Reconstructing Ultrametric Trees from Noisy Experiments

Eshwar Ram Arunachaleswaran ESHWAR@SEAS.UPENN.EDU
Anindya De ANINDYAD@CIS.UPENN.EDU
Sampath Kannan KANNAN@CIS.UPENN.EDU
Department of Computer and Information Science, University of Pennsylvania

Editors: Shipra Agrawal and Francesco Orabona

Abstract

The problem of reconstructing evolutionary trees or phylogenies is of great interest in computational biology. A popular model for this problem assumes that we are given the set of leaves (current species) of an unknown weighted binary tree and the results of 'experiments' on triples of leaves (a, b, c), which return the pair with the deepest least common ancestor. If the tree is assumed to be an *ultrametric* (i.e., with all root-leaf paths of the same length), the experiment can be equivalently seen to return the closest pair of leaves. In this model, efficient algorithms are known for reconstructing the tree.

In reality, since the data on which these 'experiments' are run is itself generated by the stochastic process of evolution, it is noisy. In all reasonable models of evolution, if the branches leading to the three leaves in a triple, separate from each other at common ancestors that are very close to each other in the tree, the result of the experiment should be close to uniformly random. Motivated by this, in the current paper, we consider a model where the noise in an experiment on any triple is just dependent on the three pairwise distances (referred to as *distance-based noise*). Our results are the following:

1. Suppose the length of every edge in the unknown tree is at least $\tilde{O}(\frac{1}{\sqrt{n}})$ fraction of the length of a root-leaf path, where n is the number of leaves. Then, we give an efficient algorithm to reconstruct the topology of the unknown tree for a broad family of distance-based noise models. Further, we show that if the edges are asymptotically shorter, then topology reconstruction is information-theoretically impossible.
2. Further, for a specific distance-based noise model – which we refer to as the *homogeneous noise model* – we show that the edge weights can also be approximately reconstructed under the same quantitative lower bound on the edge lengths. Note that in the noiseless case, such reconstruction of edge weights is impossible.

The phylogeny reconstruction problem is essentially the problem of hierarchical clustering. Our result here apply to a suitably defined version of this problem.

Keywords: Ultrametric Binary Trees, Noisy Experiments, phylogenetic trees, distance based noise, hierarchical clustering

1. Introduction

The problem of clustering is an important computational problem and a primitive that is used in multiple domains with the goal of grouping elements based on some underlying notion of distance in order to understand the relationship among them. In the standard clustering problem, the set of given elements is to be partitioned into a few sets with the goal of putting similar elements in the same partition (captured by minimizing an objective function). A natural and well studied variant (and

generalization) of this problem is *hierarchical clustering*, where the goal is to find a hierarchical partition of the elements, in which groups of elements form a nested structure. Equivalently, a hierarchical clustering can be thought of as a rooted tree with the elements at the leaves. Thus, the task of hierarchical clustering can be seen as the task of recovering the underlying unknown rooted tree.

Naturally, canonical applications of the problem of hierarchical clustering are settings where there is an underlying tree structure – examples include learning evolutionary trees of a set of species and evolutionary trees of languages. In particular, the problem of reconstructing evolutionary trees or phylogenies from data about extant species is an important one in computational biology Saitou and Nei (1987); Semple and Steel (2003); Farach and Kannan (1999); Mossel (2007); Kearney et al. (1997); Ailon and Charikar (2005); Amir Ben-Dor and Benny Chor and Dan Graur and Ron Ophir and Dan Pelleg (1998); Daskalakis et al. (2006); Erdős et al. (1999); Kannan et al. (1996); Emamjomeh-Zadeh and Kempe (2018) and is the principal motivation for this paper.

In order to define a hierarchical clustering problem, we need the precise notion of similarity, as well as the mode by which the algorithm gets access to this information. The formulation that is closest to the one in this paper is from Kannan et al. (1996). Here, the evolutionary tree is assumed to be an *ultrametric* binary tree, which is a weighted, rooted tree in which all root-leaf paths have the same length. This assumption is often justified in the computational biology literature based on the so-called molecular clock hypothesis, whereby the lengths of edges correspond to the evolutionary time that they represent. Then since all extant species are alive today and they are the leaves of this tree, they have all evolved for the same length of time. Consequently, all root to leaf paths have the same length. The model in Kannan et al. (1996) assumes that we are able to perform experiments on any 3 extant species (leaves) a, b, and c and the result of the experiment (alternately, referred to as query) is the pair that is closest together, i.e., has the most recent least common ancestor. In this model, the authors Kannan et al. (1996) give efficient algorithms for reconstruction of the tree topology – in fact, they give several procedures, each obtaining a different tradeoff between the running time and the number of experiments (i.e., queries).

A principal shortcoming of Kannan et al. (1996) is the assumption that the experiments do not have any noise – on any triple (a, b, c), the queries always returns the closest pair. However, experiments are often noisy and thus it is natural to ask if we can design a tree-reconstruction algorithm which is tolerant to this noise. Several works such as Brown and Truszkowski (2013); Gang Wu and Ming-Yang Kao and Guohui Lin and Jia-Huai You (2008); Emamjomeh-Zadeh and Kempe (2018) have explored this theme. In particular, in Emamjomeh-Zadeh and Kempe (2018), the authors gave an algorithm to reconstruct the (topology of the) ultrametric tree with $O(n \log n)$ queries even in presence of noise. However, in all the previous works Brown and Truszkowski (2013); Gang Wu and Ming-Yang Kao and Guohui Lin and Jia-Huai You (2008); Emamjomeh-Zadeh and Kempe (2018), the probability of success is identical across queries – in other words, each experiment is assumed to succeed with some fixed probability $p > 1/2$.

In this paper, we study the tree reconstruction problem under a broad family of noise models where the noise on any triple (a, b, c) is just dependent on the three pairwise distances between the leaves a, b and c – note that by the ultrametric assumption, the two largest distances are the same. We refer to such noise models as *distance-based noise models*. Our motivation is that if the data at the leaves is generated by a process like evolution then the data at each leaf is the result of a set of stochastic mutations encountered on the path from the root to that leaf. If we have three leaves a, b, and c where the least common ancestor of a and b is at distance 1 from the leaves, while the

least common ancestor of c with either a or b is at distance $1+\epsilon$, then the expected number of mutational differences between a or b on the one hand and c on the other hand, is quite close to the expected number of mutational differences between a and b. Any experiment trying to assess which pair is closest based on mutational differences would therefore have a good probability of identifying the wrong pair in this situation. Finally, similar to Brown and Truszkowski (2013) (as well as many other results in the phylogenetic reconstruction literature), we assume that the noise in each experiment is *permanent* – i.e., repeating the same experiment always yields the same outcome. Since repetitions of an experiment will use the same noisy data, this assumption is justified. This naturally rules out repeating the same experiment as a way to *denoise the answers*, thus making the algorithm design more challenging.

Our results: We now give an overview of our results. First of all, by rescaling the edge lengths (alternatively referred to as weights), we can assume that all root to leaf paths are of unit length. For our first algorithmic result, we define a so-called *general noise model*. This is any distance-based noise model that satisfies some mild properties we refer to as the *monotonicity* and *anti-Lipschitzness* properties. Informally, these properties say that the probability of getting the correct pair is always greater than $1/3$ (i.e. the experiments are better than uniformly random answers) and that for each value of the maximum distance in a triple, the probability of the experiment returning the closest pair in the triple is sufficiently sensitive to the distance between this pair.

More specifically, this success probability is monotonically decreasing as a function of the distance between the closest pair, and anti-Lipschitzness guarantees that this rate of decrease is at least a constant. We argue that one should expect to see these conditions satisfied qualitatively by any reasonable distance-based noise model for experiments to find the closest pair of a triple. The exact conditions are described in Section 3.

Our first algorithmic result shows that in the general noise model, as long as each edge length is at least $\tilde{\Omega}(n^{-1/2})$ (where n is the number of leaves), there is an algorithm that takes the results of the $\binom{n}{3}$ experiments and reconstructs the topology of tree with high probability (Theorem 1). We also show a matching lower bound – namely, if the minimum edge length is $\tilde{o}(n^{-1/2})$, then it is information-theoretically impossible to recover the topology of the tree (Theorem 6). Thus, together Theorem 1 and Theorem 6 give the minimal requirements under which topology of the tree can be recovered in the general noise model. Intuitively, these theorems quantify how "well-separated" the vertices of an ultrametric tree need to be, for us to be able to exactly reconstruct the topology of the tree under distance-based noise.

For our second algorithmic result, we explore a special instance of the general noise model which we refer to as the *homogeneous noise model*. Let us denote this model by $\mathbf{Q}_h(\cdot)$ – then, on the triple (a,b,c), the probability of returning the pair (a,b) is given by

$$\Pr[\mathbf{Q}_h(a,b,c)=(a,b)] = \frac{d(a,c)+d(b,c)}{2(d(a,b)+d(b,c)+d(a,c))},$$

where $d(\cdot,\cdot)$ denotes the distance function on the tree. Under natural 'boundary conditions' that the probability of any pair being returned should approach 1/3 as the 3 pairwise distances approach each other, that the probability of the closest pair being returned should approach some higher constant value when the other two distances tend to infinity, and that the probability of returning either wrong pair is equal, $\mathbf{Q}_h(\cdot)$ is essentially the only probability function that is a ratio of linear functions. One particularly appealing feature is that the model is invariant upon rescaling of the distance function $d(\cdot,\cdot)$. For the homogeneous noise model, we can achieve a significantly stronger

result than Theorem 1. In particular, in Theorem 5, we show that as long as all the edge weights are at least $\tilde{\Omega}(n^{-1/2})$, there is an efficient algorithm to approximately reconstruct the edge weights. In other words, for the homogeneous noise model, we can not just recover the topology of the tree but the actual distances between leaves. Such a reconstruction of the edge weights is information-theoretically impossible in models such as Kannan et al. (1996); Brown and Truszkowski (2013); Gang Wu and Ming-Yang Kao and Guohui Lin and Jia-Huai You (2008); Emamjomeh-Zadeh and Kempe (2018) where either the queries have no noise or the probability of success does not depend on the pairwise distances. We remark that our techniques for reconstructing distances are quite general, and should be applicable to a broader class of noise models. Determining the conditions under which the entire ultrametric can be reconstructed is left as a topic for future work.

2. Related Work

The problem of reconstructing evolutionary trees has received a lot of attention over the years. There are many formulations of this problem based on the type of data available and the objective function being optimized. Most formulations assume that the observed data is on extant species or leaves of an unknown tree. Objective functions seek to capture properties of the evolutionary process, with the hope that the optimal tree under an objective function is in fact the true evolutionary tree. The most popular formulations are distance-based methods Farach et al. (1995); Erdős et al. (1999); Erdös et al. (1999); Saitou and Nei (1987); Mossel and Roch (2017); Daskalakis and Roch (2013) (where we are given a matrix of distances between leaves and we want to find the best-fitting edge-weighted tree), character-based methods Agarwala and Fernández-Baca (1994); Kannan and Warnow (1994); Mossel and Roch (2005); Mossel and Steel (2007); Steel (2016), where we want to explain the evolution of different characters, each taking on a state in each extant species using the fewest number of state changes, and likelihood methods Neyman (1971); Felsenstein (1981); Farach and Kannan (1999); Roch and Sly (2017), where we assume that evolution is a stochastic process drawn from a family of processes, and want to estimate the most likely parameters. In all cases the data observed at the leaves is the result of the stochastic process of evolution. These formulations and other related types lend themselves naturally to the model considered in this paper.

As noted earlier, the closest formulation to that in our paper is the one introduced by Kannan et al. (1996) on learning an ultrametric tree through experiments involving three leaves. In this paper the authors seek to reconstruct an ultrametric tree given the outcomes of noise-free experiments on triples. They show 3 different algorithms whose run-times vary from $O(n^2)$ to $O(n \log n)$. The algorithm that takes $O(n^2)$ performs is a divide-and-conquer algorithm that performs a number of experiments that is asymptotic to $n \log_2 n$, while the algorithm that runs in $O(n \log n)$ performs a number of experiments that is bounded by $4n \log n$. Since experiments might be a lot more expensive than computational time, the first algorithm might be preferable to the second. This paper motivated several follow-ups such as Emamjomeh-Zadeh and Kempe (2018); Brown and Truszkowski (2013); Gang Wu and Ming-Yang Kao and Guohui Lin and Jia-Huai You (2008) with closely related models. In particular, in Emamjomeh-Zadeh and Kempe (2018), the authors considered a noisy version of these experiments to learn hierarchical clusterings, with each experiment succeeding with some probability $p > 1/2$ or failing adversarially. In the current paper, we consider the same problem but with a different, incomparable noise model. Similar to Emamjomeh-Zadeh and Kempe (2018), the noise in each experiment is independent. However, the noise in our model depends upon the pairwise distances of the triples involved, in contrast to Emamjomeh-Zadeh and

Kempe (2018). This leads to significant differences in the behavior of the noise model - in particular, for three leaves a, b and c, where the pairwise distances are close to each other (in the ultrametric tree), the probability of getting the correct answer can be as small as $1/3 + \theta(\frac{\log n}{\sqrt{n}})$ in our model. An additional feature of our model is that each experiment can only be performed once (similar to Brown and Truszkowski (2013)). In contrast, Emamjomeh-Zadeh and Kempe (2018) allows for repetition of the same experiment multiple times with fresh randomness each time. Finally, we remark that while Emamjomeh-Zadeh and Kempe (2018) allows for repetition of the same experiment, they view the number of experiments (equivalently the query complexity) as a key measure of performance of their algorithm – in fact, their topology reconstruction algorithm has query complexity $O(n \log n)$ (which is essentially optimal). In contrast, the focus of this paper is to identify a broad class of noise models under which tree reconstruction is even possible.

