{z,p_t}||\hat{\alpha}_{yu}| - |\hat{\alpha}_{zy}||\hat{\alpha}_{u,p_t}|) \geq |\hat{\alpha}_{w,p_t}|(|\hat{\alpha}_{z,p_t}||\hat{\alpha}_{yu}| - |\hat{\alpha}_{zy}||\hat{\alpha}_{u,p_t}|)$$

Since we have assumed that $z, u \notin I_t$, Fig. 11 implies that

$$\begin{aligned}|\hat{\alpha}_{hz}||\hat{\alpha}_{yu}| - |\hat{\alpha}_{zy}||\hat{\alpha}_{hu}| &= |\hat{\alpha}_{h,p_t}|(|\hat{\alpha}_{z,p_t}||\hat{\alpha}_{yu}| - |\hat{\alpha}_{zy}||\hat{\alpha}_{u,p_t}|) \\ &\geq |\hat{\alpha}_{w,p_t}|(|\hat{\alpha}_{z,p_t}||\hat{\alpha}_{yu}| - |\hat{\alpha}_{zy}||\hat{\alpha}_{u,p_t}|) > 20t\varepsilon - 6\varepsilon\end{aligned}$$

By the closeness of $\alpha, \hat{\alpha}$, this in turn implies that

$$|\alpha_{hz}||\alpha_{yu}| - |\alpha_{zy}||\alpha_{hu}| > 20t\varepsilon - 10\varepsilon > 20(t-1)\varepsilon \tag{17}$$

Clearly, leaves h, z, y, u do not change position during epoch t, hence the quartet $\{h, z, y, u\}$ does not change topology during epoch t. Now, there are two possibilities:

Case 1: Suppose $\{h, z, y, u\}$ has changed topology at least once in some previous epoch $t' < t$. Then, by the inductive hypothesis, we should have

$$\Delta_{h,z,y,u}(\alpha) \leq 20t'\varepsilon$$

Since $t' \leq t - 1$, this contradicts (17).

Case 2: Suppose $\{h, z, y, u\}$ has not changed topology until epoch t. This means that the topology of $\{h, z, y, u\}$ in T is $\{(zy)(hu)\}$, since that is the topology in T^{qrs}. This implies that

$$|\alpha_{hz}||\alpha_{yu}| - |\alpha_{zy}||\alpha_{hu}| < 0$$

which again contradicts (17).

Hence, in all cases we obtain a contradiction and the inductive step is proved. Now, let's consider the case $u \in I_t$. Then, we can assume w.l.o.g. that this is the first epoch where $\{w, z, y, u\}$ changes topology, otherwise the inductive step applies. This means that the topology of this quartet in T is $\{(wz)(yu)\}$. Hence,

$$\Delta_{w,z,y,u}(\alpha) = |\alpha_{wz}||\alpha_{yu}| - |\alpha_{zy}||\alpha_{wu}|$$

Clearly, the topology of this quartet is $\{(zy)(wu)\}$ in $\hat{T}$. This implies that

$$|\hat{\alpha}_{wz}||\hat{\alpha}_{yu}| - |\hat{\alpha}_{zy}||\hat{\alpha}_{wu}| < 0$$

In turn, this means

$$|\alpha_{wz}||\alpha_{yu}| - |\alpha_{zy}||\alpha_{wu}| < 4\varepsilon < 20t\varepsilon$$

and this concludes the claim in that case as well.

■

We now formulate our main result, which bounds the Lipschitzness of the function f_x^T in terms of local changes in the topology of T. We will use it to relate the changes in TV to the changes of quartet topologies along this interpolation process.

Lemma 20 *Let $T = G(V, E)$ and $\hat{T} = G(\hat{V}, \hat{E})$ be two trees and suppose that the diameter of $\hat{T}$ is D. Let α be some tree metric induced by T. We assume that $\|\alpha - \hat{\alpha}\|_\infty \leq \varepsilon/(40Dn^5)$ for some $\varepsilon < 1$. Suppose we run the procedure 3 with input $T, \hat{T}, \alpha, \hat{\alpha}$. Let T^m, T^{m+1} be two arbitrary consecutive topologies in this process, corresponding to move m. Let U_m be the set of quartets where T^m, T^{m+1} disagree. Then,*

$$\sum_{x\in\{-1,1\}^n} \left| f_x^{T^{m+1}}(\alpha) - f_x^{T^m}(\alpha) \right| \leq \frac{|U_m|\varepsilon}{Dn^4} \tag{18}$$

Proof We are going to prove this inductively on the total number of moves m. Suppose we are in the first move, $m = 1$ of round $q = 1$ and epoch $t = 1$. By applying Theorem 18, we have that

$$\begin{aligned}\sum_{x\in\{-1,1\}^n} \left| f_x^{T^1}(\alpha) - f_x^{T^0}(\alpha) \right| &= \sum_{x\in\{-1,1\}^n} \left| \sum_{\{w,z,y,u\}\in U_1} x_w x_z x_y x_u f_x^{T^1\setminus\{w,z,y,u\}}(\alpha) \right| \\ &\leq \sum_{\{w,z,y,u\}\in U_{111}} \Delta_{w,z,y,u}(\alpha) \sum_{x\in\{-1,1\}^n} \left| f_x^{T^1\setminus\{w,z,y,u\}}(\alpha) \right|\end{aligned}$$

Now, notice that since there have not been any other changes to the topology of T except for the first move, $T^0 = T$ and furthermore, the expression $f_x^{T^1\setminus\{w,z,y,u\}}(\alpha)$ is actually the probability distribution on the leaves of a tree that has edge weights that agree with α, except for edges that belong to some path of the quartet $\{w, z, y, u\}$, which have weight 0. Hence, we can remove the absolute value and this gives us

$$\sum_{x\in\{-1,1\}^n} \left| f_x^{T^1}(\alpha) - f_x^{T^0}(\alpha) \right| \leq \sum_{\{w,z,y,u\}\in U_1} \Delta_{w,z,y,u}(\alpha)$$

Finally, by applying Theorem 19 we get

$$\sum_{x\in\{-1,1\}^n} \left| f_x^{T^1}(\alpha) - f_x^{T^0}(\alpha) \right| \leq |U_1|20n \cdot \frac{\varepsilon}{40Dn^5} = \frac{\varepsilon|U_1|}{2Dn^4}$$

hence the base case is true.

Now suppose the claim holds for all moves $m' < m$. First, we define the vector $\alpha^{wzyu} \in [0,1]^{\binom{n}{2}}$ as

$$\alpha_{kl}^{wzyu} = \begin{cases} \alpha_{kl} & \text{, if the path } P_{kl} \text{ has no common edges with any paths of the quartet } \{w,z,y,u\} \\ 0 & \text{, otherwise} \end{cases}$$

Clearly, α^{wzyu} is also a metric on T, which is induced by the same weights as α, except that all edges on paths of the quartet $\{w,z,y,u\}$ have weight 0. By again applying Theorem 18 and Theorem 19, we can get

$$\begin{aligned} \sum_{x\in\{-1,1\}^n} \left| f_x^{T^{m+1}}(\alpha) - f_x^{T^m}(\alpha) \right| &\leq \sum_{\{w,z,y,u\}\in U_m} \Delta_{w,z,y,u}(\alpha) \sum_{x\in\{-1,1\}^n} \left| f_x^{T^m\setminus\{w,z,y,u\}}(\alpha^{wzyu}) \right| \\ &\leq \frac{\varepsilon}{2Dn^4} \sum_{\{w,z,y,u\}\in U_m} \left| f_x^{T^m\setminus\{w,z,y,u\}}(\alpha^{wzyu}) \right| \end{aligned} \tag{19}$$

Now, let's fix a quartet $\{w,z,y,u\} \in U_{qrs}$. The graph $T^m \setminus \{w,z,y,u\}$ is a tree where all edges that belong to some path of the quartet $\{w,z,y,u\}$ have been removed. The problem is that other changes have happened in the topology of T before it reaches the current state T^m. Therefore, the quantity $f_x^{T^m\setminus\{w,z,y,u\}}(\alpha^{wzyu})$ is no longer a distribution over a tree, since α^{wzyu} corresponds to the initial tree metric on T, with some edges set to 0. Hence, we cannot get rid of the absolute value and claim that this quantity sums up to 1. Instead, our strategy will be to relate this quantity to some other quantity that is a probability distribution. To describe this probability distribution, consider the collection of subtrees that are obtained from T^m by removing all paths of the quartet $\{w,z,y,u\}$. These partition the set of leaves into subsets S_i, one for each subtree. Let G_m be the forest that is obtained by taking for each subset on leaves S_i the subtree induced by T(when we say induced, it is implicit that the function BINARY is applied to make the subtree have all non-leaves with degree 3). We will show how to relate $f_x^{T^m\setminus\{w,z,y,u\}}(\alpha^{wzyu})$ with $f_x^{G_m}(\alpha^{wzyu})$. Notice that by definition, α^{wzyu} is clearly a metric induced from G_m and so the latter quantity is a probability distribution.

Our strategy for relating these two quantities will be to interpolate between $T^m \setminus \{w,z,y,u\}$ to G_m. The way to do this interpolation is using Algorithm 3. In particular, the following Lemma shows that in order for Algorithm 3 to transform G_m to $T^m \setminus \{w,z,y,u\}$, it will need strictly less moves than the ones needed to transform T to T^{m+1}.

