Proceedings of Machine Learning Research vol 125:1–37, 2020 33rd Annual Conference on Learning Theory

Parallels Between Phase Transitions and Circuit Complexity?

Ankur Moitra MOITRA@MIT.EDU
MIT Department of Mathematics

Elchanan Mossel ELMOS@MIT.EDU
MIT Department of Mathematics

Colin Sandon CSANDON@COMCAST.NET
MIT Department of Mathematics

Editors: Jacob Abernethy and Shivani Agarwal

Abstract

In many natural average-case problems, there are or there are believed to be critical values in the parameter space where the structure of the space of solutions changes in a fundamental way. These phase transitions are often believed to coincide with drastic changes in the computational complexity of the associated problem.

In this work, we study the circuit complexity of inference in the broadcast tree model, which has important applications in phylogenetic reconstruction and close connections to community detection. We establish a number of qualitative connections between phase transitions and circuit complexity in this model. Specifically we show that there is a $\mathbf{TC}^0$ circuit that competes with the Bayes optimal predictor in some range of parameters above the Kesten-Stigum bound. We also show that there is a 16 label broadcast tree model beneath the Kesten-Stigum bound in which it is possible to accurately guess the label of the root, but beating random guessing is $\mathbf{NC}^1$-hard on average. The key to locating phase transitions is often to study some intrinsic notions of complexity associated with belief propagation – e.g. where do linear statistics fail, or when is the posterior sensitive to noise? Ours is the first work to study the complexity of belief propagation in a way that is grounded in circuit complexity.

Keywords: Belief propagation, circuit complexity, phase transitions, Kesten-Stigum bound

1. Introduction

1.1. Background

In many basic problems in high-dimensional statistics and machine learning, there appear to be fundamental gaps between the performance of the information-theoretically best estimator and the best estimator that can be computed in polynomial time. These are called *computational vs. statistical tradeoffs*. Recently, there has been an effort to study these gaps in a systematic fashion, in particular by forging reductions between some of these problems. For example, finding sparse directions with large variance in the spiked covariance model turns out to be at least as hard as finding small planted cliques, see e.g. Berthet and Rigollet (2013); Ma and Wu (2015); Brennan et al. (2018). However, these reductions leave much to be desired as there are relatively few examples where reductions are known that map natural distributions on one problem to natural distributions on another.

In this paper, we will explore other popular methodologies for predicting where average-case problems become hard, which come from statistical physics and revolve around a powerful algorithm called belief propagation. Our key example originates from the following special case of community detection in the stochastic block model. We start with a fixed partition of n nodes into q (almost) equal sized communities. The probability of connecting any pair of nodes with an edge is $kq\theta/n + k(1-\theta)/n$ if they belong to the same community and otherwise is $k(1-\theta)/n$, where edges in the graph are sampled independently. It is easy to see that the average degree in this graph is k and that θ is a measure of the strength of the communities.

The goal is, given a graph sampled from this model, to find a q-partition of its nodes whose parts have non-trivial correlation (i.e. better than random) with the true communities. A striking prediction from statistical physics (Decelle et al., 2011) is that the problem is efficiently solvable when $k\theta^2 > 1$ while the information theory threshold for the problem is different for large values of q. By now the existence of efficient algorithms when $k\theta^2 > 1$ has been established (Mossel et al., 2015; Massoulié, 2014; Mossel et al., 2018; Bordenave et al., 2015; Abbe and Sandon, 2015) as well as the fact that for $q > 5$, the information theory threshold is strictly below this bound (Abbe and Sandon, 2015; Banks et al., 2016).

The threshold of $k\theta^2 > 1$ is called the *Kesten-Stigum bound* and will play an important role in our paper. It is believed that for some problems, like the block model, the structure of the space of solutions changes in a fundamental way beneath the Kesten-Stigum bound, and this is the basis for the predictions about computational hardness. Fundamentally, these predictions of computational difficulty all revolve around studying the behavior of belief propagation. In what follows we will explain some of the intuition behind belief propagation along with how computational versus statistical phase transitions are predicted. See also Mézard and Montanari (2006); Krząkała et al. (2007).

The way to think about belief propagation in the stochastic block model is to start with a local view around a node. With high probability, its neighborhood will be tree-like. In fact, we can model it (along with which community each node belongs to) as a Markov process on a tree. This model is called the *broadcast tree model*. We start with a complete k-regular tree of height $d < \log_k(n)/2$ (or alternatively we generate a random tree of height d in which the number of children of each node is a Poisson random variable with expectation k). The root is assigned one of the q possible labels at random. Next we propagate labels from the root to the leaves by, at each step, assigning a child the same label as its parent with probability θ and otherwise assigning it a uniformly random label. At the end, we are given the labels of the leaves and the goal is to use this information to guess the label of the root. We want our guess to be correct with some advantage over random guessing, and we want the advantage to be bounded away from zero independently of d. Belief propagation is an iterative algorithm that provably computes the posterior distribution on the label of the root given the labels of the leaves. So when belief propagation fails at guessing the label of the root with some nonzero advantage that is independent of d, it is because the problem is information-theoretically impossible. Belief propagation is based on the idea that conditioned on the label of some node, the labels of its neighbors are independent.

This is exactly true on a tree and approximately true in a sparse random graph with few short cycles.

> *The key to using belief propagation to locate phase transitions is that it has its own intrinsic notions of complexity.*

In the broadcast tree model, the Kesten-Stigum bound is the threshold $k\theta^2 > 1$. (The Kesten-Stigum bound in the stochastic block model is usually stated in terms of a and b but they are actually the same, which can be seen by relating a, b, θ and k). It turns out that the Kesten-Stigum bound coincides with where linear statistics stop working. In fact, in the seminal work of Kesten and Stigum (Kesten and Stigum, 1967, 1966), they showed that it is possible to guess the label of the root (and beat random guessing) just by tallying the number of labels of each type among the leaves. Moreover, it is not too hard to deduce from their results (Mossel and Peres, 2003) that below the Kesten-Stigum bound, this method fails. Perhaps surprisingly, it is still possible to guess the label of the root and beat random guessing beneath the Kesten-Stigum bound when $q \geq 5$. However, this requires to use *higher-order* information about which labels appear where in the tree (Mossel, 2001; Sly, 2009a,b).

Alternatively, the Kesten-Stigum bound can be thought of through the lens of robustness. Suppose we inject random noise at the leaves. In particular, suppose we overwrite the label of each leaf to a random value with probability η. Then above the Kesten-Stigum bound, reconstructing the root in the face of noise is still possible, but beneath the Kesten-Stigum bound it is information-theoretically impossible (Janson and Mossel, 2004). Thus the Kesten-Stigum bound is the location in parameter space where the typical posterior distribution on the label of the root becomes highly sensitive to noise.

Fundamentally, each of these methodologies represents a way to extract information from belief propagation about where the posterior distribution on the label of the root becomes highly complex. The notion of complexity is expressed in many different ways – for example, the failure of linear statistics, lack of robustness, or (in the physics language) stability of the trivial fixed point. In this paper, we take an approach that is grounded in computational complexity for studying the posterior distribution in the broadcast tree model. (Alternatively, we take a circuit complexity approach to studying the complexity of the problem that belief propagation is actually solving).

> *We establish some tantalizing parallels between phase transitions (in the traditional meaning of the phrase, where it refers to changes in the structure of the solution space) and phase transitions in the circuit complexity of the inference problem.*

1.2. Our Results

In this paper, we study the circuit complexity of various tasks performed by belief propagation on the broadcast tree model. We will be interested in four main problems: (1) detection, where the goal is to guess the label of the root, given leaves generated at random, with probability $1/q + \epsilon$ with $\epsilon > 0$ independent of the depth (2) inference, where the goal is to compete with the Bayes optimal predictor asymptotically in an average-case sense over samples from the model (3) computing the posterior, which is the analogous question for worst-case inputs on the labels of the leaves. And finally we study (4) the complexity of the forward problem of generating samples from the model. These tasks can all naturally be solved in $\mathbf{NC}^1$ the class of logarithmic depth circuits with AND, OR and NOT gates with constant fanin. However it will turn out that in some cases (conjecturally) weaker classes with constant depth will suffice and in others logarithmic depth is inherently necessary.

The next major circuit complexity class below $\mathbf{NC}^1$ is $\mathbf{TC}^0$, the class of constant depth circuits made from NOT and majority gates. One can build a circuit in $\mathbf{NC}^1$ that determines whether or not the majority of its inputs are 1 by counting the number of 1s in its inputs, so $\mathbf{TC}^0 \subseteq \mathbf{NC}^1$, while these classes are believed but not proven to be distinct. It is well known that for the broadcast tree model on two labels – also called the Ising model on trees – beneath the Kesten-Stigum bound detection is information-theoretically impossible. What this means is that taking the majority vote of the labels of the leaves solves the detection problem whenever it is information-theoretically possible to do so. However it is also well-known that majority vote is suboptimal in how often it guesses the label of the root correctly. In other words, the majority vote attains an asymptotic accuracy that is greater than 1/2 but less than the asymptotic accuracy attained by belief propagation. Intuitively, this is because there is more information about the label of the root contained not just in the number of labels of each type but also in the structure of where in the tree they are relative to each other. We prove that there are more complex circuits, but still ones in $\mathbf{TC}^0$, that can solve the inference problem:

Theorem 1 (informal, see Theorem 15) *There is a constant $C > 1$ so that if $k\theta^2 > C$ then the inference problem in the Ising model ($q = 2$) on trees can be solved in $\mathbf{TC}^0$.*

Our approach is based on Mossel et al. (2014) that shows belief propagation (suitably above the Kesten-Stigum bound) is robust to label noise. This allows to construct a $\mathbf{TC}^0$ circuit by using majority on the leaves of subtrees to get noisy estimates of their roots. We then bootstrap these estimates to get an asymptotically optimal estimate of the label of the overall root. It is conjectured that belief propagation works with noisy labels all the way down to the Kesten-Stigum bound (i.e. $k\theta^2 > 1$) in which case we could improve the above theorem analogously.

As we discussed earlier, belief propagation works even in a worst-case sense and computes the true posterior. We show that the worst-case problem is much harder and is $\mathbf{NC}^1$-complete:

Theorem 2 (informal, see Theorem 14) *There are constants θ and k for which computing the posterior in the Ising model on trees is $\mathbf{NC}^1$-complete.*

However there is something unsatisfying about a circuit complexity lower bound that applies to the problem of computing the posterior distribution on the label of the root for a worst-case configuration of labels on the leaves. The broadcast tree model is a generative model, and the properties of belief propagation that are used to locate phase transitions are really average-case properties – or rather, properties about the posterior distribution on the label of the root, for a typical realization of the labels of the leaves. Now we come to what we believe to be our most significant result. We study the average-case circuit complexity of guessing the label of the root in a broadcast tree model whose parameters are beneath the Kesten-Stigum bound. We prove:

Theorem 3 (informal, see Theorem 47) *There is a* 16 *label broadcast tree model where it is possible to guess the label of the root with probability* ≥ 0.999 *but where detection is* $\mathbf{NC}^1$*-complete.*

For a general Markov process on a k-regular tree with a transmission matrix M, i.e. one in which a child of a vertex with label i has label j with probability $M_{j,i}$, the Kesten-Stigum bound is $k(\lambda_2(M))^2 > 1$ where $\lambda_2(M)$ is the second largest eigenvalue of M. In our construction, the transmission matrix has a second eigenvalue equal to zero and thus no matter how large k is, we are operating below the Kesten-Stigum bound. (Equivalently, no matter how large k is, linear statistics are not enough to guess the label of the root with positive advantage over random guessing). More broadly, we conjecture that the detection problem is $\mathbf{NC}^1$-complete *anywhere* beneath the Kesten-Stigum bound, which is consistent with the fractal way that information is stored in such settings (Mossel, 2001), but we are only able to prove it for this particular 16 label broadcast tree model.

Barrington famously showed that the word problem over nonsolvable groups is $\mathbf{NC}^1$-complete (Barrington, 1989). This leads to a natural average-case $\mathbf{NC}^1$-complete problem via telescopically multiplying by random group elements. We construct a model where the labels of the children can be multiplied to get the labels of the parents. While we can solve detection by multiplying group elements in some way, what is less obvious is how to show that any circuit for detection can be used to solve the word problem. The key idea is we can define an alternative but equivalent generation procedure that starts by labelling the root implicitly as the product of many group elements, and as we follow the process down the levels of the tree, the product simplifies and involves fewer elements until at the leaves it is a random function of a single group element. In this way, the generative process expresses the label of the root as a random function of the labels of the leaves, as opposed to the other way around. This is our most challenging result and perhaps the most surprising.

Finally, we study the circuit complexity of some of the remaining tasks associated with the broadcast tree model to complete the picture. First, it is natural to wonder if weaker circuit models can ever solve the detection problem. We show an unconditional lower bound against $\mathbf{AC}^0$, the class of constant depth circuits made of AND, OR, and NOT gates:

Theorem 4 (informal, see Theorem 13) *For any* $-1 < \theta < 1$*, there is no* $\mathbf{AC}^0$ *circuit for solving the detection problem in the Ising model on trees.*

The proof is based on the observation that the generative process for the broadcast tree model can itself be thought of as a series of random projections — a variant of a classic tool for proving circuit lower bounds (Furst et al., 1984b). The main difference is that we do not get to choose the parameters of the projection ourselves, it is dictated by the model and only sets a constant fraction of the inputs as we go up one level of the tree.

Despite the fact that $\mathbf{AC}^0$ circuits do not solve even the most basic type of inference problem in any interesting range of parameters, it turns out that, somewhat surprisingly, they can solve the forward problem of generation.

Theorem 5 (informal, see Theorem 21) *For any $\theta = a/2^b$ where a and b are integers, given uniformly random bits as input, there is an $\mathbf{AC}^0$ circuit for sampling from the Ising model on trees.*

Thus the broadcast tree model on two labels is an interesting example where there is a wide discrepancy between the depth needed for generation vs. inference. This is reminiscent of the work of Babai (Babai, 1987) and Boppana and Lagarias (Boppana and Lagarias, 1987) who show that, while $\mathbf{AC}^0$ cannot compute parity on the uniform distribution, there is a depth one circuit whose outputs depend on two bits each that samples from a distribution whose first n bits are uniform and whose last bit is their parity. It also has a resemblance to Kearns and Valiant (1994), which shows that certain efficiently computable functions cannot be learned efficiently if standard cryptographic assumptions are true.