Besides evolutionary biology, "distance based noise models" have also been studied in other reconstruction problem. In Tamuz et al. (2011), Tamuz *et al.* study the following problem: there are n elements with an unknown embedding in the Euclidean space. The algorithm gets noisy answers to *relative similarity queries* and the goal is to reconstruct this embedding. More precisely, the algorithm can query any triple (a, b, c) with the underlying semantics being "Is a closer to b or to c?". On such a query, it gets the pair (a, b) with probability $\frac{d(a,c)}{d(a,b)+d(a,c)}$ (and otherwise the pair (a, c) is returned). Here $d(\cdot)$ is the underlying distance metric. We note that the model is both in form and spirit, very similar to the homogeneous noise model studied in Theorem 5. Indeed, as the distances $d(a, b)$ and $d(a, c)$ approach each other, the response to the query (a, b, c) is basically a coin flip. On the other hand, if one of the distances is much smaller than the other, then the probability of returning the closer pair approaches 1. Along similar lines Van Der Maaten and Weinberger (2012) study the problem of learning a low dimensional embedding of a set of elements in Euclidean space based upon seeing the closest pair in a triplet. Just as in our model, each possible closest pair appears with a probability that depends upon an underlying dissimilarity/ distance function relating these elements.

Distance-based models are also quite popular in the ranking literature. In particular, in the well-known Bradley-Terry-Luce (BTL) model Bradley and Terry (1952); Luce (1959), there are n elements where the i^{th} element is assigned (an unknown) weight $w_i \in [0, 1]$ – thus defining a total order on these elements. The algorithm queries pairs (i, j) and is returned i with probability $w_i/(w_i + w_j)$. Note that this can be interpreted as a noisy comparison query where the probability of returning the larger element depends on the relative scores of the two elements. The goal in the BTL model is to recover the underlying ranking given these noisy comparison queries. Again, the BTL model bears strong syntactic resemblance to our homogeneous noise model (from Theorem 5). In fact, similar to the current paper, in the BTL model, each query can be made at most once. We also note that higher arity generalizations of the BTL model have also been explored in literature, for example, in Plackett (1975); McFadden (1973) (under the name multinomial logistic model).

Outside of distance-based noise models, there is large body of literature in computer science which aims to model relations between elements by an (unknown) embedding in some metric space. The algorithm makes relational queries and gets noisy responses where the noise is governed by the hidden embedding. Several models including the famous stochastic block model McSherry (2001); Mossel et al. (2018); Decelle et al. (2011); Abbe (2017) and its variants Chen et al. (2020) fit this motif and the current paper can be seen as yet another instantiation of this general framework. Further examples from the world of machine learning include Jamieson and Nowak (2011), Klein-

dessner and Luxburg (2014), Agarwal et al. (2007), Hoffer and Ailon (2015), Schultz and Joachims (2003).

3. Model, Notation, and Preliminaries

There is an underlying weighted tree T with the weights constrained such that the distance function between leaves is an ultrametric. We refer to any tree having this property as an ultrametric tree. We assume that the height of the tree $h(T)$ is normalized to 1.

Distance-Based Noise Model: For each triple of leaves (a, b, c), we perform an experiment and get back one of the pairs (a, b), (b, c), or (c, a) probabilistically. Such an experiment will be denoted by $Q(a, b, c)$. Repeating an experiment produces the same answer, and results of distinct experiments are independent of each other. Recall that in an ultrametric, the largest two of the three pairwise distances are equal. Thus we will model the probabilities of different answers as a function of just two distances — d_1, the distance between the closest pair of leaves and d_2 the distance between either of the two other pairs in the triple (So $d_1 \leq d_2$). In our model, the two incorrect pairs have equal probabilities of being returned, which is justified because their pairwise distances are the same. Thus, we define two probability functions: p_{CORRECT} and $p_{\text{INCORRECT}}$ where p_{CORRECT} denotes the probability that the closest pair is returned and $p_{\text{INCORRECT}}$ denotes the probability that each of the other pairs is returned. Thus $\forall d_1, d_2,\ p_{\text{CORRECT}}(d_1, d_2) + 2p_{\text{INCORRECT}}(d_1, d_2) = 1$.

We impose some mild conditions on the probability functions. First, we naturally insist that $p_{\text{CORRECT}} > p_{\text{INCORRECT}}$, since otherwise the output of the experiment is not useful. Second, we require that the probability of returning the correct pair is sufficiently sensitive to the change in distance. In mathematical terms, for any d_2 and $0 < d_1 < d_2$, $\frac{\partial p_{\text{CORRECT}}(d_1,d_2)}{\partial d_1} \leq -\varepsilon$ for some constant $\varepsilon > 0$. We are implicitly also assuming that the probability functions are continuous in each coordinate since we are assuming that their partial derivatives are defined. The latter assumption captures both the monotonicity and anti-Lipschitzness properties mentioned in the introduction.

We will refer to this model as the **distance-based noise model** or the **general noise model**.

When it comes to reconstructing weights (Section 5), we show that this is possible for a specific instantiation of the distance based model, called the **homogeneous model**, denoted by $\mathbf{Q}_h(a, b, c)$. Recall the definition of this model from the introduction - on the triple (a, b, c), the probability of returning the pair (a, b) is given by

$$\Pr[\mathbf{Q}_h(a, b, c) = (a, b)] = \frac{d(a, c) + d(b, c)}{2(d(a, b) + d(b, c) + d(a, c))},$$

where $d(\cdot, \cdot)$ denotes the distance function on the tree.

If $d(a, b)$ is the smallest of the 3 pairwise distances, then the probability of the experiment returning (a, b) is between 1/3 and 1/2. Since the other two distances, $d(a, c)$ and $d(b, c)$ are equal in an ultrametric, the experiment has equal probability of returning (a, c) or (b, c). Thus for pairs a, b that are very close and c that is much farther, the probability of getting the result (a, b) approaches .5, while for triples (a, b, c) whose least common ancestors are very close, the probability of getting any pair approaches $1/3$. In the introduction we provided intuition on why this is a natural model.

Even in this simple model, we cannot hope to reconstruct arbitrary ultrametrics as the following example shows. Suppose the underlying tree is a balanced binary tree, where the edge at depth i has weight $C \cdot 2^{2^i}$. Let the height of the tree be $h = \log n$. (In this paragraph 'height' denotes the number of edges on the longest root-leaf path, and ignores the weights of these edges.) The constant C is

chosen so that any root to leaf path has weight $C \cdot \sum_{i=0}^{h-1} 2^{2^i} = 1$. Now, consider any three leaves a, b and c such that the least common ancestor for any of the pairs is at height at least $h/2$. Further, for any three leaves x, y, z, let us call $\mathbf{Q}_h(x, y, z)$ to be δ-random if any of the pairs is returned with probability $1/3 \pm \delta$. Then, the following can be easily verified. (i) The experiment $\mathbf{Q}_h(a, b, c)$ is $\exp(-\Theta(n))$-random. (ii) If $x \neq a, b, c$ is any other leaf, then the experiments, $\mathbf{Q}_h(a, b, x)$, $\mathbf{Q}_h(b, c, x)$ and $\mathbf{Q}_h(a, c, x)$ are $\exp(-\Theta(n))$-random. (iii) If $x, y \neq a, b, c$ are two other leaves, then the experiments, $\mathbf{Q}_h(a, x, y)$, $\mathbf{Q}_h(b, x, y)$ and $\mathbf{Q}_h(c, x, y)$ are $\exp(-\Theta(n))$-random. From the above, it easily follows that using just $\binom{n}{3}$ experiments, the relative topology of the leaves a, b and c cannot be resolved.

Thus, we will need to impose some conditions on the ultrametric to make the problem tractable. Specifically, we show that a lower bound on the length of each edge is necessary and sufficient (up to log factors) for reconstructing the topology in the general model, and reconstructing the weights in the homogeneous model.

Without loss of generality, we will assume that the tree is a full binary tree, since an internal node with 1 child does not affect the response to any experiment and can be eliminated.

Through this paper, when we refer to events having overwhelmingly high probability, we mean a probability of at least $1 - \frac{1}{n^6}$. Since we will consider at most $o(n^5)$ such events, using the union bound, we can assume that all of them happen with high probability (at least $1 - \frac{1}{n}$), and condition our analysis upon this event.

We fix some standard notation for full binary trees that will be used in our algorithms.

Subtrees: By a *subtree* of some tree T, we will mean the entire tree rooted at some internal node of T. (Thus we use the term "subtree" in a more restrictive manner than usual.) For any tree T, $L(T)$ denotes the set of leaves of T. We will also refer to the set of all leaves in the tree by L.

Subtree-Induced Partition: If T_B is a subtree of T, it naturally partitions $L(T) - L(T_B)$ into buckets $S_1, S_2, \cdots S_k$, where x and y are in the same bucket if and only if for any $z \in L(T_B)$, the least common ancestors of x and z, and of y and z are the same. An alternative characterization of these buckets is that x and y are in the same bucket if and only if for any z in T_B, the closest pair out of the triple (x, y, z) is (x, y). Each bucket can be thought of as a subtree hanging off from the path from the root of T_B to the root of T. Thus there is a natural order on the buckets that is defined by this path, with S_1 being the bucket closest to T_B and S_k the farthest. A visual depiction is shown in Figure 1.

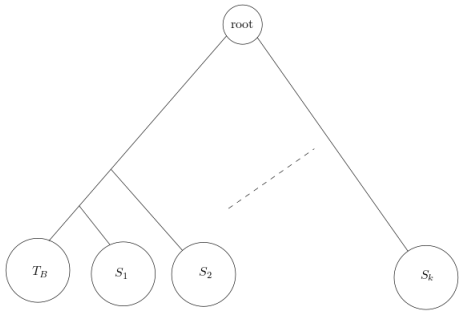

Figure 1: Partition of buckets with respect to Subtree T_B

For any $j \in [k]$, the set of leaves in $T_B, S_1, S_2 \cdots S_j$ form the leaves of a subtree.

Induced Topology: For any subset of leaves S, the induced topology on S is defined by removing all leaves outside of S and removing internal nodes that now have only one child. (Since we are talking about weighted trees, when we have an internal node v with one child, we replace the two edges incident on v by a single edge whose weight is the sum of the weights of the two edges.) It is not hard to see that the weighted tree obtained by this process will define an ultrametric on S. As a special case, when T_B is a subtree of T, we will denote the induced topology on the leaves **not** in T_B by $T - T_B$. We will also define a slightly different induced topology where we replace T_B by a single leaf (of T_B). We think of this operation as taking the quotient of T with respect to T_B, and denote the resulting topology by $T \,/\, T_B$.

By the *topology of a triple* of leaves (x, y, z) we mean the induced subtree of T with just these leaves. This topology is completely specified by specifying the pair among x, y, and z that has the least common ancestor of smallest height.

Finally, we use some standard concentration inequalities and results about measures of statistical distance in our paper. These can be found in Appendix D .

4. Reconstructing Full Binary Trees

This section is devoted to the proof of the following result.

Theorem 1 *There exists an efficient algorithm* Topology-Reconstruction*, that works as follows: Given access to the* ***general model*** *on the leaves of a weighted full binary ultrametric tree T where all root-leaf paths are of length 1, and each edge is of weight at least $\frac{\tau}{\varepsilon}\sqrt{\frac{\log n}{n}}$ for some large constant τ,* Topology-Reconstruction *exactly reconstructs the topology of this tree with high probability.*

For simplicity's sake, we normalize ε to be 1, the proof can be easily modified for general values of ε by scaling the constant τ in the edge weight lower bound by $\frac{1}{\varepsilon}$.

We start by providing a high-level description of the algorithm Topology-Reconstruction. We want to infer an unknown tree T on a given set of leaves L. We are given the result of the experiments on each triple $(a, b, c) \in L$. This result is one of the three possible pairs with probabilities specified by the distance-based noise model (defined in Section 3). We use the phrase 'resolving the topology' of a subtree T' to mean that we know the rooted tree representation of T'.

Before describing the algorithm, we make an important observation regarding the probability of getting the correct answer.

Observation 1 *The assumption that all edge weights are at least $\tau\sqrt{(\log n/n)}$ implies that $d_1 + 2\tau\sqrt{(\log n/n)} \leq d_2$ for the distances d_1, d_2 involved in every experiment $Q(a, b, c)$. Thus, using the properties of the model, we observe that $p_{\text{CORRECT}} \geq p_{\text{INCORRECT}} + 2\tau\sqrt{(\log n/n)}$.*

Our algorithm works by resolving the topologies of small subtrees, and then stitching these together until all of T is resolved.

1. In a bottom-up manner by combining sibling subtrees, we build a "base" tree T_B containing between $\sqrt{n}$ and $2\sqrt{n}$ leaves that is a subtree of T with high probability. (This is as large a tree as we can build to be confident that we have a subtree of T.)

2. We use the same idea to build a "pivot" tree T_P on about the same number of leaves outside of $L(T_B)$. With high probability, T_P will also be a subtree of the induced tree on $L - L(T_B)$.

3. Using the fact that $|L(T_B)| \times |L(T_P)| = \Omega(n)$, we partition the leaves in $L(T) - L(T_B)$ into 3 parts (some possibly empty) — leaves in buckets to the left of T_P (i.e., buckets with smaller indices than the bucket that the leaves of T_P come from), leaves in the same bucket as T_P, and leaves in buckets to the right of T_P (i.e., buckets with larger indices than the bucket(s) that the leaves of T_P come from).

4. We show that if a subtree excludes $\Omega(n)$ leaves then we can infer its entire topology with high probability. Likewise, if it contains $\Omega(n)$ leaves, we can infer the topology of the complement of the subtree with high probability. Using these two facts, we fully resolve the topology of all but one of the 3 parts in the previous step and recurse on the unresolved part. When this part has fewer than $\frac{11n}{12}$ leaves, we can infer its topology directly from the leaves outside, and the recursion bottoms out.

We will now elaborate on the algorithm. All proofs in the section have been moved to Appendix A in the interest of space. To begin, we state a lemma that establishes the claims made in Step 4 above.

Lemma 2 *There exists an algorithm* Completion *that takes as input a subtree T' of an ultrametric tree T with n leaves and has the following properties.*

1. *Given the set $L(T')$, if $|L(T')| \geq \frac{n}{24}$,* Completion *resolves the topology of the quotient $T \,/\, T'$ with high probability. (Recall that this quotient is arrived at by collapsing T' to a single leaf and taking the induced topology on the resulting set of leaves.)*

2. *Given the set $L(T')$, if $|L(T')| \leq n - \frac{n}{25}$,* Completion *resolves the induced topology on this set with high probability.*

We now fill in the details of Step 1 in our algorithm outline above, describing and analyzing an algorithm that constructs an approximately $\sqrt{n}$ sized subtree within the induced tree of a large enough set of leaves.

