Large Deviation Bounds for Decision Trees and Sampling Lower Bounds for AC0-circuits

Chris Beck*
Princeton University
cbeck@princeton.edu

Russell Impagliazzo†
Institute for Advanced Study
russell@cs.ucsd.edu

Shachar Lovett‡
Institute for Advanced Study
slovett@math.ias.edu

August 14, 2012

Abstract

There has been considerable interest lately in the complexity of distributions. Recently, Lovett and Viola (CCC 2011) showed that the statistical distance between a uniform distribution over a good code, and any distribution which can be efficiently sampled by a small bounded-depth $\mathbf{AC^0}$ circuit, is inverse-polynomially close to one. That is, such distributions are very far from each other. We strengthen their result, and show that the distance is in fact exponentially close to one. This allows us to strengthen the parameters in their application for data structure lower bounds for succinct data structures for codes.

From a technical point of view, we develop new large deviation bounds for functions computed by small depth decision trees, which we then apply to obtain bounds for $\mathbf{AC^0}$ circuits via the switching lemma. We show that if such functions are Lipschitz on average in a certain sense, then they are in fact Lipschitz almost everywhere. This type of result falls into the extensive line of research which studies large deviation bounds for the sum of random variables, where while not independent, exhibit large deviation bounds similar to these obtained by independent random variables.

1 Introduction

Perhaps the earliest use of randomized (Monte Carlo) methods in algorithms was not to solve decision problems, but to sample from distributions as a simulation. The complexity

*Research supported by NSF grants CCF-0832797, CCF-1117309.
†Research supported by NSF grants DMS-0835373, CCF-0832797, and The Oswald Veblen Fund.
‡Research supported by NSF grant DMS-0835373.

theory of randomized sampling algorithms was introduced by Jerrum, Valiant and Vazirani [JVV86], and there have been a huge number of algorithmic results on sampling, especially via the Monte Carlo Markov Chain method [JS89]. However, the first lower bounds on the complexity of sampling have been relatively recent. Explicitly, the challenge of exhibiting a distribution which cannot be efficiently sampled was raised by Goldreich, Goldwasser and Nussboim [GGN10] and by Viola [Vio10]. Such a distribution was given recently by Lovett and Viola [LV11], who showed that the uniform distribution over good codes cannot be sampled, or even approximately sampled, by bounded depth circuits (i.e. $\mathbf{AC^0}$ circuits). Our work was motivated by improving the parameters obtained by [LV11], but has led us to discover certain large deviation bounds which hold for bounded depth circuits and for decision trees, which we believe should have other applications.

Let us start by describing the result of [LV11]. In the following, an (n, k, d)-code is a subset $C \subset \{0,1\}^n$ of size $|C| = 2^k$, such that the hamming distance between any two distinct codewords is at least d. A code is called *good* if $k, d = \Omega(n)$. A distribution D over $\{0,1\}^n$ is said to be *sampled* by an $\mathbf{AC^0}$ circuit of depth d and size s, if there exists a function $F : \{0,1\}^m \to \{0,1\}^n$ for some m, computed by an $\mathbf{AC^0}$ circuit of depth d and size s, such that D is the output distribution of F given uniform input. One may think of such distributions as distributions which can be sampled efficiently *in parallel* given access to truly uniform bits.

Theorem 1.1 ([LV11])**.** The statistical distance between the uniform distribution over a good code $C \subset \{0,1\}^n$ and any distribution sampled by an $\mathbf{AC^0}$ circuit of depth d and size $\exp(n^{O(1/d)})$ is at least $1 - n^{-\Omega(1)}$.

This result achieves the "correct" tradeoff between the size and depth of the circuit. However, a shortcoming of the parameters achieved is that the statistical distance between the distributions is only guaranteed to be inverse-polynomial close to 1, while in theory one could hope for it to be exponentially close to 1. This can be seen as the analog of correlation bounds in the world of distributions: statistical distance $1 - \varepsilon$ between distributions can be seen as the analog of two functions having correlation of at most ε. In this work, we improve the statistical distance guarantee to indeed be exponentially close to 1.

Theorem 1.2 (This work)**.** The statistical distance between the uniform distribution over a good code $C \subset \{0,1\}^n$ and any distribution sampled by an $\mathbf{AC^0}$ circuit of depth d and size $\exp(n^{O(1/d)})$ is at least $1 - \exp(-n^{\Omega(1/d)})$.

Applications to data structures One application of [LV11] to the sampling lower bounds they obtained, is a corollary which shows that data structures for codes, which allow to compute the codewords given their internal storage efficiently, must have some redundancy in their internal storage.

Corollary 1.3 ([LV11])**.** Let C be an (n, k, d) code with $kd \geq n^{1+\epsilon}$. Suppose we can store codewords of C using only $k + r$ bits so that each bit of the codeword can be computed by an $\mathbf{AC^0}$ circuit of depth $O(1)$ and size $\text{poly}(n)$. Then $r \geq \Omega(\log n)$.

Plugging in our improved bound on the statistical distance, we obtain the following improvement.