Lemma 21 *Let M be the number of moves needed for Algorithm 3 to transform T to T^{m+1}. Then, it is possible to transform G_m to $T^m \setminus \{w,z,y,u\}$ using a number of moves that is strictly smaller than M.*

Proof The idea of the proof is very simple and relies on the fact that we can simply "copy" the moves made from T to T^{m+1}, except when these moves aim at making a cherry with two leaves that belong to different subtrees of G_m, in which case no move is necessary. To be more formal, let R be the number of epochs that Algorithm 3 needs to reach T^{m+1} starting from T. Then, we will show that we can reach $T^m \setminus \{w,z,y,u\}$ using a number of epochs R' such that $R' < R$. Furthermore, we will argue that each of the R' epochs has at most the same number of moves as the corresponding one starting from R. We do this by examining one by one the R epochs from T to T^{m+1} and deciding how to potentially change it. First of all, let's remember that at the start of each epoch t, Algorithm 3 chooses two nodes i_t, j_t and makes them siblings with parent p_t, thus making a larger subtree that

agrees with $\hat{T}$. The relative topology inside this subtree will never be altered by the algorithm again. We call this process *fixing* the subtree with root p_t. When we refer to the subtree of i_t and j_t we mean the connected component that results when we remove path $P_{i_t j_t}$ from the graph at epoch t. We denote $\{i_t, j_t\}_t$ the sequence of epochs produced from T to T^{qrs} and $\{i'_t, j'_t\}_t$ the sequence of epochs that transforms G_m to $T^m \setminus \{w, z, y, u\}$.

We will inductively prove that at any epoch in the sequence $\{i_t, j_t\}$, if a subtree with root p_t has been fixed after that epoch and if this subtree is contained in some component of G_m, then this subtree will also be fixed under sequence $\{i'_t, j'_t\}$. In proving the inductive step, we will also describe how to define the sequence of epochs $\{i'_t, j'_t\}$. After proving this claim, we will explain why it implies the statement of the Lemma.

Suppose the claim holds for all epochs prior to t(for $t = 1$ the claim is trivial). Suppose then that at epoch t i_t and j_t become cherries with parent p_t. Let T^t be the topology at the start of the epoch and let G^t_m be the corresponding topology at the start of epoch t under the sequence $\{i'_t, j'_t\}$. Suppose first that the subtrees of i_t, j_t belong to the same component of T^{qrs}. Then, inductively, we know that i_t, j_t exist also in G^t_m and their subtrees have already been fixed by the sequence $\{i'_t, j'_t\}$. In that case, we set $i'_t = i_t, j'_t = j_t$ and set the movement of i'_t to j'_t to be the same as the one from i_t to j_t, but on the induced component of G^t_m that i_t, j_t belong to. We call this a *true* epoch. Since the paths in an induced subtree can only stay the same or become smaller than the ones in the original tree (after applying operation BINARY), the number of moves required to move i'_t to j'_t is at most the number of moves required to move i_t to j_t. Once we move i'_t to become sibling with j'_t, the new subtree with root p'_t has also been fixed for the sequence $\{i'_t, j'_t\}$, proving the inductive hypothersis in that case. Now, suppose that i_t, j_t belong in different subtrees of G_m. There are two cases: either both i_t, j_t exists as nodes in G^t_m, or at least one of them does not exist. If they both exist, then again by the inductive hypothesis, it follows that the subtrees i_t, j_t have also been fixed in G^t_m. In that case, it must be the case that the entire component of i_t in G^t_m is equal to that subtree (otherwise we would be able to connect i_t to some other sibling and enlarge it). Hence, in that case no movement takes place and we trivially set $i'_t = j'_t = i_t$ to denote that this is not a true epoch. The point is that there is no need to move them again until we reach $T^m \setminus \{w, z, y, u\}$, so our choice not to move them is correct. Now, let's examine the case that either i_t or j_t does not exist in G^t_m. Suppose i_t does not exist w.l.o.g. Then, this means that it is a parent of two subtrees that do not belong to the same component of G^t_m. Thus, this means that there is no reason to connect these subtrees, hence we also trivially set $i'_t = j'_t = i_t$. We call this epoch *fake*. The inductive step is now complete.

The induction we just proved shows that the sequence $\{i'_t, j'_t\}$ leads to $T^m \setminus \{w, z, y, u\}$ when started from G_m. Also, it is clear that the true number of epochs in $\{i'_t, j'_t\}$ at any given time is at most the ones in $\{i_t, j_t\}$, since some epochs might be fake. In fact, if $\{i_t, j_t\}$ reaches T^{m+1} at epoch R, the number of true epochs R' in $\{i'_t, j'_t\}$ should be strictly smaller than R. The reason is that at epoch R, i_R and j_R belong to different components of $T^{qrs} \setminus \{w, z, y, u\}$ by definition (since we remove the path from w to u). Hence, the last epoch R will not be a true epoch for $\{i'_t, j'_t\}$. Since we have also argued that the number of moves in epochs of $\{i'_t, j'_t\}$ is at most the corresponding number for epochs in $\{i_t, j_t\}$,this concludes the proof of the Lemma.

■

Let M be the total number of moves required by Algorithm 3 to transform T to T^{m+1} and M' the moves to transform G_m to $T^m \setminus \{w, z, y, u\}$. The point of Theorem 21 is that $M' < M$. Let $G_m = G^0_m, G^1_m, \ldots, G^{M'}_m = T^m \setminus \{w, z, y, u\}$ be the sequence of graphs in the interpolation. By

triangle inequality, we have

$$\left|f_x^{T^m\setminus\{w,z,y,u\}}(\alpha^{wzyu})\right| \leq \left|f_x^{G_m^0}(\alpha^{wzyu})\right| + \sum_{s=1}^{M'}\left|f_x^{G_m^s}(\alpha) - f_x^{G_m^{s-1}}(\alpha)\right|$$

As we have already explained, the first term on the right hand side corresponds to a distribution, hence we can remove the absolute values. The remaining terms have the form of the left hand side of (18), which is what we want to bound in general. However, these differences are applied to graphs that are obtained after at most M' moves of Algorithm 3. Hence, we can apply the inductive hypothesis (18). If U_s' is the set of quartets that change from G_m^{s-1} to G_m^s, then,

$$\sum_{x\in\{-1,1\}^n}\sum_{s=1}^{M'}\left|f_x^{G_m^s}(\alpha) - f_x^{G_m^{s-1}}(\alpha)\right| \leq \frac{\varepsilon}{Dn^4}\sum_{s=1}^{M'}|U_s'|$$

It remains to bound the sum $\sum_{s=1}^{M'}|U_s'|$. This is equal to the total number of quartets that have changed topology until move M', starting from G_m (if a quartet has changed multiple times, we count the number of times it has changed in this sum). We argue that

$$\sum_{s=1}^{M'}|U_s'| \leq Dn^4$$

The reason is the following: there is a total of $\binom{n}{4}$ quartets, so it suffices to bound the number of times that any specific quartet $\{w,z,y,u\}$ changes topology. First of all, we have already argued that a quartet changes topology at most once every epoch. In order for a quartet to change topology during some epoch, at least one of it's leaves should be moved to some different position. By Observation 5 we know that a leaf is moved at most $\lceil D/2\rceil$ times in total. Hence, a quartet changes topology at most $4\lceil D/2\rceil$ times in total. Hence,

$$\sum_{s=1}^{M'}|U_s| \leq \binom{n}{4}2D \leq Dn^4$$

Combining everything together, we get

$$\sum_{x\in\{-1,1\}^n}\left|f_x^{T^m\setminus\{w,z,y,u\}}(\alpha^{wzyu})\right| \leq \sum_{x\in\{-1,1\}^n} f_x^{G_m^0}(\alpha^{wzyu}) + \frac{\varepsilon}{Dn^4}Dn^4 = 1+\varepsilon$$

This holds for all $\{w,z,y,u\}\in U_m$. Hence, by using (19) we get

$$\begin{aligned}\sum_{x\in\{-1,1\}^n}\left|f_x^{T^{m+1}}(\alpha) - f_x^{T^m}(\alpha)\right| &\leq \frac{\varepsilon}{2Dn^4}\sum_{\{w,z,y,u\}\in U_m}\left|f_x^{T^m\setminus\{w,z,y,u\}}(\alpha^{wzyu})\right| \\ &\leq \frac{\varepsilon}{2Dn^4}|U_m|(1+\varepsilon) \leq \frac{|U_m|\varepsilon}{Dn^4}\end{aligned}$$

since $\varepsilon \leq 1$. This is the inductive claim that we wanted to prove.

■

We are now ready to conclude the proof of Theorem 13. To do it, we simply use Theorem 20 to transition from T to $\hat{T}$. Then, we use the bound for the fixed topology to change α to $\hat{\alpha}$.

Proof [Proof of Theorem 13] We can assume without loss of generality that $\hat{T}$ has a smaller diameter than T, otherwise we just reverse the roles of $T, \hat{T}$. We run Algorithm 3 with input $T, \hat{T}, \alpha$, which produces a sequence $T = T^0, T^2, \ldots, T^M = \hat{T}$, where each element of the sequence corresponds to some move. We have that

$$\begin{aligned} TV(\mu, \hat{\mu}) &= \sum_{x \in \{-1,1\}^n} \left| f_x^T(\alpha) - f_x^{\hat{T}}(\hat{\alpha}) \right| \\ &\leq \sum_{x \in \{-1,1\}^n} \left| f_x^T(\alpha) - f_x^{\hat{T}}(\alpha) \right| + \sum_{x \in \{-1,1\}^n} \left| f_x^{\hat{T}}(\alpha) - f_x^{\hat{T}}(\hat{\alpha}) \right| \end{aligned} \tag{20}$$

Define $\varepsilon' = 40Dn^5\varepsilon$. Let us divide into cases.

Case 1: Suppose $\varepsilon' < 1$. Then, the first term of the RHS of (20) can be bounded using the successive steps of the interpolation process. In particular, since $\varepsilon = \varepsilon'/(40Dn^5)$, we can apply Theorem 20 to get

$$\sum_{x \in \{-1,1\}^n} \left| f_x^T(\alpha) - f_x^{\hat{T}}(\alpha) \right| \leq \sum_{m=1}^{M} \sum_{x \in \{-1,1\}^n} \left| f_x^{T^m}(\alpha) - f_x^{T^{m-1}}(\alpha) \right| \leq \sum_{m=1}^{M} \frac{|U_m|\varepsilon'}{Dn^4}$$

By the proof of Theorem 20, this implies that

$$\sum_{x \in \{-1,1\}^n} \left| f_x^T(\alpha) - f_x^{\hat{T}}(\alpha) \right| \leq \frac{\varepsilon'}{Dn^4} Dn^4 = \varepsilon' = 40Dn^5\varepsilon$$

As for the second term of the RHS of (20), it is essentially the difference when we substitute $\hat{\alpha}$ with α in the fixed topology $\hat{T}$. Hence, we can directly apply Theorem 10 to get

$$\sum_{x \in \{-1,1\}^n} \left| f_x^{\hat{T}}(\alpha) - f_x^{\hat{T}}(\hat{\alpha}) \right| \leq 2n^2\varepsilon$$

Overall, this gives us

$$TV(\mu, \hat{\mu}) \leq 42Dn^5\varepsilon$$

which proves inequality (15) in that case.