Lemma 3 *There exists an algorithm* Build-Subtree *that given a subset $S \subseteq L$ of leaves with $|S| \geq \frac{n}{12}$, finds a subtree T' of T_S such that $|L(T')| \in [\sqrt{n}, 2\sqrt{n}]$ w.h.p. Here T_S denotes the (unknown) induced topology on S.*

We show how to partition the leaves within a contiguous interval of buckets onto either side of a $\sqrt{n}$ sized subtree in the induced topology on the leaves within this interval.

Lemma 4 *There exists an algorithm* Partition *that takes the following inputs: 1) A subtree T_B of T with $|T_B| \geq \sqrt{n}$, 2) A contiguous interval I in the partition of buckets with respect to T_B, and 3) A subtree T_P of the tree induced on $L(I)$ such that $\sqrt{n} \leq |L(T_P)| \leq 2\sqrt{n}$ and w.h.p. partitions $L(I) - L(T_P)$ into 3 sets P_1, P_2, and P_3 such that P_1 consists of all leaves in lower indexed buckets than the leaves of T_P, P_2 consists of leaves in the same bucket(s) as T_P, and P_3 consists of leaves in higher numbered buckets. If $L(T_P)$ comprises leaves from more than one bucket, then P_1 and P_2 are empty.*

Before describing the algorithm, we introduce some notation that will aid in describing and tracking the progress of the algorithm.

We have seen that we can bucket the complement of a subtree T_B. We can also find a subtree of the induced tree on one of the buckets, and partition the rest of the leaves in this bucket. We can recurse on this idea, repeatedly descending into one of the buckets in a partition, finding a subtree in this bucket, and partitioning the leaves in the bucket not in the subtree. We call this a recursive bucketing of the tree.

We make a straightforward observation about the buckets in a recursive bucketing, which can be seen by induction over the depth of the recursion.

Observation 2 *The leaves in the bucket of a recursive bucketing form the leaves of a subtree of the entire tree.*

As an immediate corollary, a subtree in the induced topology of the leaves of a bucket of a recursive bucketing is in fact a subtree of the entire tree.

Consider a recursive bucketing within a bucket $\mathcal{B}$ containing at least $\frac{11n}{12}$ leaves with a base tree T_B of size in $[\sqrt{n}, 2\sqrt{n}]$. Let T_P be a pivot tree of size in $[\sqrt{n}, 2\sqrt{n}]$ in the induced topology of $L(\mathcal{B}) \setminus L(T_B)$. On using the algorithm Partition from Lemma 4, we get three partitions P_1, P_2 and P_3 of the leaves in the bucket and not in the base or pivot tree. We show that any partition with less than $\frac{11n}{12}$ leaves can be partially resolved, i.e., added to the set of leaves for which we know the answer to any closest pair of three leaves.

Claim 1 *There exists an algorithm Partial-Resolution that resolves any partition P with fewer than $\frac{11n}{12}$ leaves.*

Proof At most one of the partitions can be larger than $\frac{11n}{12}$. First, we deal with the case where one of the partitions has at least $\frac{11n}{12}$ leaves. If P_1 is this partition, then $T_B \cup P_1$ forms a subtree in the induced topology of the recursive bucket $\mathcal{B}$ and hence is a subtree of the entire tree with over $\frac{11n}{12}$ leaves. We can thus use Lemma 2 to resolve P_2 and P_3. When P_2 has at least $\frac{11n}{12}$ leaves, $T_B \cup P_1 \cup P_2$ is a subtree with at least $\frac{11n}{12}$ leaves and can therefore be used (through Lemma 2) to resolve P_3. Further, $T_B \cup P_1$ is a subtree with at most $\frac{n}{12}$ leaves, therefore we can use Lemma 2 to resolve this subtree and hence P_1. If P_3 is the largest partition, then $T_B \cup P_1 \cup P_2$ is a subtree with at most $\frac{n}{12}$ leaves, which allows us to resolve P_1 and P_2.

Now, consider the case where all three partitions have fewer than $\frac{11n}{12}$ leaves. Since the recursive bucket $\mathcal{B}$ has $\frac{11n}{12}$ leaves and the base and pivot trees have at most $2\sqrt{n}$ leaves, the largest partition has at least $\frac{n}{3}$ leaves. In the first case, let P_1 be the largest partition. Then the subtree $T_B \cup P_1$ has at least $\frac{n}{3}$ leaves and at most $\frac{11n}{12} + 2\sqrt{n}$ leaves, allowing us to resolve the entire tree using Lemma 2. If P_2 is the largest partition, the algorithm again branches into two cases – either P_1 has at least $\frac{n}{24}$ leaves or it does not. If P_1 has at least $\frac{n}{24}$ leaves, then the subtree $T_B \cup P_1$ has at least $\frac{n}{24}$ leaves and at most $\frac{11n}{12}$ leaves, allowing us to resolve the entire tree using Lemma 2. Else, if P_1 has less than $\frac{n}{24}$ leaves, then the subtree $T_B \cup P_1 \cup T_P \cup P_2$ has at least $\frac{n}{3}$ leaves and at most $\frac{11n}{12} + \frac{n}{24} + 4\sqrt{n}$ leaves, allowing us to resolve the entire tree using Lemma 2. Finally, consider P_3 being the largest partition. Consider the case where P_1 and P_2 have at least $\frac{n}{24}$ leaves together. In this case, the subtree $T_B \cup P_1 \cup T_P \cup P_2$ has at least $\frac{n}{24}$ leaves and at most $\frac{2n}{3}$ leaves, allowing us to resolve the entire tree using Lemma 2. If P_1 and P_2 do not have $\frac{n}{24}$ leaves in total, then the subtree

$T_B \cup P_1 \cup T_P \cup P_2 \cup P_3$ has at least $\frac{n}{3}$ leaves and at most $\frac{11n}{12} + \frac{n}{24} + 4\sqrt{n}$ leaves, allowing us to resolve the entire tree using Lemma 2. ■

Algorithm 1 Algorithm Topology-Reconstruction

Initialization: Create a base tree T_B with no. of leaves in $[\sqrt{n}, 2\sqrt{n}]$ using Build-Subtree (Lemma 3). Let the resolved leaves be $R = L(T_B)$ and the unresolved leaves U be the rest of the tree. Set the current recursive bucket $\mathcal{B}$ to be the entire tree

while U *is non empty* **do**

1. Use Algorithm Build-Subtree to build a subtree T_P of the induced topology of U, with $\Theta(\sqrt{n})$ leaves.
2. Use Algorithm Partition to partition the leaves $L(\mathcal{B}) \setminus (L(T_P) \cup L(T_B))$ into the partitions P_1, P_2 and P_3
3. Use algorithm Partial-Resolution to resolve any of the partitions with less than $\frac{11n}{12}$ leaves.
4. If either P_1 or P_3 has more than $\frac{11n}{12}$ leaves, then set $U \leftarrow (U \setminus T_P) \cap P$ where P is the large partition.
5. If P_2 has more than $\frac{11n}{12}$ leaves, set the new recursive bucket $\mathcal{B} \leftarrow T_P \cup P_2$, set the base tree $T_B \leftarrow T_P$ and set $U \leftarrow U \cap P_2$.

end

In the course of our algorithm, we maintain a set of "resolved leaves" R with the property that we know the answers to all queries involving only leaves from this set. The complement set of "unresolved" leaves U has the additional invariant property that it is exactly the set of leaves contained in some contiguous interval of buckets in a recursive bucketing. Our algorithm works by shrinking the set U and growing the set R by at least $\sqrt{n}$ in each step. Additionally, we keep track of the current recursive bucketing (through the variable $\mathcal{B}$) and ensure that it always has at least $\frac{11n}{12}$ leaves. In fact, we maintain the stronger property that the unresolved set U has at least $\frac{11n}{12}$ leaves at the start of each iteration of the for loop and $\mathcal{B}$ contains U. The progress of the algorithm can be seen from the fact that each iteration of the while loop reduces the size of the unresolved set by at least $\sqrt{n}$ (the minimum size for the pivot tree T_P). We note two subtle points used in the execution of the algorithm. First, we use the algorithm Partition to partition the leaves of a large (at least $\frac{11n}{12}$ leaves) recursive bucket rather than the entire tree as the algorithm is originally described. However, it is easy to see that the algorithm and its analysis in facts works as is for this altered application. Second, we critically use the invariant that the unresolved set has at least $\frac{11n}{12}$ leaves (or it is resolved using Partial-Resolution) in using Build-Subtree to construct a pivot tree T_P in the induced topology of U (since the precondition is that the set must have at least $\frac{n}{12}$ leaves).

To conclude the proof of Theorem 1, we argue that the algorithm Topology-Reconstruction succeeds with high probability. The key observation is that we use the various subroutines Completion, Build-Subtree and Partition at most $O(\sqrt{n})$ times each. This is because we reduce the size of the unresolved part by $\sqrt{n}$ using only a constant number of calls to these subroutines. Each of them assume at most $O(n^3)$ overhwelmingly high probability events to be simultaneously true to succeed,

implying that we only need a total of $O(n^{7/2})$ overwhwelmingly high probability events to all be true for the algorithm to correctly recover the topology of the tree. Since each of them occurs with probability at least $1 - \frac{1}{n^6}$, an application of the union bound gives us the desired result. Further, since each subroutine runs in polynomial time, the overall algorithm is also efficient.

5. Weight Reconstruction

In the previous section, we gave an algorithm to reconstruct the topology of the tree in the general model. In this section, we will show how to approximately reconstruct the edge weights in the homogeneous model. The precise theorem is stated below (Theorem 5). Assuming that each root to leaf path has unit weight, our algorithm can reconstruct the tree as long as each edge has weight at least $\tau \cdot \log n/\sqrt{n}$. Note that the condition required here is stronger than Theorem 1 where it suffices that each edge weight is $\Omega(\sqrt{\log n/n})$.

Theorem 5 *There exists an algorithm Tree-reconstruct-weight, that works as follows: Given access to the **homogeneous model** on the leaves of a weighted full binary ultrametric tree T where all root-leaf paths have length 1 and all edges have weight at least $\tau\frac{\log n}{\sqrt{n}}$ for some large constant τ, Tree-reconstruct-weight reconstructs the weight of each edge with high probability within an additive error $\kappa\frac{\log n}{\sqrt{n}}$, where $\kappa \ll \tau$.*

We explain the high-level strategy for procedure Tree-reconstruct-weight. Instead of estimating the weight of each edge, we will give a procedure which for any vertex v, will estimate the weight of the path from root to v up to an additive $\kappa \log n/2\sqrt{n}$. This trivially implies reconstruction of the weight of each edge up to $\pm\kappa \log n/\sqrt{n}$. Our algorithm assumes that $\tau \gg \kappa$.

Since the homogeneous model is a special case of the general model (with $\epsilon = 1/6$) and the edge weights satisfy the condition required by Theorem 1 on the edge weights, we can first run Topology-Reconstruction to reconstruct the topology of the tree (with high probability). The rest of the proof assumes we have access to the topology of the tree. The main workhorse for weight reconstruction is the idea that if we have two leaves a and b that lie in the left subtree of some node v, and the right subtree of v has $\Omega(n)$ leaves, then we can get a good approximation to the height of the least common ancestor of a and b. A related idea is that if we have an internal node v such that the product of the number of leaves in its left and right subtrees is $\Omega(n)$, then we can also get a good approximation to the height of v. This leaves the case of nodes v for which neither of these properties is true and much of the technical difficulty of our algorithm is in handling such nodes (Lemma 13). For such a node we get several coarse approximations of its height, which need to be combined carefully because of subtle dependence between these approximations.

In interest of space, we move the detailed description of the algorithm and its associated proof to Appendix B.

6. Necessary Conditions

The goal of this section is to show that to reconstruct the topology of the tree, it is necessary for each edge to have weight $\Omega(1/\sqrt{n})$ – thus, essentially matching the lower bound assumption in Theorem 1. Recall that we normalize the edge weights so that the height (weighted root to leaf distance) of the tree is 1.

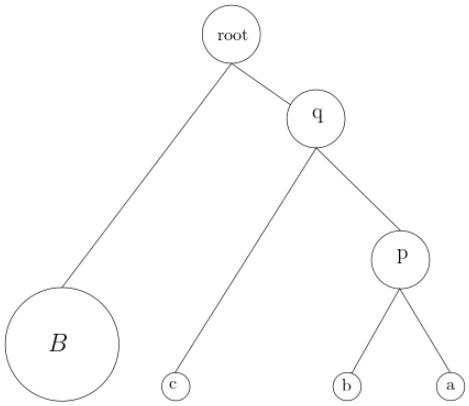

Figure 2: Lower Bound Instance Tree T_1

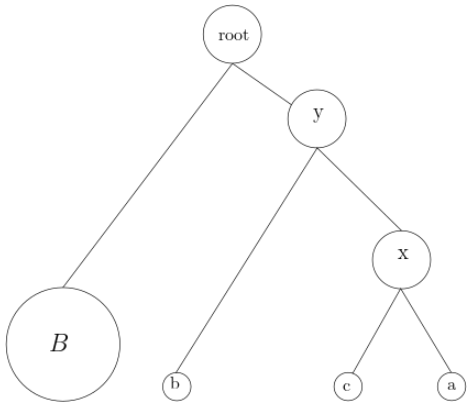

Figure 3: Lower Bound Instance Tree 2

In the theorem below, we give a nearly matching lower bound on the minimum weight of each edge even for topology reconstruction in our model. We also note that such edge weight lower bounds are commonplace in the literature on phylogenetic reconstruction Felsenstein (1981); Farach and Kannan (1999); Erdős et al. (1999).

Theorem 6 *Let T be the set of weighted full binary trees tree such that the weights induce an ultrametric on the distances between leaves (within each tree). Then, for any algorithm with access to the **homogeneous model**, as described in Section 3), there exist two trees with edge weights allowed to be as small as $\frac{\rho}{\sqrt{n}}$ that the algorithm cannot distinguish with better than .51 probability. Here ρ is a sufficiently small constant ($\rho \leq \frac{1}{100}$). A fortiori, we obtain the same lower bound for the general model as well.*

To prove this result, we construct two trees with the following properties:

1. They have distinct topologies and the weight of each edge is at least $\rho/\sqrt{n}$.
2. It is information-theoretically impossible to distinguish between the trees with probability more than 0.51 (in the homogeneous model).