1.3. More Related Work

We note that while our depth lower bounds results apply to a natural inference problem, the results proving logarithmic lower bounds are conditional (on the fact that $\mathbf{NC}^1 \neq \mathbf{TC}^0$). This should be compared to the unconditional lower bounds for deep nets (Telgarsky, 2016) and to worst case (Håstad, 1987) and average case (Håstad et al., 2017) lower bounds in circuit complexity. In fact, part of the motivation for our work comes from Mossel (2019) which suggested that the broadcast model is a particularly natural data generative model that has provable reconstruction algorithms and for which one can prove rigorously that depth is needed for inference. The reconstruction algorithms of the broadcast process are often referred to as phylogenetic reconstruction algorithms. Polynomial time algorithms for reconstructing phylogenies were established in Erdös et al. (1999a,b) and phase transitions related to the Kesten-Stigum bound in the model were established in Mossel (2003, 2004b) and follow up work. The paper Mossel (2019) does not prove depth lower bounds in the sense of the current paper. Rather, it shows that for an interval of values of θ, in a semi-supervised broadcast setting, algorithms that can only access low moments of the labelled data are unable to classify better than random, while there exist algorithms that use high moments and are able to label accurately. In a different recent work Jain et al. (2019), it was shown that message passing algorithms that use only bounded memory of bits per node, do not achieve the Kesten-Stigum Bound even for the Ising Model ($q = 2$). This proves a conjecture from Evans et al. (2000). However, these results do not have an implications for the circuit complexity of the problem.

There is also a close connection between the types of problems we study here and the coin problem in pseudorandomness (Brody and Verbin, 2010), which asks: Suppose we are given a coin which is promised to have bias either $1/2+\delta$ or $1/2-\delta$ along with n independent tosses and our goal is to guess which way the coin is biased and to guess correctly with (say) probability at least $2/3$. What is the smallest δ for which a given computational model (e.g. $\mathbf{AC}^0$ (Shaltiel and Viola, 2010; Aaronson, 2009; Limaye et al., 2019), width w ROBPs (Brody and Verbin, 2010)) can succeed? In fact we can think of this as a broadcast problem on a n-ary depth one tree with two labels where the label of the root represents whether the coin has positive or negative bias.

With an unrestricted computational model, the majority function is optimal. And thus the coin problem is interesting in models that cannot compute the majority function and in turn leads to bounds on the fourier coefficients of the functions that they can compute and is a key ingredient in various PRGs. In the broadcast tree model, the labels of the leaves are no longer independent conditioned on the root but rather have a hierarchical structure to the strength of their dependencies. As it turns out, in light of our results, this problem can be much harder. We show that it is $\mathbf{NC}^1$-complete for a particular broadcast problem on 16 labels. Optimistically, and in analogy with the coin problem, we could ask: Could proving unconditional lower bounds against $\mathbf{TC}^0$ for the broadcast tree problem lead to non-trivial PRGs?

2. Preliminaries

2.1. The broadcast tree model

In this paper we consider the classical tree broadcast model on regular trees and binary labels. Throughout we will use the following notation. We write $T_k(d)$ for the d-level k-ary tree. We will identify such a tree $T_k(d)$ with a subset of $\mathbb{N}^*$, the set of finite strings of natural numbers, with the property that if $v \in T$ then any prefix of v is also in T. In this way, the root of the tree is naturally identified with the empty string, which we will denote by ρ. We will write uv for the concatenation of the strings u and v, and $L_r(u)$ for the rth-level descendants of u; that is, $L_r(u) = \{uv \in T : |v| = r\}$. Also, we will write $\mathbb{C}(u) \subset \mathbb{N}$ for the indices of u's children relative to itself. That is, $i \in \mathbb{C}(u)$ if and only if $ui \in L_1(u)$. We write L_r for $L_r(\rho)$ and $\mathrm{par}(v)$ for the parent of node v.

Definition 6 (Broadcast process on a tree) *Given a parameter $\theta \in [-1, 1]$ and a k-ary tree of d level $T_k(d)$, the* broadcast process on T *is a two-state Markov process $\{\sigma_u : u \in T\}$ defined as follows: let σ_ρ be 1 or 0 with probability $\frac{1}{2}$. Then, for each u such that σ_u is defined, independently for every $v \in L_1(u)$ let $\sigma_v = \sigma_u$ with probability $\theta + (1-\theta)/2$ and $\sigma_v = 1 - \sigma_u$ otherwise.*

In other words, in the broadcast model, the root is randomly assigned a label in $\{0, 1\}$, and then each other vertex is assigned its parent's label with probability θ and an independent uniformly chosen label with probability $1 - \theta$. Of course, this is equivalent to keeping the bit with probability $1/2 + \theta/2$ and flipping it to the opposite value with probability $1/2 - \theta/2$.

This broadcast process has been extensively studied in probability, where the major question is whether the labels of vertices far from the root of the tree give any information on the label of the root (Kesten and Stigum, 1966; Bleher et al., 1995). See also Evans et al. (2000); Mossel (2004a); Mézard and Montanari (2006). A similar question was studied in various communities including bio-informatics (Felsenstein, 2004) and AI (Pearl, 1988) from an algorithmic perspective, where the goal is to estimate (the posterior) of the root given the labels of vertices far from the root. It is well known that Belief Propagation is an exact linear time algorithm for computing the posterior.

We will mainly be focusing on the asymptotic behavior of the broadcast model as d increases with all other parameters held constant, and we will commonly set $n = k^d$. We will be discussing the circuit complexity of multiple tasks associated with the broadcast model on the tree. To simplify notation we write $X^{(r)}$ for the vector of labels at level r: $X^{(r)} := (\sigma_v : |v| = r)$.

The most important task associated with the model is inference of the root given $X^{(d)}$. As mentioned earlier, Belief propagation is used for this task. The output of Belief propagation is a posterior distribution $\mathbb{P}[X^{(0)} = \cdot | X^{(d)} = x]$. For a fixed d and k the posterior is always bounded away from 0 and 1. Indeed if k is even, the posterior can often assign equal probability to the two root values. Rounding the posterior allows to determine the more likely root value. The probabilistic nature of the inference problem, leads to a number of complexity formulations. First, in the worst-case formulation, we are looking for circuits that estimate the root correctly whenever the posterior is far enough from $(1/2, 1/2)$. In terms of average case, there is a natural distribution over the inputs, i.e, the distribution given by the broadcast process. It is thus natural to formulate an average case version of the problem where the inputs are drawn from the broadcast distribution and the objective is to estimate the root correctly with almost the same probability that BP does. Finally, in the average case setup we may settle for less, i.e., inferring the root correctly with probability bounded away from 1/2. The formal definitions of the 3 problems follow.

Definition 7 *We say that a series of functions $f_d : \{0,1\}^{L_d} \to \{0,1\}$ are* posterior *functions if*

$$\mathbb{P}[X^{(0)} = f_d(x) | X^{(d)} = x] \geq \mathbb{P}[X^{(0)} = \mathrm{BP}(x) | X^{(d)} = x] - \delta_d$$

for every d and every $x \in \{0,1\}^{L_d}$, where $\mathrm{BP}(x) := \mathrm{argmax}_{a \in \{0,1\}} P[X^{(0)} = a | X^d = x]$ is the optimal Bayes posterior, i.e., the one obtained by applying Belief Propagation and rounding, and $\delta_d \to 0$ as $d \to \infty$.

Definition 8 *We say that a series of functions $f_d : \{0,1\}^{L_d} \to \{0,1\}$ are* inference *functions if*

$$\mathbb{P}[f_d(X^{(d)}) = X^{(0)}] \geq \mathbb{P}[\mathrm{BP}(X^{(d)}) = X^{(0)}] - \delta_d,$$

where $\delta_d \to 0$ as $d \to \infty$

Thus a function is an inference function if it finds the most likely root with (almost) the same overall probability as Belief Propagation does.

Definition 9 *We say that a series of functions $f_d : \{0,1\}^{L_d} \to \{0,1\}$ are* detection *functions if there exists $\delta > 0$ and d_0 such that for all $d \geq d_0$,*

$$\mathbb{P}[f_d(X^{(d)}) = X^{(0)}] \geq 1/2 + \delta$$

In other words, a series of detection functions determines the root's label with accuracy $1/2 + \Omega(1)$, a series of inference functions determines the root's label with an accuracy within $o(1)$ of the best possible, and a series of posterior functions determines the root's label with an accuracy within $o(1)$ of the best possible conditioned on any possible value of $X^{(d)}$. Clearly posterior functions are also inference functions. When the reconstruction problem is unsolvable, there are no detection functions. If it is solvable, then inference functions are also detection functions.

In addition to inference problem, we are also interested in the generation problem, in other words, what is the computation complexity of generating $X^{(d)}$ given access to random bits. We address the generation question in appendix A.

2.2. Circuit Classes

Here we give the formal definitions for the circuit classes that we will be interested in:

Definition 10 *The circuit class* $\mathbf{AC}^0$ *is the class of constant depth circuits with a polynomial number of AND, OR, and NOT gates, where the AND and OR gates can have arbitrary fan-in.*

Definition 11 *The circuit class* $\mathbf{NC}^1$ *is the class of circuits with depth logarithmic in the number of inputs and a polynomial number of AND, OR, and NOT gates, where the AND and OR gates have fan-in two.*

We remark that in the broadcast tree model, the depth of the tree is logarithmic in the number of leaves, and each vertex has a constant number of children. This suggests that the posterior distribution on the root should be computable in $\mathbf{NC}^1$, and we prove that this is the case in Theorem 14. It is well-known that there are explicit functions (such as the parity function) for which we can prove lower bounds against $\mathbf{AC}^0$ (Furst et al., 1984b).

Definition 12 *A linear threshold function* $f : \{0,1\}^m \to \{0,1\}$ *takes the form* $f(x) = sgn(w^T x - \theta)$ *where* $w \in \mathbb{R}^m$ *and* $\theta \in \mathbb{R}$. *The circuit class* $\mathbf{TC}^0$ *is the class of constant depth circuits with a polynomial number of linear threshold function gates with unbounded fan-in.*

The class $\mathbf{TC}^0$ is contained in $\mathbf{NC}^1$ and can compute any symmetric function of its inputs. In many ways, $\mathbf{TC}^0$ represents the frontier in circuit complexity. Impagliazzo, Paturi and Saks (Impagliazzo et al., 1997) showed that depth d $\mathbf{TC}^0$ circuits with m inputs need at least $m^{1+c^{-d}}$ wires to compute the parity function for some constant $c > 0$. Chen and Tell (Chen and Tell, 2019) showed that bootstrapping $\mathbf{TC}^0$ lower bounds just beyond this would yield super-polynomial lower bounds. Miles and Viola (Miles and Viola, 2015) gave a candidate pseudorandom function computable in $\mathbf{TC}^0$ which helps explain the difficulty in proving lower bounds against $\mathbf{TC}^0$.

3. Lower bounds against $\mathbf{AC}^0$ for detection

We show that there is no $\mathbf{AC}^0$ circuit that solves the detection problem for any non-trivial choice of parameters. In order to prove this, we are going to define a series of random projections that preserve the probability distribution of $X^{(d)}$ but reduce any circuit in $\mathbf{AC}^0$ to a constant with high probability. For the most part, the proof that these projections reduce the circuit to a constant will be a fairly standard argument using the switching lemma (Furst et al., 1984a; Yao, 1985; Hastad, 1986). However, due to the nature of the $X^{(d')}$, each projection will only fix a constant fraction of the variables, which will force us to apply $\Theta(\log n)$ successive projections every time we wish to reduce the circuit depth by one. Also, we use random projections as defined in Rossman et al. (2015) instead of the more generic random restrictions to reflect the fact that multiple vertices at each layer are affected by each vertex in the layer above. The key observation is that we can preserve the probability distribution of $X^{(d)}$ by setting each vertex's label to its parent's label with probability θ and a random value otherwise. We prove:

Theorem 13 *Let $-1 < \theta < 1$ and $f : \{0,1\}^{L_d} \to \{0,1\}$ be computed by an* $\mathbf{AC}^0$ *circuit. Then there exists $\delta > 0$ such that $\mathbb{P}[f(X^{(d)}) = X^{(0)}] = 1/2 + O(n^{-\delta})$*

We defer the proof to Appendix B. As usual, the key idea is to prove that f can be approximated by a small DNF, although here the input to f comes from the broadcast tree model.

4. $\mathbf{NC}^1$-completeness of posterior functions

In appendices D and E we will prove that

Theorem 14 *For all θ and k and in the Ising tree model, there are posterior functions in* $\mathbf{NC}^1$*. Moreover there are θ and k for which all posterior functions in the Ising model are* $\mathbf{NC}^1$*-hard to compute.*

Interestingly, the proof that the posterior is in $\mathbf{NC}^1$ uses a random construction. The proof of the completeness is a reduction from $\mathbf{NC}^1$ circuits to broadcast processes where each vertex in the broadcast process correspond to a variable or a gate in the circuit.

5. A $\mathbf{TC}^0$ circuit for inference

The previous result implies that if $\mathbf{TC}^0 \neq \mathbf{NC}^1$ then no $\mathbf{TC}^0$ circuit can compute a posterior function in the Ising tree model. However we can still hope that $\mathbf{TC}^0$ circuits attempting to determine $X^{(0)}$ can perform well in the average case and can compute an inference function.

Theorem 15 *There exists $C' > 0$ such that if $k\theta^2 > C'$ then there exists a function f for which* LinearizedBP *run on f is an inference function for the Ising model on trees.*

We prove this in Appendix C

6. $\mathbf{NC}^1$ hardness of detection with many labels

So far, we have been assuming that there are only two labels that could be assigned to a vertex. However, we could instead have q labels for arbitrary q. That leads to the following definition

Definition 16 (Generalized broadcast process on a tree) *Given parameters $q > 0$ and a $q \times q$ matrix M with nonnegative entries and columns that add up to 1, the* generalized broadcast process on T *is a q-state Markov process $\{\sigma^\star_u : u \in T\}$ defined as follows: let $\sigma^\star_\rho$ be drawn uniformly at random from $\{1, \cdots, q\}$. Then, for each u such that $\sigma^\star_u$ is defined, independently for every $v \in L_1(u)$ let $\sigma^\star_v = i$ with probability $M_{i,\sigma^\star_u}$ for each i.*

In other words, in the generalized broadcast model, the root is randomly assigned a label in $\{1, \cdots, q\}$, and then each other vertex is assigned a label with a probability distribution corresponding to the column of M indexed by its parent's label. Note that the previous case is simply the instance of this where $q = 2$ and $M = \theta I + \frac{1-\theta}{2} J$, where J is the matrix with all entries equal to 1. There is an important difference between the case when there are just two labels and when there are more. It turns out there are many natural cases where it is possible to detect the label of the root, but not by taking the majority vote of the labels of the leaves. The function computed by Belief Propagation is generally more complicated and the main result of this section is to show that this manifests as a phase transition in the circuit complexity of solving detection. When there are many labels, we will show that the problem becomes $\mathbf{NC}^1$ hard.