Corollary 1.4 (This work)**.** Let C be a good code of size $|C| = 2^k$. Suppose we can store codewords of C using only $k + r$ bits so that each bit of the codeword can be computed by an $\mathbf{AC^0}$ circuit of depth d and size $\exp(n^{O(1/d)})$. Then $r \geq n^{\Omega(1/d)}$.

We note that the bound of [LV11] holds for codes for which $dk \geq n^{1+\epsilon}$, while our bounds as stated hold only for good codes, i.e. codes for which $k, d \geq \Omega(n)$. A careful examination of our proof shows that the proof can be extended to the case of $d^4k^5 \geq n^{8+\epsilon}$. We leave it as an open problem whether our result can be extended to the case of $dk \geq n^{1+\epsilon}$.

New tools We next describe the new tools we develop in this work. The proof of [LV11] was based on analyzing the effect of noise on the circuit which samples the distribution. Let $F : \{0,1\}^m \to \{0,1\}^n$ be a function computed by a small $\mathbf{AC^0}$-circuit whose output distribution is somewhat close to the uniform distribution over a code. Let $x \in \{0,1\}^m$ be a uniform input, and $y \in \{0,1\}^m$ be a correlated input chosen so $\Pr[x_i = y_i] = 1 - p$ leaving the noise rate p as a parameter. The idea is to bound the probability that $F(x), F(y)$ are two distinct codewords, in two different ways. On the one hand, if F is somewhat successful in sampling a code, then this probability will be noticeably large, simply because of the expansion of subsets of the noisy hypercube, and without consideration of the complexity of computing F. (Lovett and Viola argue using hypercontractivity estimates, and for reasonable settings of noise and code parameters, the argument shows that the probability to get distinct codewords is comparable to one minus the statistical distance.) On the other hand, if each output bit F_i of F is computed by a small $\mathbf{AC^0}$-circuit, then by the noise sensitivity results of [LMN93], each output bit has low noise sensitivity.

$$\Pr[F_i(x) \neq F_i(y)] \leq p \cdot \log^{O(1)} n.$$

Using this and a Markov argument, we see that

$$\Pr[\mathrm{dist}(F(x), F(y)) \geq \Omega(n)] \leq p \cdot \log^{O(1)} n.$$

Hence we obtain an upper bound on the probability to obtain two distinct codewords, and thus the overlap of the distribution with a code, as desired.

To improve on the analysis, we focus on this last step where there seems to be the most slack. Suppose we could show that the output bits of F are not only noise insensitive, but also uncorrelated in their response to noise, so that the probability that t bits flip in response to noise falls off exponentially in t. Then we would have

$$\Pr[\mathrm{dist}(F(x), F(y)) \geq \Omega(n)] \leq \exp(-n^{\Omega(1)}).$$

This would imply the desired exponential improvement in the statistical distance bound. However, in general this need not be the case. For instance each of the F_i might be identical, or they could all depend on a tiny subset of the variables and hence be highly correlated.

In general if there is even one influential variable, then the noise response will be large noticeably often.

We show that in the weaker model where each bit of F is computed by a small decision tree, this is the only way that concentration can fail. If there are no influential variables, we show that an inequality as above holds, and so by the previous considerations, *balanced* collections of decision trees, collections with no influential variables, cannot sample a good code with better than exponentially small overlap.

Then, we essentially reduce the general case to this case. First, we use random restrictions and the Håstad Switching Lemma [Hås86] (which underlies [LMN93]) to show that if an $\mathbf{AC^0}$-circuit samples a good code well, then a collection of small decision trees samples a fairly good code fairly well. Then, we similarly reduce general decision trees to balanced decision trees – the idea is that a small decision tree can only have a small number of influential variables, so we can restrict each of these randomly until none are left and obtain a balanced tree.

The reduction argument can also be understood roughly as follows. Since balanced decision trees satisfy concentration, they can only sample distributions which place significant weight on at most one codeword of any code. On the other hand, a general decision tree of height say n^{ε} will have at most $\approx n^{\varepsilon}$ influential variables, so it will sample a convex combination of $2^{n^{\epsilon}}$ distributions sampled by balanced decision trees. Thus it can only place significant weight on at most $2^{n^{\epsilon}}$ codewords, far less than $2^{O(n)}$, and cannot be sampling a good code.

The main technical step in our work is the large deviation bound for balanced collections of decision trees, which we discuss and prove in Section 2. The two reduction lemmas are proved in Section 3, where we also prove the main theorem.

2 Large Deviation Bound for Decision Forests

One of the most common mistakes in reasoning about probabilities is to identify a random variable with its expectation. Fortunately, in many circumstances, there are large deviation bounds that tell us that the variable is approximately its expectation with high probability. Examples of such large deviation bounds are the Chernoff-Hoeffding bounds for sums of independent Boolean variables, Azuma's inequality for Martingales of bounded difference, Talagrand's inequality, and the Kim-Vu inequality for low degree polynomials. For a discussion, see [ASE92].

We show a similar concentration bound for the sum of Boolean variables that are computed as relatively small height decision trees over a common set of variables. As far as we are aware, there is no previous work giving such a bound. The Kim-Vu bound [KV00] is closest to our situation, since decision trees of height h can also be written as polynomials of degree h. However, their bound, while useful for a host of combinatorial applications, deteriorates sharply in the degree, and so does not seem useful when the height is greater than logarithmic in the number of bits output.