The two trees T_1 and T_2 are as follows. Both have roots with identical weighted left subtrees B. The right subtrees of both T_1 and T_2 both have three leaves a, b, and c but with different induced topologies. In T_1, a and b are sibling leaves with parent p. The parent of c and p is the node q, which is the right child of the root. In T_2, a and c are siblings, whose parent is x. x and b have a parent y, which is the right child of the root. All 'corresponding' edge lengths are identical and in particular, the edges (p, q) and (x, y) have weight $\frac{\rho}{\sqrt{n}}$. The edges to the sibling pair of leaves in the right subtree of both trees have the same weight, say $\frac{1}{3}$. The two trees we construct are shown in Figures 2 and 3.

The rest of the proof of Theorem 6 has been moved to Appendix C.

Acknowledgements

The first author was supported by NSF Award CCF 1910534 and "start up support from the University of Pennsylvania", the second author was supported by NSF Awards CCF 1926872, CCF 1910534 and CCF 2045128 and the third author was supported by NSF Award CCF 1763307.

References

Gang Wu and Ming-Yang Kao and Guohui Lin and Jia-Huai You. Reconstructing phylogenies from noisy quartets in polynomial time with a high success probability. *Algorithms for Molecular Biology*, 3(1):377–390, 2008.

Emmanuel Abbe. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531, 2017.

Sameer Agarwal, Josh Wills, Lawrence Cayton, Gert Lanckriet, David Kriegman, and Serge Belongie. Generalized non-metric multidimensional scaling. In *Artificial Intelligence and Statistics*, pages 11–18. PMLR, 2007.

Richa Agarwala and David Fernández-Baca. A polynomial-time algorithm for the perfect phylogeny problem when the number of character states is fixed. *SIAM J. Computing*, 23:1216–1224, January 1994.

Nir Ailon and Moses Charikar. Fitting tree metrics: Hierarchical clustering and phylogeny. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 73–82. IEEE, 2005.

Amir Ben-Dor and Benny Chor and Dan Graur and Ron Ophir and Dan Pelleg. Constructing phylogenies from quartets: elucidation of eutherian superordinal relationships. . *Journal of Computational Biology*, pages 377–390, Jan 1998.

Boaz Barak, Moritz Hardt, Ishay Haviv, Anup Rao, Oded Regev, and David Steurer. Rounding parallel repetitions of unique games. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 374–383. IEEE, 2008.

Ralph Allan Bradley and Milton E. Terry. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Daniel Brown and Jakub Truszkowski. Fast error-tolerant quartet phylogeny algorithms. *Theoretical Computer Science*, 483:104–114, 2013.

Yu Chen, Sampath Kannan, and Sanjeev Khanna. Near-perfect recovery in the one-dimensional latent space model. In *Proceedings of The Web Conference 2020*, pages 1932–1942, 2020.

Constantinos Daskalakis and Sebastien Roch. Alignment-free phylogenetic reconstruction: Sample complexity via a branching process analysis. *The Annals of Applied Probability*, 23(2):693–721, 2013.

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.

Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84(6):066106, 2011.

Ehsan Emamjomeh-Zadeh and David Kempe. Adaptive Hierarchical Clustering Using Ordinal Queries. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, page 415–429, 2018.

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.

Péter L Erdös, Michael A Steel, LászlóA Székely, and Tandy J Warnow. A few logs suffice to build (almost) all trees: Part II. *Theoretical Computer Science*, 221(1-2):77–118, 1999.

M. Farach, S. Kannan, and T. Warnow. A robust model for finding optimal evolutionary trees. *Algorithmica*, 13(1-2):155–179, February 1995.

Martin Farach and Sampath Kannan. Efficient algorithms for inverting evolution. *Journal of the ACM (JACM)*, 46(4):437–449, 1999.

Joseph Felsenstein. Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of Molecular Evolution*, 17(6):368–376, 1981.

Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, pages 84–92. Springer, 2015.

Kevin G Jamieson and Robert Nowak. Active ranking using pairwise comparisons. *Advances in neural information processing systems*, 24, 2011.

Sampath Kannan and Tandy Warnow. Inferring evolutionary history from dna sequences. *SIAM J. Computing*, 23(4):713–737, August 1994.

Sampath K Kannan, Eugene L Lawler, and Tandy J Warnow. Determining the evolutionary tree using experiments. *Journal of Algorithms*, 21(1):26–50, 1996.

Paul E. Kearney, Ryan B. Hayward, and Henk Meijer. Inferring Evolutionary Trees from Ordinal Data. In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 418–426, 1997.

Matthäus Kleindessner and Ulrike Luxburg. Uniqueness of ordinal embedding. In *Conference on Learning Theory*, pages 40–67. PMLR, 2014.

R Duncan Luce. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 1959.

Daniel McFadden. Conditional logit analysis of qualitative choice behavior. 1973.

Frank McSherry. Spectral partitioning of random graphs. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 529–537. IEEE, 2001.

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.

Elchanan Mossel and Sebastien Roch. Distance-based species tree estimation under the coalescent: information-theoretic trade-off between number of loci and sequence length. *The Annals of Applied Probability*, 27(5):2926–2955, 2017.

Elchanan Mossel and Mike A. Steel. How much can evolved characters tell us about the tree that generated them? In Olivier Gascuel, editor, *Mathematics of Evolution and Phylogeny*. Oxford University Press, 2007.

Elchanan Mossel, Joe Neeman, and Allan Sly. A proof of the block model threshold conjecture. *Combinatorica*, 38(3):665–708, 2018.

Jerzy Neyman. Molecular studies of evolution: a source of novel statistical problems. In *Statistical decision theory and related topics*, pages 1–27. Elsevier, 1971.

Robin L Plackett. The analysis of permutations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 24(2):193–202, 1975.

David Pollard. *A User's Guide to Measure Theoretic Probability*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2001. doi: 10.1017/CBO9780511811555.

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.

Naruya Saitou and Masatoshi Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, 4(4):406–425, 1987.

Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. *Advances in neural information processing systems*, 16, 2003.

C. Semple and M. Steel. *Phylogenetics. Oxford Lecture Series in Mathematics and Its Applications*. Oxford University Press, 2003.

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

Omer Tamuz, Ce Liu, Serge Belongie, Ohad Shamir, and Adam Tauman Kalai. Adaptively learning the crowd kernel. In *International Conference on Machine Learning (ICML)*, 2011.

Laurens Van Der Maaten and Kilian Weinberger. Stochastic triplet embedding. In *2012 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6. IEEE, 2012.

Appendix A. Proofs from Section 4 (Topology Reconstruction)

Before describing the proofs of the different parts used in our algorithm, we state and prove a simple lemma about sums of independent random variables that recurs in multiple proofs.

Lemma 7 *Let $L \geq \frac{n}{25}$, let $X_1, X_2, \cdots X_L$ and $Y_1, Y_2, \cdots Y_L$ be two sets of independent $0-1$ random variables such that for all $i \in [L]$, $\mathbb{E}[X_i] \leq \mathbb{E}[Y_i] - c\sqrt{(\log n/n)}$, for a sufficiently large constant c. Let $X = \sum_{i=1}^{l} X_i$ and let $Y = \sum_{i=1}^{l} Y_i$ Then, with overhwelmingly high probability, i.e. at least $1 - 1/n^6$, we have $Y > X + 24\sqrt{n \log n}$.*

Alternatively, if X_i and Y_i are identical random variables for each i, with overhwelmingly high probability, we have $|X - Y| < 12\sqrt{n \log n}$.

Proof Since X and Y are each sums of at least $\frac{n}{25}$ independent $0, 1$ random variables, we use the Hoeffding bound (Theorem 16) to argue that the events $|X - \mathbb{E}[X]| \leq 6\sqrt{n \log n}$ and $|Y - \mathbb{E}[Y]| \leq 6\sqrt{n \log n}$ each happen with probability at least $1 - \frac{1}{2n^6}$. Using the union bound, both events happen with probability at least $1 - \frac{1}{n^6}$.

For the first part of the lemma, since for each $i \in [L]$, $\mathbb{E}[X_i] \leq \mathbb{E}[Y_i] - c\sqrt{\frac{\log n}{n}}$, using linearity of expectation gives : $\mathbb{E}[X] \leq \mathbb{E}[Y] - c\sqrt{\frac{\log n}{n}}$. Using the triangle inequality, we conclude that $|X - Y| < 12\sqrt{n \log n}$ is true with probability at least $1 - \frac{1}{n^6}$.

For the second part of the lemma, since X_i and Y_i are identical random variables, $\mathbb{E}[X] = \mathbb{E}[Y]$. Using the triangle inequality, we conclude that $Y > X + 24\sqrt{n \log n}$ is true with probability at least $1 - \frac{1}{n^6}$. ■

A.1. Proof of Lemma 2

Part 1: First, we describe how Completion reconstructs the buckets $S_1, S_2, \cdots S_k$ that partition $L(T) - L(T')$. Let L' be the set of leaves in T'.

The following test is used to resolve the relative order of two leaves $x, y \in L \setminus L'$. For each $a \in L'$, the random variable X_a is set to 1 if the result of the experiment $Q(a, x, y)$ is (a, x), and to 0 otherwise. Similarly, the random variable Y_a is set to 1 if the result of this experiment is (a, y), and to 0 otherwise. Let $X = \sum_{a \in L'} X_a$ and $Y = \sum_{a \in L'} Y_a$.

If $X - Y > 24\sqrt{n \log n}$, then we declare that x's bucket has a lower index than y's. Otherwise, if $|X - Y| \leq 24\sqrt{n \log n}$, we say that x and y are in the same bucket. We prove that this algorithm is correct with high probability.

We first show that the above method of comparison works correctly for any pair of leaves x, y in different buckets. Without loss of generality, $x \in S_i$ and $y \in S_j$ with $i < j$. Using Observation 1 about experiment $Q(a, x, y)$, we conclude that $\mathbb{E}[X_a] - \mathbb{E}[Y_a] \geq 2\tau\sqrt{\frac{\log n}{n}}$. Using Lemma 7, we get the desired result about the comparison of X and Y, with high probability.

We next extend the result to pairs of leaves x, y in the same bucket S_i. We know that $d(a, x) = d(a, y) > d(x, y)$ for all $a \in L'$.Thus, X_a and Y_a are identically distributed random variables (since our noise model guarantees that the two incorrect answers to a experiment appear with equal probability). Using the second part of Lemma 7, we get the desired result about the comparison of X and Y, with high probability.

Finally, Completion labels the buckets generated by doing all pairwise tests using a standard topological sorting algorithm to recover the original labels. $O(n^2)$ high probability events are assumed to simultaneously occur as part of this proof, since we are comparing all pairs of leaves.

In the next phase, the algorithm Completion resolves the closest pair of three leaves from the same bucket. A score s_{ab} is associated with each pair of leaves $a, b \in S_i$ and a $0-1$ random variables X^x_{ab} is defined for each $x \in L'$. Each pair $a,\ b$ has at least $\frac{n}{25}$ associated random variables. X^x_{ab} is set to 1 if $Q(a,b,x)$ is (a,b) and 0 otherwise. Let $s_{ab} = \sum_{x \in L'} X^x_{ab}$. Given three leaves $a, b, c \in S_i$ the pair with the highest score is declared to be the closest pair.

We argue that this test succeeds with high probability. Let (a,b) be the closest pair among (a,b,c). We will show that with high probability $s_{ab} > s_{bc}$. (An identical argument helps establish that $s_{ab} > s_{ac}$.) We observe that $d(b,c) \geq d(a,b) + 2\tau\sqrt{\frac{\log n}{n}}$ and $d(a,b) < d(b,c) < d(a,x) = d(b,x) = d(c,x)$ for all $x \in L'$. By the properties of the noise model (the bound on the partial derivative), we get $\mathbb{E}[X^x_{ab}] - \mathbb{E}[X^x_{ac}] \geq 2\tau\sqrt{\frac{\log n}{n}}$. Using Lemma 7, we get the desired result about the comparison of s_{ab} and s_{bc}, with high probability.

$O(n^3)$ high probability events are assumed to simultaneously occur as part of this proof, since we are separately arguing correct recovery of the closest pair for every possible set of three leaves.

Part 2: Let L' be the set of leaves in T'. We describe how the algorithm Completion recovers the topology of T'.

A score s_{ab} is associated with each pair of leaves $a, b \in L'$ and a $0-1$ random variables X^x_{ab} is defined for each $x \in L \setminus L'$. Each leaf pair a, b has at least $\frac{n}{25}$ associated random variables. X^x_{ab} is set to 1 if $Q(a,b,x)$ is (a,b) and 0 otherwise. Let $s_{ab} = \sum_{x \in L \setminus L'} X^x_{ab}$. Given three leaves $a, b, c \in L'$, the closest pair is chosen as the pair with the largest score. The argument that this test succeeds with high probability uses Lemma 7 in a manner similar to the proof of Part 1 and hence we omit it. $O(n^3)$ high probability events are assumed to simultaneously occur as part of this proof, since we are separately arguing correct recovery of the closest pair for every possible set of three leaves.

A.2. Proof of Lemma 3

The algorithm Build-Subtree builds subtrees of T_S from the ground up, starting with each leaf in S as a singleton subtree. Let $T_1, T_2, \cdots T_l$ represent subtrees of T_S whose leaves form a disjoint partition of S - in each step the algorithm, with high probability, combines two of these subtrees to form a larger subtree of T_S. Let L_i represent the set of leaves of T_i. Using this procedure and starting with the initial configuration, the algorithm repeatedly keeps applying this step until the size of the largest subtree exceeds $\sqrt{n} - 1$. Since each step can at most double the size of the largest subtree, we obtain a subtree of T_S of size in the range $[\sqrt{n}, 2\sqrt{n}]$.