First, we need a problem that is $\mathbf{NC}^1$-hard in the average case. In a celebrated result, Barrington showed that deciding whether the word problem (i.e. if a given word is the identity or not) over a finite nonsolvable group is $\mathbf{NC}^1$-complete (Barrington, 1989). In order to avoid causing ambiguity by multiplying group elements in an unclear order, we give the following clarification of what our product notation means.

Definition 17 *Given elements of a group $a_1, a_2, \ldots$ and positive integers $m \leq m'$, we define the product $\prod_{i=m}^{m'} a_i$ as having earlier elements to the left of later ones. For instance, $\prod_{i=1}^{3} a_i = a_1 \cdot a_2 \cdot a_3$.*

With this definition, we can state the hardness result of the word problem on the alternating group A_5 as follows.

Proposition 18 *(Barrington, 1989) For every $c \in A_5$ such that $c \neq 1$, determining whether a product of elements of A_5, $\prod_{i=1}^{m} \sigma_i$ is c or the identity given that it is one of them is $\mathbf{NC}^1$-complete.*

Conveniently, this problem has a simple worst-case to average-case reduction:

Theorem 19 *Let $f_r : A_5^r \to A_5$ be a family of functions. Suppose there exists $\epsilon > 0$ independent of r such that when $\Sigma_1, \cdots, \Sigma_r$ are independently drawn from A_5 according to the uniform distribution,*

$$\mathbb{P}[f_r(\Sigma_1, \cdots, \Sigma_r) = \prod_{i=1}^{r} \Sigma_i] \geq 1/60 + \epsilon$$

If $\mathbf{TC}^0 \neq \mathbf{NC}^1$ then there is no $\mathbf{TC}^0$ circuit that computes f.

Proof For the sake of contradiction, we will assume that there is a $\mathbf{TC}^0$ circuit that computes f. Let $h_n : \{0,1\}^n \to \{0,1\}$ be an $\mathbf{NC}^1$-complete family of functions. Consider the following randomized algorithm attempting to compute $h_n(x)$. First, generate a random $c \in A_5 \backslash \{1\}$. Next, the completeness of h_n implies there there exists r polynomial in n and $\sigma \in A_5^r$ such that $\prod_{i=1}^{r} \sigma_i = c$ if $h_n(x) = 1$ and $\prod_{i=1}^{r} \sigma_i = 1$ if $h_n(x) = 0$ (note that σ depends on c and x and the computation of σ is in $\mathbf{NC}^0$). Now randomly select $b_i \in A_5$ for each $1 \leq i \leq r$. Next compute

$$f(\sigma_1 b_1, b_1^{-1} \sigma_2 b_2, b_2^{-1} \sigma_3 b_3, \cdots, b_{r-1}^{-1} \sigma_r b_r).$$

If it is equal to b_r, conclude that $h_n(x) = 0$, if it is cb_r then conclude that $h_n(x) = 1$, and output nothing otherwise. No matter what the value of σ is, the probability distribution of $(\sigma_1 b_1, b_1^{-1} \sigma_2 b_2, \cdots, b_{r-1}^{-1} \sigma_r b_r)$ is the uniform distribution on A_5^r. Hence we have that

$$\mathbb{P}[f(\sigma_1 b_1, \cdots, b_{r-1}^{-1} \sigma_r b_r) = \sigma_1 \sigma_2, \cdots, \sigma_r b_r] \geq 1/60 + \epsilon$$

Thus, this algorithm computes $h_n(x)$ correctly with a probability of at least $1/60 + \epsilon$. Furthermore, c is independent of $(\sigma_1 b_1, \cdots, b_{r-1}^{-1}\sigma_r b_r)$, and thus of what f will return if it computes the product incorrectly. So, this algorithm computes $h_n(x)$ incorrectly with a probability of at most $1/60$. Thus if we repeat this process a large polynomial number of times and take the majority vote, we can compute $h_n(x)$ correctly with probability at least $1 - o(2^{-n})$. Thus there must be some choices of our random variables for which this computes $h_n(x)$ correctly for every x. This whole procedure can be carried out by a $\mathbf{TC}^0$ circuit, so $\mathbf{TC}^0$=$\mathbf{NC}^1$. ■

Now that we have a problem that is $\mathbf{NC}^1$-hard in the average case, we need a way to reduce this to the problem of determining the label of the root for some choice of parameters. In order to do that, we consider the following instance of the generalized broadcast process on a tree. There is one label for every ordered pair $(\sigma, \sigma') \in A_5^2$, and $k = 60000$. Given a vertex with a parent with label (σ, σ'), we select a random $b \in A_5$. Then, we set its label to $(b, b^{-1}\sigma)$ with probability $2/3$ and $(b, b^{-1}\sigma')$ with probability $1/3$. In other words, each child of a vertex is assigned a random ordered pair that multiplies to σ with probability $2/3$ and a random ordered pair that multiplies to σ' with probability $1/3$. For the rest of this section, we will assume that $\sigma^\star$ was generated by the generalized broadcast process with these parameters.

Note that it is straightforward to implement this process with an $\mathbf{NC}^1$ circuit because the tree has logarithmic depth. Moreover, we argue that detection is information-theoretically possible. The key idea is for any d', if we can determine the labels of the vertices at depth d' so that each label is correct (independently) with probability 0.99 then for any vertex at depth $d' - 1$ we can determine its label with probability at least 0.99. We do this by taking the two most common products of the elements among its children's suspected labels and by a Chernoff bound it is easy to see that this procedure succeeds with probabiity at least 0.99. Furthermore because the subtrees of each vertex at depth $d' - 1$ are disjoint, the probability our guess is correct is independent. Now we can continue this process until we reach the root. This type of recursive reconstruction arguments are by now standard, see e.g. Mossel and Peres (2003)

Next we will give an alternative procedure, called ProductTreeConstructionAlgorithm, for sampling from the generalized broadcast tree model. The intuition behind it is as follows: Instead of writing the label of the root explicitly as a pair of elements in A_5 we will express it as a pair of sequences of products of elements in A_5. As we traverse the tree from top to bottom the label of each intermediate node will be a pair of shorter and shorter sequences of products. In fact, they will essentially be subsequences of the sequences at the root. Now we need the precise way we do this to be faithful to the original sampling procedure in the sense that we assign a label whose product is equal to the first permutation in its parent's label with probability $2/3$, and otherwise its product is equal to the second one. Moreover the pair of permutations should be chosen uniformly at random subject to this constraint. We accomplish this through conjugating by random elements in a careful way. Ultimately, at the leaves, we are left with sequences that are only three elements long, which we can multiply out, and we have embedded the word problem for A_5 equivalently as the problem of guessing the label of the root. The details of this algorithm and the theorem's proof are in appendix G.

References

Scott Aaronson. Bqp and the polynomial hierarchy. *arXiv preprint arXiv:0910.4698*, 2009.

Emmanuel Abbe and Colin Sandon. Detection in the stochastic block model with multiple clusters: proof of the achievability conjectures, acyclic bp, and the information-computation gap. *arXiv preprint arXiv:1512.09080*, 2015.

Lászió Babai. Random oracles separate pspace from the polynomial-time hierarchy. *Information Processing Letters*, 26(1):51–53, 1987.

Jess Banks, Cristopher Moore, Joe Neeman, and Praneeth Netrapalli. Information-theoretic thresholds for community detection in sparse networks. In *Conference on Learning Theory*, pages 383–416, 2016.

David A Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in nc1. *Journal of Computer and System Sciences*, 38(1):150–164, 1989.

Quentin Berthet and Philippe Rigollet. Complexity theoretic lower bounds for sparse principal component detection. In *Conference on Learning Theory*, pages 1046–1066, 2013.

P. M. Bleher, J. Ruiz, and V. A. Zagrebnov. On the purity of the limiting Gibbs state for the Ising model on the Bethe lattice. *J. Statist. Phys.*, 79(1-2):473–482, 1995.

Ravi B Boppana and JC Lagarias. One-way functions and circuit complexity. *Information and Computation*, 74(3):226–240, 1987.

Charles Bordenave, Marc Lelarge, and Laurent Massoulié. Non-backtracking spectrum of random graphs: community detection and non-regular ramanujan graphs. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 1347–1357. IEEE, 2015.

Matthew Brennan, Guy Bresler, and Wasim Huleihel. Reducibility and computational lower bounds for problems with planted sparse structure. In *Conference On Learning Theory*, pages 48–166, 2018.

Joshua Brody and Elad Verbin. The coin problem and pseudorandomness for branching programs. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 30–39. IEEE, 2010.

Lijie Chen and Roei Tell. Bootstrapping results for threshold circuits just beyond known lower bounds. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 34–41. ACM, 2019.

A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physics Review E*, 84:066106, Dec 2011. doi: 10.1103/PhysRevE.84.066106. URL http://link.aps.org/doi/10.1103/PhysRevE.84.066106.

P. L. Erdös, M. A. Steel;, L. A. Székely, and T. A. Warnow. A few logs suffice to build (almost) all trees (part 1). *Random Structures Algorithms*, 14(2):153–184, 1999a.

P. L. Erdös, M. A. Steel;, L. A. Székely, and T. A. Warnow. A few logs suffice to build (almost) all trees (part 2). *Theoretical Computer Science*, 221:77–118, 1999b.

W. S. Evans, C. Kenyon, Yuval Y. Peres, and L. J. Schulman. Broadcasting on trees and the Ising model. *Ann. Appl. Probab.*, 10(2):410–433, 2000.

J. Felsenstein. *Inferring Phylogenies*. Sinauer, New York, New York, 2004.

M. Furst, J. Saxe, and M. Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984a.

Merrick Furst, James B Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical systems theory*, 17(1):13–27, 1984b.

Johan Håstad. Computational limitations of small-depth circuits. 1987.

Johan Håstad, Benjamin Rossman, Rocco A Servedio, and Li-Yang Tan. An average-case depth hierarchy theorem for boolean circuits. *Journal of the ACM (JACM)*, 64(5):35, 2017.

John Hastad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 6–20. Citeseer, 1986.

Russell Impagliazzo, Ramamohan Paturi, and Michael E Saks. Size–depth tradeoffs for threshold circuits. *SIAM Journal on Computing*, 26(3):693–707, 1997.

Vishesh Jain, Frederic Koehler, Jingbo Liu, and Elchanan Mossel. Accuracy-memory tradeoffs and phase transitions in belief propagation. In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 1756–1771, Phoenix, USA, 25–28 Jun 2019. PMLR. URL http://proceedings.mlr.press/v99/jain19b.html.

S. Janson and E. Mossel. Robust reconstruction on trees is determined by the second eigenvalue. *Ann. Probab.*, 32:2630–2649, 2004. URL http://front.math.ucdavis.edu/0406.5447.

Michael Kearns and Leslie Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *JOURNAL OF THE ACM*, 41:433–444, 1994.

H. Kesten and B. P. Stigum. Additional limit theorems for indecomposable multidimensional Galton-Watson processes. *Ann. Math. Statist.*, 37:1463–1481, 1966.

H. Kesten and B. P. Stigum. Limit theorems for decomposable multi-dimensional Galton-Watson processes. *J. Math. Anal. Appl.*, 17:309–338, 1967.

F. Krząkała, A. Montanari, F. Ricci-Tersenghi, G. Semerjian, and L. Zdeborová. Gibbs states and the set of solutions of random constraint satisfaction problems. *Proceedings of the National Academy of Sciences*, 104(25):10318–10323, 2007.

Nutan Limaye, Karteek Sreenivasaiah, Srikanth Srinivasan, Utkarsh Tripathi, and Venkitesh S. A fixed-depth size-hierarchy theorem for ac 0 [] via the coin problem. pages 442–453, 06 2019. doi: 10.1145/3313276.3316339.

Zongming Ma and Yihong Wu. Computational barriers in minimax submatrix detection. *The Annals of Statistics*, 43(3):1089–1116, 2015.

Laurent Massoulié. Community detection thresholds and the weak ramanujan property. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 694–703. ACM, 2014.

M. Mézard and A. Montanari. Reconstruction on trees and the spin glass transition. *Journal of Statistical Physics*, 124:1317–1350, 2006.

Eric Miles and Emanuele Viola. Substitution-permutation networks, pseudorandom functions, and natural proofs. *Journal of the ACM (JACM)*, 62(6):46, 2015.

E. Mossel. Reconstruction on trees: beating the second eigenvalue. *Ann. Appl. Probab.*, 11 (1):285–300, 2001. ISSN 1050-5164. URL http://www.stat.berkeley.edu/~mossel/publications/second.pdf.

E. Mossel. On the impossibility of reconstructing ancestral data and phylogenies. *Jour. Comput. Bio.*, 10(5):669–678, 2003. URL http://www.stat.berkeley.edu/~mossel/publications/jcb_impossibility.pdf.

E. Mossel. Survey: Information flow on trees. In J. Nestril and P. Winkler, editors, *Graphs, Morphisms and Statistical Physics. DIMACS series in discrete mathematics and theoretical computer science*, pages 155–170. 2004a. URL http://front.math.ucdavis.edu/0406.5446.

E. Mossel. Phase transitions in phylogeny. *Trans. Amer. Math. Soc.*, 356(6):2379–2404 (electronic), 2004b. ISSN 0002-9947. URL http://front.math.ucdavis.edu/0304.5491.

E. Mossel and Y. Peres. Information flow on trees. *Ann. Appl. Probab.*, 13(3):817–844, 2003. ISSN 1050-5164. URL http://front.math.ucdavis.edu/math.PR/0107033.

E. Mossel, J. Neeman, and A. Sly. Belief propagation, robust reconstruction, and optimal recovery of block models (extended abstract). *JMLR Workshop and Conference Proceedings (COLT proceedings)*, 35:1–35, 2014. Winner of best paper award at COLT 2014.

E. Mossel, J. Neeman, and A. Sly. Reconstruction and estimation in the planted partition model. *Probability Theory and Related Fields*, (3-4):431–461, 2015. The Arxiv version of this paper is titled Stochastic Block Models and Reconstruction.

E. Mossel, J. Neeman, and A. Sly. Belief propagation, robust reconstruction, and optimal recovery of block models. *Annals of Applied Probability*, 26(4):2211–2256, 2016.

Elchanan Mossel. Deep learning and hierarchal generative models. arXiv preprint arXiv:1612.09057, 2019.

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

J. Pearl. *Probabilistic reasoning in intelligent systems.* Morgan Kaufman, San Mateo, 1988.

B. Rossman, R. A. Servedio, and L. Tan. An average-case depth hierarchy theorem for boolean circuits. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 1030–1048, 2015.