Case 2: Assume $\varepsilon' \geq 1$. Then,

$$TV(\mu, \hat{\mu}) \leq 1 \leq 40Dn^5\varepsilon$$

which means that (15) trivially holds in that case too. The proof is now complete. ■

Appendix D. Proof of Theorem 3 (unknown topology)

D.1. Outline

In this section, we will present an algorithm that takes samples from the leaves of some tree Ising model with tree T^* and weight θ^* and estimates a topology $\hat{T}$ together with weights on the edges, so that the distributions on the leaves of T and $\hat{T}$ are ε-close in TV distance. The number of samples will be polynomial in n and $1/\varepsilon$.

We will use the results in Daskalakis et al. (2009) about learning phylogenetic trees without assuming any upper/lower bounds on the edge weights. We will show how we can use the guarantees of this prior work to obtain an algorithm for finding a good enough topology. Before explaining the result formally, we first want to give an intuition. We discuss how the output might be different from the original tree. Firstly, without sufficient samples, it is impossible to determine the existence of edges that are far away from leaves. In such cases, the algorithm of Daskalakis et al. (2009) will omit those edges and output a forest, as shown in Fig. 12.

To compare with Cryan et al. (2001), they employ a similar process of splitting the tree into subtrees. However, they then rely on learning the weights within in each subtree accurately. This yields a bound in total variation between the learned and true distribution within each tree. The difference between these approaches is that we learn in total variation only the *leaf distribution* within each subtree, while their analysis relies on learning the distribution on both the leaves and the *internal nodes* of each subtree. This requires them to cut the original tree into significantly smaller subtrees, which harms the complexity.

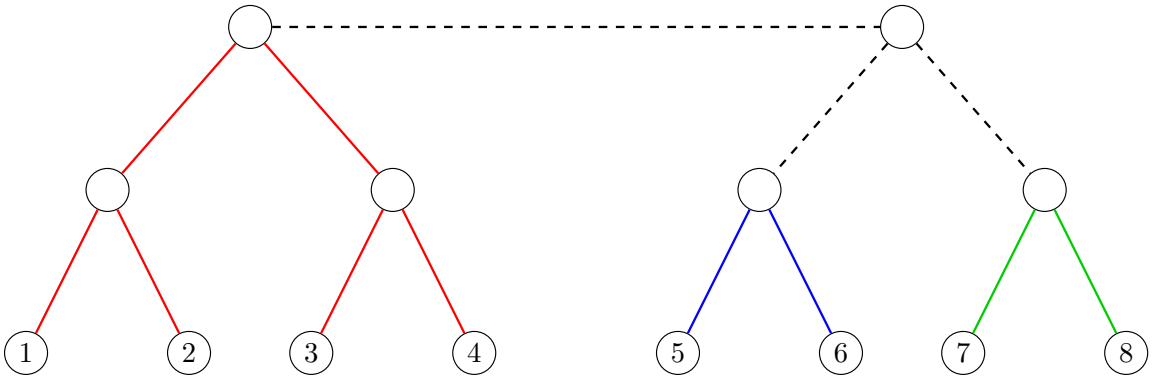


Figure 12: A forest that is created by deleting edges: the true tree contains all the solid and dashed edges. Yet, the algorithm of Daskalakis et al. (2009) might not be able to identify some of the edges because they do not sufficiently correlate with the leaves, and so it will return a forest. In the example here, the forest is obtained from the original tree by removing the dashed edges. The three connected components of the output forest are colored red, blue and green.

Secondly, the algorithm may fail to split the tree in a topologically sensible way. This means that it will return a forest, yet, in contrast with the example in Fig. 12, it is *impossible* to obtain this forest by cutting some edges in the ground truth tree. Still, the topology *within each connected component*

in this forest is preserved, as illustrated in Fig. 13 (a)-(b). After splitting the tree into two subtrees, those subtrees might contain some internal nodes of degree 2. Such nodes cannot be identified from the leaves, and they will be contracted, as shown in Fig. 13 (c).

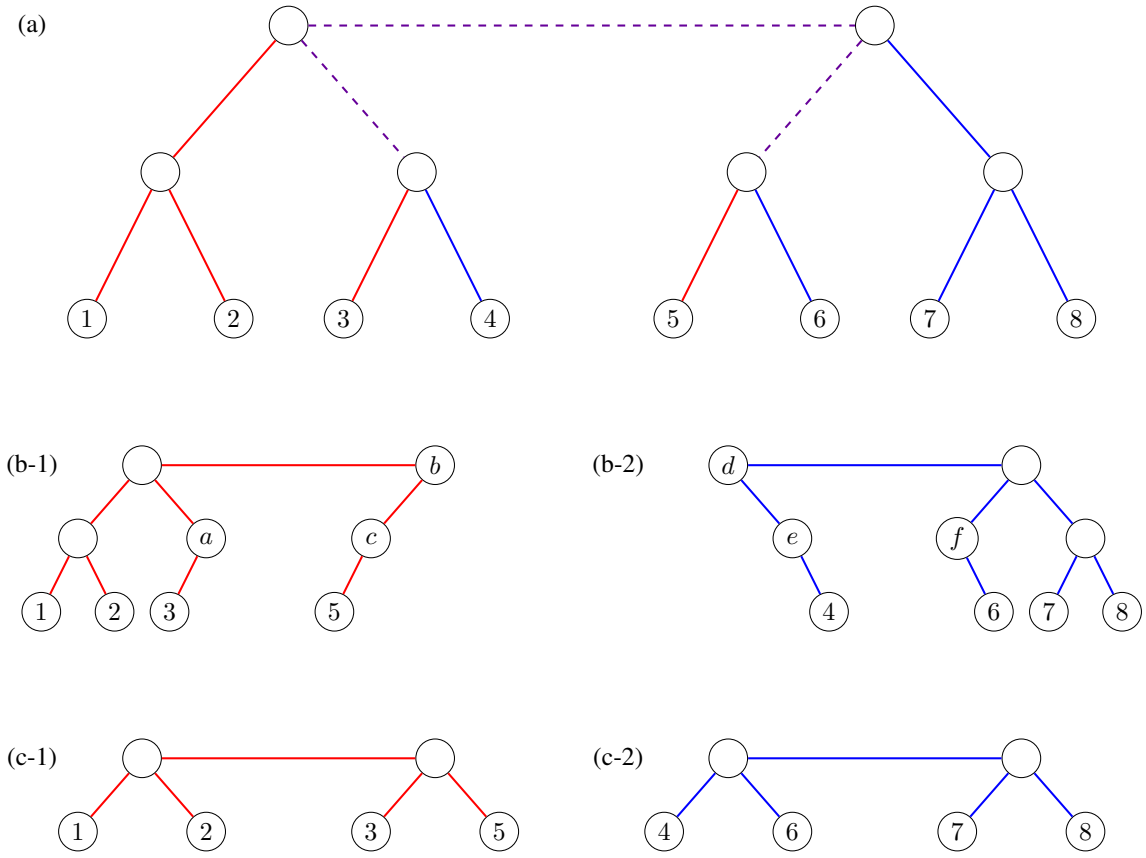


Figure 13: Splitting a tree into subtrees in a more complicated fashion: In (a), the true tree contains both the solid and dashed edges. Yet, the algorithm might not identify completely the topology. Instead it will return two subtrees. Yet, the subtrees are not topologically sensible: one tree will contain the leaves $\{1, 2, 3, 5\}$ and the other will contain $\{4, 6, 7, 8\}$. The red edges correspond to the first tree, the blue edges to the second tree, while the dashed edges are shared by both trees. In figure (b), we split the original tree into the two subtrees, one containing the red and dashed edges and the other the blue and dashed edges. Notice that in Figure (b), the nodes labeled a-f have degree 2. Information theoretically, it is impossible to identify hidden nodes of degree 2. Indeed, the same leaf distribution is obtained by removing each of these nodes, connecting its neighbors, and adjusting the weight of the new edges. Hence, the output of the algorithm will not have degree-2 nodes: instead, the transformation described above, that removes degree-2 nodes, will be performed. In (c) we demonstrate the result of removing such degree-2 nodes.

The two transformations that are applied to a tree in Fig. 12 and Fig. 13 can be viewed as a single transformation: separating the tree into subtrees, while preserving the topology within each subtree. This can be seen in Fig. 14.