Now, we describe the key step (combining subtrees) of the algorithm Build-Subtree. For each pair T_i, T_j, a score s_{ij} is generated in the following manner. Fix arbitrary $a \in L_i$, and $b \in L_j$ and define a $0-1$ random variable X^x_{ij} for each leaf $x \in S \setminus (L_i \cup L_j)$ to be 1 if the experiment $Q(a,b,x)$ gives (a,b) as the answer, and 0 otherwise.

$$s_{ij} := \sum_{x \in S \setminus (L_i \cup L_j)} X^x_{ij}$$

We claim that (whp) the tree pair with the highest score is in fact a "sibling - tree pair", i.e., a pair of subtrees of T_S that can be combined through a shared parent internal node to form a subtree of

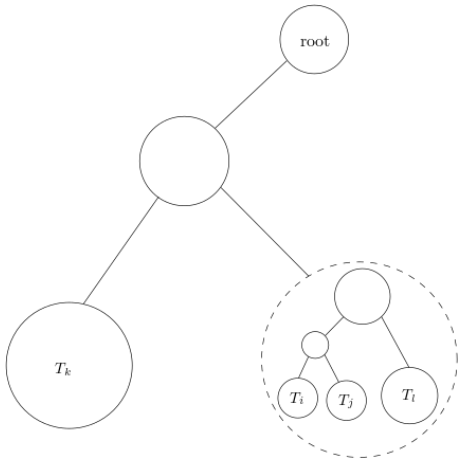

Figure 4: Sibling-Tree Pair between Non Sibling-Tree Pair

T_S. There always exists such a sibling-tree pair, since these leaf-disjoint subtrees of T_S themselves form a full binary tree.

We start with a simple observation: For any pair T_k, T_l that is not a sibling pair, there is a sibling pair T_i, T_j such that their least common ancestor is strictly below the root of either T_k or T_l.

To see this we consider the full binary tree resulting from starting with T and collapsing each set of leaves L_i into a single node and look at the subtree rooted at the least common ancestor of T_k and T_l. There must be a deepest internal node in this tree whose children are sibling subtrees T_i and T_j, completing the proof. Figure 4 shows such a set of trees.

Claim 2 *With high probability,* $s_{kl} < s_{ij}$.

Proof

Intuitively, this claim is true because the number of leaves that are used to generate the scores of both pairs T_i, T_j and T_k, T_l is sufficiently large (at least $\frac{n}{16}$), and that these leaves all favor the pair T_i, T_j (by at least $\theta(\sqrt{(\log n/n)})$ each). Further, the number of leaves that are used to generate only one of the scores is bounded by $2\sqrt{n}$. Consequently, these leaves do not alter the signal created by comparing the action of the leaves used to generate both scores.

We introduce some notation to formalize the above intuition. Let $S_1 := S \setminus (L_k \cup L_l)$ and $S_2 := S \setminus (L_i \cup L_j)$. S_1 is the set of leaves used to compute the score s_{kl} and S_2 is the set of leaves used to compute the score s_{ij}. We can write $S_1 = A \cup B_1$ and $S_2 = A \cup B_2$ where $A := S_1 \cap S_2$, $B_1 := L_i \cup L_j$ and $B_2 := L_k \cup L_l$. Since $|L_k|, |L_l|, |L_i|, |L_j| \leq \sqrt{n}$, we can assume that $|A| \geq \frac{n}{16}$ and $|B_1|, |B_2| \leq 2\sqrt{n}$.

Using the condition on minimum edge length and the bound on the partial derivative of the function $p_{\text{CORRECT}}(d_1, d_2)$, it is easy to see (Ref Fig 4) that :

$$\forall x \in A : \mathbb{E}[X^x_{ij}] - \mathbb{E}[X^x_{kl}] \geq \frac{\tau\sqrt{\log n}}{9\sqrt{n}}$$

Expanding S_1 and S_2, we get:

$$s_{kl} = \sum_{x \in A} X^x_{kl} + \sum_{y \in B_1} X^y_{kl}$$

$$s_{ij} = \sum_{x \in A} X_{ij}^x + \sum_{x \in B_2} X_{ij}^x$$

Taking the difference and using linearity of expectation, we get:

$$\begin{aligned}
\mathbb{E}[s_{ij}] - \mathbb{E}[s_{kl}] &= \sum_{x \in A} (\mathbb{E}[X_{ij}^x] - \mathbb{E}[X_{kl}^x]) + \sum_{y \in B_2} E[X_{ij}^y] - \sum_{y \in B_1} E[X_{kl}^y] \\
&\geq \sum_{x \in A} (\mathbb{E}[X_{ij}^x] - \mathbb{E}[X_{kl}^x]) - \sum_{y \in B_1} E[X_{kl}^y] \\
&\geq \frac{\tau \sqrt{n \log n}}{144} - 2\sqrt{n}
\end{aligned}$$

where the last inequality derives from the fact that each X_{kl}^y is a $0-1$ random variable and hence has expectation upper bounded by 1.

Since both s_{kl} and s_{ij} are each sums of at least $n/16$ independent $0-1$ random variables, we can use the Chernoff bound to conclude that their expectations do not differ from their value by more than $24\sqrt{n \log n}$ with very high probability. Putting together these inequalities, we conclude that, for large enough τ, we have $s_{kl} < s_{ij}$ with high probability. The proof of this claim assumes that $O(n^2)$ high probability events occur simultaneously, since it suffices that a correct subtree pair beats every wrong subtree pair, of which there are at most n^2.

■

An immediate corollary of this claim, via an application of the union bound is that the highest scoring pair is in fact a sibling-tree pair. This gives an efficient algorithm that correctly combines subtrees of T_S to form a larger subtree of T_S. The proof of correctness of one call to the algorithm Build-Subtree assumes the simultaneous occurrence of $O(n^3)$ high probability events - since the subroutine to combine subtrees is used at most n times (from the fact that each such subroutine introduces an internal vertex, and there are most n of them).

A.3. Proof of Lemma 4

When P_2 is non empty, all leaves in T_P are equidistant from T_B, implying that P_2 must be contained within a single bucket S_i.

We describe the algorithm Partition that does the partitioning. For each leaf $x \in L - (T_B \cup T_P)$, introduce $0-1$ random variables X_{ab}^x and Y_{ab}^x for each $a \in T_B, b \in T_P$. Each such leaf x has $2n$ such associated random variables, since $|T_B|, |T_P| \geq \sqrt{n}$. X_{ab}^x is set to 1 if $Q(a,b,x)$ is (a,x) and 0 otherwise. Y_{ab}^x is set to 1 if $Q(a,b,x)$ is (a,b) and 0 otherwise. Let $X^x = \sum_{a \in T_B, b \in T_P} X_{ab}^x$ and $Y^x = \sum_{a \in T_B, b \in T_P} Y_{ab}^x$. If $X^x - Y^x > 24\sqrt{n \log n}$, then, x is placed in set P_1'. If If $Y^x - X^x > 24\sqrt{n \log n}$, then, x is placed in set P_3'. Otherwise if $|Y^x - X^x| \leq 24\sqrt{n \log n}$, x is placed in P_2'. We claim that for $i \in [3], P_i' = P_i$ with high probability.

The first case is $x \in P_1$, we know that $d(a,b) \geq d(a,x) + 2\tau\sqrt{\frac{\log n}{n}}$, for all $a \in T_B, b \in T_P$, by the definition of S_1 and the minimum weight condition. Using Observation 1, we conclude that $\mathbb{E}[X_{ab}^x] - \mathbb{E}[Y_{ab}^x] \geq \frac{\tau\sqrt{\log n}}{3\sqrt{n}}$. Using Lemma 7, we conclude that the $X^x - Y^x > 24\sqrt{n \log n}$, with high probability.

Next, consider $x \in P_3$, we know that $d(a,x) \geq d(a,b) + 2\tau\sqrt{\frac{\log n}{n}}$, for all $a \in T_B, b \in T_P$, by the definition of S_1 and the minimum weight condition (Observation 1). Through similar analysis as the previous case, we conclude x is correctly placed in P_3' whp.

Finally, we see the case of $x \in P_2$. We know that $d(a,x) = d(a,b) > d(b,x)$ for all $a \in T_B, b \in T_P$, by the definition of S_2.Thus, X_{ab}^x and Y_{ab}^x are identical random variables, since the incorrect answers to any query appear with equal probability. Using the second part of Lemma 7, we get the desired result about the comparison of X^x and Y^x, with high probability. Thus, x is correctly placed in P_2' whp.

In total, we need $O(n)$ high probability events to happen simultaneously (one for partitioning each leaf) in this proof of correctness.

We index the buckets of the contiguous interval I as $S_{j_1}, S_{j_1+1}, \cdots$. For the final part of the claim, we analyze the case where T_P consists of leaves from more than one bucket. Let us index these buckets as $S_{i_1}, S_{i_1+1}, \cdots$ where $i_1 \geq j_1$. Clearly, T_P must consist of all leaves in an interval of buckets, since leaves within a bucket are always closer to each other than leaves in another bucket (recall that the leaves of each bucket are the leaves of a subtree of T, Ref Fig 1). Due to this property, P_2 must be empty. Recall that the set of leaves in $T_B, S_1, S_2 \cdots S_i$ forms a tree for any index i. Consequently, every leaf $l \in L(S_i)$ is closer to any leaf in $T_B, S_1, S_2, \cdots S_{i-1}$ than to a leaf in S_j with $j > i$. Now, consider the case in which T_P consists of leaves contained in a union of buckets - $S_{i_1}, S_{i_1+1} \cdots S_{i_2}$ where $i_1 > j_1$. The leaves of T_P cannot be a set of leaves of a tree in the induced topology of $L(I)$, since any leaf in S_{i_1} is closer to a leaf in S_{j_1} (which is part of the set $L(I)$) than to a leaf in S_{i_2}. We have a contradiction, invalidating the assumption $i_1 > j_1$. Thus, we must have $i_1 = j_1$, implying that P_1 is also empty.

Appendix B. Proof of Theorem 5 (Weights Reconstruction)

We start by fixing some notation. Let $\mathsf{NL}(v)$ denote the number of leaves in the sub-tree rooted at vertex v. We will adopt the convention that the children of each node are ordered so that for any node v with right child rc and left child lc, $\mathsf{NL}(\mathsf{rc}) \geq \mathsf{NL}(\mathsf{lc})$. Further, without loss of generality, we can also assume that the binary tree is full – otherwise, if an internal node v has exactly one child, we can collapse the two edges incident on v into one. This modification does not affect the output probabilities associated with any triple of leaves. Finally, for any vertex v, we define h_v as the "height" of v – i.e., the total weight of any path from v to a leaf in its subtree. Because the distance on the tree is an ultrametric, the choice of the leaf is immaterial.

Definition 8 *A vertex v is said to be* heavy *if $\mathsf{NL}(v) \geq \alpha n + 1$ where $\alpha = \frac{1}{6}$. Otherwise, the vertex is said to be* light*. By definition, the root r is a heavy vertex.*

Next, given a rooted tree as above, we identify a distinguished vertex v_f as follows. Let P be the rightmost path from the root to a leaf . Starting from the root r, let the vertices in P (in order) be $r, v_1, \ldots, v_{\ell^*}$ where v_{ℓ^*} is the rightmost leaf. We define f to be maximum index such that v_f is a heavy vertex.

Consider the root r, with left child lc_r and right child rc_r. Observe that by our convention, rc_r is a heavy vertex. We now give a procedure to (approximately) compute h_v for any vertex v in the subtree rooted at lc_r.

Lemma 9 *There is a procedure* Compute-light-tree *which with high probability, computes h_v for all vertices v in the tree rooted at lc_r with accuracy $\Theta(\sqrt{\log n/n})$.*

Proof Consider any non-leaf vertex v in the subtree of lc_r. Since v is a non-leaf vertex, there is a leaf a in its left subtree and b in the right subtree. Additionally, since rc_r is a heavy vertex, there are at least αn leaves under it - indexed by $\{c_i\}_{i=1}^{k}$ where $k \geq \alpha n$. Let $\mathbf{X}_i$ be the indicator random variable that query $Q(a, b, c_i)$ returns (a, b). Each $\mathbf{X}_i$ is an independent (and in fact identically distributed) random variable such that

$$\Pr[\mathbf{X}_i = 1] = \frac{d(a, c_i) + d(b, c_i)}{2(d(a, b) + d(a, c_i) + d(b, c_i))} = \frac{4}{2(4 + 2h_v)} = \frac{1}{2 + h_v}.$$

Let us define $\mathbf{A} := (1/k) \cdot (\sum_{i=1}^{k} \mathbf{X}_i)$. Define h'_v to be such that $\mathbf{A} = \frac{1}{2+h'_v}$. Now, by Hoeffding's inequality (Theorem 16), we get that with overwhelmingly high probability,

$$\left| \mathbf{A} - \frac{1}{2 + h_v} \right| \leq \kappa_2 \sqrt{\frac{\log n}{n}}.$$

Thus

$$\left| \frac{1}{2 + h'_v} - \frac{1}{2 + h_v} \right| \leq \kappa_2 \sqrt{\frac{\log n}{n}}.$$

This means that

$$|h_v - h'_v| \leq \kappa_2 \sqrt{\frac{\log n}{n}} (2 + h_v)(2 + h'_v).$$

Note that $h_v \leq 1$. Because $\left|\frac{1}{2+h'_v} - \frac{1}{2+h_v}\right| \leq 1/6$, it follows that $h'_v \leq 4$. Thus, it follows that $|h_v - h'_v| \leq \Theta(\sqrt{\log n/n})$. ■

Recall that we had define P as the rightmost path in the tree with vertices $\{v_i\}_{i=1}^{\ell^*}$ such that v_f is the *last* heavy vertex in the sequence. Next, we have the following lemma.

Lemma 10 *There is a procedure* Reconstruct-right-path *which with high probability computes h_v up to $\pm\Theta(\sqrt{\log n/n})$ for any v_ℓ in the path P where $\ell \leq f$.*

Proof Consider vertex v_ℓ, since $\ell \leq f$, we know that $\mathsf{NL}(v_\ell) \geq \alpha n + 1$. Let the number of leaves in the left (resp. right) subtree of v_ℓ be s_ℓ (resp. s_r). We have $s_\ell + s_r \geq \alpha n + 1$. It immediately follows that the number of pairs of the form (a_i, b_i) where a_i is in the left subtree and b_i is in the right subtree is at least αn. Now, let c be a leaf in the left subtree of the root r – we know that there exists at least one.