Ronen Shaltiel and Emanuele Viola. Hardness amplification proofs require majority. *SIAM Journal on Computing*, 39(7):3122–3154, 2010.

A. Sly. Reconstruction of random colourings. *Comm. Math. Phys.*, 288, 2009a.

A. Sly. Reconstruction of symmetric potts models. In *Proceedings of the 41st ACM Symposium on Theory of Computing*, pages 581–590, 2009b.

Matus Telgarsky. benefits of depth in neural networks. In *Conference on Learning Theory*, pages 1517–1539, 2016. arXiv preprint arXiv:1602.04485.

Andrew Chi-Chih Yao. Separating the polynomial-time hierarchy by oracles. In *26th Annual Symposium on Foundations of Computer Science (sfcs 1985)*, pages 1–10. IEEE, 1985.

Appendix A. Difficulty of generation

In this paper, we are mostly concerned with depth lower (and upper) bounds for estimating $X^{(0)}$ given $X^{(d)}$. However, we also study the generation problem, i.e., the complexity of generating $X^{(d)}$ given a sequence of random bits as an input. More formally:

Definition 20 *We say that a series of functions $f_d : \{0,1\}^{m(d)} \to \{0,1\}^{L_d}$ are* generation *functions if under the uniform distribution over the inputs, it holds that $f_d(x)$ has the distribution $X^{(d)}$ for all d. We call such functions $(\delta_d)_{d=1}^{\infty}$-approximate-generation functions if the total variation distance between the distribution of $f_d(x)$ and $X^{(d)}$ is bounded by δ_d for all d*

Despite the fact that the tree has logarithmic depth, it turns out that generation can be accomplished in $\mathbf{AC}^0$ easily.

Theorem 21 *If θ is a dyadic number: $\theta = a/2^b$ for some integers a and b, then there exists generation functions in $\mathbf{AC}^0$. Moreover, for all θ, and any constant $c > 0$, there exists 2^{-n^c}–approximate-generation functions in $\mathbf{AC}^0$.*

Proof Assume first that θ and therefore $(\theta \pm 1)/2$ are dyadic. This means that there exists a function $g : \{0,1\}^s \to \{0,1\}$ of a bounded number of bits such that $\mathbb{P}[g = 1] = (\theta + 1)/2$. We apply a copy of g independently for each vertex of the tree thus obtaining a collection of independent random variables (Y_v). So, if we set $Y'^{(0)}$ to be a uniformly random bit and then define $X'_v = \prod_{w \in \text{path}(\rho,v)} Y_w$, then the probability distribution of X' is identical to the probability distribution of X. Furthermore, for each v, there are at most d elements of Y

that effect the value of X'_v. That means that there are only $2^{d+1} \leq 2n$ possible values of $X^{(0)}$ and the elements of Y that effect $X'^{(d)}_i$. As such, we only need $O(n)$ gates to have an AND for every possible combination of values of $X^{(0)}$ and these Y_v, at which point we can OR together all of the ones for values that result in $X'_v = 1$. Doing this for every v merely multiplies the number of gates by n, and this clearly has constant depth. This proves the first part of the theorem.

The second part of the theorem is similar, except we now approximate coin tosses of bias $(\theta+1)/2$. It is easy to see that an approximation to error 2^{-n^c} is achievable in $\mathbf{AC}^0$ in constant depth and size polynomial in n. This is done by generating a polynomial number of unbiased bits $Z_1, \ldots, Z_{n^c+\lceil \log_2(2n) \rceil}$ and considering them as the binary expansion of a number in $[0, 1]$. We then declare the bias-coin toss to be 1 if the resulting number is bigger than $(1+\theta)/2$ and 0 otherwise. The threshold computation $\sum Z_i > (1+\theta)/2$ can be carried out by an OR of AND gates. ∎

Remark 22 *If we consider a computational model where the inputs have bias θ instead of $1/2$, then the proof above provides generation functions in $\mathbf{AC}^0$.*

Now that we have established that $\mathbf{AC}^0$ circuits are capable of drawing strings from the correct probability distribution, the logical next question is whether or not $\mathbf{NC}^0$ circuits can do the same. As it turns out, they generally cannot. The key issue is that each bit output by an $\mathbf{NC}^0$ circuit is affected by a constant number of input bits.

Theorem 23 *Let $f_n : \{0,1\}^{m_n} \to \{0,1\}^{L_d}$ be a series of functions that can be computed by an $\mathbf{NC}^0$ circuit. Also, let $W_1, \cdots, W_{m_n}$ be independently generated random variables and $X' = f_n(W)$. If $0 < \theta < 1$ then*

$$\sum_{x \in \{0,1\}^{L_d}} \min\left(\mathbb{P}[X^{(d)} = x], \mathbb{P}[X' = x]\right) = O\left(e^{-\sqrt{n}}\right)$$

It turns out to be much easier to prove the simpler result that $\mathbf{NC}^0$ fails when it is given uniformly random bits as input, just because some pairs of bits in the output of the broadcast tree model have weak but non-zero correlations.

If the random bits are each set to 1 with probability $1/2$, then this means that the probability that any pair of outputs take on any two values must be an integer multiple of 2^{-2c} where c is the largest number of input bits affecting a single output bit. However, some of the elements of $X^{(d)}$ have correlations that are less than 2^{-2c}, so this cannot get the probability distribution right. More formally, we have the following.

Lemma 24 *Let $f : \{0,1\}^m \to \{0,1\}^n$ be a function that can be computed by an $\mathbf{NC}^0$ circuit, $W_1, \cdots, W_m$ be independent random variables that are set to 1 with probability $1/2$ and 0 with probability $1/2$, and $X' = f(W)$. If $0 < \theta < 1$ then*

$$d_{TV}(X', X^{(d)}) = \Omega(1).$$

Proof There must exist a constant c such that each output of f is affected by at most c of its inputs. Now, let d' be the smallest positive integer such that $\theta^{2d'} \le 2^{-2c}$. For any $n \ge k^{d'}$,

$$\mathbb{P}[X^{(d)}_{1^d} = X^{(d)}_{1^{d-d'}k^{d'}}] = 1/2 + \theta^{2d'}/2$$

However, $\mathbb{P}[X'_{1^d} = X'_{1^{d-d'}k^{d'}}]$ must be an integer multiple of 2^{-2c}. So, it must be the case that

$$|\mathbb{P}[X^{(d)}_{1^d} = X^{(d)}_{1^{d-d'}k^{d'}}] - \mathbb{P}[X'_{1^d} = X'_{1^{d-d'}k^{d'}}]| \ge \theta^{2d'}/2$$

The desired conclusion follows. ■

This lemma is somewhat unsatisfying in that it leaves open the possibility that the generation process might be doable in $\mathbf{NC}^0$ if we are given access to independent bits with any desired given biases. We study this case next.

To prove lower bounds against $\mathbf{NC}^0$ in this more general setup, we will still use the property that each bit output by the $\mathbf{NC}^0$ circuit is affected by a constant number of input bits. Also, each input bit could effect anywhere from 1 output bit to all of them. That means that if we divide the interval $[1, n]$ into a sufficiently large collection of subintervals, there must be at least one, $[a, b]$, such that less than half of the outputs of the circuit are affected by an input that affects a number of outputs in that range. Then, we can find a set of $\Omega(n/a)$ outputs that only have dependencies as a result of inputs that affect more than b outputs. That allows us to show that for any fixed assignment of values to those inputs the overlap between the probability distributions of $X^{(d)}$ and the output of the circuit is very small. Then, we can add together these overlaps for every assignment of values to those variables and show that it is still small because there are at most $\Omega(n/b)$ inputs that affect that many outputs. Our first step towards proving this will be to show that any $\mathbf{NC}^0$ circuit with a large number of outputs has a large subset of its outputs that are independent conditioned on the values of a relatively small number of inputs. More formally, we have:

Lemma 25 *Let $f_n : \{0,1\}^{m_n} \to \{0,1\}^n$ be a series of functions that can be computed by an $\mathbf{NC}^0$ circuit, and c be the maximum number of inputs that any output is affected by. Also, let $W_1, \cdots, W_{m_n}$ be independently generated random variables and $X' = f_n(W)$. Next, let $n \ge b_0 \ge b_1 \ge b_2 \ge \cdots \ge b_{2c} \ge 1$. For any given n, there exists $0 < i \le 2c$, $S \subseteq \{1, \cdots, n\}$ and $T \subseteq \{1, \cdots, m\}$ such that $|S| \ge \frac{n}{2cb_i}$, $|T| \le cn/b_{i-1}$, and $\{X'_j : j \in S\}$ are independent conditioned on any fixed value of $\{W_j : j \in T\}$.*

Proof Choose an n, refer to m_n as m, and for each j, let s_j be the number of bits in the output of f_n that are affected by the value of W_j. Also, assume without loss of generality that $s_1 \ge s_2 \ge \cdots \ge s_m$. Next, for each $0 \le i \le 2c$, let j_i be the smallest positive integer such that $s_{j_i} \le b_i$, or $m+1$ if $s_j > b_i$ for all j. Now, observe that

$$\sum_{i=1}^{2c} \sum_{j=j_{i-1}}^{j_i-1} s_j = \sum_{j=j_0}^{j_{2c}-1} s_j \le \sum_{j=1}^{m} s_j \le cn$$

So, there must exist i such that $\sum_{j=j_{i-1}}^{j_i-1} s_j \le n/2$. That means that there are at least $n/2$ elements of X that are not affected by W_j for any $j_{i-1} \le j < j_i$. For any such element of

X, there are at most $c(b_i - 1)$ other elements of X that are affected by any of the elements of $W_{j_i}, W_{j_i+1}, \cdots, W_m$ that affect it. So, we can find at least $n/2cb_i$ elements of X such that for all $j_{i-1} \leq j \leq j_i - 1$, W_j does not affect any of them, and for all $j \geq j_i$, at most one of these elements is affected by W_j. Also, $j_{i-1} \leq cn/b_{i-1} + 1$. So, that leaves at most cn/b_{i-1} elements of W that affect more than one of these elements of X. ■

That establishes that the output of any such $\mathbf{NC}^0$ circuit contains a large number of elements that are independent conditioned on the values of a relatively small number of inputs. Ultimately, we will want to show that the probability distribution of the corresponding elements of $X^{(d)}$ must have negligible overlap with the probability distribution of these outputs. In order to do this, we will need to establish that the probability distribution of any large subset of the elements of $X^{(d)}$ has very low overlap with the probability distribution of any set of independent random variables. The main idea behind that argument is that elements of $X^{(d)}$ corresponding to nearby leaves are correlated. So, any two independent random variables corresponding to nearby leaves must either be excessively biased towards one label or have too low a probability of being equal to each other. As such, we state the following result:

Lemma 26 *For any fixed values of $0 < \theta < 1$ and $k > 1$, there exist constants $c_1, c_2 > 0$ such that the following holds. Let $S \subseteq L_d$, and let $X'_i \in \{0,1\}$ be a random variable for each $i \in L_d$ such that $\{X'_i : i \in S\}$ are independent. Then*

$$\sum_{x \in \{0,1\}^S} \min\left(\mathbb{P}\left[X_i^{(d)} = x_i \text{ for } i \in S\right], \mathbb{P}\left[X'_i = x_i \text{ for } i \in S\right]\right) \leq 2e^{-c_1|S|^{1+c_2}/n^{c_2}}$$

Proof First, let $d' = \lfloor \log_k(|S|/6) \rfloor$. Next, let $\delta = \theta^{d-d'}/4$ and $d'' = d' - \lceil -\log(4)/\log\theta \rceil$. We will break up our analysis into two cases.

First consider the case where $E[X'_i] \geq 1/2 + \delta$ for at least $1/3$ of the i in S or $E[X'_i] \leq 1/2 - \delta$ for at least $1/3$ of the i in S. Assume without loss of generality that $E[X'_i] \geq 1/2 + \delta$ for at least $1/3$ of the i in S. In this case, let $S' = \{i \in S : E[X'_i] \geq 1/2 + \delta\}$. Next, let S'' be a maximal subset of S such that $\text{par}^{(d-d'')}(i) \neq \text{par}^{(d-d'')}(i')$ for all distinct $i, i' \in S''$. Clearly, there are at most $k^{d-d''}$ elements of S' that have any given ancestor in $L_{d''}$, so $|S''| \geq |S'|/k^{d-d''} \geq |S|/3k^{d-d''}$. Also, $E[X'_i] \geq 1/2 + \delta$ for every $i \in S''$. However, for any $x \in \{0,1\}^{L_{d''}}$ and any $i \in S''$, it must be the case that

$$\begin{aligned} E[X_i^{(d)} | X^{(d'')} = x] &\leq 1/2 + \theta^{d-d''}/2 \\ &\leq 1/2 + \theta^{d-d'}/8 = 1/2 + \delta/2 \end{aligned}$$

Also, these elements of $X^{(d)}$ are independent conditioned on any value of $X^{(d'')}$ because S'' does not contain the indices of any pair of vertices with a common ancestor closer than $X^{(d'')}$. So, by a Chernoff bound,

$$P\left[\frac{1}{|S''|}\sum_{i \in S''} X_i^{(d)} \geq 1/2 + 3\delta/4\right] \leq e^{-\delta^2|S''|/96}$$

On the flip side,

$$P\left[\frac{1}{|S''|}\sum_{i\in S''}X'_i \le 1/2+3\delta/4\right] \le e^{-\delta^2|S''|/64}$$

So, the overlap between the probability distributions of X' and $X^{(d)}$ is at most $2e^{-\delta^2|S''|/96}$. Now, observe that

$$\begin{aligned}\delta^2|S''| &\ge \theta^{2(d-d')}|S|/48k^{d-d''}\\ &\ge \theta^{2(d-d')}|S|/48k^{d-d'}k^{1+\log(4)/\log\theta}\\ &\ge \theta^2[|S|/6n]^{1-2\log_k\theta}|S|/48k^{2+\log(4)/\log\theta}\\ &= \frac{\theta^2}{48\cdot 6^{1-2\log_k\theta}k^{2+\log(4)/\log\theta}}\cdot\frac{|S|^{2-2\log_k\theta}}{n^{1-2\log_k\theta}}\end{aligned}$$

So, the overlap between the probability distributions of X' and $X^{(d)}$ is at most

$$2e^{-\frac{\theta^2}{4608\cdot 6^{1-2\log_k\theta}k^{2+\log(4)/\log\theta}}\cdot\frac{|S|^{2-2\log_k\theta}}{n^{1-2\log_k\theta}}}$$

Now we consider the remaining case when $1/2-\delta \le E[X'_i] \le 1/2+\delta$ for at least 1/3 of the i in S. In this case, let $S'=\{i\in S: 1/2-\delta\le E[X'_i]\le 1/2+\delta\}$. We know that $|S'|\ge|S|/3\ge 2k^{d'}$. So, there must be at least $(|S'|-k^{d'})/k^{d-d'}\ge|S|/6k^{d-d'}$ values of $j\in L_{d'}$ such that more than one of the elements of $L_{d-d'}(j)$ are in S'. Now, pick $i,i'\in L_{d-d'}(j)\cap S'$ for each such j, and let S'' be the set of all such pairs (i,i'). For any such i,i', we know that X'_i is independent of $X'_{i'}$, so $\mathbb{P}[X'_i=X'_{i'}]\le 1/2+2\delta^2$. Also, $\mathbb{P}[X_i^{(d)}=X_{i'}^{(d)}]\ge 1/2+\theta^{2d-2d'}/2=1/2+8\delta^2$, and this probability is independent of the labels of any leaves not descended from $\text{par}^{(d-d')}(i)$. So, by a Chernoff bound,

$$\mathbb{P}[|\{(i,i')\in S'': X_i^{(d)}=X_{i'}^{(d)}\}|/|S''|\le 1/2+5\delta^2]\le e^{-9\delta^4|S''|/4}$$

and

$$\mathbb{P}[|\{(i,i')\in S'': X'_i=X'_{i'}\}|/|S''|\ge 1/2+5\delta^2]\le e^{-9\delta^4|S''|/6}$$

So, the overlap between the probability distributions of X' and $X^{(d)}$ is at most $2e^{-9\delta^4|S''|/6}$. Now, observe that

$$\begin{aligned}9\delta^4|S''|/6 &\ge \delta^4|S|/4k^{d-d'}\\ &= \theta^{4(d-d')}|S|/1024k^{d-d'}\\ &\ge \frac{\theta^4}{1024k}\cdot|S|(|S|/6n)^{1-4\log_k\theta}\end{aligned}$$

Thus, the overlap between the probability distributions is at most $2e^{-\frac{\theta^4}{1024k}\cdot|S|(|S|/6n)^{1-4\log_k\theta}}$.