One reason why there may be no previous concentration bounds is that, in general, such

bounds are false. The decision trees could all be identical, or more generally, all depend on a small set of variables and so be highly correlated. What we show is that, essentially, when these pathological cases are ruled out, concentration around the expectation holds.

We state our result more precisely below:

A *decision tree* is a binary tree whose leaves are labeled with values $\{0,1\}$ and whose internal nodes are labeled with Boolean variables $x_1, \ldots, x_m$. Given an input assignment of $\{0,1\}$ to the variables $x_1, \ldots, x_m$, a path is determined from the root to one of the leaves by identifying 0 with left and 1 with right and moving from each node labeled x_i to the child as indicated by the value of the assignment on x_i. The decision tree is then said to compute the value corresponding to this leaf on that input. If the path passes through a node labeled x_i, then x_i is said to be queried on this input. A decision tree queries a variable at most once on a path. The *height* of the decision tree is the height of the underlying binary tree.

A *decision forest* is a collection of decision trees. Given a forest $\mathcal{F}$ of n trees reading variables $x_1, \ldots, x_m$, it computes the function $\mathcal{F} : \{0,1\}^m \to \{0,1\}^n$ whose i'th bit is the function computed by the i'th tree.

Definition 2.1. For a decision forest $\mathcal{F}$ and an input $\vec{x}$, a Boolean variable x_i has *significance* α if an α fraction of trees query x_i on input $\vec{x}$. We notate this

$$\mathrm{sig}_{\mathcal{F}}(\vec{x}, x_i) = \alpha \ .$$

The *average significance* of a variable x_i with respect to $\mathcal{F}$ is the expected significance of x_i for a uniformly random assignment $\vec{x}$, notated $\mathrm{sig}_{\mathcal{F}}(x_i)$.

Significance seems very related to the influence of variables. The influence of a variable on a boolean function is the probability that changing that variable changes the function value, starting from a random input. For functions computing multiple bits, a natural generalization is the expected fraction of outputs bits which flip. However, whereas influence is a blackbox definition depending only on the function computed by $\mathcal{F}$, significance is a "whitebox" definition and may be different for two forests even if they compute the same function. The significance of a variable upper bounds its influence – if a decision tree does not query variable x_i on some input $\vec{x}$, then flipping x_i cannot change the output value. Intuitively, the stronger assumption of bounded average significances rather than bounded influences permits us to show that on a random input, the paths followed in each of the trees of $\mathcal{F}$ are "decoupled" and behave mostly independently of one another.

Definition 2.2. For $\vec{x}$ a string in $\{0,1\}^n$, $W(\vec{x})$ is the number of ones, i.e., the hamming weight of $\vec{x}$ and $w(\vec{x})$ is the fractional hamming weight, $W(\vec{x})/n$.

Theorem 2.3. Let $\mathcal{F}$ be a decision forest of height at most h and with all average significances at most β. Then, for any $\epsilon > 0$,

$$\Pr_{\vec{x}}\left[w(\mathcal{F}(\vec{x})) \geq O\left(\mathop{\mathrm{E}}_{\vec{x}}[w(\mathcal{F}(\vec{x}))] + h\sqrt{\beta \log(h^4/\epsilon)}\right)\right] \leq \epsilon \ .$$

While this result already has several interesting applications and is reminiscent of the polynomial setting, it is quite different from the Kim and Vu result in several important ways. In applications, the value β might be on the order of $n^{-\delta}$ where n is the number of trees, so in such cases h can also be polynomially large in n while still giving a strong bound on deviations. It is also interesting that the bound we obtain does not depend explicitly on the number of input variables, a fact which is convenient for us later. On the other hand, we will mainly apply our bound to situations where the expectation is comparable to β, where it is not a true concentration bound, since the error term will be much larger than the expectation.

The theorem has the following important corollary which we will use for our primary applications:

Corollary 2.4. Let $\mathcal{F}$ be a decision forest of height at most h, with all average significances at most β. For any $\epsilon > 0$,

$$\Pr_{\vec{x}}\left[\max_i \mathrm{sig}_{\mathcal{F}}(\vec{x}, x_i) \geq O\left(h\sqrt{\beta \log(2h^5/\beta\epsilon)}\right)\right] \leq \epsilon \ .$$

Thus, if $\mathcal{F}$ is small height and with all significances small on average, then in a strong quantitative sense $\mathcal{F}$ has all significances small almost always. Loosely, if such an $\mathcal{F}$ is "Lipschitz on average", usually a relatively benign condition, $\mathcal{F}$ is automatically "Lipschitz almost everywhere" for an appropriate small Lipschitz constant. This relatively strong condition permits further analysis to take place via the well-known tools described earlier. This automatic boosting of a Lipschitz on average condition to a Lipschitz almost everywhere condition is a rare and interesting feature of our work.

Now we will prove Theorem 2.3 and Corollary 2.4. To build intuition, we first prove a special case of the theorem where the sets of variables queried at different heights of the trees are disjoint. Then we show the general case by in essence reducing to the special case.