Let us now define $\mathbf{X}_i$ to be the indicator variable that the query on triple (a_i, b_i, c) returns (a_i, b_i). Observe that each $\mathbf{X}_i$ is an i.i.d. Bernoulli random variable such that $\Pr[\mathbf{X}_i = 1] = \frac{1}{2+h_{v_\ell}}$. Now repeating the same calculation as done at end of Lemma 9, we obtain an estimate h'_v such that with overwhelmingly high probability, $|h'_v - h_v| = \Theta(\sqrt{\log n/n})$.

■

Lemma 11 *Let v be an internal vertex of the tree and let v_ℓ be a heavy vertex on path P such that (i) v_ℓ is an ancestor of vertex v; (ii) There are $k \geq 100$ leaves in the sub-tree of v_ℓ that does not contain v. Let a and b be leaves in distinct subtrees of v and c a vertex in the subtree of v_l not containing v. Let p be the probability that a query on the triple (a, b, c) returns (a, b). Note that p is independent of the particular choices of a, b, and c as constrained above. Then, there is an algorithm that outputs an estimate $\hat{p}$ such that*

1. $|\hat{p} - p| = O\big(\sqrt{\log k/k}\big)$.
2. *$\hat{p}$ is an unbiased estimator of p.*

We say that vertex v has been anchored to vertex v_l to obtain this estimate.

Proof

We index the leaves in the sub-tree of v_ℓ not containing v as $c_1, c_2, ..c_k$. Let $c \in \{c_1, \ldots, c_k\}$. As noted in the statement of the lemma, the probability that $Q(a, b, c)$ returns (a, b) is a constant p independent of c.

The pairwise distances are $d(a, b) = 2h_v$ and $d(a, c) = d(b, c) = 2h_{v_\ell}$. This implies that $p = \frac{h_{v_\ell}}{2h_{v_\ell}+h_v}$. Let $\mathbf{X}_i$ be the indicator random variable that $Q(a, b, c_i)$ returns (a, b). Note that

$$\Pr[\mathbf{X}_i = 1] = \frac{h_{v_\ell}}{2h_{v_\ell} + h_v}.$$

Now, define $\hat{p} := (\sum_{i=1}^{k} \mathbf{X}_i)/k$. Hoeffding's inequality (Theorem 16) and linearity of expectation now immediately imply the claim. ■

Lemma 12 *Suppose v is an internal vertex which lies in the left subtree of v_ℓ – where v_ℓ lies on the path P and $1 \leq \ell \leq f$. There is a procedure* Reconstruct-height-left-heavy *which given as input such a vertex v, reconstructs the height of v up to error $O(\alpha^{-1} \cdot \sqrt{\log n/n})$.*

Proof We are given that v_ℓ is an ancestor of v. The right subtree of v_ℓ has at least αn leaves. Thus, applying Lemma 11, we can obtain an estimate $\hat{p}$ such that

$$\left|\hat{p} - \frac{h_{v_\ell}}{h_v + 2h_{v_\ell}}\right| = O\left(\frac{1}{\alpha} \cdot \sqrt{\frac{\log n}{n}}\right).$$

Now, h_v can be expressed in terms of h_{v_ℓ} and p as follows:

$$h_v = \frac{h_{v_\ell} \cdot (1 - 2p)}{p} := f(v_\ell, p).$$

Using Lemma 10, we can obtain an estimate $\widehat{h_{v_\ell}}$ such that

$$\big|h_{v_\ell} - \widehat{h_{v_\ell}}\big| = \Theta\left(\sqrt{\frac{\log n}{n}}\right).$$

The estimate of the procedure is $\widehat{h_v}$ defined as $f(\widehat{h_{v_\ell}}, \hat{p})$. We now observe that by triangle inequality,

$$\big|f(\widehat{h_{v_\ell}}, \hat{p}) - f(h_{v_\ell}, p)\big| \leq \big|f(h_{v_\ell}, \hat{p}) - f(h_{v_\ell}, p)\big| + \big|f(\widehat{h_{v_\ell}}, \hat{p}) - f(h_{v_\ell}, \hat{p})\big|. \tag{1}$$

To bound the terms on the right hand side, observe that $h_{v_\ell} \leq 1$ and $p \in [1/3, 1]$. Thus, by Lemma 11, we can assume that $\hat{p} \geq 1/4$. Consequently,

$$\left|f(h_{v_\ell}, \hat{p}) - f(h_{v_\ell}, p)\right| = O\left(\frac{|p - \hat{p}|}{p^2}\right) = O(|p - \hat{p}|). \tag{2}$$

$$\left|f(h_{v_\ell}, \hat{p}) - f(\widehat{h_{v_\ell}}, \hat{p})\right| = O\left(\frac{\left|h_{v_\ell} - \widehat{h_{v_\ell}}\right|}{\hat{p}^2}\right) = O\left(\left|h_{v_\ell} - \widehat{h_{v_\ell}}\right|\right). \tag{3}$$

Plugging (2) and (3) into (1) gives us the stated claim. ■

Using Lemma 10 and Lemma 12, we have constructed h_v approximately (to additive $O(\pm\sqrt{\log n/n})$) for all vertices except the ones in the subtree of v_{f+1}. For such vertices v, we next show how to compute h_v up to an additive $O(\log n/\sqrt{n})$. Note that this estimate is slightly worse than the ones achieved in Lemma 12 and Lemma 10.

The above lemmas reconstruct h_v for any vertex in the left subtree of any heavy vertex except the last, as well as any vertex in the "rightmost path". We next describe how to compute h_v for the remaining vertices, i.e., the vertices in the subtree rooted at v_{f+1}. At this point, we also note that if there was a vertex v with two heavy children, then the obvious adaptation of Lemma 9 gives us the weight of every edge in the subtree rooted at v, and hence the entire tree T. However, in general, there is no guarantee that this will happen and we have to go for a more significantly more complicated procedure. Towards this, we prove the following crucial lemma.

Lemma 13 *There is a procedure* Reconstruct-Internal-left *which given any internal vertex v in the subtree of v_{f+1}, outputs an estimate of h_v that has additive error $O(\log n/\sqrt{n})$.*

This lemma completes the proof of Theorem 5, since we have successfully reconstructed the heights of all internal vertices of the tree within additive error $O(\frac{\log n}{\sqrt{n}})$.

Recall that the vertices in path P are $v_0, \ldots, v_f$ (in order) where v_0 is the root and v_f is the last heavy vertex. By definition, this implies that the total number of leaves in the left subtrees of v_0, v_1, $\ldots$, v_f is at least $(1-\alpha)n - 1$. For $0 \leq i \leq f$, let C_i be the set of leaves in the left subtree of v_i. Then $\sum_{i=0}^{f} |\mathsf{C}_i| \geq (1-\alpha)n - 1 > n/2$ (as $\alpha \leq 0.49$). Let $\mathcal{A} := \{v_i : |\mathsf{C}_i| < n/(4(1+f))\}$. It easily follows that $\sum_{i \notin A} |\mathsf{C}_i| > n/4$.

Let the elements of $\overline{\mathcal{A}}$ (in order) be $t_1, \ldots, t_k$. Let $c \in \mathsf{C}_{t_i}$ and let a and b be vertices in different subtrees of v. Then, note that the probability with which $Q(a, b, c)$ returns (a, b) is p_i where $p_i = h_{t_i}/(2h_{t_i} + h_v)$. We now apply Lemma 11 and using v_{t_i} as an anchor for v – with overwhelmingly high probability$(1 - \frac{1}{n^6})$, this gives us an estimate $\hat{p}_i$ such that

$$|\hat{p}_i - p_i| \leq 4\left(\sqrt{\frac{\log |\mathsf{C}_{t_i}|}{|\mathsf{C}_{t_i}|}}\right)$$

At this point, one might ask whether any of the estimates $\hat{p}_i$ is good enough to construct a good estimate $\widehat{h}_v$ for the height of vertex v. However, note that the best guarantee that we can give for any $|C_{t_i}|$ is at most $\theta(\sqrt{n}\log n)$, since the number of vertices on a root to leaf path (to which f may be comparable) can potentially be as bad as $\frac{\sqrt{n}}{\log n}$. To demonstrate such an instance, consider the tree where each edge the rightmost path from the root to leaf path has weight $\theta(\frac{\sqrt{n}}{\log n})$, and each heavy

vertex has $\theta(\sqrt{n}\log n)$ vertices in its left subtree. With such a guarantee, using similar techniques as in the proof of Lemma 12, we can only obtain an estimate for h_v such that the additive error is upper bounded by $O(\frac{1}{n^{1/4}\log n})$. Such an estimate falls well short of our target of $O(\frac{\log n}{n})$ additive error.

To get a better estimate, we use the fact that the set of random variables $\{\hat{p}_i\}_{i\in[k]}$ are independent, owing to the fact that they are functions of disjoint sets of queries. We expect to see that errors in these random variables balance out when we aggregate them in some fashion. A natural approach is to use each $\hat{p}_i$ to construct an estimator $\widehat{h}_v^i = \widehat{h}_{t_i}(\frac{1}{\hat{p}_i} - 2)$ for h_v and take the weighted average (weighted by the $|C_{t_i}|$s). However, this approach runs into difficulties, triggered by the fact that the random variables $\widehat{h}_v^i$ are not unbiased estimators for h_v. To avoid this issue, we aggregate the $\hat{p}_i$'s directly and then recover a single estimate $\widehat{h}_v$ for h_v from the aggregated quantity. In particular, we focus on estimating the quantity $Q := \frac{\sum_i |C_{t_i}|p_i}{\sum_i |C_{t_i}|}$ through an estimator $\widehat{Q}$ and recovering a good estimate $\widehat{h}_v$ forh_v using the estimate $\widehat{Q}$. The proof is complete if we prove the following two claims.

Claim 3 *From the estimators $\{\hat{p}_i\}_{i\in[k]}$, there exists an algorithm to recover an estimator $\widehat{Q}$ for $Q = \frac{\sum_i |C_{t_i}|p_i}{\sum_i |C_{t_i}|}$ such that $|\widehat{Q} - Q| \leq \theta\left(\frac{\log n}{\sqrt{n}}\right)$.*

Claim 4 *Given a good estimate $\widehat{Q}$ for $Q = \frac{\sum_i |C_{t_i}|p_i}{\sum_i |C_{t_i}|}$, i..e, with additive error $O\left(\frac{\log n}{\sqrt{n}}\right)$, there exists an algorithm* Final- Estimate *that uses estimators $\{\widehat{h}_{t_i}\}_{i\in[k]}$ to construct an estimator $\widehat{h}_v^f$ for h_v with additive error $O\left(\frac{\log n}{\sqrt{n}}\right)$.*

Proof of Claim 3: Assume, for sake of a thought experiment, that each $\hat{p}_i$ is restricted to a range of $\theta\left(\sqrt{\frac{\log|C_{t_i}|}{|C_{t_i}|}}\right)$ around p_i. Then, if we take the weighted average $\frac{\sum_i |C_{t_i}|\hat{p}_i}{\sum_i |C_{t_i}|}$ of the $\hat{p}_i$s, the size of the range of each term $\frac{|C_{t_i}|\hat{p}_i}{\sum_i |C_{t_i}|}$ is at most $\theta\left(\frac{\sqrt{|C_{t_i}|\log n}}{n}\right)$. Thus, the variance of the sum is upper bounded by $O(\frac{\log n}{n})$ (this uses the fact that $n \geq \sum_i |C_{t_i}| \geq \frac{n}{4}$). Using an exponential tail bound would lead us to the desired result under this thought experiment.

Consequently, the natural approach is to directly take the weighted average of the $\hat{p}_i$s, since this would be an unbiased estimator for Q. However, we only have the guarantee that each $\hat{p}_i$ is at most $\theta\left(\sqrt{\frac{|C_{t_i}|\log n}{|C_{t_i}|}}\right)$ away from p_i as a high probability event rather than as absolute truth. To get around this roadblock, for each $i \in [k]$, we define a real valued random variable $\widetilde{p}_i$, and introduce the following coupling between $\hat{p}_i$ and $\widetilde{p}_i$ based on truncating $\hat{p}_i$:

$$\widetilde{p}_i = \begin{cases} \hat{p}_i, \text{ when } |\hat{p}_i - p_i| \leq 4\left(\sqrt{\frac{\log|C_{t_i}|}{|C_{t_i}|}}\right) \\ p_i + 4\left(\sqrt{\frac{\log|C_{t_i}|}{|C_{t_i}|}}\right), \text{ when } \hat{p}_i - p_i > 4\left(\sqrt{\frac{\log|C_{t_i}|}{|C_{t_i}|}}\right) \\ p_i - 4\left(\sqrt{\frac{\log|C_{t_i}|}{|C_{t_i}|}}\right), \text{ otherwise} \end{cases}$$

We complete the description of the algorithm by defining our final estimator : $\widehat{Q} := \frac{|C_{t_i}|\hat{p}_i}{\sum_i |C_{t_i}|}$.

We observe that the random variables $\{\widetilde{p}_i\}_{i\in[k]}$ are independent. This follows from the fact that the random variables $\{\hat{p}_i\}_{i\in[k]}$ are themselves independent. As a consequence of this doing this truncation, the resultant $\widetilde{p}_i$ is no longer an unbiased estimator of p_i, however we show that this does not functionally affect us. To do so, we prove a claim showing that the expectations of the coupled random variables are very close to each other.

Claim 5 *For each $i \in [k]$, $|\mathbb{E}[\widetilde{p}_i] - \mathbb{E}[\hat{p}_i]| \leq \frac{2}{n^6}$.*

Proof We already know that $\Pr[|\hat{p}_i - p_i| \geq \kappa_1\left(\sqrt{\frac{\log|\mathsf{C}_{t_i}|}{|\mathsf{C}_{t_i}|}}\right)] \leq \frac{1}{n^6}$. Additionally, the random variable $\hat{p}_i$ takes its value in the range $[0,1]$ since it is an empirical average. This also implies that $\mathbb{E}[\hat{p}_i] = p_i \in [0,1]$. Thus, we have:

$$|\mathbb{E}[\widetilde{p}_i] - \mathbb{E}[\hat{p}_i]| \leq |\frac{1-p_i}{n^6}| + |\frac{p_i}{n^6}|$$

which gives us the desired result. ■

Now, define $j_i := |C_{t_i}|$ for each $i \in [k]$. Define the function $g(\widetilde{p}_1, \widetilde{p}_2, ..\widetilde{p}_k) := \frac{\sum_{i=1}^k j_i\widetilde{p}_i}{\sum_{i=1}^k j_i}$ on the domain $\mathbb{R}^k$ Each random variable $\frac{j_i\widetilde{p}_i}{\sum_{l=1}^k j_l}$ is within the interval $[a_i, b_i]$ such that $L_i = |b_i - a_i| = \theta(\frac{\sqrt{j_i \log n}}{n})$. We know that $\sum_{i=1}^k L_i^2 = \sum_{i=1}^k \theta(\frac{j_i \log n}{n^2}) = \theta(\frac{\log n}{n})$.