So, the desired conclusion holds with

$$c_1=\min\left(\frac{\theta^2}{4608\cdot 6^{1-2\log_k\theta}k^{2+\log(4)/\log\theta}},\frac{\theta^4}{1024k\cdot 6^{1-4\log_k\theta}}\right)$$

and $c_2 = 1 - 4\log_k \theta$. ■

So, at this point we have established that any $\mathbf{NC}^0$ circuit with independent random inputs must have a large set of outputs that are independent conditioned on any assignment of values to a relatively small set of inputs. Also, we know that the overlap between the probability distribution of the outputs conditioned on an assignment of value to these inputs and the probability distribution of $X^{(d)}$ must be small. Now, we just need to add up the overlaps for every possible assignment of values to these inputs in order to bound the overall overlap between the probability distribution of $X^{(d)}$ and the probability distribution of the circuit's output.

We are now ready to prove Theorem 23:

Proof First, let c be the maximum number of inputs that any output of f is ever affected by. Next, for each integer $0 \le i \le 2c$, let $b_i = e^{\ln(n)^{(2c-i)/2c}}$. There must exist $0 < i \le 2c$, $S \subseteq Ł_d$ and $T \subseteq \{1, \cdots, m_n\}$ such that $|S| \ge \frac{n}{2cb_i}$, $|T| \le cn/b_{i-1}$, and $\{X'_j : j \in S\}$ are independent conditioned on any fixed value of $\{W_j : j \in T\}$. Now, choose c_1 and c_2 satisfying the conditions of the previous lemma. Then, for every $w \in \{0,1\}^{|T|}$, let E_w be the event that the elements of W with indices in T take on the values given by w. Observe that

$$\begin{aligned}
&\sum_{x\in\{0,1\}^{L_d}} \min\left(\mathbb{P}[X^{(d)} = x], \mathbb{P}[X' = x]\right) \\
&\le \sum_{x\in\{0,1\}^{L_d}} \sum_{w\in\{0,1\}^{|T|}} \min\left(\mathbb{P}[X^{(d)} = x], \mathbb{P}[X' = x, E_w]\right) \\
&\le \sum_{w\in\{0,1\}^{|T|}} \sum_{x\in\{0,1\}^{L_d}} \min\left(\mathbb{P}[X^{(d)} = x], \mathbb{P}[X' = x|E_w]\right) \\
&\le \sum_{w\in\{0,1\}^{|T|}} 2e^{-c_1|S|^{1+c_2}/n^{c_2}} \\
&= 2^{|T|+1}e^{-c_1|S|^{1+c_2}/n^{c_2}} \\
&\le 2^{cn/b_{i-1}+1}e^{-c_1 n/(2cb_i)^{c_2}} \\
&= 2e^{\ln(2)cn/b_{i-1} - c_1 n/(2cb_i)^{c_2}}
\end{aligned}$$

Also, $b_i^{c_2} = o(b_{i-1})$. So, there exists n_0 such that for all $n \ge n_0$ and all integers $0 < i \le 2c$, we have that $\ln(2)cn/b_{i-1} \le c_1 n/(2cb_i)^{c_2}/2$.That means that for all $n \ge n_0$,

$$\begin{aligned}
&\sum_{x\in\{0,1\}^{L_d}} \min\left(\mathbb{P}[X^{(d)} = x], \mathbb{P}[X' = x]\right) \\
&\le 2e^{-c_1 n/(2cb_i)^{c_2}/2} \\
&\le 2e^{-c_1 n/(2cb_1)^{c_2}/2}
\end{aligned}$$

$\ln(b_1) = \ln(n)^{1-1/2c} = o(\ln(n))$, so

$$\sum_{x \in \{0,1\}^{L_d}} \min\left(\mathbb{P}[X^{(d)} = x], \mathbb{P}[X' = x]\right) = O\left(e^{-\sqrt{n}}\right)$$

as desired. ■

Appendix B. Random restrictions in the broadcast tree model

Here we prove Theorem 13. The usual approach to proving lower bounds against $\mathbf{AC}^0$ is through *random restrictions* where for every input x_i we leave it unset with probability p and otherwise we set it to zero with probability $\frac{1-p}{2}$ and set it to one with the remaining probability $\frac{1-p}{2}$. The main insight is that if the parameters are chosen appropriately, with high probability the $\mathbf{AC}^0$ circuit becomes much simpler (while the parity function remains a parity on fewer inputs). The key to our lower bound is an alternative but equivalent way to generate samples from the broadcast tree model. We will need the following definitions:

Definition 27 *Let $d' > 0$. Let $\phi_{d'} : \{0,1\}^{L_{d'-1}} \times \{0,1,*\}^{L_{d'}} \to \{0,1\}^{L_{d'}}$ be the function such that for all $x \in \{0,1\}^{L_{d'-1}}, r \in \{0,1,*\}^{L_{d'}}$ and $v \in L_{d'}$, we have that*

$$(\phi_{d'}(x,r))_v = \begin{cases} r_v & \text{for } r_v \in \{0,1\} \\ x_{\mathrm{par}(v)} & \text{for } r_i = *, \end{cases}$$

For the tree broadcast process the natural distribution for r is given by independent copies of the following distribution:

Definition 28 *For any $0 \le \theta < 1$, let R_θ be the probability distribution over $\{0,1,*\}$ such that a variable drawn from R_θ will be 0 with probability $(1-\theta)/2$, 1 with probability $(1-\theta)/2$, and '*' with probability θ.*

Replacing θ with $-\theta$ is equivalent to inverting all entries of σ at odd depths, so we can assume without loss of generality that $\theta \ge 0$.

Definition 29 *Let $d' > 0$. Let $\Phi_{d'} : \{0,1\}^{L_{d'-1}} \to \{0,1\}^{L_{d'}}$ be the random function such that for all $x \in \{0,1\}^{L_{d'-1}}$ and $v \in L_{d'}$, we let $\Phi_{d'}(x) = \phi_{d'}(x,r)$, where r is drawn from $R_\theta^{L_{d'}}$*

One can easily check that for all k, θ, and d' and all $x \in \{0,1\}^{L_{d'-1}}$ and $x' \in \{0,1\}^{L_{d'}}$, if we have $r \sim R_\theta^{L_{d'}}$ then

$$\mathbb{P}[\Phi(x) = x'] = \mathbb{P}[\phi(x,r) = x'] = \mathbb{P}[X^{(d')} = x' | X^{(d'-1)} = x].$$

For any $x \in \{0,1\}$, the probability distribution of $\Phi_d \circ \Phi_{d-1} \cdots \circ \Phi_1(x)$ is identical to the probability distribution of $X^{(d)}$ given that $X^{(0)} = x$. So as in classical applications of the switching lemma, we want to show that for any $f \in \mathbf{AC}^0$, $f \circ \Phi_d \circ \Phi_{d-1} \cdots \circ \Phi_1$ is a constant function with high probability. The first step towards doing that will be to prove that applying a logarithmic number of these projections to an $\mathbf{AC}^0$ circuit is enough to reduce the fan-in of all gates in its bottom layer to a constant with high probability. For that, we will need the following.

Lemma 30 *Let m, d', h, and c be positive integers. Also, let $f : \{0,1\}^{L_{d'}} \to \{0,1\}$ be a function such that there are only m inputs that ever affect its value. Next, let $f' = f \circ \Phi_{d'} \circ \Phi_{d'-1} \circ \cdots \circ \Phi_{d'-h+1}$ With probability at least $1-(m\theta^h)^c$, there are fewer than c inputs that affect the value of f'.*

Proof Each time we compose the function with Φ_i, each of its inputs is independently set to a constant with probability $1-\theta$, and then some of the inputs might be set to the same variable. If f' depends on c or more inputs, then there must be a set of c of the inputs of f that affect it such that none of these inputs get set to a constant or set to the same variable by any of the projections. There are at most m^c sets of c inputs of f that affect its value, and for any such set, the probability that none of them get set to a constant or merged is at most θ^{ch}. The desired conclusion follows. ■

Corollary 31 *Let b and h' be positive constants, $d' > h'\ln(n)$ be a positive integer, and $f : \{0,1\}^{L_{d'}} \to \{0,1\}$ be a function that takes an AND or OR of some subset of its inputs and their negations. Also, let $f' = f \circ \Phi_{d'} \circ \Phi_{d'-1} \circ \cdots \circ \Phi_{d'-\lceil h'\ln(n)\rceil+1}$. With probability at least $1-O(n^{-b})$, f' is an AND or OR of $-b/h'\ln\theta+1$ or fewer inputs and negations of inputs.*

Proof First of all, observe that if f takes an AND/OR of more than $2b\ln(n)$ variables, one of the projections will set one of its inputs to the value that reduces the function to a constant with probability $1-o(n^{-b})$. Otherwise, the desired conclusion follows by the lemma and the fact that a projection of an AND or OR must still be an AND or OR. ■

Now that we know that applying a logarithmic number of projections to the circuit will reduce the fan-in of all gates in the bottom layer to a constant with high probability, our next step is to prove that one more projection is enough to reduce all gates on the second layer to decision trees of logarithmic depth. In order to do that, we will need to prove that the projection of one of these gates can be represented by a decision tree of height $O(\log(n))$ with probability $1-n^{\Omega(1)}$. In other words, we need:

Lemma 32 *Let w be a positive constant. There exists a constant $h > 0$ such that if $p > 0$ is a function of n and f is a w-DNF on $\{0,1\}^{L_{d'}}$ then $f \circ \Phi_{d'}$ can be represented as a decision tree of height at most $h\ln(p)$ with probability $1-O(1/p)$.*

Proof We proceed by induction on w. If $w=0$, then every w-DNF is a constant function, and is thus expressible as a decision tree of height 0. Now, assume this result holds for $w-1$. If f is a w-DNF with more than $(\frac{1-\theta}{2})^{-w}\ln(p)$ clauses that do not share any variables, then with probability $1-O(1/p)$, the projection sets at least one of these clauses to 1, with the result that f becomes a constant function. Otherwise, there exists a set of at most

$$w \cdot \left(\frac{1-\theta}{2}\right)^{-w} \ln(p)$$

variables such that at least one of these variables appears in every clause. As such, any assignment of values to these variables would reduce f to a $(w-1)$-DNF. By the induction hypothesis, there exists a constant h' such that composing any resulting $(w-1)$-DNF with $\Phi_{d'}$ yields a decision tree of height at most $h' \ln(p)$ with probability at least

$$1 - O(p^{-1-\ln(2)w\cdot(\frac{1-\theta}{2})^{-w}})$$

That means that all assignments of values to these variables reduce $f \circ \Phi_{d'}$ to a decision tree of depth $h' \ln(p)$ with probability at least $1 - O(1/p)$. Therefore, $f \circ \Phi_{d'}$ can be represented as a decision tree of depth

$$\left[w \cdot \left(\frac{1-\theta}{2}\right)^{-w} + h'\right] \ln(p)$$

with probability $1 - O(1/p)$. This completes the proof. ■

At this point, we know that applying a logarithmic number of projections is enough to reduce every gate on the second level of an $\mathbf{AC}^0$ circuit to a decision tree of logarithmic depth with high probability. Any such decision tree can be computed by a polynomial size AND of ORs and by a polynomial size OR of ANDs. So, we can replace it by whichever allows us to reduce the circuit depth by 1. We are applying $\Omega(\log(n))$ projections in total, so if the circuit has depth b we can divide the projections into b serieses of $\Omega(\log(n))$ projections each. That is enough to reduce the entire circuit to a decision tree of logarithmic depth with a logarithmic number of projections left over. In fact, we can prove that with high probability, the depth of the decision tree is low enough that it must be unaffected by the values of most of the variables. Then, we can show that the remaining projections set all of the variables that the output does depend on to constants with high probability. As such, we can prove:

Lemma 33 *Let $f : \{0,1\}^{L_d} \to \{0,1\}$ be in $\mathbf{AC}^0$. Then there exists $\delta > 0$ such that $f \circ \Phi_d \cdots \circ \Phi_1$ is a constant function with probability $1 - O(n^{-\delta})$.*

Proof First, let $f^{(0)} = f$ and $f^{(i+1)} = f^{(i)} \circ \Phi_{d-i}$ for each i. Also, let b be the depth of f, and $\delta_1 > 0$ be a constant. We claim that $f^{(\lfloor id/b \rfloor)}$ can be expressed as a polynomial-sized circuit of depth $b - i$ with probability $1 - O(n^{-\delta_1})$ for each $0 \leq i < b - 1$, and prove this by induction on i. This is clearly true for $i = 0$. For $i > 0$, if $f^{(\lfloor (i-1)d/b \rfloor)}$ can be expressed as a polynomial-sized circuit of depth $b - i + 1$, then by corollary 1 there exists a constant c_i such that $f^{(\lfloor id/b \rfloor - 1)}$ can be expressed as a polynomial-sized circuit of depth $b - i + 1$ in which every gate at the bottom level has fanin at most c_i with probability $1 - O(n^{-\delta_1})$. Then by lemma 32, composing this with $\Phi_{d-\lfloor id/b \rfloor+1}$ allows us to replace all gates two levels from the bottom with decision trees of depth $O(\ln(n))$ with probability $1 - O(n^{-\delta_1})$. Every such decision tree can be converted to a DNF or CNF of size polynomial in n, so we can apply this transformation to all such gates in order to switch the order of the ORs and ANDs, thus allowing us to reduce the depth of the circuit by 1. Thus, $f^{(\lfloor id/b \rfloor)}$ can be expressed as a polynomial-sized circuit of depth $b - i$ with probability $1 - O(n^{-\delta_1})$, as desired.