2.1 Special Case

First we need some preliminaries. Say that a node is at height i in a tree if it is i steps from the root. The i'th layer is the set of nodes at height i. The way in which we will use bounds on average significance is to bound the number of times a variable is queried in any given layer. Generally we will speak of the leaves of a decision tree as the "bottom" of the tree.

Observation 2.5. For any forest $\mathcal{F}$,

$$\mathrm{sig}_{\mathcal{F}}(x_i) = \frac{1}{|\mathcal{F}|} \cdot \sum_{j=0}^{h} 2^{-j} \cdot \{\#\text{ nodes at height } j \text{ querying } x_i\} \ .$$

Proposition 2.6. Let $\mathcal{F}$ be a decision forest of height at most h, with average significances at most β, and with expected hamming weight of an output at most α. Suppose further that no input variable occurs at multiple layers in the forest. Then,

$$\Pr_{\vec{x}}\left[w(\mathcal{F}(\vec{x})) \geq \mathop{\mathrm{E}}_{\vec{x}}[w(\mathcal{F}(\vec{x}))] + h\sqrt{\frac{\beta}{2}\log\frac{h}{\epsilon}}\right] \leq \epsilon \ .$$

The idea of the proof is to reveal the input variables one layer at a time, starting at the bottom. Suppose we choose values just for the inputs corresponding to the bottom layer. Since these inputs aren't queried anywhere else, the upper portion of the tree remains the same. The bottom layer nodes simplify, since the variable they query has now been assigned, and become the new leaves. Thus the decision forest becomes one smaller in height each time we reveal a layer in this manner. We track how the expected hamming weight of an input changes as we reveal all the layers one by one, and show that at each step it is unlikely to increase by much. In analyzing this, we are really only thinking about trees of height 1; the following lemma encapsulates our reasoning here, which is just a simple application of Hoeffding's inequality:

Fact 2.7 (Hoeffding's Inequality)**.** Let $X_1, \ldots, X_n$ be independent random variables, such that for each i, $X_i \in [a_i, b_i]$. Then

$$\Pr\left[\sum_i X_i - \mathrm{E}\left[\sum_i X_i\right] > \delta\right] \leq \exp\left(-\frac{2\delta^2}{\sum_i (b_i - a_i)^2}\right) .$$

Lemma 2.8. Let $\mathcal{F}$ be a decision forest of height 1, with expected weight α and average significances at most β. For any $\delta > 0$,

$$\Pr_{\vec{x}}\left[w(\mathcal{F}(\vec{x})) \geq \mathrm{E}_{\vec{x}}[w(\mathcal{F}(\vec{x}))] + \delta\right] \leq \exp\left(-\frac{2\delta^2}{\beta}\right) .$$

Proof. Let X denote the fractional weight of $\mathcal{F}(\vec{x})$. Write

$$X = X_0 + \sum_{i : x_i \text{ is a variable}} X_i ,$$

for the contributions to X of trees querying a particular variable x_i and constant trees. Thus X_0 is a constant and the X_i's are independent. Each X_i is at most $\mathrm{sig}_{\mathcal{F}}(x_i)$ and all are at least 0, so Hoeffding's inequality gives the bound

$$\Pr_{\vec{x}}\left[w(\mathcal{F}(\vec{x})) - \mathrm{E}_{\vec{x}}[w(\mathcal{F}(\vec{x}))] \geq \delta\right] < \exp\left(-\frac{2\delta^2}{\sum_i \mathrm{sig}_{\mathcal{F}}(x_i)^2}\right) .$$

We bound the sum of the squares of the significances by the sum of the significances times the maximum significance, the former being at most 1 and the latter being at most β by assumption. This finishes the proof. □

We can apply this argument recursively to prove Proposition 2.6.

Proof of Proposition 2.6. For simplicity throughout this argument we will assume that all trees are complete trees of height h, that is, no query path terminates early. This can be achieved by padding the trees with dummy queries without changing anything important; we think of these dummy queries as all being answered randomly and independent of one another and the input. It is convenient to do this because when the trees are complete trees,

the expected fractional hamming weight of an output is exactly equal to the fraction of leaves which are labeled 1.

We apply Lemma 2.8 h times in succession, each time to the bottom layer of the current forest; by revealing this bottom layer, the forest becomes a forest of complete trees one height smaller, and the average significances of the variables don't increase. Since the expected fractional hamming weight of a forest is the fractional weight of the leaves, Lemma 2.8 bounds exactly the fractional weight of the leaves of the reduced tree. Additionally, it is easy to see using Observation 2.5 that these bottom level decision trees also have average significances at most β. If we apply the bound with the same value of δ each time, we obtain

$$\Pr_{\vec{x}}\left[w(\mathcal{F}(\vec{x})) > \mathop{\mathrm{E}}_{\vec{x}}[w(\mathcal{F}(\vec{x}))] + h\delta\right] \leq h\exp\left(-\frac{2\delta^2}{\beta}\right) .$$

or, rewriting in terms of error probability,

$$\Pr_{\vec{x}}\left[w(\mathcal{F}(\vec{x})) > \mathop{\mathrm{E}}_{\vec{x}}[w(\mathcal{F}(\vec{x}))] + h\sqrt{\frac{\beta}{2}\log\frac{h}{\epsilon}}\right] \leq \epsilon .$$

□

2.2 General Case

In the general case, the plan is again to prove the result by induction on the height. We fix the following notation for two operations on decision forests. Here $\mathcal{F}$ is a decision forest of n trees of height h.