Using the above property, we appeal to the generalized Hoeffding bound (Theorem 15).With probability at most $\frac{1}{n^6}$, we have $|g(\widetilde{p}_1, \widetilde{p}_2, .., \widetilde{p}_k) - \mathbb{E}[g(\widetilde{p}_1, \widetilde{p}_2, .., \widetilde{p}_k)]| \geq \theta\left(\frac{\log n}{\sqrt{n}}\right)$. Using Claim 5, we know that $|\mathbb{E}[g(\widetilde{p}_1, \widetilde{p}_2, .., \widetilde{p}_k)] - \mathbb{E}[\frac{\sum_i j_i\hat{p}_i}{\sum_i j_i}]| \leq \frac{2}{n^6}$. Using linearity of expectation, we know that $\mathbb{E}[\frac{\sum_i j_i\hat{p}_i}{\sum_i j_i}] = \frac{\sum_i j_i p_i}{\sum_i j_i} = Q$. Using the triangle inequality, the event $|\widehat{Q} - Q| \leq \theta\left(\frac{\log n}{\sqrt{n}}\right)$ happens with high probability.

■

Proof of Claim 4:

First, we describe how to recover estimator $\widehat{h}_v^f$ for h_v from $\widehat{Q}$ using the estimates $\{\widehat{h}_{t_i}\}_{i\in[k]}$.

Define $j_i := |C_{t_i}|$ for each $i \in [k]$. We define a multivariate function F that operates on inputs $a \in \mathbb{R}$ and $b = (b_1, b_2, \cdots b_k) \in \mathbb{R}^k$.

$$F(a,b) := \frac{\sum_{i=1}^k \frac{j_i b_i}{2b_i + a}}{\sum_i j_i}$$

Alternatively, we write $F(a,b) = \frac{\sum_{i=1}^k j_i f(a,b_i)}{\sum_i j_i}$ where $f(a,b_i) = \frac{b_i}{2b_i+a}$, i.e., we write f as a convex combination of k functions. For ease of notation, we call $\alpha_i = \frac{j_i}{\sum_{a=1}^k j_a}$. Note that all $\alpha > 0$ and $\sum_i \alpha_i = 1$.

Let $H = \{h_{t_1}, h_{t_2}, \cdots h_{t_k}\}$ and $\widehat{H} = \{\widehat{h}_{t_1}, \widehat{h}_{t_2}, \cdots \widehat{h}_{t_k}\}$. Observe that $Q = F(h_v, H)$. Intuitively, we want to invert the value $\widehat{Q}$ of the function F using our estimates H' to obtain an estimate $\widehat{h}_v$ that is close to h_v. We prove a claim below that tells us how to do so.

Claim 6 *There exists a unique real number $\widehat{h}_v$ such that $F(\widehat{h}_v, \widehat{H}) = \widehat{Q}$. Additionally, this real number can be found using numerical methods from $\widehat{Q}$ and $\widehat{H}$.*

Proof Fix the second argument of F to be $\widehat{H}$, F is now a univariate function on the first argument. This function is monotone decreasing since each univariate function $f(a, \widehat{h}_{t_i})$ is monotone decreasing and $F(a, \widehat{H})$ is a convex combination of these functions. Thus, F is invertible. Further, simple numerical methods such as binary search can be used to find $\widehat{h}_v$ upto arbitrary precision such that $F(\widehat{h}_v, \widehat{H}) = \widehat{Q}$. ■

The algorithm then makes the following final correction to the estimate : $\widehat{h}_v^f \leftarrow \min\{\widehat{h}_v, \min_i \widehat{h}_{t_i}\}$ - there is a compelling reason to make such a correction - to ensure that the vertex v always has estimated height that is no larger than the estimated height of one of its ancestors. Note that since we eventually show that the error in estimating edge weight is only a fraction of the minimum edge weight, such an event never happens in practice - however, we include this correction for the sake of the analysis leading to the proof of that result.

First, we show a simplified analysis, for the case that $\widehat{h}_v \leq \min_i \widehat{h}_{t_i}$. Here the final estimator is $\widehat{h}_v^f = \widehat{h}_v$. We will later show a more involved analysis for when our estimate does not satisfy these conditions.

For the sake of contradiction, assume that $|\widehat{h}_v - h_v| \geq \Delta \frac{\log n}{\sqrt{n}}$, where Δ is a sufficiently large constant. Since $|\widehat{Q} - Q| \leq \theta(\frac{\log n}{\sqrt{n}})$, we get:

$$|\sum_i \alpha_i \{f(h_v, h_{t_i}) - f(\widehat{h}_v, \widehat{h}_{t_i})\}| \leq \theta(\frac{\log n}{\sqrt{n}})$$

$$\implies |\sum_i \alpha_i \{(f(h_v, h_{t_i}) - f(\widehat{h}_v, h_{t_i})) + (f(\widehat{h}_v, h_{t_i}) - f(\widehat{h}_v, \widehat{h}_{t_i}))\}| \leq \theta(\frac{\log n}{\sqrt{n}}) \qquad (*)$$

Our approach is to show that each $A_i := |f(h_v, h_{t_i}) - f(\widehat{h}_v, h_{t_i})|$ is sufficiently larger (by at least some $\Omega(\frac{\log n}{\sqrt{n}})$ than any $B_i := |f(\widehat{h}_v, h_{t_i}) - f(\widehat{h}_v, \widehat{h}_{t_i})|$. Using the triangle inequality would finish the proof. To this purpose, recall that $|\widehat{h}_{t_i} - h_{t_i}| \leq \theta(\sqrt{\frac{\log n}{n}})$.

Intuitively, we wish to show that the function $f(a, b_i)$ changes rapidly with change in the first input a (large partial derivative in our interval of interest) while it is relatively more stable with change in the second input b_i (small partial derivative in our interval of interest). To do so, we prove some properties of the function $f(a, b_i)$.

$$\frac{\partial f(a, b_i)}{\partial b_i} = \frac{a}{(2b_i + a)^2}$$

$$\frac{\partial f(a, b_i)}{\partial a} = \frac{-b_i}{(2a_i + b)^2}$$

We know that $h_{t_i} > h_v$ for all indices i. Additionally, $h_{t_i} \geq \tau \frac{\log n}{\sqrt{n}}$ where τ is a large constant. We argue that it is not possible to independently bound each quantity A_i and B_i - to see why this is the case - observe that the partial derivative with respect to b_i can only be upper bounded by a constant while the partial derivative with respect to a can be as small as $\theta(\frac{\log n}{\sqrt{n}})$. These bounds do not suffice to prove the desired comparison between the quantities A_i and B_i. Therefore, it is important that we carefully compare these partial derivatives when establishing that A_i is significantly larger than b_i. More formally : Using the intermediate value theorem on the (continuous) univariate

function $f(a, h_{t_i})$ (Second argument fixed to be h_{t_i}) in the interval $[h_v, h'_v]$, we know there exists $x \in [h_v, h'_v]$ such that:

$$\begin{aligned} A_i = |f(h_v, h_{t_i}) - f(\widehat{h}_v, h_{t_i})| &= \left| \frac{\partial f(a, h_{t_i})}{\partial a} \right|_{a=x} .|h_v - \widehat{h}_v| \\ &= \left| \frac{h_{t_i}}{(2h_{t_i} + x)^2} \right| .|h_v - \widehat{h}_v| \end{aligned}$$

Similarly, using the intermediate value theorem on the univariate function $f(\widehat{h}_v, b)$ (first argument fixed to be $\widehat{h}_v$) in the interval $[h_{t_i}, \widehat{h}_{t_i}]$, we know there exists $y \in [h_{t_i}, \widehat{h}_{t_i}]$ such that:

$$\begin{aligned} B_i = |f(\widehat{h}_v, h_{t_i}) - f(\widehat{h}_v, \widehat{h}_{t_i})| &= \left| \frac{\partial f(\widehat{h}_v, b)}{\partial b} \right|_{b=y} .|h_{t_i} - \widehat{h}_{t_i}| \\ &= \left| \frac{\widehat{h}_v}{(2y + \widehat{h}_v)^2} \right| .|h_{t_i} - \widehat{h}_{t_i}| \end{aligned}$$

Since $|\widehat{h}_{t_i} - h_{t_i}| \leq \theta(\sqrt{\frac{\log n}{n}})$ and $h_{t_i} \geq \tau \frac{\log n}{\sqrt{n}}$, we know that $2\widehat{h}_{t_i} \geq h_{t_i} \geq \widehat{h}_{t_i}/2$. We also know that $\widehat{h}_v \leq \widehat{h}_{t_i}$ by our assumption and hence $\widehat{h}_v \leq 2h_{t_i}$ which would imply that $x \leq 2h_{t_i}$. Thus, we have $\frac{h_{t_i}}{(2h_{t_i}+x)^2} \geq \frac{h_{t_i}}{16h_{t_i}^2} = \frac{1}{16h_{t_i}}$ and $\frac{\widehat{h}_v}{(2y+\widehat{h}_v)^2} \leq \frac{2}{h_{t_i}}$. Note that using the triangle inequality $|A_i| - |B_i| \geq ||A_i| - |B_i||$. Thus, as long as τ is large enough, we conclude that $|A_i| - |B_i| \geq \tau/20 \frac{\log n}{\sqrt{n}}$. Substituting this into inequality *, we get a contradiction, leading us to conclude that $|h'_v - h_v| \leq \Delta \frac{\log n}{\sqrt{n}}$.

The final part of the proof is for the case where there exists some i such that $\widehat{h}_v > \widehat{h}_{t_i}$. In this event, recall that our algorithm uses the smallest such height $U = \min_i \{\widehat{h}_{t_i}\}$ as the final estimate $\widehat{h}_v^f$ for h_v. We show that $|U - h_v| \leq \Delta \frac{\log n}{\sqrt{n}}$ with high probability. First, we note that $U > h_v$.Assume that $|U - h_v| > \Delta \frac{\log n}{\sqrt{n}}$, for sake of contradiction. We will show similar bounds on the expressions A_i and B_i from the above proof. First, we lower bound $|A_i|$. Consider the (continuous) univariate function $f(a, h_{t_i})$ on the intervals $[h_v, U]$ and $[U, \widehat{h}_v]$. Using the intermediate value theorem on both intervals, there exists $x_1 \in [h_v, U]$ and $x_2 \in [U, \widehat{h}_v]$ such that:

$$\begin{aligned} A_i := |f(h_v, h_{t_i}) - f(\widehat{h}_v, h_{t_i})| &= \left| \frac{\partial f(a, h_{t_i})}{\partial a} \right|_{a=x_1} .|U - h_v| + \left| \frac{\partial f(a, h_{t_i})}{\partial a} \right|_{a=x_2} .|\widehat{h}_v - U| \\ &\geq \left| \frac{\partial f(a, h_{t_i})}{\partial a} \right|_{a=x_1} .|U - h_v| \\ &= \left| \frac{h_{t_i}}{(2h_{t_i} + x_1)^2} \right| .|U - h_v| \end{aligned}$$

Thus, with similar reasoning as before, we have $\frac{h_{t_i}}{(2h_{t_i}+x_1)^2} \geq \frac{h_{t_i}}{16h_{t_i}^2} = \frac{1}{16h_{t_i}}$ since $x_1 \leq \widehat{h}_{t_i}$ for all i.

Now, to upper bound B_i, we consider the (continuous) univariate function $f(\widehat{h}_v, b)$ on the interval $[h_{t_i}, \widehat{h}_{t_i}]$, we know there exists $y \in [h_{t_i}, \widehat{h}_{t_i}]$ such that:

$$B_i := |f(\widehat{h}_v, h_{t_i}) - f(\widehat{h}_v, \widehat{h}_{t_i})| = \left| \frac{\partial f(\widehat{h}_v, b)}{\partial b} \right|_{b=y} .|h_{t_i} - \widehat{h}_{t_i}|$$
$$= \left| \frac{\widehat{h}_v}{(2y + \widehat{h}_v)^2} \right| .|h_{t_i} - \widehat{h}_{t_i}|$$

Consider the univariate function $g(x) = \frac{x}{(2c+x)^2}$. Note that $g(x) > 0$ if $c, x > 0$. Thus, $|g(x)|$ (for $c, x > 0$) is maximized at the same point as $g(x)$ which is at $x = 2c$. Thus, $\frac{\widehat{h}_v}{(2y+\widehat{h}_v)^2} \leq \frac{2y}{16y^2} = \frac{1}{8y} \leq \frac{1}{4h_{t_i}}$. Similar analysis as before leads us to a contradiction (we can show that $|A_i| \geq 2|B_i|$, which suffices to prove the result).

■

Appendix C. Proof of Theorem 6 (Lower Bound on Edge Weights)

Let us assign an arbitrary ordering on the set of all possible triples of leaves (r, s, t) from 1 to k where $k = \binom{n}{3}$. Let $\mathbf{Q}_j^{(1)}$ (resp. $\mathbf{Q}_j^{(2)}$) denote the random variable corresponding to the response on the j^{th} query (i.e., triple) from tree T_1 (resp. T_2). Let $\mathbf{Q}^{(1)}$ (resp. $\mathbf{Q}^{(2)}$) denote the $\binom{n}{3}$-dimensional random variable whose j^{th} coordinate is $\mathbf{Q}_j^{(1)}$ (resp. $\mathbf{Q}_j^{(2)}$). Both $\mathbf{Q}^{(1)}$ and $\mathbf{Q}^{(2)}$ follow product distributions. To prove our theorem, it suffices to prove the following

$$\Delta(\mathbf{Q}^{(1)}, \mathbf{Q}^{(2)}) \leq 0.01.$$

Here $\Delta(\cdot, \cdot)$ refers to the total variation distance. The above inequality follows from Lemma 14 which we prove next.