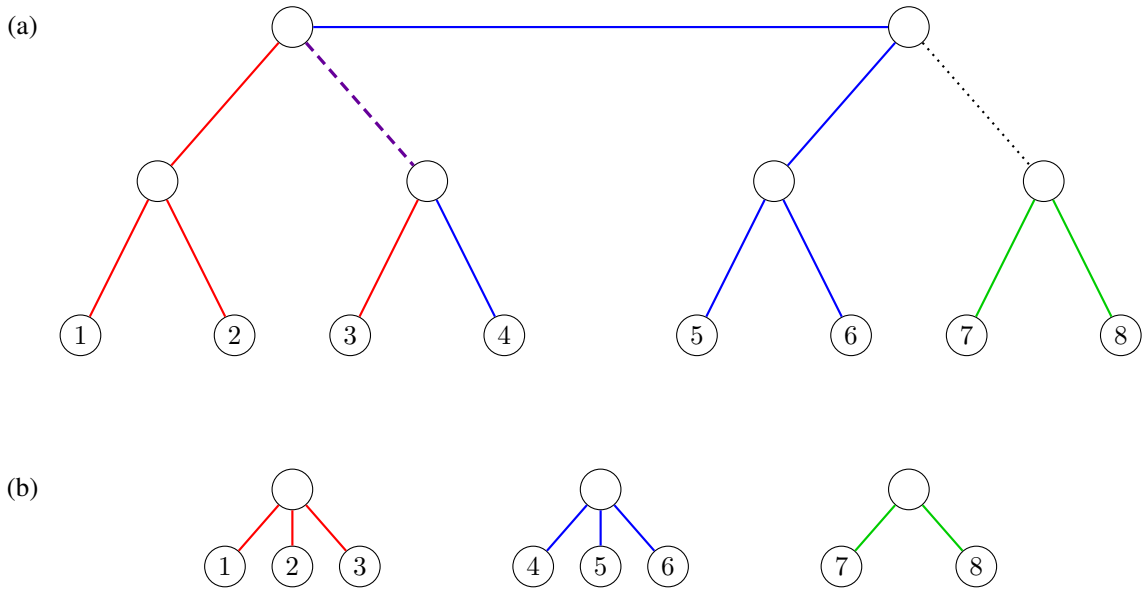


Figure 14: Separating a tree into multiple subtrees: In (a), the original tree contains all solid, dashed and dotted edges. It is split by the algorithm to three subtrees. The first subtree contains leaves $\{1, 2, 3\}$ and the red and dashed edges. The second subtree contains $\{4, 5, 6\}$ and the blue and dashed edges. The third contains $\{7, 8\}$ and the green edges. Notice that the dashed purple edge is contained in two trees, while the dotted black edge is contained in no tree. Notice that the first and second subtrees intersect while the third subtree is disjoint from the other subtrees. In (b) we see the output of the algorithm.

Yet, the algorithm of Daskalakis et al. (2009) might not be able to tell the exact topology within each subtree. In this case, the output will just contract some of the internal edges of this subtree. See Fig. 15 (a)-(c) for an example of a contracted subtree. In Fig. 15 (d) we depict some of the possible topologies that the algorithm could have confused between, which led to contracting a specific edge.

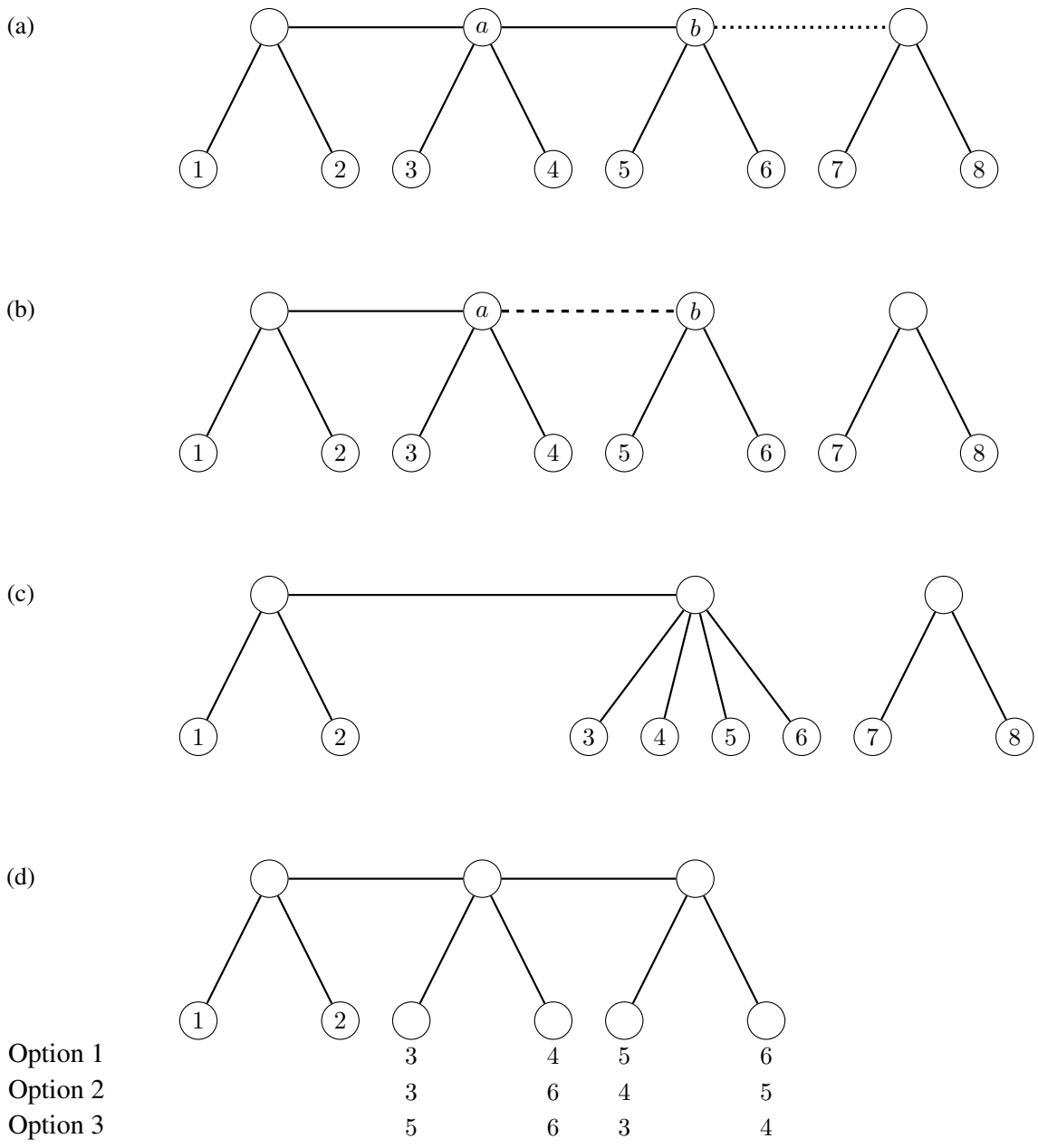


Figure 15: An example of edge contraction. In (a), we see the original tree. In (b), the algorithm splits the tree into two subtrees. Yet, the algorithm could not figure exactly the topology of the left subtree. Instead, it contracts the dashed edge, and the result is shown in (c). The reason for the contraction is: the algorithm could not tell the true topology. In (d), we show multiple topologies that the algorithm might confuse between, leading it to contract the edge. These are labeled Option 1-3.

Before we give the formal proof, let us give an imprecise intuition of the approach. In particular, we show how to compare between the output of the algorithm and the true topology. For the simple case where the output is obtained from the true tree by only deleting edges, these edges are "distant" from the leaves, and so they cannot influence the leaf distribution significantly. Thus, it suffices to study the more complicated case where the subtrees are *not* obtained by simply cutting edges from the true tree. Here, we first compare the true tree to a tree where all edges that appear in two different subtrees have been contracted, as shown in Fig. 16 (b). In the next step, each edge corresponds only to one subtree, and we can detach the different subtrees, as shown in Fig. 16 (c). Lastly, we can reconstruct the edges that were previously contracted, as shown in Fig. 16 (d).

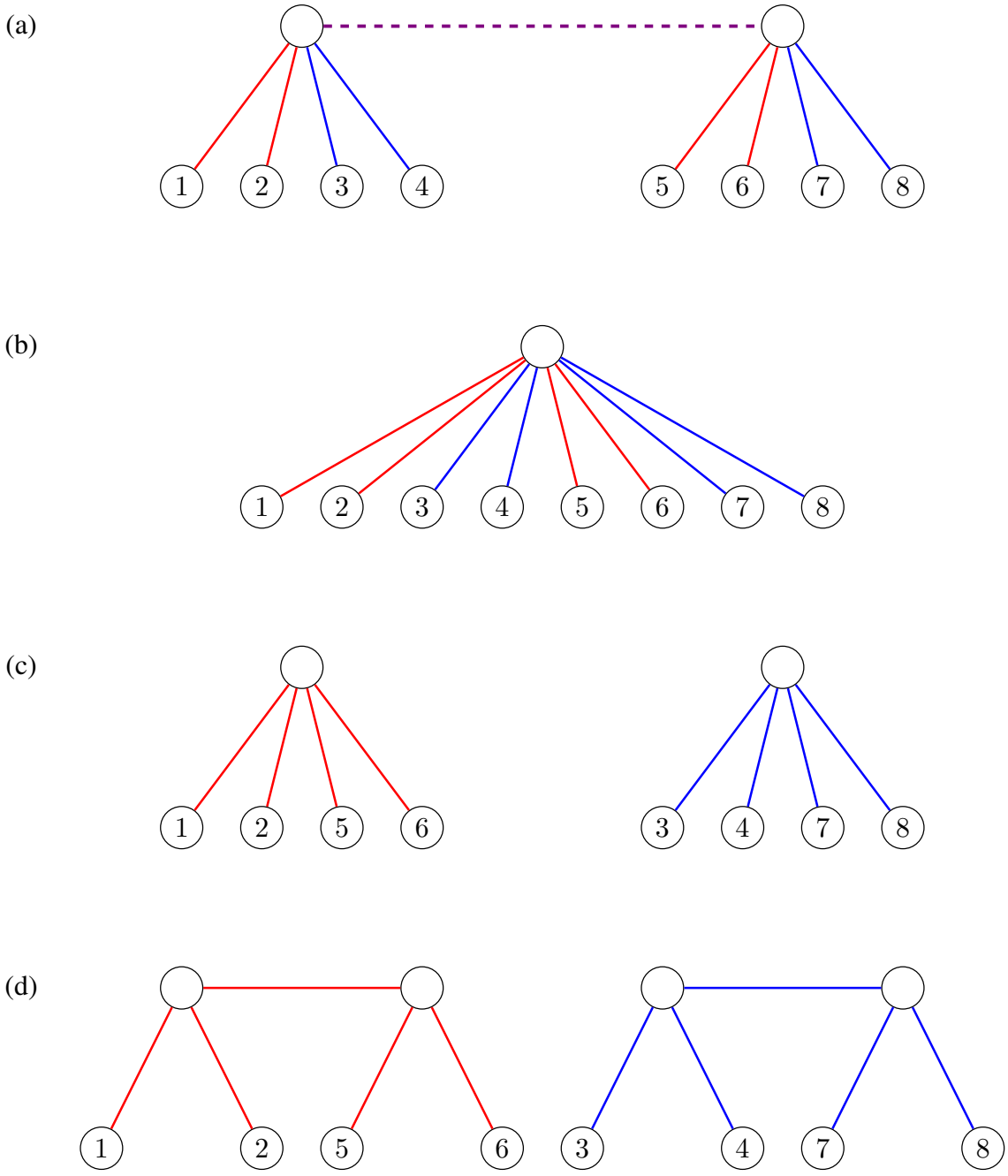


Figure 16: Analyzing the difference between the true topology and the output of the algorithm. In (a), the true tree contains the union of solid and dashed lines. Yet, the algorithm splits the tree into two subtrees, with leaves $\{1, 2, 5, 6\}$ and $\{3, 4, 7, 8\}$. In (d), we can see the output of the algorithm. In order to compare between the true and output topologies, we construct two auxiliary forests: In (b), we contract all the edges that are shared between the two subtree. In this example, this amounts to contracting the dashed edge. Then, in (c), we split the different subtrees. In (d), we reconstruct the edge that was previously contracted. Yet, we reconstruct this edge separately for each subtree.