That leaves us with the conclusion that $f^{(\lfloor (b-2)id/b \rfloor)}$ can be expressed as a polynomial-sized circuit of depth 2 with probability $1 - O(n^{-\delta_1})$. Then, by another application of corollary 1, we have that $f^{(\lfloor (b-1)id/b \rfloor - 1)}$ can be expressed as a DNF or CNF of constant fanin with probability $1 - O(n^{-\delta_1})$. Then, by lemma 32, we have that there exists a constant h such that $f^{(\lfloor (b-1)id/b \rfloor)}$ can be expressed as a decision tree of depth $h\delta_2 \log_2(n)$ with probability $1 - O(n^{-\delta_2})$ for any $\delta_2 > 0$. Such a decision tree can only be affected by $n^{h\delta_2}$ variables, so by lemma 30, $f^{(d)}$ is a constant function with probability $1 - O(n^{-\delta_1} + n^{-\delta_2} + \theta^{d/b} n^{h\delta_2})$. For $\delta_2 < -\ln\theta/[b(h+1)\ln(k)]$ and $\delta \leq \min(\delta_1, \delta_2)$, that means that $f^{(d)}$ is a constant with probability $1 - O(n^{-\delta})$. ■

Recall that for any fixed value of $X^{(0)}$, the probability distribution of $\Phi_d \cdots \circ \Phi_1(X^{(0)})$ is identical to the probability distribution of $X^{(d)}$. So, the probability that $f(X^{(d)}) = X^{(0)}$ is the same as the probability that $f \circ \Phi_d \cdots \circ \Phi_1(X^{(0)}) = X^{(0)}$. That means that the fact that $f \circ \Phi_d \cdots \circ \Phi_1$ is probably a constant implies that f is only accurate about half of the time.

Now we are ready to prove Theorem 13:

Proof Let $(r_1, \cdots, r_d)$ be bad if $f \circ \Phi_d \cdots \circ \Phi_1$ is a constant function and good otherwise. Then we have that

$$\begin{aligned}\mathbb{P}[f(X^{(d)}) = X^{(0)}] &= \mathbb{P}[f \circ \Phi_d \cdots \circ \Phi_1(X^{(0)}) = X^{(0)}] \\ &\leq 1/2 + \mathbb{P}[(r_1, \cdots, r_d) \text{ is good}]/2 \\ &= 1/2 + O(n^{-\delta})\end{aligned}$$

which completes the proof. ■

Corollary 34 *For every $c > 0$, there is no function in $\mathbf{AC}^0$ that computes whether more than half of its inputs are 1 whenever at least $n/2 + n^{1-c}/2$ of its inputs are the same.*

Proof For any such c, there is a choice of $0 \leq \theta < 1$ and $k > 0$ such that more than $n/2 + n^{1-c}/2$ of the entries in $X^{(d)}$ will equal $X^{(0)}$ with a probability of at least $2/3$. So, any such function would be capable of computing $X^{(0)}$ from $X^{(d)}$ with nontrivial accuracy. ■

Appendix C. A $\mathbf{TC}^0$ circuit for inference

Here we prove theorem 15. A natural approach to attempting to determine the root label is to guess that the root has the same label as the majority of the leaves, which gives the right answer with probability $1/2 + \Omega(1)$ if $\theta > 1/\sqrt{k}$. However, this is not an inference function. In particular, it achieves worse error even in an average-case sense. Alternatively we could compute an inference function using belief propagation but the naive way to encode this as a circuit would lead to logarithmic depth. The key idea is that the function computed by belief propagation is robust to injecting noise at the leaves. We use this idea by first guessing that each node at depth $\lfloor \log_k(\log_2(n)) \rfloor$ has the same label as the majority of the leaves descended from it. Then we guess the value of $X^{(0)}$ by computing the output of belief propagation (on the smaller depth tree) using a look up table. We are able to prove that this circuit is indeed a posterior function when $k\theta^2$ is sufficiently large and we conjecture that it is for any $k\theta^2 > 1$.

More precisely we will build a $\mathbf{TC}^0$ circuit that encodes the following algorithm.

LinearizedBP
Input : A function f and the values of d, k, θ, and $X^{(d)}$
Output: A guess of the value of $X^{(0)}$

1. Let $d' = \lfloor \log_k(\log_2(n)) \rfloor$.
2. For each $i \in L_{d'}$, randomly select $x_i^\star \in \{0,1\}$ and set

$$x_i = \begin{cases} 1 & \text{if } \sum_{j \in L_{d-d'}(i)} X_j^{(d)} > k^{d-d'}/2 \\ 0 & \text{if } \sum_{j \in L_{d-d'}(i)} X_j^{(d)} < k^{d-d'}/2 \\ x_i^\star & \text{if } \sum_{j \in L_{d-d'}(i)} X_j^{(d)} = k^{d-d'}/2 \end{cases}$$

3. Output $f(x)$

First of all, note that each value of n has a unique corresponding value of d', and each of the x_i can be computed from the inputs and a random bit by a threshold gate. $k^{d'} \leq \log_2(n)$, so there are at most n possible values of x. That means that we can use an AND gate to check for each possible value of x and then OR together the ones for which $f(x) = 1$. That means that for any fixed series of functions $f_d : \{0,1\}^{k^{\lfloor \log_k(\ln(n)) \rfloor}} \to \{0,1\}$, there is a $\mathbf{TC}^0$ circuit that computes LinearizedBP(d, k, θ, $X^{(d)}$, f) given access to $\log_2(n)$ random bits. Furthermore, we conjecture the following.

Conjecture 35 *There exists a series of functions $f_d : \{0,1\}^{k^{\lfloor \log_k(\ln(n)) \rfloor}} \to \{0,1\}$ such that if $X' = LinearizedBP(d, k, \theta, X^{(d)}, f_d)$ then*

$$\lim_{n \to \infty} \mathbb{P}[X' = X^{(0)}] - \mathbb{P}[\mathrm{BP}(X^{(d)}) = X^{(0)}] = 0,$$

where $\mathrm{BP}(x) : \{0,1\}^{L_d} \to \{0,1\}$ *returns the more likely posterior label of the root*

$$BP(x) = a \;\; \textit{if} \;\; \mathbb{P}[X^{(0)} = a | X^{(d)} = x] > \mathbb{P}[X^{(0)} = 1 - a | X^{(d)} = x]$$

In other words, we believe that LinearizedBP can compute $X^{(0)}$ with optimal accuracy. If $k\theta^2 \leq 1$, then it is known that no algorithm can compute $X^{(0)}$ from $X^{(d)}$ with nontrivial accuracy, so this algorithm uninterestingly attains optimal accuracy. In this section, we will prove that there exists $C > 1$ such that LinearizedBP can attain optimal accuracy whenever $k\theta^2 > C$. The case where $1 < k\theta^2 \leq C$ remains open. The first step towards proving that it can attain optimal accuracy for large values of $k\theta^2$ is to prove that when the algorithm is run, x is a reasonably accurate approximation of $X^{(d')}$. For that, we need the following standard second moment lemma which we include for completeness in Appendix F (similar lemmas were proven in previous work including Evans et al. (2000)).

Lemma 36 *For any d, k, and θ such that $k\theta^2 > 2$,*

$$\mathbb{P}\left[\sum_{i=1}^{k^d} X_i^{(d)} \leq k^d/2 \,\middle|\, X^{(0)} = 1 \right] \leq \frac{1}{\theta^2 k - 1}$$

By symmetry, this also implies that $\mathbb{P}\left[\sum_{i=1}^{k^d} X_i^{(d)} \geq k^d/2 \middle| X^{(0)} = 0\right] \leq \frac{1}{\theta^2 k - 1}$. So, that gives us a bound on $\mathbb{P}[x_i \neq X_i^{(d')}]$ when the algorithm is run. That leaves the task of showing that we can determine $X^{(0)}$ with optimal accuracy from a noisy version of $X^{(d')}$. In order to discuss the accuracy with which one can do that, we will need to define the following.

Definition 37 *Let $0 \leq s \leq 1/2$ and d be a positive integer. Also, let $X' \in \{0,1\}^{L_d}$ such that for each i, X'_i is independently set equal to $1 - X_i^{(d)}$ with probability s and $X_i^{(d)}$ otherwise.*

$$P_{s,d} = \sum_{x \in \{0,1\}^{L_d}} \max(\mathbb{P}[X^{(0)} = 0, X' = x], \mathbb{P}[X^{(0)} = 1, X' = x])$$

In other words, $P_{s,d}$ is the maximum accuracy with which we can determine $X^{(0)}$ from a noisy version of $X^{(d)}$ in which each bit is flipped with probability s. Mossel et al. (Mossel et al., 2016) show the following:

Proposition 38 *(Mossel et al., 2016) There exists $C > 0$ such that if $k\theta^2 > C$ then*

$$\lim_{s \to 1/2} \inf \lim_{d \to \infty} \inf P_{s,d} = \lim_{d \to \infty} \inf P_{0,d}$$

In other words, if $k\theta^2$ is sufficiently large then the maximum accuracy with which $X^{(0)}$ can be determined from a highly noisy estimate of $X^{(d')}$ is the same as the maximum accuracy with which $X^{(0)}$ can be determined from $X^{(d')}$. That allows us to prove theorem 15 as follows.

Proof First, let $C' = \max(C, 4)$. First we observe that for any d, when LINEARIZEDBP is run, each bit x_i is independently set equal to $X_i^{(d')}$ with some advantage over random guessing and set to the opposite value otherwise. Let $s_d = \mathbb{P}[x_i \neq X_i^{(d')}]$. Next, let f_d be the function that maximizes the probability that LINEARIZEDBP outputs the correct label for the root. Let q be the probability that it succeeds. Then we have

$$q = \sum_{x' \in \{0,1\}^{L_{d'}}} \max(\mathbb{P}[X^{(0)} = 0, x = x'], \mathbb{P}[X^{(0)} = 1, x = x']) = P_{s_d, d'}$$

Now, let $s' = \frac{1}{\theta^2 k - 1}$. Proposition 38 shows that $s_d \leq s'$ for all d, and adding more noise can never make it easier to determine $X^{(0)}$, so for every d, it must be the case that

$$P_{0,d'} \geq P_{s_d,d'} \geq P_{s',d'}$$

Combining that with the previous theorem shows that

$$\lim_{d' \to \infty} \inf P_{0,d'} \geq \lim_{d' \to \infty} \inf P_{s',d'} \geq \lim_{s \to 1/2} \inf \lim_{d' \to \infty} \inf P_{s,d'} = \lim_{d' \to \infty} \inf P_{0,d'}$$

Also, $P_{0,d'}$ is a nonincreasing function of d', so $P_{0,d'}$ converges. So,

$$\lim_{s \to 1/2} \sup \lim_{d' \to \infty} \sup P_{s,d'} \leq \lim_{d' \to \infty} \sup P_{s',d'} \leq \lim_{d' \to \infty} \sup P_{0,d'} = \lim_{d' \to \infty} P_{0,d'}$$

That implies that all of these sequences converge to $\lim_{d' \to \infty} P_{0,d'}$, and thus that LINEARIZEDBP computes $X^{(0)}$ with optimal accuracy. ■

Appendix D. Computing the posterior in $\mathbf{NC}^1$

Here we prove that there is an $\mathbf{NC}^1$ circuit for computing the posterior. This is the first part of Theorem 14. Our plan is to essentially create randomized circuits that allow us to sample from the probability distributions of vertices' labels, and then count up how many of the samples for the root are 1. These circuits will occasionally fail, but they will be reasonably reliable in the following sense.

Lemma 39 *For every $-1 < \theta < 1$ and positive integer k, there exists $h > 0$ such that for every $d' \geq 0$, there exists a probability distribution $P_F^{d'}$ over functions from $\{0,1\}^{L_{d'}} \to \{0,1,?\}$ with the following properties:*

- *Every function drawn from $P_F^{d'}$ can be computed by an NC circuit of depth at most hd'.*
- *For every $x \in \{0,1\}^{L_{d'}}$, if $F \sim P_F^{d'}$ then $\mathbb{P}[F(x) \in \{0,1\}] \geq 1 - 1/2k$.*
- $\mathbb{P}[F(x) = 1 | F(x) \in \{0,1\}] = \mathbb{P}[X^{(0)} = 1 | X^{(d')} = x]$

Proof We proceed by induction on d'. For $d' = 0$, we can always return the function f such that $f(0) = 0$ and $f(1) = 1$. Now, assume that this holds for $d' - 1$. Prior to defining $P_F^{d'}$, we will define a preliminary probability distribution $P_F^{d'\star}$, such that in order to draw a function $F^\star$ from $P_F^{d'\star}$, we do the following. First, draw $F_1, \cdots, F_k$ independently from $P_F^{d'-1}$. Also, independently choose $\delta_1, \cdots, \delta_k$ such that for each i, δ_i is 1 with probability $(1-\theta)/2$ and 0 otherwise.

If there exists i such that $F_i(x(L_{d'-1}(i))) \notin \{0,1\}$, then $F^\star(x) ='?'$. Otherwise, let $x_i^\star = F_i(x(L_{d'-1}(i)))$ for each i. Then, set $F^\star(x)$ equal to 0 if $x_i^\star$ xor $\delta_i = 0$ for all i, set it to 1 if $x_i^\star$ xor $\delta_i = 1$ for all i, and set it to $'?'$ otherwise.