Definition 2.9 (truncating)**.** $\mathcal{F}'$ is the forest of $2n$ subtrees rooted at the immediate children of the roots of the trees of $\mathcal{F}$, thus $\mathcal{F}'$ has height $h-1$. If one of the trees of $\mathcal{F}$ is a constant, then corresponding to it $\mathcal{F}'$ will have 2 copies of this constant tree.

Definition 2.10 (pruning)**.** For $\mathcal{P}$ a partition of the variables, $\mathcal{F}_{\mathcal{P}}$ is the pruned forest which never reads a variable in any tree which is in the same class as the root variable in that tree. That is, for each tree in $\mathcal{F}$ if the variable read at the root is in part $\mathcal{P}_i$, then the corresponding tree in $\mathcal{F}_{\mathcal{P}}$ has all non-root nodes labeled with variables from $\mathcal{P}_i$ deleted and instead replaced with leaves assigning the value 0.

The idea is to repeatedly prune and truncate the forest, using standard techniques to ensure that with high probability, a large deviation can only occur in the original forest when it occurs in the pruned and truncated forest. When the height is small we don't lose much by iterating this. Very important to this strategy is the observation that pruning and truncating never increase the significance of a variable.

The proof of the inductive hypothesis will have two steps. First we prune $\mathcal{F}$ using a random partition $\mathcal{P}$ of the variables into h^3 parts. Via an averaging argument, we can select a partition such that if $\mathcal{F}$ has significant probability for a large deviation, then $\mathcal{F}_{\mathcal{P}}$ also

has similar probability for a similarly large deviation. We observe that since we were only pruning nodes, $\mathcal{F}_\mathcal{P}$ has small average significances if $\mathcal{F}$ does.

Second, we consider each part of the pruned forest $\mathcal{F}_\mathcal{P}$ and analyze as we did in the special case. We show that no part is likely to deviate much from the corresponding part of the truncated forest $(\mathcal{F}_\mathcal{P})'$. Aggregating across the different parts, we conclude $\mathcal{F}_\mathcal{P}$ rarely deviates much from $(\mathcal{F}_\mathcal{P})'$, which is controlled by the inductive assumption.

2.2.1 Choosing a Partition

Lemma 2.11. For any height h forest $\mathcal{F}$, if $\Pr_{\vec{x}}[w(\mathcal{F}(\vec{x})) > \alpha] > \epsilon$, then for some partition $\mathcal{P}$ of the variables into h^3 parts, $\Pr_{\vec{x}}[w(\mathcal{F}_\mathcal{P}(\vec{x})) > \alpha(1-h^{-1})] > \epsilon(1-h^{-1})$.

Proof. The proof is an averaging argument. Let $\mathcal{P}$ be a random coloring of the input variables with h^3 colors. Fix $\vec{x}$ and consider what fraction of ones of $\mathcal{F}(\vec{x})$ are pruned by $\mathcal{P}$, that is, are zeros of $\mathcal{F}_\mathcal{P}(\vec{x})$. A particular one is pruned if in the corresponding tree, one of the nonroot variables queried on $\vec{x}$'s path is colored the same as the root variable. There are at most h variables on a path, so by a union bound the probability that it is pruned is at most h^{-2}.

Let $S := \{\vec{x} : w(\mathcal{F}(\vec{x})) > \alpha\}$ be the set of inputs resulting in high hamming weight. By assumption S has measure exceeding ϵ.

Now, choose $\vec{x}$ randomly from S and a random $\mathcal{P}$. In expectation, at most a fraction h^{-2} of the ones are pruned by $\mathcal{P}$, so by averaging there is a fixed choice of $\mathcal{P}$ for which this holds. By Markov's inequality, the probability that more than h^{-1} are pruned by $\mathcal{P}$ from a random element of S is at most h^{-1}. Thus, $\mathcal{F}_\mathcal{P}$ has fractional weight exceeding $(1-h^{-1})\alpha$ with probability exceeding $\epsilon(1-h^{-1})$, as desired. □

Since we are only going to perform pruning h times, we are over all only going to lose multiplicative factors of $(1-h^{-1})^h = O(1)$ in the probability and magnitude of the deviation overall to these steps.

2.2.2 Bounding deviations in $\mathcal{F}_\mathcal{P}$

Lemma 2.12. Fix a forest $\mathcal{F}_\mathcal{P}$ of n trees of height h, with $\mathcal{P}$ a partition having h^3 parts. Then, for any $\delta > 0$,

$$\Pr_{\vec{x}}\left[w(\mathcal{F}_\mathcal{P}(\vec{x})) - w(\mathcal{F}'_\mathcal{P}(\vec{x})) > \delta\right] \leq h^3 \exp\left(\frac{2\delta^2}{\beta}\right) .$$

Proof. The idea of the proof is to break the forest into parts according to $\mathcal{P}$, and bound the growth of each separately using Hoeffding's inequality as before. Note that we can safely ignore any constant trees in $\mathcal{F}_\mathcal{P}$, as before.