We can bound the total variation distance for each of these steps using the following guarantees: (1) edges that appear in multiple subtrees have weight close to 1. Hence, contracting them, which is equivalent to changing their weight to 1, does not influence significantly the total variation distance. (2) Leaves of different trees have a small correlation. Hence, detaching the different subtrees does not incur a high cost in total variation.

D.2. Definitions and probabilistic lemmas

We begin with some notations that are specific for this section: given a tree $T = (V, E)$ with weights θ_e on the edges $e \in E$, denote the resulting distribution over the values $x = (x_1, \ldots, x_n)$ on the leaves by $\Pr_{T,\theta}[x]$. The total variation distance between two leaf-distributions of two models, (T, θ) and (T', θ') is denoted by $\mathrm{TV}(\Pr_{T,\theta}[x], \Pr_{T',\theta'}[x])$. The values on the internal nodes are denoted by y_v for each internal node $v \in V$, and $\Pr_{T,\theta}[x, y]$ denotes the joint distribution over the leaves and internal nodes. We continue with some definitions:

Definition 22 (Edge contraction) *Given a graph $G = (V, E)$ and an edge $e = \{u, v\}$ in the graph, we say that a graph $G' = (V', E')$ is obtained from G by contracting e if G' is the result of removing e from the graph and identifying u and v as a single vertex. Namely, if the new vertex is denoted z, then*

$$V' = (V \setminus \{u, v\}) \cup \{z\}$$

and

$$E' = \left(E \setminus \left\{\{y, w\} \in E : y = \{u, v\} \wedge w \in V\right\}\right) \cup \left\{\{z, w\} : \{u, w\} \in E \vee \{v, w\} \in E\right\}.$$

For an example of an edge-contraction, see Fig. 15 (b)-(c): The dashed edge in (b) is contracted, resulting in the graph shown in (c).

We now define the contraction of all degree-2 nodes:

Definition 23 *Given a graph $G = (V, E)$, we say that $G' = (V', E')$ is obtained from G by contracting all the degree-2 nodes, if G' is obtained from G using the following process:*

- *$G' \leftarrow G$.*
- *While G' contains a node w of degree 2:*
  - *Contract one of the edges incident with w.*

For an example of a contraction of degree-2 nodes, see Fig. 13, (b)-(c): In (b), there are some nodes of degree 2, labeled a-f. In (c), we can see the result of contracting these nodes.

Definition 24 (Subtrees induced by a set of leaves) *Given a tree T and a subset S of the leaves of T, the* subtree of T induced by S *is the tree that is obtained from T by removing all the edges and all the nodes that are not in any path between two leaves $i, j \in S$.*

In other words, the subtree of T induced by S is the minimal subtree of T that contains S. For an example, in Fig. 13 (a), a tree is depicted, and its subtrees induced by $\{1, 2, 3, 5\}$ and $\{4, 6, 7, 8\}$ are depicted in Fig. 13 (b-1) and (b-2), respectively.

Throughout the proof, we will modify graphs by contracting edges. Whenever we contract an edge, we identify its two endpoints as a single vertex. If we contract multiple edges, the resulting graph may identify even more than two edges of the original as one edge. If a graph T' is the result of multiple edge-contractions applied on a graph T, then, for each vertex v' of T', *the set of preimages of v' under the transformation from T to T'* is defined as the set of all vertices of T that were identified into v'. To be more formal, we provide the following definition:

Definition 25 *Let $T' = (V', E')$ be obtained from $T = (V, E)$ via a sequence of edge contractions. For any vertex $v' \in T'$,* the set of preimages of v under the transformation from T to T' *is defined as the following set, which we denote here by $A_{v'}$:*

- *Start with the tree T, and define $A_v = \{v\}$ for each $v \in V$.*
- *For any contraction of an edge (u, v) into a single vertex w:*
  - *Define $A_w = A_u \cup A_v$.*
- *Return $A_{v'}$ for each vertex $v' \in V'$.*

Lastly, notice that if we contract edges then some of the nodes might change names. Hence, an edge that was connecting between two nodes (u, v) in the original tree, might connect two other nodes in the contracted tree. Yet, the edge's function remain the same. Hence, we define *the analogue of an edge e* in the contracted graph:

Definition 26 *Let T' be a tree that results from another tree T via a sequence of edge contractions. Let (u, v) be an edge in T that is not contracted. Then,* the analogue of (u, v) in T' *is the obtained from (u, v) in the following fashion:*

- *Set $e' \leftarrow (u, v)$.*
- *For any contraction of edge (z, w) into a node q that T undergoes:*
  - *If $u \in \{z, w\}$ then $e' \leftarrow (q, v)$.*
  - *Otherwise, if $v \in \{z, w\}$ then $e' \leftarrow (u, q)$,*
- *Return e'.*

We continue with presenting some auxiliary lemmas that will be used for the proof.

Lemma 27 *Let $T = (V, E)$ and let $(\theta_e)_{e \in E}$ denote some weight-vector on the edges. Let θ' denote a weight vector that differs only on one edge e'. Then,* $\mathrm{TV}(\Pr_{T,\theta}[x], \Pr_{T,\theta'}[x]) \leq |\theta'_{e'} - \theta_{e'}|/2$.

Proof Due to the equivalent definition of total variation in terms of coupling, it is sufficient to produce a coupling between $x \sim \Pr_{T,\theta}$ and $x' \sim \Pr_{T',\theta'}$ such that $\Pr[x \neq x'] \leq |\theta'_{e'} - \theta_{e'}|/2$. While x and x' denote the values of the leaves, we will use y and y' to denote the values on the internal nodes, such that (x, y) and (x', y') are jointly sampled from $\Pr_{T,\theta}$ and $\Pr_{T',\theta'}$, respectively. We will produce a coupling between (x, y) and (x', y') such that $\Pr[(x, y) \neq (x', y')] \leq |\theta'_{e'} - \theta_{e'}|/2$ and this suffices to conclude the proof.

Let $e' = (u, v)$ denote the *single* edge where θ and θ' differ. We produce the coupling as follows:

- We start by sampling y_u uniformly from $\{-1, 1\}$ and $y'_u = y_u$
- Then, we sample y_v such that $\Pr[y_v = y_u] = (1 + \theta_e)/2$. Similarly, we sample y'_v such that $\Pr[y'_v = y'_u] = \Pr[y'_v = y_u] = (1 + \theta'_e)/2$. Note that we can couple y_v and y'_v such that $\Pr[y_v \neq y'_v] = \mathrm{TV}(y_v \mid y_u,\ y'_v \mid y_u) = |\theta_e - \theta'_e|/2$.
- Next, we will sample the remaining values of x, y conditioned on y_u and y_v, and the remaining values of x', y' conditioned on y'_u and y'_v. If $y_u = y'_u$ and $y_v = y'_v$, then these two conditional distributions are the same, hence, we can sample such that $(x, y) = (x', y')$. Otherwise, we will sample x, y, x' and y' arbitrarily.

Notice that with probability $1 - |\theta_{e'} - \theta'_{e'}|/2$, $y_u = y'_u$ and $y_v = y'_v$. Hence, $\Pr[(x, y) = (x', y')] \geq 1 - |\theta_{e'} - \theta'_{e'}|/2$, as required to complete the proof. ■

Lemma 28 *Let T be a tree and θ a weight-function on its edges. Let T' be obtained from T by contracting an edge e with $\theta_e = 1$ and let θ' denote the restriction of θ to the edges of T'. Then, for any values $x = (x_1, \ldots, x_n)$ on the leaves, $\Pr_{T,\theta}[x] = \Pr_{T',\theta'}[x]$.*

Proof Notice that by contracting the edge, the pairwise correlations α_{ij} between any two leaves do not change, hence the leaf distributions are identical (this is a known fact and it also follows directly from Lemma 9). ■

D.3. Proof body

We are ready to present the results of Daskalakis et al. (2009). They are written in a slightly different way than was originally present, but we translate their guarantees to our notation. (See Section D.5 for translating their guarantees).