For any fixed value of x, when $F^\star \sim P_F^{d'\star}$, the values of the $x_i^\star$ are independent. As such,

$$\mathbb{P}[F^\star(x) = 0] = \prod_{i=1}^{k} \left[\left(\frac{1+\theta}{2}\right) \mathbb{P}[x_i^\star = 0] + \left(\frac{1-\theta}{2}\right) \mathbb{P}[x_i^\star = 1] \right]$$

and

$$\mathbb{P}[F^\star(x) = 1] = \prod_{i=1}^{k} \left[\left(\frac{1+\theta}{2}\right) \mathbb{P}[x_i^\star = 1] + \left(\frac{1-\theta}{2}\right) \mathbb{P}[x_i^\star = 0] \right]$$

By the requirement that $\mathbb{P}[F_i(x) \in \{0,1\}] \geq 1 - 1/2k$, the $x^\star$ are all assigned values in $\{0,1\}$ with probability at least 1/2, which also implies that

$$\mathbb{P}[F^\star(x) \in \{0,1\}] \geq [(1-\theta^2)/4]^{k/2}$$

By the induction hypothesis,

$$\mathbb{P}[x_i^\star = 1 | x_i^\star \neq' ?', X^{(d')} = x] = \mathbb{P}[X_i^{(1)} = 1 | (X_{(i-1)k^{d'-1}+1}, \cdots, X_{ik^{d'}}) = (x_{(i-1)k^{d'-1}+1}, \cdots, x_{ik^{d'}})]$$

This implies that

$$\mathbb{P}[F^\star(x) = 1 | F^\star(X^{(d')}) \in \{0,1\}] = \mathbb{P}[X^{(0)} = 1 | X^{(d')} = x]$$

Furthermore, $F^\star(x)$ can be computed from the values output by the F_i by an NC circuit of some constant depth.

So, $P_F^{d'\star}$ has all of the properties that we want, except that its functions return $'?'$ with excessively high probability. So, in order to draw a function from $P_F^{d'}$, we simply draw $\lceil [(1-\theta^2)/4]^{-k/2}\ln(2k)\rceil$ independent functions from $P_F^{d'\star}$. Then, we compute them all on the input we are given, and return the first output in $\{0,1\}$, if any. This leaves the relative probability of returning 0 and 1 unchanged, reduces the probability of returning $'?'$ to $1/2k$ or less, and only increases the circuit depth by a constant. So, $P_F^{d'}$ has all of the desired properties. ■

That means that we have randomized $\mathbf{NC}^1$ circuits that can essentially draw a sample from the probability distribution of $X^{(0)}$ given that $X^{(d)}=x$, with the complication that they occasionally fail to return a value in $\{0,1\}$. So, if we use a large number of them, we can count up how many of them return 1 and how many return 0 in order to estimate $\mathbb{P}[X^{(0)}=1|X^{(d)}=x]$. With enough of these circuits, this estimate will be within $\delta/2$ of the true probability at least $1-o(2^{-n})$ of the times, which means that there must be some choice of the randomness for which it is always right. That allows us to prove:

Proposition 40 *For every k and θ and in the Ising tree model, there is a posterior function that can be computed by an $\mathbf{NC}^1$ circuit.*

Proof Consider independently drawing $F_1,\cdots,F_{n^4}$ from P_F^d. Also, consider any $x\in\{0,1\}^n$ such that $\mathbb{P}[X^{(0)}=1|X^{(d)}=x]>1/2+1/n$. For each i, it is the case that $\mathbb{P}[F_i(x)\in\{0,1\}]\geq 1/2$ and $\mathbb{P}[F_i(x)=1|F_i(x)\in\{0,1\}]\geq 1/2+1/n$. So, there will be more i for which $F_i(x)=1$ than i for which $F_i(x)=0$ with probability $1-o(2^{-n})$. That in turn implies that this holds for every such x with probability $1-o(1)$. By the same logic, there will be more i for which $F_i(x)=0$ than i for which $F_i(x)=1$ for every x such that $\mathbb{P}[X^{(0)}=1|X^{(d)}=x]<1/2-1/n$ with probability $1-o(1)$. That means that there must exist a specific choice of $F_1,\cdots,F_{n^4}$ for which both of these properties hold. These functions can each be computed by an $\mathbf{NC}^1$ circuit, and a logarithmic additional depth is sufficient to determine whether more of them output 1 or 0. Thus, the function that returns 1 if more of them output 1 than 0 and 0 otherwise is an $\mathbf{NC}^1$ posterior function.

■

Appendix E. Gadgets in the broadcast tree model

Here we prove that being able to compute the posterior allows us to implement any $\mathbf{NC}^1$ circuit. This is the second part of Theorem 14. Our plan is to encode the circuit as a tree where each vertex corresponds to an input or gate. The probability that the vertex's label is 1 given the values of its leaves will be at least .95 if the corresponding input or gate outputs 1 and at most .05 if the corresponding input or gate outputs 0. Our first step towards doing this is to prove that with appropriate parameters, each vertex will encode the same value as the majority of its children. More formally, we have the following.

Lemma 41 *Let* $\theta = \frac{9}{10}$ *and* $k = 6$*. Next, choose* $x \in \{0,1\}^{L_d}$ *such that there are at least* 4 *choices of* i *such that* $\mathbb{P}[X_i^{(1)} = 1 | X^{(d)}(L_{d-1}(i)) = x(L_{d-1'}(i))] \geq .95$*. Then* $\mathbb{P}[X^{(0)} = 1 | X^{(d)} = x] \geq \frac{19}{20}$*.*

Proof First of all, for each i, let $p_i = \mathbb{P}[X_i^{(1)} = 1 | X^{(d)}(L_{d-1}(i)) = x(L_{d-1}(i))]$. Given these values of θ and k, it must be the case that

$$\begin{aligned}
\mathbb{P}[X^{(0)} = 1 | X^{(d)} = x] &= \frac{\prod_{i=1}^{6}\left(\frac{19}{20}p_i + \frac{1}{20}(1-p_i)\right)}{\prod_{i=1}^{6}\left(\frac{19}{20}p_i + \frac{1}{20}(1-p_i)\right) + \prod_{i=1}^{6}\left(\frac{1}{20}p_i + \frac{19}{20}(1-p_i)\right)} \\
&\geq \frac{\left((\frac{19}{20})^2 + (\frac{1}{20})^2\right)^4 \cdot (\frac{1}{20})^2}{\left((\frac{19}{20})^2 + (\frac{1}{20})^2\right)^4 \cdot (\frac{1}{20})^2 + \left(\frac{19}{20}\cdot\frac{1}{20} + \frac{1}{20}\cdot\frac{19}{20}\right)^4 \cdot (\frac{19}{20})^2} \\
&> \frac{19}{20}
\end{aligned}$$

■

That lets us make a vertex encode the majority of the values encoded by 3 other vertices, which in turn allows us to make it encode an AND or OR of two other vertices by setting the third to a constant. That allows us to prove the following.

Proposition 42 *Let* $\theta = \frac{9}{10}$ *and* $k = 6$ *and consider the Ising tree model. For every* **NC** *circuit of depth* d*, there exists a way to define* $x \in \{0,1\}^{L_d}$ *so that* x_i *is set to* 0*,* 1*, an input to the circuit, or the negation of the input to the circuit, such that for every choice of inputs to the circuit,* $\mathbb{P}[X^{(0)} = 1 | X^{(d)} = x]$ *is at least* $\frac{19}{20}$ *if the circuit outputs* 1 *on this input, and at most* $\frac{1}{20}$ *if the circuit outputs* 0 *on this input.*

Proof We proceed by induction on d. This is clearly true for $d = 0$. Now, assume that it holds for $d-1$, and consider a function f that is computable by an **NC** circuit of depth d. There must exist functions f_1 and f_2 that are computable by **NC** circuits of depth $d-1$ such that either $f = NOT(f_1)$, $f = f_1$ AND f_2, or $f = f_1$ OR f_2. By the induction hypothesis, for each i, j there is a way to set all of the entries in $\{x_{i'} : i' \in L_{d-1}(i)\}$ equal to constants, inputs to the circuit, or negations of inputs in such a way that $\mathbb{P}[X_i^{(1)} = 1 | X^{(d)}(L_{d-1}(i)) = x(L_{d-1}(i))]$ is always at least $\frac{19}{20}$ if f_j outputs 1 and at most $\frac{1}{20}$ if it outputs 0. Also, $\mathbb{P}[X_i^{(1)} = 1 | X^{(d)}(L_{d-1}(i)) = (1, \ldots, 1)] > \frac{19}{20}$ and $\mathbb{P}[X_i^{(1)} = 1 | X^{(d)}(L_{d-1}(i)) = (0, \ldots, 0)] < \frac{1}{20}$ by repeated application of the previous lemma.

In particular, if we set x so that

$$\mathbb{P}[X_i^{(1)} = 1 | X^{(d)}(L_{d-1}(i)) = x(L_{d-1}(i))]$$

tracks f_1 for $i = 1, 2$, f_2 for $i = 3, 4$, and 1 for $i = 5, 6$ then $\mathbb{P}[X^{(0)} = 1 | X^{(d)} = x]$ will track $f = f_1$ OR f_2 by the lemma. If we have it track 0 for $i = 5, 6$ instead, then it will track $f = f_1$ AND f_2 instead. That leaves the case where $f = NOT(f_1)$. In that case, we can simply start with the assignment of value to x that we would use if $f = f_1 = f_1$ OR f_1, and then invert every entry in x in order to switch the probability that $X_i^{(1)} = 1$ with the probability that it is 0. So, the desired conclusion holds for d. ■

Remark 43 *More generally, given any $\theta, k, \delta > 0$ such that $\lim_{d\to\infty} \mathbb{P}[X^{(0)} = 1 | X^{(d)} = 1, 1, \cdots, 1] > 1/2 + \delta$, determining whether $\mathbb{P}[X^{(0)} = 1 | X^{(d)} = x] > 1/2$ whenever this probability is greater than $1/2 + \delta$ or less than $1/2 - \delta$ is $\mathbf{NC}^1$-hard. This can be proven by a variant of the above argument. In it we would argue that there exists δ' such that if f_1 and f_2 are functions that can be tracked by trees of depth d' with an accuracy of $1/2 + \delta$, we can construct a tree of depth $d' + 2$ that tracks f_1 AND f_2 with accuracy $1/2 + \delta'$. Then we would argue that we can amplify the accuracy back up to $1/2 + \delta$ by constructing a tree such that all of its subtrees at some suitable depth are copies of that tree. That would allow us to prove the desired result by induction on circuit depth the same way we do in the theorem above.*

Combining this with the previous theorem shows that posterior computation is $\mathbf{NC}^1$-complete, as desired.

Appendix F. Deviation bounds for the broadcast tree model

Here we prove Lemma 36:

Proof First, observe that

$$E\left[\sum_{i \in L_d} X_i^{(d)} \middle| X^{(0)} = 1\right] = k^d/2 + k^d\theta^d/2$$

Now, for each $0 \le d' \le d$, let $v_{d'} = Var\left[\sum_{i \in L_{d-d'}(1^{d'})} X_i^{(d)} \middle| X_{1^{d'}}^{(d')} = 1\right]$. Clearly, $v_d = 0$, and for each $d' < d$, it must be the case that

$$v_{d'} = kVar\left[X_{1^{d'+1}}^{(d'+1)} \middle| X_{1^{d'}}^{(d')} = 1\right] \cdot k^{2d-2d'-2}\theta^{2d-2d'-2} + k \cdot v_{d'+1}$$

And hence we have

$$\begin{aligned} v_0 &= \sum_{d'=0}^{d-1} \frac{1-\theta^2}{4} k^{d'+1} \cdot k^{2d-2d'-2}\theta^{2d-2d'-2} \\ &\le \frac{1-\theta^2}{4} \sum_{d'=0}^{\infty} k^{2d-d'-1}\theta^{2d-2d'-2} \\ &= \frac{1-\theta^2}{4} k^{2d}\theta^{2d}/[\theta^2 k - 1] \end{aligned}$$

In particular, this implies that

$$\begin{aligned} \mathbb{P}\left[\sum_{i \in L_d} X_i^{(d)} \le k^d/2 \middle| X^{(0)} = 1\right] &\le \text{Var}\left[\sum_{i \in L_d} X_i^{(d)} \middle| X^{(0)} = 1\right] \Big/ \left(\mathbb{E}\left[\sum_{i \in L_d} X_i^{(d)} \middle| X^{(0)} = 1\right] - k^d/2\right)^2 \\ &= \frac{1-\theta^2}{\theta^2 k - 1} \le \frac{1}{\theta^2 k - 1} \end{aligned}$$

■

Appendix G. Average case $\mathbb{NC}^1$ hardness

Here we prove that detection is $\mathbb{NC}^1$-complete in the following instance of the generalized broadcast process on a tree. There is one label for every ordered pair $(\sigma, \sigma') \in A_5^2$, and $k = 60000$. Given a vertex with a parent with label (σ, σ'), we select a random $b \in A_5$. Then, we set its label to $(b, b^{-1}\sigma)$ with probability $2/3$ and $(b, b^{-1}\sigma')$ with probability $1/3$. For the rest of this section, we will assume that $\sigma^\star$ was generated by the generalized broadcast process with these parameters. Recall that it is $\mathbb{NC}^1$ complete to determine the product of a random series of elements in A_5 with nontrivial accuracy. Given such a series of elements, we can encode the problem of finding there product in an instance of the broadcast tree model using the following algorithm.

PRODUCTTREECONSTRUCTIONALGORITHM
Input : A positive integer d
Output: A formula for $X^{(d)}$ as a function of $\sigma_1, ..., \sigma_{2^{d+1}} \in A_5$.

1. Set $\overline{X}^{(0)} = \left(\prod_{i=1}^{2^d} \sigma_i, \prod_{i=2^d+1}^{2^{d+1}} \sigma_i\right)$.
2. **for** $d' = 1$ *to* d **do**
   for $i \in L_{d'}$ **do**
   (a) There will exist a constant $0 \le j \le 2^{d+1}$ and constants $b, b', b'' \in A_5$ such that

$$\overline{X}_{\text{par}(i)}^{(d'-1)} = \left(b' \cdot \left(\prod_{i=j+1}^{j+2^{d-d'+1}} \sigma_i\right) \cdot b, b^{-1} \cdot \left(\prod_{i=j+2^{d-d'+1}+1}^{j+2^{d-d'+2}} \sigma_i\right) \cdot b''\right)$$

   (b) Randomly select $b''' \in A_5$.
   (c) With probability 2/3, set

$$\overline{X}_i^{(d')} = \left(b' \cdot \left(\prod_{i=j+1}^{j+2^{d-d'}} \sigma_i\right) \cdot b''', (b''')^{-1} \cdot \left(\prod_{i=j+2^{d-d'}+1}^{j+2^{d-d'+1}} \sigma_i\right) \cdot b\right)$$

   Otherwise, set

$$\overline{X}_i^{(d')} = \left(b^{-1} \cdot \left(\prod_{i=j+2^{d-d'+1}+1}^{j+3\cdot 2^{d-d'}} \sigma_i\right) \cdot b''', (b''')^{-1} \cdot \left(\prod_{i=j+3\cdot 2^{d-d'}+1}^{j+2^{d-d'+2}} \sigma_i\right) \cdot b''\right)$$

   end
   end
3. Return $\overline{X}^{(d)}$.