Let p be any part of $\mathcal{P}$, and let $\mathcal{F}_p$ denote the set of trees of $\mathcal{F}$ whose root is in p. For any fixed assignment to the variables outside p, $\mathcal{F}_p$ becomes a forest of height 1, with variables of p at the root. No variable occurs more than βn times, or else it has significance exceeding

β in the overall forest. Applying Hoeffding's bound essentially as we did in Lemma 2.8, we have that for any $\delta > 0$,

$$\begin{aligned}\Pr_{\vec{x}}\left[W(\mathcal{F}_p(\vec{x})) - \mathop{\mathrm{E}}_{\vec{x}}[W(\mathcal{F}_p(\vec{x}))] > \delta|\mathcal{F}_p|\right] &\leq \exp\left(-\frac{2(\delta|\mathcal{F}_p|)^2}{\sum_i \left(|\mathcal{F}_p|\mathrm{sig}_{\mathcal{F}_p}(x_i)\right)^2}\right) \\ &\leq \exp\left(-\frac{2\delta^2|\mathcal{F}_p|^2}{\left(\sum_i |\mathcal{F}_p|\mathrm{sig}_{\mathcal{F}_p}(x_i)\right)\cdot\left(\max_i |\mathcal{F}_p|\mathrm{sig}_{\mathcal{F}_p}(x_i)\right)}\right) \\ &\leq \exp\left(-\frac{2\delta^2}{\beta}\right) .\end{aligned}$$

We saw before that the expected fractional hamming weight of a height one forest (in which every root makes a query) is the fractional hamming weight of the leaves, and the string appearing at the leaves is computed by the truncated forest $\mathcal{F}'_p$, so this gives

$$\Pr_{\vec{x}}[w(\mathcal{F}_p(\vec{x})) - w(\mathcal{F}'_p(\vec{x})) > \delta] \leq \exp\left(-\frac{2\delta^2}{\beta}\right) .$$

This bound holds for any fixed value of the variables outside p, so it holds when these are chosen randomly as well. The sum of the hamming weight for each part $\mathcal{F}_p$ is the hamming weight for $\mathcal{F}_\mathcal{P}$, and the sum of the hamming weight for each part $\mathcal{F}'_p$ is the hamming weight for $\mathcal{F}'_\mathcal{P}$, so by a union bound over each $p \in \mathcal{P}$,

$$\Pr_{\vec{x}}\left[w(\mathcal{F}_\mathcal{P}(\vec{x})) - w(\mathcal{F}'_\mathcal{P}(\vec{x})) > \delta\right] \leq h^3 \exp\left(-\frac{2\delta^2}{\beta}\right) ,$$

as desired. □

2.2.3 Putting the pieces together

Now we prove Theorem 2.3.

Proof of Theorem 2.3. Following the sketch earlier, the proof is by induction on the height. Fix some α, β later.

Suppose that $\mathcal{F}$ is of height h with $\mathrm{E}_{\vec{x}}[w(\mathcal{F}(\vec{x}))] \leq \alpha$, $\max_i \mathrm{sig}_\mathcal{F}(x_i) \leq \beta$, and $\Pr_{\vec{x}}[w(\mathcal{F}(\vec{x})) > \alpha_h] > \epsilon_h$. By Lemma 2.11, there is a partition $\mathcal{P}$ with h^3 parts so that $\Pr_{\vec{x}}[w(\mathcal{F}_\mathcal{P}(\vec{x})) > (1-h^{-1})\alpha_h] > (1-h^{-1})\epsilon_h$, where $\mathcal{F}_\mathcal{P}$ the same or smaller expected fractional weight and average significances. Fix some $\delta > 0$ later; by Lemma 2.12, and a union bound, the probability that $\mathcal{F}'_\mathcal{P}(\vec{x})$ has fractional weight exceeding $(1-h^{-1})\alpha_h - \delta$ is at least $(1-h^{-1})\epsilon_h - h^3 \exp\left(-\frac{2\delta^2}{\beta}\right)$. Let $\alpha_{h-1} = (1-h^{-1})\alpha_h - \delta, \epsilon_{h-1} = (1-h^{-1})\epsilon_h - h^3 \exp\left(-\frac{2\delta^2}{\beta}\right)$, and apply this recursively to $\mathcal{F}'_\mathcal{P}$.

When the height is reduced to one, Lemma 2.8 bounds α_1 as at most $\alpha + \delta$ and ϵ_1 as at most $\exp\left(-\frac{2\delta^2}{\beta}\right)$, so we deduce a constraint on α_h, ϵ_h. For convenience, we use the same

value of δ at every step and also the same value of h when dividing into partitions. When we unfold the depth h recursion above, an additive term may be multiplied by as many as h factors of $(1-h^{-1})^{-1}$, however as noted earlier $(1-h^{-1})^{-h} = \Theta(1)$, so up to $O(1)$ factors

$$\begin{aligned}\alpha_h &\leq O(\alpha + h\delta) \ , \\ \epsilon_h &\leq O\left(h^4 \exp\left(-\frac{2\delta^2}{\beta}\right)\right) \ .\end{aligned}$$