Theorem 29 *There is a polynomial-time algorithm, for learning some unknown tree $T^* = (V^*, E^*)$, whose properties are presented below. Its inputs are:*

- *Approximate correlations, $\hat{\alpha}_{ij}$, for any two leaves $i, j \in [n]$. These satisfy the guarantee that there exists an Ising model $\Pr_{T^*,\theta^*}$, whose correlations α^*_{ij} satisfy: $|\alpha^*_{ij} - \hat{\alpha}_{ij}| \leq \eta$, for any two leaves i, j and for some $\eta \in (0, 1/2]$.*
- *Parameters $\xi, \delta > 0$ such that $\xi\delta \geq \eta$.*

The algorithm outputs a forest, whose connected components are trees, $\tilde{T}_1 = (\tilde{V}_1, \tilde{E}_1), \ldots, \tilde{T}_R = (\tilde{V}_R, \tilde{E}_R)$ with the following guarantees: (below, $C > 0$ is a universal constant)

- *Let S_r denote the set of leaves of $\tilde{T}_r$ for any $r \in [R]$. Then, $\{S_1, \ldots, S_R\}$ is a partition of the set of leaves of T^*.*
- *For all $r = 1, \ldots, R$, denote by $T_r = (V_r, E_r)$ the subtree of T^* induced by S_r, as defined in Definition 24. Then, each tree $\tilde{T}_r$ is obtained from T_r using the following operations:*
  - *Contract a subset of the edges. Only edges e of weight $\theta^*_e \geq 1 - C\xi$ can be contracted.*

– *Contract all the nodes of degree-2 from the resulting tree.*

- *Any edge e that is common to more than one of the trees $\{T_1, \ldots, T_R\}$, satisfies $\theta^*_e \geq 1 - C\xi$.*
- *Any leaves i, j that belongs to different sets from $\{S_1, \ldots, S_R\}$, satisfy $|\alpha^*_{ij}| \leq C\sqrt{\delta}$.*

For example, in Fig. 15 (a) a tree is depicted, whereas the output of the algorithm is given in Fig. 15 (c). Since the dashed edge e connecting nodes a and b is contracted in the output, its weight must satisfy $\theta^*_e \geq 1 - \Omega(\xi)$. Further, since nodes 1 and 7 reside in different subtrees in the output of the algorithm, their correlation must satisfy $|\alpha^*_{17}| \leq O(\sqrt{\delta})$. For another example, see Fig. 13: in (a), the original tree is depicted, whereas, the output is the forest in (c). The induced trees T_1 and T_2 are shown in (b). Since the dashed edges in (a) are shared by both induced subtrees, their weight satisfies $\theta^*_e \geq 1 - \Omega(\xi)$.

Below, we will prove the following central Lemma:

Lemma 30 *Let $\eta > 0$ be a parameter and T^* be an unknown tree with weight vector θ^* and pairwise correlations α^* between the leaves. Suppose we execute Algorithm 2 with the following inputs:*

- *Pairwise correlations $\hat{\alpha}_{ij}$ that satisfy $|\hat{\alpha}_{ij} - \alpha^*_{ij}| \leq \eta$.*
- *Parameters δ, ξ such that $\xi\delta \geq \eta$ and $\xi = C_1/n$, for some universal constant $C_1 > 0$.*
- *Parameter $\hat{\eta}$ that satisfies $\hat{\eta} = C_2 n\xi + \eta$, for some universal constant $C_2 > 0$.*

Recall that the algorithm outputs a weighted forest, and denote the forest by $\tilde{F}$ and the weights by $\hat{\theta}$. Then, $\mathrm{TV}(\Pr_{T^*,\theta^*}[x], \Pr_{\tilde{F},\hat{\theta}}[x]) \in O(n^3\xi + n^2\sqrt{\delta} + \eta)$. (We note that an Ising model over a forest is defined by taking the different tree components to be independent.)

The remainder of this section is dedicated to the proof of Lemma 30. In Section D.4 we conclude the proof of Theorem 3 (unknown topology), by substituting the parameters ξ, δ and η appropriately using the finite-sample estimates.

To analyze the algorithm, we will create auxiliary trees $T^{(i)} = (V^{(i)}, E^{(i)})$ with weight function $\theta^{(i)}$ and pairwise correlations $\alpha^{(i)}_{ij}$ (for $i = \{1, 2, 3\}$) that interpolate between the true parameters T^*, θ^*, and the Algorithm 2's output $(\tilde{F}, \tilde{\theta})$. To bound the total variation distance between (T^*, θ^*) and $(\tilde{F}, \tilde{\theta})$, we apply triangle inequality after individually bounding the total variation of the leaf distributions (i) between (T^*, θ^*) and $(T^{(1)}, \theta^{(1)})$, (ii) between $(T^{(1)}, \theta^{(1)})$ and $(T^{(2)}, \theta^{(2)})$, (iii) between $(T^{(2)}, \theta^{(2)})$ and $(T^{(3)}, \theta^{(3)})$, and (iv) between $(T^{(3)}, \theta^{(3)})$ and $(\tilde{F}, \tilde{\theta})$.

Before defining the first intermediate distribution, $(T^{(1)}, \theta^{(1)})$, we recall some definitions. First, recall that the algorithm of Daskalakis et al. (2009) returns a forest whose connected components are $(\tilde{T}_1, \ldots, \tilde{T}_R)$. Each $\tilde{T}_r$ is a modification of T_r, which is defined as the subtree of T^* that is induced by the set of leaves of $\tilde{T}_r$. As an intermediate step, we start by modifying T^* according to the induced subtrees $T_1, \ldots, T_R$. Note that initially, we consider T_r instead of $\tilde{T}_r$, as T_r is *closer* to T^* than $\tilde{T}_r$.

We start by defining $T^{(1)}$ as the tree that is obtained from T^* by contracting all the edges that appear in more than one induced tree T_r Further, the edge-weight for $\theta^{(1)}$ equals θ^* on all the remaining (non-contracted) edges. Note that $T^{(1)}$ still has a single connected component. (For example, in Fig. 17, we contract the purple-dashed lines in (a) because they appear both in the induces red and blue trees. This results in the tree depicted in (b).)

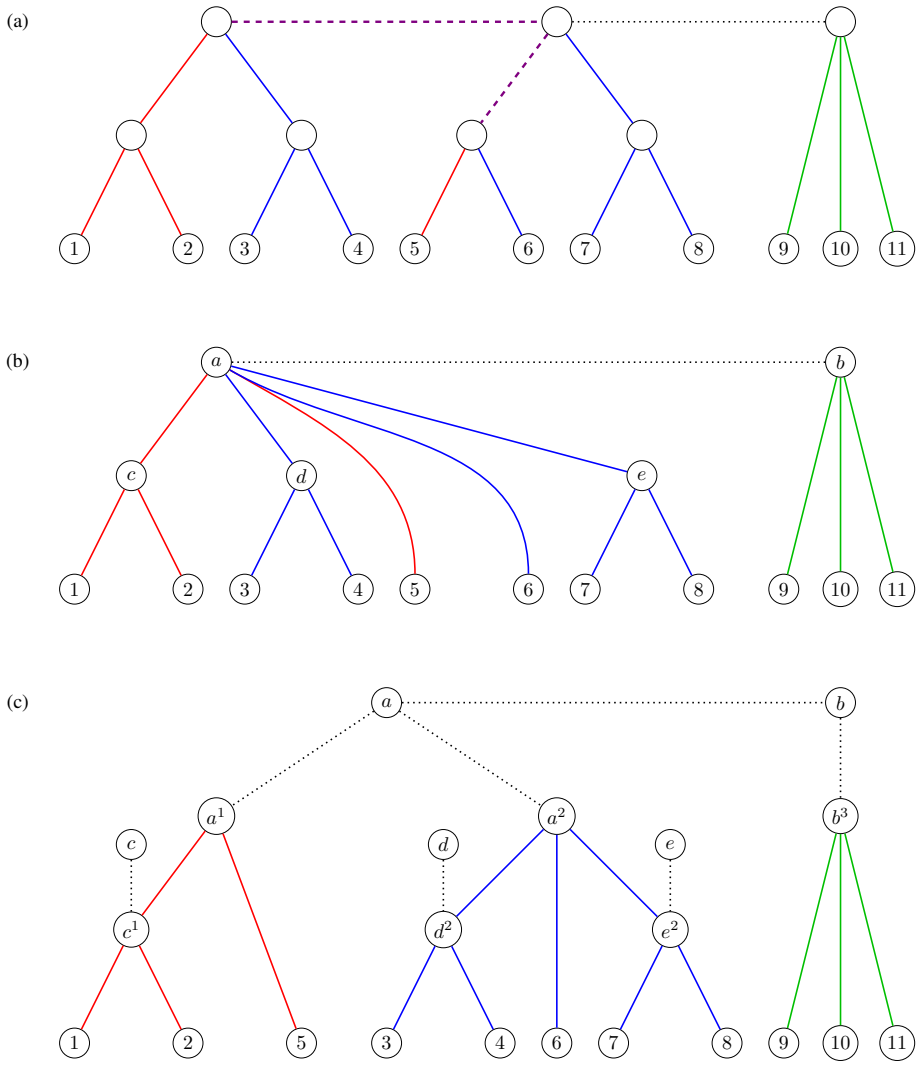


Figure 17: A depiction of the intermediate trees, $T^{(1)}, T^{(2)}$ and $T^{(3)}$ in the analysis of the algorithm. The original tree T^* contains all the edges in (a). The output forest partitions the leaves into three connected components: $S_1 = \{1, 2, 5\}$, $S_2 = \{3, 4, 6, 7, 8\}$ and $S_3 = \{9, 10, 11\}$. The tree T_1, which is the subtree of T^* induced by S_1, contains the red and purple-dashed edged. The tree T_2 contains the blue and purple-dashed edges. And T_3 contains the green edges. In (b), we depict $T^{(2)}$, which is obtained by contracting all the edges that are shared among multiple subtrees T_r, for $r = 1, 2, 3$. In this example, those are the purple-dashed edges. In (c), we depict $T^{(2)}$, which is obtained from $T^{(2)}$ by adding auxiliary nodes: for each internal node v and each subtree T_r that it touches, we create a new node v^r, connect it with v with an edge of weight 1, and further, connect to v^r all the edges in T_r that were incident with v in $T^{(1)}$. The forest $T^{(3)}$ is obtained from $T^{(2)}$ by removing all the edges that are incident with the nodes a, b, c, d, e. These are the dashed edges.