In step 2.a we asserted that every element of $\overline{X}^{(d'-1)}$ will have the form

$$\left(b' \cdot \left(\prod_{i=j+1}^{j+2^{d-d'+1}} \sigma_i\right) \cdot b, b^{-1} \cdot \left(\prod_{i=j+2^{d-d'+1}+1}^{j+2^{d-d'+2}} \sigma_i\right) \cdot b''\right)$$

It is easy to see that this is true for $\overline{X}^{(0)}$ and throughout the process, $\overline{X}^{(d'-1)}$ will always be set to an expression of this form. The key fact is:

Lemma 44 *Let $\sigma \in A_5^{2^{d+1}}$ and $x_0 = \left(\prod_{i=1}^{2^d} \sigma_i, \prod_{i=2^d+1}^{2^{d+1}} \sigma_i\right)$. Then for every $x \in (A_5^2)^n$,*

$$\mathbb{P}\left[X^{(d)} = x \middle| X^{(0)} = x_0\right] = \mathbb{P}\left[\overline{X}^{(d)}(\sigma) = x\right]$$

Thus ProductTreeConstructionAlgorithm(d) is an equivalent way to sample from the generalized broadcast tree model that we defined earlier.
Proof We will prove by induction on d' that the distribution of $X^{(d')}$ given $X^{(0)} = x_0$ is identical to the distribution of $\overline{X}^{(d')}(\sigma)$ for every d'. If $d' = 0$, then $\overline{X}^{(d')} = x_0$, so the base case holds. Now, assume that it holds for $d' - 1$.

It is easy to check that the way we have defined step 2.c every vertex at depth d' is assigned a label whose product is equal to the first permutation in its parent's label with probability 2/3 and the second permutation in its parent's label with probability 1/3. Moreover the pair of permutations is chosen uniformly at random subject to this constraint. Finally each element of $\overline{X}^{(d')}$ is independent conditioned on the value of its parent. These are exactly the key properties that defined our generalized broadcast tree model, and hence completes the proof. ■

Now we are ready to prove that any algorithm for solving the detection problem for our generalized broadcast tree model can be used to solve the word problem over A_5 with some advantage over random guessing:

Theorem 45 *Let $g_d : (A_5^2)^{k^d} \to A_5$ be a family of functions. Suppose there exists $\epsilon > 0$ independent of d such that*

$$\mathbb{P}[g_d(X^{(d)}) = X^{(0)}] \geq \frac{1}{|A_5|^2} + \epsilon$$

If $\mathbf{TC}^0 \neq \mathbf{NC}^1$ then g is not in $\mathbf{TC}^0$.

Proof For the sake of contradiction we will assume that $g \in \mathbf{TC}^0$. Let $\Sigma_1, \cdots, \Sigma_{2^{d+1}}$ be chosen randomly. We can interpret ProductTreeConstructionAlgorithm(d) as outputting a random formula that labels the leaves of the generalized broadcast tree model. The key point is both the depth of the tree and the number of bits of randomness that determine the value at any leaf are both logarithmic. Thus $X^{(d)}$ can be computed by a $\mathbf{TC}^0$ circuit. Now let g'_d be the composition of g_d and ProductTreeConstructionAlgorithm(d).

Because g_d solves the detection problem we have that

$$\mathbb{P}\left[g'_d(\Sigma_1, \cdots, \Sigma_{2^{d+1}}) = \left(\prod_{i=1}^{2^d} \Sigma_i, \prod_{i=2^d+1}^{2^{d+1}} \Sigma_i\right)\right] \geq \frac{1}{|A_5|^2} + \epsilon$$

where the randomness is over both the choice of the Σ_i's and g' which depends on the generation process. For the sake of simplifying the notation, let $g'_d(\sigma) = (g_d^{[1]}(\sigma), g_d^{[2]}(\sigma))$. Now there are two cases:

In the first case suppose that $g_d^{[1]}$ gets nontrivial advantage over random guessing. In particular suppose

$$\mathbb{P}\left[g_d^{[1]}(\Sigma_1, \cdots, \Sigma_{2^{d+1}}) = \prod_{i=1}^{2^d} \Sigma_i\right] \geq \sqrt{\frac{1}{|A_5|^2} + \epsilon}$$

There must exist a specific choice $\Sigma_{2^d+1} = \sigma_{2^d+1}, \cdots, \Sigma_{2^{d+1}} = \sigma_{2^{d+1}}$ and setting of the randomness in the generation process which achieves nontrivial advantage over random guessing. Even when we fix these values, the function is still in $\mathbf{TC}^0$ and hence we conclude $\mathbf{TC}^0 = \mathbf{NC}^1$.

In the second case, we must have

$$\mathbb{P}\left[g_d^{[2]}(\Sigma_1, \cdots, \Sigma_{2^{d+1}}) = \prod_{i=2^d+1}^{2^{d+1}} \Sigma_i \middle| g_d^{[1]}(\Sigma_1, \cdots, \Sigma_{2^{d+1}}) = \prod_{i=1}^{2^d} \Sigma_i\right] \geq \sqrt{\frac{1}{|A_5|^2} + \epsilon}$$

The idea is we want to use $g_d^{[2]}$ to solve an $\mathbf{NC}^1$ hard problem, but to do so using the above inequality we need to decide if the output of $g_d^{[1]}$ is correct. Now we can once again use an average-case reduction to reduce to the case when we know the product of the inputs to $g_d^{[1]}$ and thus check its own output.

In particular for any $\sigma_1, \cdots, \sigma_{2^{d+1}} \in A_5$ and randomly generated $B_1, \cdots, B_{2^{d+1}}$, let

$$\Sigma' = (\sigma_1 B_1, B_1^{-1}\sigma_2 B_2, B_2^{-1}\sigma_3 B_3, \cdots, B_{2^{d+1}-1}^{-1}\sigma_{2^{d+1}} B_{2^{d+1}})$$

The distribution of Σ' is uniform on $A_5^{2^{d+1}}$ so we have

$$\mathbb{P}\left[g_d^{[2]}(\Sigma') = B_{2^d}^{-1}\left(\prod_{i=2^d+1}^{2^{d+1}} \sigma_i\right) B_{2^{d+1}} \middle| g_d^{[1]}(\Sigma') = \left(\prod_{i=1}^{2^d} \sigma_i\right) B_{2^d}\right] \geq \sqrt{\frac{1}{|A_5|^2} + \epsilon}$$

Now we can choose $\sigma_1, \cdots, \sigma_{2^d}$ such that we already know their product and we can repeatedly generate $B_1, \cdots, B_{2^{d+1}}$ until we find one for which

$$g_d^{[1]}(\Sigma') = \left(\prod_{i=1}^{2^d} \sigma_i\right) B_{2^d}$$

Now if we guess that $\prod_{i=2^d+1}^{2^{d+1}} \sigma_i$ is equal to $B_{2^d} g_d^{[2]}(\Sigma') B_{2^{d+1}}^{-1}$ we will get nontrivial advantage over random guessing. As before there must be some choice of the randomness (in this case the values of $B_1, \cdots, B_{2^{d+1}}$ and the randomness in the generation process) where the probability of computing the product is at least average. This again implies that $\mathbf{TC}^0 = \mathbf{NC}^1$. ■

So, this is a set of parameters for which one can determine the root's label from the leaves' labels with very high accuracy in the average case. However, unless $\mathbf{TC}^0$=$\mathbf{NC}^1$, there is no $\mathbf{TC}^0$ algorithm that can determine the root's label with an accuracy that is nontrivially higher than that attained by guessing blindly. With some more work, we could prove that this also holds for sufficiently slight perturbations of these parameters. Next, we show how to reduce the number of labels to 16 by using symmetry arguments and working with conjugacy classes of permutations instead.

G.1. Reducing the number of labels

It turns out that we will be able to exploit the symmetries in our generalized broadcast tree model in the previous section to be able to drastically reduce the number of labels from $|A_5|^2 = 3600$ corresponding to all pairs of even permutations to 16 corresponding to pairs of *conjugacy classes* of even permutations in S_5 — namely two even permutations σ and τ are in the same conjugacy class if there is a permutation c (not necessarily even) for which $\tau = c^{-1}\sigma c$. Intuitively, knowing the conjugacy class fixes the cycle structure of a permutation.

The main technical ingredient in this section is to show that if the the labels can be grouped into collections in such a way that the probability that a vertex has a child in a given collection depends only on what collection that vertex is in, then we can replace the labels with the collections without making it easier to determine $X^{(0)}$ with nontrivial accuracy. More formally, we have the following.

Lemma 46 *Consider a generalized broadcast tree model with parameters q, k and M. Suppose there is a partition $S_1, \cdots, S_{q'}$ of $\{1, \cdots, q\}$ with the following property: Let $w^{(i)} = \sum_{j \in S_i} e_j$ for each i. Then for all $1 \leq i, i' \leq q'$ and $j, j' \in S_i$ we have*

$$w^{(i')} \cdot Me_j = w^{(i')} \cdot Me_{j'}$$

Finally let M' be the $q' \times q'$ matrix such that for each i, i', $M'_{i,i'} = w^{(i)} \cdot Me_j$ for some $j \in S_{i'}$. If there is a $\mathbf{TC}^0$ detection function for the generalized broadcast process with parameters (q', M') then there is a $\mathbf{TC}^0$ detection function for (q, M) as well.

Proof First fix any d and let $(X^{(0)}, \cdots, X^{(d)})$ be vectors of labels generated by the generalized broadcast process with parameters (q, M). The labels of the generalized broadcast process with parameters (q', M') will naturally be associated with parts of the partition. Let $(X'^{(0)}, \cdots, X'^{(d)})$ be the result of replacing each label with the part it belongs to. Also, let $(X^{\star(0)}, \cdots, X^{\star(d)})$ be vectors of labels generated by the generalized broadcast process with parameters (q', M'). We claim that the distribution of $(X'^{(0)}, \cdots, X'^{(d)})$ conditioned on a fixed value of $X'^{(0)}$ is the same as that of $(X^{\star(0)}, \cdots, X^{\star(d)})$ conditioned on the same value for the root label. This is because by assumption, when we only care about which part of the partition each child belongs to, it only matters what part of the partition the parent belongs to.

Now suppose that f is a $\mathbf{TC}^0$ function that solves the detection problem for the generalized broadcast process with parameters (q', M'). Finally let $\hat{X}$ be a random label contained in $S_{f(X'^{(d)})}$. Then

$$\begin{aligned}
\mathbb{P}[\hat{X} = X^{(0)}] &= \sum_{i=1}^{q'} \mathbb{P}[\hat{X} = X^{(0)} | X'^{(0)} = i] \mathbb{P}[X'^{(0)} = i] \\
&= \sum_{i=1}^{q'} \frac{\mathbb{P}[f(X'^{(d)}) = X'^{(0)} | X'^{(0)} = i]}{|S_i|} \cdot \frac{|S_i|}{q} \\
&= \frac{1}{q} \sum_{i=1}^{q'} \mathbb{P}[f(X^{\star(d)}) = X^{\star(0)} | X^{\star(0)} = i] = \frac{q'}{q} \mathbb{P}[f(X^{\star(d)}) = X^{\star(0)}]
\end{aligned}$$

Also note that we can compute $X'^{(d)}$ from $X^{(d)}$ using an $\mathbf{NC}^0$ circuit. Putting it all together, because $f \in \mathbf{TC}^0$, and there must be a specific way to choose a value of $\hat{X}$ for each possible value of $f(X'^{(d)})$ such that

$$\mathbb{P}[\hat{X} = X^{(0)}] \geq \frac{q'}{q}\mathbb{P}[f(X^{\star(d)}) = X^{\star(0)}]$$

Hence there is a $\mathbf{TC}^0$ circuit that computes $X^{(0)}$ from $X^{(d)}$ with nontrivial advantage. ■

In particular, we can now reduce the number of labels in our generalized broadcast process as follows. We will call this final model the generalized broadcast process on conjugacy classes. There are 16 labels corresponding to the ordered pairs of conjugacy classes of even permutations in S_5. In order to assign a label to a vertex's child, first let σ be a random element of the vertex's label's first conjugacy class with probability 2/3 and a random element of its label's second conjugacy class with probability 1/3. Then, select random $\sigma', \sigma'' \in A_5$ such that $\sigma' \cdot \sigma'' = \sigma$. Finally, set the child's label equal to the pair of the conjugacy classes of σ' and σ''.

Theorem 47 *If there is an $\mathbf{TC}^0$ detection function for the generalized broadcast process on conjugacy classes then $\mathbf{TC}^0 = \mathbf{NC}^1$.*

Proof What we need to do is verify that partitioning A_5 into conjugacy classes in S_5 satisfies the conditions in Lemma 46. First, observe that given any $\sigma, \overline{\sigma}$ in the same conjugacy class of S_5, there exists $c \in S_5$ such that $\overline{\sigma} = c^{-1}\sigma c$. So, if $\sigma' \cdot \sigma'' = \sigma$ then $(c^{-1}\sigma' c) \cdot (c^{-1}\sigma'' c) = \overline{\sigma}$. That gives us a bijection between pairs of permutations in any given pair of conjugacy classes with a product of σ and pairs of permutations in that pair of conjugacy classes with a product of $\overline{\sigma}$. So, if we set $S_1, \cdots, S_{16}$ equal to the sets of pairs of permutations in each even conjugacy class of S_5 then by Lemma 46 we have that if $\mathbf{TC}^0 \neq \mathbf{NC}^1$ there is no $\mathbf{TC}^0$ detection function for this instance of the generalized broadcast process on conjugacy classes. ■

Finally we show that the detection problem can be solved in $\mathbf{NC}^1$ where as before we set $k = 60000$. First we note that one of the conjugacy classes contains only the identity, so if a vertex's label is (S, S'), then each of its children have a label of $(\{1\}, S)$ with probability 1/90, $(\{1\}, S')$ with probability 1/180, and no other possibility of having a label with its first entry equal to the identity's conjugacy class. As such, if we can determine the labels of the vertices at depth d' with accuracy .999 then for each vertex at depth $d'-1$, we can estimate how many children it has with label $(\{1\}, S'')$ for each conjugacy class S'' and use that to determine its label with accuracy at least .999. Therefore, by induction we can determine the root's label correctly with probability at least .999. Also, this can clearly be done in $\mathbf{NC}^1$.