Lemma 14 *For $\mathbf{Q}^{(1)}$ and $\mathbf{Q}^{(2)}$ as described above, $\Delta(\mathbf{Q}^{(1)}, \mathbf{Q}^{(2)}) \leq 0.01$.*

Proof We start by partitioning the set $[k]$ into five different sets defined below:

1. Let $L' = L \setminus \{a, b, c\}$. Define $\mathcal{A}_1 \subseteq [k]$ to be the set of those indices which correspond to triples (r, s, t) such that $|\{r, s, t\} \cap L'| \geq 2$.

2. Define $\mathcal{A}_2 \subseteq [k]$ to be the set of those indices which correspond to triples (r, b, c) where $r \in L'$.

3. Let j^* be the index which corresponds to the triple (a, b, c). Let $\mathcal{A}_3 = \{j^*\}$.

4. Let $\mathcal{A}_4$ be the set of those indices which correspond to the triples of the form (a, b, x) for $x \in L'$.

5. Let $\mathcal{A}_5$ be the set of those indices which correspond to the triples of the form (a, c, x) for $x \in L'$.

In order to analyze $\Delta(\mathbf{Q}^{(1)}, \mathbf{Q}^{(2)})$, it is more convenient to look at the notion of Hellinger distance (see Definition 18). We start with the following easy claims.

Claim 7 *Let $j \in \mathcal{A}_1$. Then the random variables $\mathbf{Q}_j^{(1)}$ and $\mathbf{Q}_j^{(2)}$ are identical. Thus, $H^2(\mathbf{Q}_j^{(1)}, \mathbf{Q}_j^{(2)}) = 0$.*

Proof Let j correspond to the triple (r, s, t). Note that the random variables $\mathbf{Q}_j^{(1)}$ and $\mathbf{Q}_j^{(2)}$ are just dependent on the pairwise distances between the leaves r, s and t. It is easy to observe that as long as two of these leaves are in L', their pairwise distances are same both in trees T_1 and T_2. This proves the claim. ■

Claim 8 *Let $j \in \mathcal{A}_2$. Then the random variables $\mathbf{Q}_j^{(1)}$ and $\mathbf{Q}_j^{(2)}$ are identical. Thus, $H^2(\mathbf{Q}_j^{(1)}, \mathbf{Q}_j^{(2)}) = 0$.*

Proof Let j correspond to the triple (r, c, b). As in Claim 7, the pairwise distances between the vertices r, c and b are the same in both T_1 and T_2. The claim now follows. ■

Using Claim 8 and Claim 7, it follows tha ■

Claim 9 *Let $\mathcal{A}_3 = \{j^*\}$. Then, $H^2(\mathbf{Q}_{j^*}^{(1)}, \mathbf{Q}_{j^*}^{(2)}) \leq \frac{4\rho^2}{n}$.*

Proof Observe that both random variables $\mathbf{Q}_{j^*}^{(1)}$ and $\mathbf{Q}_{j^*}^{(2)}$ are supported on the set $\{(a, b), (b, c), (a, c)\}$. Let $p_1 := \Pr[\mathbf{Q}_{j^*}^{(1)} = (a, b)]$, $p_2 := \Pr[\mathbf{Q}_{j^*}^{(1)} = (b, c)]$ and $p_3 := \Pr[\mathbf{Q}_{j^*}^{(1)} = (a, c)]$. Similarly, let $q_1 := \Pr[\mathbf{Q}_{j^*}^{(2)} = (a, b)]$, $q_2 := \Pr[\mathbf{Q}_{j^*}^{(2)} = (b, c)]$ and $q_3 := \Pr[\mathbf{Q}_{j^*}^{(2)} = (a, c)]$.

Let $d_1(\cdot, \cdot)$ be the distance metric on tree T_1 and $d_2(\cdot, \cdot)$ be the distance metric on tree T_2. Now, define $\alpha := d_1(a, b) = d_2(a, c)$ and $\beta := d_1(a, c) = d_1(b, c) = d_2(a, b) = d_2(b, c)$.

Observe that $\alpha \geq \frac{2}{3}$ since $d_1(a, p) = d_2(a, p) = \frac{1}{3}$. Also, $\beta - \alpha = \frac{2\rho}{\sqrt{n}}$.

$$H^2(\mathbf{Q}_{j^*}^{(1)}, \mathbf{Q}_{j^*}^{(2)}) = \frac{1}{2}((\sqrt{p_1} - \sqrt{q_1})^2 + (\sqrt{p_2} - \sqrt{q_2})^2 + (\sqrt{p_3} - \sqrt{q_3})^2)$$

.

We rewrite the probabilities in terms of the distances α and β:

$$p_1 = q_3 = \frac{2\beta}{2(\alpha + 2\beta)};\ p_2 = p_3 = q_1 = q_2 = \frac{\alpha + \beta}{2(\alpha + 2\beta)}$$

Next, we upper bound the quantity: $\sqrt{p_1} - \sqrt{q_1}$.

$$\begin{aligned}
\sqrt{p_1} - \sqrt{q_1} &= \sqrt{\frac{2\beta}{2(\alpha+2\beta)}} - \sqrt{\frac{\alpha+\beta}{2(\alpha+2\beta)}} \\
&= \frac{\sqrt{2\beta} - \sqrt{\alpha+\beta}}{\sqrt{2(\alpha+2\beta)}} \\
&\leq \sqrt{\frac{4}{3}}(\sqrt{2\beta} - \sqrt{\alpha+\beta}) \qquad (\text{ since } \alpha \geq \frac{2}{3}) \\
&= \sqrt{\frac{4}{3}}\frac{\beta-\alpha}{(\sqrt{2\beta}+\sqrt{\alpha+\beta})} \qquad (\text{ multiplying and dividing by } (\sqrt{2\beta}+\sqrt{\alpha+\beta})) \\
&\leq \frac{2\rho}{\sqrt{n}} \qquad (\text{ since } \alpha \geq \frac{2}{3})
\end{aligned}$$

We know $p_2 = q_2$. The final term of interest is $|\sqrt{p_3} - \sqrt{q_3}|$. Since $p_1 = q_3$ and $p_3 = q_1$, we get the upper bound - $|\sqrt{p_3} - \sqrt{q_3}| \leq \frac{2\rho}{\sqrt{n}}$.

Putting together these inequalities, we get $H^2(\mathbf{Q}_{j^*}^{(1)}, \mathbf{Q}_{j^*}^{(2)}) \leq \frac{1}{2}(\frac{8\rho^2}{n})$. ■

Claim 10 *Let $j \in \mathcal{A}_4$. Then, $H^2(\mathbf{Q}_j^{(1)}, \mathbf{Q}_j^{(2)}) \leq \frac{\rho^2}{4n}$.*

Proof Let j correspond to the triple (a, b, x). The support space of both $\mathbf{Q}_j^{(1)}$ and $\mathbf{Q}_j^{(2)}$ is $\{(a,b),(b,x),(a,x)\}$. Let $p_1 := \Pr[\mathbf{Q}_j^{(1)} = (a,b)]$, $p_2 := \Pr[\mathbf{Q}_j^{(1)} = (b,x)]$ and $p_3 := \Pr[\mathbf{Q}_j^{(1)} = (a,x)]$. Similarly, let $q_1 := \Pr[\mathbf{Q}_j^{(2)} = (a,b)]$, $q_2 := \Pr[\mathbf{Q}_j^{(2)} = (b,x)]$ and $q_3 := \Pr[\mathbf{Q}_j^{(2)} = (a,x)]$.

As in Claim 9, let $d_1(\cdot,\cdot)$ be the distance metric on tree T_1 and $d_2(\cdot,\cdot)$ be the distance metric on T_2. Let $\alpha := d_1(a,b)$ and $\beta := d_2(a,b)$. We have $d_1(a,x) = d_1(b,x) = d_2(a,x) = d_2(b,x) = 2$, since the associated unique path goes through the root for each of these pairs. Finally, $\beta - \alpha = \frac{2\rho}{\sqrt{n}}$. Using the definition of Hellinger distance, we have :

$$H^2(\mathbf{Q}_j^{(1)}, \mathbf{Q}_j^{(2)}) = \frac{1}{2}((\sqrt{p_1} - \sqrt{q_1})^2 + (\sqrt{p_2} - \sqrt{q_2})^2 + (\sqrt{p_3} - \sqrt{q_3})^2)$$

. Thus, we have,

$$p_1 = \frac{4}{2(4+\alpha)};\ q_1 = \frac{4}{2(4+\beta)};\ p_2 = p_3 = \frac{2+\alpha}{2(4+\alpha)};\ q_2 = q_3 = \frac{2+\beta}{2(4+\beta)}$$

Next, we upper bound the quantity $\sqrt{p_1} - \sqrt{q_1}$ as follows:

$$\begin{aligned}
\sqrt{p_1} - \sqrt{q_1} &= \sqrt{2}\left(\frac{\sqrt{4+\beta} - \sqrt{4+\alpha}}{\sqrt{(4+\alpha)(4+\beta)}}\right) \\
&\leq \frac{1}{2\sqrt{2}}(\sqrt{4+\beta} - \sqrt{4+\alpha}) \\
&\leq \frac{1}{2\sqrt{2}(\sqrt{4+\beta}+\sqrt{4+\alpha})}(\beta - \alpha) \\
&\leq \frac{1}{4\sqrt{2}}\frac{\rho}{\sqrt{n}} \qquad (4)
\end{aligned}$$

Next, we upper bound $\sqrt{q_2} - \sqrt{p_2}$ as follows:

$$\begin{aligned}
\sqrt{q_2} - \sqrt{p_2} &= \frac{1}{\sqrt{2}} \frac{\sqrt{(2+\beta)(4+\alpha)} - \sqrt{(4+\beta)(2+\alpha)}}{\sqrt{(4+\alpha)(4+\beta)}} \\
&\leq \frac{1}{4\sqrt{2}} (\sqrt{8+\alpha\beta+4\beta+2\alpha} - \sqrt{8+\alpha\beta+2\beta+4\alpha}) \\
&= \frac{1}{4\sqrt{2}(\sqrt{8+\alpha\beta+4\beta+2\alpha} + \sqrt{8+\alpha\beta+2\beta+4\alpha})} (2\beta - 2\alpha) \\
&\leq \frac{1}{8\sqrt{2}} \frac{\rho}{\sqrt{n}} \qquad (5)
\end{aligned}$$

By an identical analysis, we have $\sqrt{q_3} - \sqrt{p_3} \leq \frac{1}{8\sqrt{2}} \frac{\rho}{\sqrt{n}}$. Combining this with (4) and (5) gives us the claim. ■

Using an analysis identical to Claim 10, we also get the following Claim.

Claim 11 *Let $j \in \mathcal{A}_5$. Then, $H^2(\mathbf{Q}_j^{(1)}, \mathbf{Q}_j^{(2)}) \leq \frac{\rho^2}{4n}$.*

Putting together these claims, we can use Lemma 20 to show that $H(\mathbf{Q}^{(1)}, \mathbf{Q}^{(2)}) \leq \frac{0.01}{\sqrt{2}}$, and then use Lemma 19 to conclude that $\Delta(\mathbf{Q}^{(1)}, \mathbf{Q}^{(2)}) \leq 0.01$.

Appendix D. Concentration Bounds and Measures of Statistical Distance

We list here some standard concentration results, that will be used to aggregate the results of stochastically independent queries

Theorem 15 (Generalized Hoeffding Bound) *Let $\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_k$ be k independent random variables, with each variable $\mathbf{y}_i$ having range $[a_i, b_i]$ and mean y_i then:*

$$\Pr\left[\left|\sum_{i=1}^{k} y_i - \sum_{i=1}^{k} \mathbf{y}_i\right| \leq t\right] \geq 1 - e^{\frac{2t^2}{\sum_i (b_i - a_i)^2}}$$

We state below a special case of the generalized Hoeffding bound that we use repeatedly in proofs, referring to it as the Hoeffding bound.

Theorem 16 (Hoeffding Bound) *Let $\mathbf{y}_1, \mathbf{y}_2, .., \mathbf{y}_k$ be k iid random variables between 0 and 1, each with mean y, then:*

$$\Pr\left[\left|y - \frac{\sum_{i=1}^{k} \mathbf{y}_i}{k}\right| \leq 4\sqrt{\frac{\log n}{k}}\right] \geq 1 - \frac{1}{n^6}$$

We also list some measures of statistical distance and connections between them, that will be employed in our lower bounds proofs. We formally define total variation distance and Hellinger distance for discrete distributions.

Definition 17 (Total Variation Distance) *Let X and Y be discrete distributions, having weight p_{x_i} and p_{y_j} respectively on points z_1, $z_2 \cdots$. Then, the total variation distance between X and Y, denoted by $\Delta(X, Y)$ is defined as:*

$$\Delta(X, Y) := \frac{1}{2} \sum_{i=1}^{\infty} |p_{x_i} - p_{y_i}|$$

An alternative, equivalent definition is as follows: Let the sample space of the two distributions X, Y be Ω, then:

$\Delta(X, Y) = \sup_{A \in \Omega} |X(A) - Y(A)|$.

Definition 18 (Hellinger Distance) *Let X and Y be discrete distributions, having weight $p_{x_1}, p_{x_2}, \cdots$ and $p_{y_1}, p_{y_2}, \cdots$ respectively on points z_1, $z_2 \cdots$. Then, the total variation distance between X and Y, denoted by $H(X, Y)$ is defined as:*

$$H(X, Y) := \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{\infty} (\sqrt{p_{x_i}} - \sqrt{p_{y_i}})^2}$$

We use the following two useful lemmas about Total Variation Distance Hellinger Distance from Barak et al. (2008).

Lemma 19 (Pollard (2001)) *For two distributions X and Y:*

$$H^2(X, Y) \leq \Delta(X, Y) \leq \sqrt{2} H(X, Y)$$

Lemma 20 (Barak et al. (2008)) *Let $X_1, X_2 \cdots X_n$ and $Y_1, Y_2 \cdots Y_n$ be two families of distributions. Then,*

$$H^2(X_1 \oplus X_2 \oplus \cdots X_n, Y_1 \oplus Y_2 \oplus \cdots Y_n) \leq \sum_{i=1}^{n} H^2(X_i, Y_i)$$

where $X \oplus Y$ denotes the product of two distributions X and Y, generated by taking independent samples of X and Y.