We would like to bound the total variation distance between $\Pr_{T^*,\theta^*}[x]$ and $\Pr_{T^{(1)},\theta^{(1)}}[x]$. Notice that for each contracted edge e, we have from Theorem 29 that $\theta_e \geq 1 - C\xi$ for some universal constant $C > 0$. Contracting each such edge e is equivalent to modifying θ_e to equal 1, as argued in Lemma 28. Hence, the total variation distance for each contraction is bounded by $C\xi/2$ from Lemma 27. By the triangle inequality, since we contract at most $O(n)$ edges, then $\mathrm{TV}(\Pr_{T^*,\theta^*}[x], \Pr_{T^{(1)},\theta^{(1)}}[x]) \leq O(n\xi)$.

Next, we continue to present the second intermediate model, parameterized by $T^{(2)} = (V^{(2)}, E^{(2)})$ and $\theta^{(2)}$. To obtain $T^{(2)} = (V^{(2)}, E^{(2)})$ and $\theta^{(2)}$, recall that each vertex of $T^{(1)} = (V^{(1)}, E^{(1)})$ may correspond to multiple vertices of T^* since $T^{(1)}$ was obtained by contracting edges in T^*. For any $v \in V^{(1)}$, we denote by A_v the set of preimages of v under the transformation from T^* to $T^{(1)}$ (see Definition 25 and the paragraph above this definition). In other words, A_v is the set of nodes of T^* that were contracted into v. Denote by $\rho(v)$ the set of indices $r \in \{1, \ldots, R\}$ of trees T_r that intersect a node that was contracted into v. In other words, $\rho(v)$ is the set of indices r such that $A_v \cap V_r \neq \emptyset$, where V_r is the set of vertices of the subtree T_r. Intuitively, for the construction of $T^{(2)}$, we would like to make $\rho(v)$ copies of node v, each for a subtree that contains v. To be precise, $T^{(2)}$ is constructed as follows:

- The set of vertices $V^{(2)}$ of $T^{(2)}$ is obtained from $V^{(1)}$ by adding a new vertex v^r, for each v and r such that $r \in \rho(v)$.
- For any $v \in V^{(1)}$ and $r \in \rho(v)$, define an edge (v, v^r) of weight $\theta^{(2)}(v, v^r) = 1$.
- For any edge $(u, v) \in E^{(1)}$, define a new edge in $E^{(2)}$ according to the following considerations:
  - If $|\rho(u) \cap \rho(v)| = 1$, denote $\{r\} = \rho(u) \cap \rho(v)$ and add an edge (u^r, v^r) in $E^{(2)}$, with weight $\theta^{(2)}_{(u^r,v^r)} = \theta^{(1)}_{(u,v)}$. This replaces the edge (u, v).
  - If $|\rho(u) \cap \rho(v)| = \emptyset$, add an edge (u, v) to $E^{(2)}$ with weight $\theta^{(2)}_{(u,v)} = \theta^{(1)}_{(u,v)}$
  - It is impossible that $|\rho(u) \cap \rho(v)| > 1$, otherwise, (u, v) would have been contracted.

(See Fig. 17 for an example of a tree $T^{(2)}$.) We note that $T^{(1)}$ can be obtained from $T^{(2)}$ by contracting all the edges (v, v^r) for $v \in V^{(1)}$ and $r \in \rho(v)$. Hence, from Lemma 28, $\Pr_{T^{(2)},\theta^{(2)}}[x] = \Pr_{T^{(1)},\theta^{(1)}}[x]$ for all values x on the leaves. In particular,

$$\mathrm{TV}\left(\Pr_{T^{(2)},\theta^{(2)}}[x], \Pr_{T^{(1)},\theta^{(1)}}[x]\right) = 0.$$

Next, we define $T^{(3)}$ and $\theta^{(3)}$. Recall that the vertices of $T^{(2)}$ are of two types: (1) vertices that are copies of those in $V^{(1)}$; and (2) vertices v^r for $v \in V^{(1)}$ and $r \in \rho(v)$. Then, $T^{(3)}$ is obtained from $T^{(2)}$ by removing all the edges incident to the vertices of category (1), as depicted in Fig. 17 (c). This creates a *forest*, and we remove from this forest each connected component that is disconnected from the leaves. We prove the following lemma:

Lemma 31 *There are R connected components in $T^{(3)}$, and the sets of leaves of the connected components are exactly the sets of leaves of $T_1, \ldots, T_R$. Namely, for each T_r there exists one connected component of $T^{(3)}$ that has the same leaf-set as T_r.*

Proof First, we argue that any two leaves that are in the same tree T_r, are also in the same connected component of $T^{(3)}$. Let i, j be two leaves of T_r and consider the edges on the path P between them in T_r. For any such edge (u, v), there are two possibilities: (1) This edge was contracted at some point, and there is some $w \in T^{(1)}$ such that $u, v \in A_w$. (2) This edge was not contracted, and there exist some (w, z) in $E^{(1)}$ that is the analogue of (u, v) in $T^{(1)}$ (see Definition 26 and its preceding discussion). By definition of $T^{(2)}$, this edge is moved in $T^{(2)}$ to connect (w^r, z^r). In particular, the path from i to j in $T^{(2)}$ contains only vertices of the form q^r. Hence, this path is not disconnected in $T^{(3)}$ from the graph.

Next, we argue why two leaves i, j in two different trees T_r and T_s, respectively, cannot be connected in $T^{(3)}$. Indeed, the parent of leaf i in $T^{(2)}$ must be of the form v^r while the parent of j must be of the form v^s. Notice that in $T^{(3)}$ there is no edge connecting a vertex from v^r with a vertex from v^3, hence, these two leaves are necessarily disconnected.

■

We would like to use Theorem 1 to bound the total variation distance between $\Pr_{T^{(2)},\theta^{(2)}}[x]$ and $\Pr_{T^{(3)},\theta^{(3)}}[x]$, by analyzing the change in the pairwise correlations. To do so, we use the fact that the pairwise correlations that have changed are only those between leaves i, j of different trees T_r and T_s, respectively, and by Theorem 29 we have that in T^* those have correlation $|\alpha^*_{ij}| \leq C\sqrt{\delta}$ where δ is defined in Theorem 29 and $C > 0$ is a universal constant. Notice that, though, the correlation between two leaves in $T^{(2)}$ can be greater than their correlation in T^*. As argued above, the correlation in $T^{(2)}$ equals that in $T^{(1)}$. To compare between the correlations in $T^{(1)}$ and T^*, recall that $T^{(1)}$ is obtained from T by contracting edges. Yet, any contracted edge has weight at least $1 - C\xi$, hence, contracting the edge can increase the correlation by at most $1/(1 - C\xi)$. Since any path between two leaves can contain at most n edges, the contraction can increase the correlation by at most $1/(1 - C\xi)^n$. By assumption, $\xi \leq 1/(Cn)$, hence this factor is at most a constant. Under that assumption, the correlation in $T^{(2)}$ between any two leaves of different subtrees T_r is bounded by $O(\sqrt{\delta})$. Since in $T^{(3)}$ their correlation becomes 0, this implies that the pairwise correlations change by at most $O(\sqrt{\delta})$.

Hence, for any i, j, $|\alpha^{(2)}_{ij} - \alpha^{(3)}_{ij}| \leq O(\sqrt{\delta})$. Since $T^{(2)}$ and $T^{(3)}$ both share the same underlying graph (as removing edges can be done by just replacing the weight with 0), it follows from Theorem 1 that the total variation distance between the leaf distributions of $T^{(2)}$ and $T^{(3)}$ is bounded by $O(n^2\sqrt{\delta})$.

Next, we would like to bound the total variation distance between the leaf distribution of $T^{(3)}$ and the output of the algorithm. Since $T^{(3)}$ and $\tilde{F}$ have the same connected components, the leaf distributions of both factorize the same. In particular, the leaf-sets $S_1, \ldots, S_R$ of $T_1, \ldots, T_r$ are independent in these product distributions. To bound the total variation distance between the leaf distributions of $T^{(3)}$ and $\tilde{F}$, it suffices to bound the total variation distance with respect to each connected component separately, and then sum the bounds for each component. Hence, we will fix some $r \in \{1, \ldots, R\}$. For this end, let us analyze the weights $\tilde{\theta}_e$ given by Algorithm 2 on the tree $\tilde{T}_r$. Recall that the last step of this algorithm is to use Algorithm 1 on the tree $\tilde{T}_r$ with some parameter $\eta' > 0$ and correlations $\hat{\alpha}_{ij}$ that were estmated from samples of the original tree T^*. Further, recall that Algorithm 1 is guaranteed to return some weights $\tilde{\theta}_e$ on the edges, such that the pairwise correlations between the leaves, which we denote by $\tilde{\alpha}_{ij}$, are η'-close to the correlations $\hat{\alpha}_{ij}$ that were given to it as input, namely, $|\tilde{\alpha}_{ij} - \hat{\alpha}_{ij}| \leq \eta$ for any pair i, j of leaves. Yet, Algorithm 1

will succeed only if there exist such weights $\tilde{\theta}_e$ that satisfy the above constraint. To that end, we claim the following:

Lemma 32 *Fix $r \in \{1, \ldots, R\}$. Then, there exist weights θ'_{ij} to the edges of $\tilde{T}_r$ such that the corresponding pairwise correlations, α'_{ij} satisfy $|\alpha'_{ij} - \hat{\alpha}_{ij}| \leq \eta + O(n\xi)$ for any leaves $i, j \in S_r$.*

Proof Notice that it is sufficient to find weights θ'_e such that $|\alpha'_{ij} - \alpha^*_{ij}| \leq O(n\xi)$, since $|\hat{\alpha}_{ij} - \alpha^*_{ij}| \leq \eta$, by the assumption in Lemma 30. Since T_r is the subtree induced by the set of leaves S_r, our goal is to show that α'_{ij} is $O(n\xi)$-close to the pairwise correlation of i and j accross T_r. Hence, this is what we will do. As a first solution, we propose to set for each edge e its weight θ'_e to equal its corresponding weight θ^*_e in T_r. Recall, though, that in the process of transforming T_i to $\tilde{T}_i$, there are two modifications, which implies that we cannot exactly match the edges of $\tilde{T}_i$ with those of T_i. We elaborate below on the transformations and how to set θ'_e given these transformations.