Let $\mathcal{F}$ be any forest of height h and take $\beta = \max_i \mathrm{sig}_{\mathcal{F}}(x_i), \alpha = \mathrm{E}_{\vec{x}}[w(\mathcal{F}(\vec{x}))]$ and apply the result. Writing δ in terms of our final ϵ, we have

$$\Pr_{\vec{x}}\left[w(\mathcal{F}(\vec{x})) \geq O\left(\mathrm{E}_{\vec{x}}[w(\mathcal{F}(\vec{x}))] + h\sqrt{\beta \log(h^4/\epsilon)}\right)\right] \leq \epsilon \ ,$$

as desired. □

2.3 Average Lipschitz to Lipschitz Almost Everywhere

Here we give the proof of Corollary 2.4.

Proof of Corollary 2.4. For any decision tree $\mathcal{T}$ of height h and any variable x_i, the function $\vec{x} \mapsto \mathrm{sig}_{\mathcal{T}}(\vec{x}, x_i)$ can be computed by a tree of height h by relabeling the leaves of $\mathcal{T}$. If $\mathcal{F}$ is a forest with average significances at most β, then relabeling the leaves of each tree of $\mathcal{F}$ this way produces a forest such that $w(\mathcal{F}(\vec{x})) = \mathrm{sig}_{\mathcal{F}}(\vec{x}, x_i)$. By assumption, the expected fractional weight of this forest is at most β, so Theorem 2.3 bounds the probability that $\mathrm{sig}_{\mathcal{F}}(\vec{x}, x_i)$ is large. Suppose there are n trees in $\mathcal{F}$. A union bound over all $n2^h$ possible variables queried by $\mathcal{F}$ yields

$$\Pr_{\vec{x}}\left[\max_i \mathrm{sig}_{\mathcal{F}}(\vec{x}, x_i) \geq O\left(h\sqrt{\beta \log(h^4 n 2^h/\epsilon)}\right)\right] \leq \epsilon \ ,$$

which while good enough for some applications, is somewhat wasteful.

To do better, first cluster the variables greedily into clusters such that the sum of the average significances of the variables is between $\beta/2$ and β. Since on any input at most h variables are queried, the sum of the average significances of all variables is at most h, so we obtain at most $2h\beta^{-1}$ clusters. For each cluster $\mathcal{C}$, relabel the leaves of each tree $\mathcal{T} \in \mathcal{F}$ so that it computes the indicator $\bigvee_{x_i \in \mathcal{C}} \mathrm{sig}_{\mathcal{T}}(\vec{x}, x_i)$. The expected fractional weight is again at most β, so we can apply Theorem 2.3 to bound the probability that on any input, many trees query a variable from $\mathcal{C}$, which also bounds the probability that many trees query any particular variable of $\mathcal{C}$. A union bound over all clusters implies

$$\Pr_{\vec{x}}\left[\max_i \mathrm{sig}_{\mathcal{F}}(\vec{x}, x_i) \geq O\left(h\sqrt{\beta \log(2h^5/\beta\epsilon)}\right)\right] \leq \epsilon \ .$$

□

3 A Lower Bound for Sampling by $\mathbf{AC^0}$-circuits

Recall that an $\mathbf{AC^0}$-circuit family is a sequence of boolean circuit using $\wedge, \vee$ gates of unbounded fan-in and $\neg$ gates, such that the depth is bounded as a constant.

Lovett and Viola [LV11] showed that even exponentially large $\mathbf{AC^0}$-circuits cannot approximate uniform distributions over good error correcting codes, where approximation is measured by the statistical distance between the distributions. For two distributions D, D',

$$\mathrm{sd}(D, D') := \max_S \left| \Pr_D[S] - \Pr_{D'}[S] \right| .$$

Let U_n denote the uniform distribution on $\{0,1\}^n$, and for $\mathcal{C}$ a subset of $\{0,1\}^n$ let $U_{\mathcal{C}}$ denote the uniform distribution on $\mathcal{C}$.

Just for convenience, we will say that the statistical distance between a function $F : \{0,1\}^m \to \{0,1\}^n$ and a set $\mathcal{C}$ is the statistical distance between $F(U_m)$ and $U_{\mathcal{C}}$.

Here we formally restate Theorems 1.1, 1.2.

Theorem 3.1 ([LV11], main result)**.** Let $F : \{0,1\}^m \to \{0,1\}^n$ be a function computable by an $\mathbf{AC^0}$ circuit of size S and depth d. For any good code $\mathcal{C}$,

$$\mathrm{sd}(F, \mathcal{C}) \geq 1 - O\left(n^{-1} \log^{d-1} S\right)^{1/3} .$$

Theorem 3.2 (This work)**.** Let $\epsilon = \min(\frac{1}{5d+17}, \frac{4}{6d+5})$. Let $F : \{0,1\}^m \to \{0,1\}^n$ be a function computable by an $\mathbf{AC^0}$-circuit of depth d and size $2^{O(n^\epsilon)}$. For any good code $\mathcal{C}$,

$$\mathrm{sd}(F, \mathcal{C}) \geq 1 - 4 \cdot 2^{-\Omega(n^\epsilon)} ,$$

the constants depending only on the quality of the code and d.