- The first transformation is obtained by contracting some edges of weight $\theta_e \geq 1 - O(\xi)$. For these contracted edges, we will not define θ'_e. Contracting these edges changes the pairwise correlations between the leaves by at most $O(n\xi)$, since there can be at most $O(n)$ edges along each path.
- The second transformation is a contraction of some nodes of degree 2. Yet, for each such contraction, there is an easy way to modify the weights such that the pairwise correlations over the leaves does not change. In particular, if u is a node and v, w are its neighbors, then u is being deleted from the graph and v, w are being connected. If we set the weight of the new edge as a multiplication of the weights of the two old edges, then the pairwise correlations between the leaves do not change. In particular, we will define θ' under this logic: we will track the changes from T_i to $\tilde{T}_i$, and whenever a degree-2 node is being contracted, we modify the weights accordingly: if edges e and e' were contracted to e'', we set the weight of e'' to equal the multiplication of weights of e and e'.

Using the above definition of θ'_e and the above analysis, it follows that $|\alpha'_{ij} - \alpha^*_{ij}| \leq O(n\xi)$. This suffices to complete the proof, as explained above. ■

It follows from Lemma 32 that the execution of Algorithm 1 succeeds, if it is run with $\eta' = Cn\xi + \eta$ and a sufficiently large $C > 0$. This implies that for any $i, j \in S_r$, $|\tilde{\alpha}_{ij} - \hat{\alpha}_{ij}| \leq \eta' \leq \eta + O(n\xi)$. By the triangle inequality, $|\tilde{\alpha}_{ij} - \alpha^*_{ij}| \leq 2\eta + O(n\xi)$. Lastly, following the analysis above, it is easy to show that also $|\alpha^{(3)}_{ij} - \alpha^*_{ij}| \leq O(n\xi)$. Indeed, the only modification from T to $T^{(3)}$ that affects the pairwise correlation between $i, j \in S_r$ is the contraction of edges of $\theta^*_e \geq 1 - O(\xi)$. This affects the pairwise correlation by $O(n\xi)$. By the triangle inequality, we derive that $|\tilde{\alpha}_{ij} - \alpha^{(3)}_{ij}| \leq O(n\xi + \eta)$. The last step would be to apply Theorem 1 (same topology) to compare between the distribution over $\tilde{T}_i$ and its corresponding connected component in $T^{(3)}$. Yet, this theorem would apply only if the two components have the same topology. While they do not, we note that both trees are contractions of the same tree T_r. Hence, we can view both distributions as defined over the tree T_r, where the contracted edges have weight 1. By Theorem 1 (same topology) we derive that the total variation distance between the two distributions over S_r is bounded by $O(n\xi|S_r|^2)$. By summing over all r, we derive that

$$\mathrm{TV}\left(\Pr_{T^{(3)},\theta^{(3)}}[x],\ \Pr_{\tilde{F},\tilde{\theta}}[x]\right) \leq O\left(n\xi \sum_{r=1}^{r} |S_r|^2\right) \leq O(n^3\xi).$$

By summing up the total variation distances between the auxiliary distributions parameterized by $T^{(i)}$, this concludes the proof of Lemma 30.

D.4. Concluding the proof of Theorem 3 (unknown topology)

We use Lemma 30. First of all, let us optimize ξ and δ for a fixed value of η. This can be achieved by selecting $\delta = \eta^{2/3}n^{2/3}$ and $\xi = \eta^{1/3}n^{-2/3}$. We note that the requirement $\xi \leq O(1/n)$ if $\eta \leq O(1/n)$. The final bound is $O(n^{7/3}\eta^{1/3})$. To get this below ε, we have to set $\eta \leq O(\varepsilon^3/n^7)$. This requires a sample of size $n \geq \Omega(\log(n/\delta)/\eta^2) = \Omega(n^{14}\log(n/\delta)/\varepsilon^6)$.

D.5. Translating the notation of Daskalakis et al. (2009)

We note that Daskalakis et al. (2009) uses a different notation. For convenience, we explain the translation in the Ferromagnetic setting where $\alpha_{ij}, \theta_{kl} \geq 0$, however, in order to transition to the non-Ferromagnetic setting one would simply have to replace these quantities with their absolute values, $|\alpha_{ij}|$ and $|\theta_{kl}|$, respectively.

Instead of edge weight $\theta_{(k,\ell)}$, Daskalakis et al. (2009) use the metric $d_{k,\ell} = -\log \theta_{(k,\ell)}$. Instead of $\alpha_{i,j}$ they use the metric $d_{i,j} = -\log \alpha_{i,j}$. They denote the true (underlying) metric by d, whereas they assume that the algorithm receives a (τ, M)-*distorted metric* on the leaves, denoted $\hat{d}$. This means that $|\hat{d}_{i,j} - d_{i,j}| \leq \tau$ whenever $d_{i,j} \leq M$. Using our notation, this means that $|\log \hat{\alpha}_{ij} - \log \alpha^*_{i,j}| \leq \tau$ whenever $\alpha^*_{ij} \geq e^{-M}$. Equivalently,

$$-\tau \leq \log \hat{\alpha}_{ij} - \log \alpha^*_{i,j} \leq \tau \quad \text{whenever } \alpha^*_{ij} \geq e^{-M}$$

which is equivalent to

$$e^{-\tau} \leq \hat{\alpha}_{ij}/\alpha^*_{i,j} \leq e^{\tau} \quad \text{whenever } \alpha^*_{ij} \geq e^{-M} \ . \tag{21}$$

We will show how their guarantees can be implied from our guarantees. First, notice that in order for (21) to hold, it is sufficient to assume that $\tau \leq 1$ and

$$(1-\tau/2) \leq \hat{\alpha}_{ij}/\alpha^*_{i,j} \leq (1+\tau/2) \quad \text{whenever } \alpha^*_{ij} \geq e^{-M} \ .$$

If we substitute $\xi = \tau/2$ and $\delta = e^{-M}$, the last inequality substitutes to

$$\alpha^*_{i,j} - \xi\alpha^*_{ij} \leq \hat{\alpha}_{ij} \leq \alpha^*_{i,j} + \xi\alpha^*_{ij} \quad \text{whenever } \alpha^*_{ij} \geq \delta \ ,$$

which is equivalent to

$$|\hat{\alpha}_{ij} - \alpha^*_{i,j}| \leq \xi\alpha^*_{ij} \quad \text{whenever } \alpha^*_{ij} \geq \delta \ . \tag{22}$$

Eq. 22 is guaranteed to hold if $|\hat{\alpha}_{i,j} - \alpha^*_{i,j}| \leq \xi\delta$. Since in Theorem 29 we assume that $|\hat{\alpha}_{i,j} - \alpha^*_{i,j}| \leq \eta$, it suffices to assume that $\eta \leq \xi\delta$ in order to imply (22), which in turn implies the conditions in the paper of Daskalakis et al. (2009).

Appendix E. Information theoretic bound

Upper bound. While the result below was known, we prove it for completeness.

Theorem 33 *There is an algorithm that, given m samples from the leaf-marginal of some tree structured Ising model with n leaves, returns another tree structured Ising model whose total variation distance to the original model is bounded by ε, with sample complexity $m = O(n \log(n/\varepsilon)/\varepsilon^2)$.*

While it is apparent that the family of tree-structured Ising models is infinite, we will select a finite set which is an ε-cover in total variation, and then we will use the following result to learn in total variation distance over a finite set:

Theorem 34 (Yatracos (1985)) *Let $\varepsilon, \delta > 0$. Given a finite family $\mathcal{C}$ of distributions and m samples from some arbitrary distribution μ, there exists an algorithm such that, with probability $1 - \delta$, returns a distribution $\hat{\mu} \in \mathcal{C}$ that satisfies:*

$$\mathrm{TV}(\mu, \hat{\mu}) \leq 3 \inf_{\nu \in \mathcal{C}} \mathrm{TV}(\mu, \nu) + \varepsilon,$$

with sample complexity $m = O(\log(|\mathcal{C}|/\delta)/\varepsilon^2)$.

In order to apply Theorem 34, we will construct an ε-cover to the set of tree structured Ising models.

Lemma 35 *For any $\varepsilon > 0$, there exists a family $\mathcal{C}$ of tree-structured Ising models of log cardinality $\log |\mathcal{C}| \leq O(n \log(n/\varepsilon))$, such that for any tree structured Ising model, there exists some model from $\mathcal{C}$ such that the total variation distance between these the two leaf distributions of these models is bounded by ε.*

Proof For completeness, we prove this lemma using Theorem 1, yet, there are more direct ways to prove this lemma.

Each element of $\mathcal{C}$ will be parameterized by the following:

- A tree topology, with n leaves labeled $1, \ldots, n$. There can be at most $n^{O(n)}$ distinct trees.
- For each edge of the tree, its weight θ_e is one of $\{0, 1/M, 2/M, \ldots, 1\}$, where $M = \Theta(n^3/\varepsilon)$. There can be at most $M^{O(n)}$ possibilities to select the weights.

We derive that $|\mathcal{C}| \leq (n/\varepsilon)^{O(n)}$.

Given some tree T and weight θ, we will find an element of $\mathcal{C}$ that approximates it. In particular, we will take the element from $\mathcal{C}$ that has the same structure and additionally, each of its weights are $1/M$ close to θ. It is easy to see from (1) that the pairwise correlations between the leaves α_{ij}, are $O(n/M)$-close in absolute value between the two models. Hence, by Theorem 1 (fixed topology), the two models are $O(n^3/M) \leq \varepsilon$ close in total variation between the leaf distributions, provided that $M \geq \Omega(n^3/\varepsilon)$. ■

To conclude the proof, we use the algorithm of Theorem 34, applying it on an $\varepsilon/4$-cover using the construction in Lemma 35.

Lower bound In order to learn latent tree-structured Ising models, when the topology is unknown, the sample complexity is lower bounded by $\Omega(n \log(n)/\varepsilon^2)$. This follows from Koehler (2020): they prove that the number of samples that are required to learn a *full tree* from samples is $\Omega(n \log(n)/\varepsilon^2)$, yet, this proof extends directly to the setting of latent nodes.