We begin with some preliminaries.

3.1 Results from previous work

The following lemma is an application of hypercontractivity, which we will use in several places. We won't use hypercontractivity except via this lemma. In this section, for S a subset of the hypercube $\{0,1\}^m$, $\mu(S)$ will denote the measure of S, $\mu(S) = |S|/2^m$.

Lemma 3.3 ([LV11], Lemma 6)**.** Let S be a subset of the hypercube $\{0,1\}^m$. Let x be a uniformly chosen point of the hypercube, and let y be chosen from the noise distribution μ_p in which each component is iid, and 1 is chosen with probability p. Then for any p,

$$(\mu(S))^{1+p} \geq \Pr_{x \in U_m, y \in \mu_p}[x \in S, x + y \in S] \geq (\mu(S))^2 ,$$

where $+$ is bitwise xor.

We also need to use the Håstad Switching Lemma. Let $x_1, \ldots, x_m$ be a set of boolean variables. A (boolean) restriction is a map ρ be a map $\{x_1, \ldots, x_m\} \to \{0, 1, \star\}$. If $\rho(x_i) = \star$, x_i is said to be *unset* by the restriction ρ, and otherwise x_i is *set* to the value $\rho(x_i)$. A *random restriction* with unset probability p is the distribution on restrictions where each variable is independently unset with probability p and otherwise independently set to a uniformly random value.

Proposition 3.4. [Håstad switching lemma [Hås86], (see also [Ajt83, FSS84, Yao85, Bea94])] Let C be a circuit on n variables of size S and depth d. For any h, let ρ be a random restriction with unset probability $p \leq \frac{1}{14} \cdot (14h)^{-d}$. Then each output of $C|_\rho$ is computed by a decision tree of height at most h, except with probability $S\left(\frac{1}{2}\right)^h$.

3.2 High Level Overview

We follow the same general strategy as [LV11]. They showed, using hypercontractivity, that if any function has significant overlap with the uniform distribution on a code $\mathcal{C}$, that there if we look at two correlated inputs x and $x' = x+y$ for noise vector y, there is a good probability that both map to codewords, and that these codewords are distinct. Since codewords are far in hamming distance, this means that the small perturbation on the inputs is responsible for a large perturbation on the outputs. Thus, with reasonable probability, the outputs have to be very sensitive to the inputs. Finally, they use the bound on the average sensitivity of $\mathbf{AC^0}$ functions by [LMN93] to get a contradiction.

It is this last step we improve. [LMN93] prove their bound on sensitivity by using the Håstad Switching Lemma, so we use the full Switching Lemma rather than just its consequence. Intuitively, this allows us to deal with decision trees rather than with formulas. If these decision trees are balanced, in that no variable has a high average significance, we can use our concentration bound to show that there is only an exponentially small probability of large sensitivity to an input. Then it follows that the probability that x and x' map to distinct codewords is exponentially small.

Unfortunately, we have no guarantee that the decision trees are balanced, even after the random restriction, and if they are not, there could well be two codewords so that half the time we map to one and the other half to the other. However, what we show is that this is essentially the only situation that can occur: once we fix a small number of inputs, we get a balanced family of decision trees. So while decision trees can compute maps that go to distinct codewords, the *number* of such codewords cannot be very large. Finally, we use hypercontractivity to show that a random restriction of a map that has a large overlap with a code also has a large intersection with a large subcode. This allows us to move from circuits to decision trees.

3.3 Measuring overlap with good sets

In going from circuits to decision trees, and decision trees to balanced decision trees, intuitively, we are also possibly moving from the uniform distribution on codewords to a distri-

bution on codewords with smaller entropy. It simplifies the argument to use the following parameters instead of keeping track explicitly of this distribution.

Definition 3.5 ("good set")**.** Let $F : \{0,1\}^m \to \{0,1\}^n$ and let $\mathcal{C}$ be a subset of $\{0,1\}^n$. We'll say that a subset $S \subseteq \{0,1\}^m$ is a (Δ, τ)-set for F with respect to $\mathcal{C}$ if

- $F(S) \subseteq \mathcal{C}$
- $\mu(S) \geq \Delta$
- For any $c \in \mathcal{C}$, $\mu(S \cap F^{-1}(c)) \leq \tau$.

The good set witnesses agreement of F with the code $\mathcal{C}$. As the below observation formalizes, when $\tau = 1/|\mathcal{C}|$, the maximum achievable Δ is exactly one minus the statistical distance. However, in our series of reductions, we will need to increase τ , intuitively moving to a distribution concentrated on a smaller subset of codewords. So our lemmas will consider not just the value $\tau = 1/|\mathcal{C}|$ that we need at the end, but the range of possible tradeoffs between τ and Δ.

Observation 3.6. A function $F : \{0,1\}^m \to \{0,1\}^n$ has statistical distance $\leq 1 - \Delta$ from $U_{\mathcal{C}}$ for some set $\mathcal{C} \subseteq \{0,1\}^n$ if and only if F has a $(\Delta, 1/|\mathcal{C}|)$-set with respect to $\mathcal{C}$.

Proof. Let $D_1(z)$ be the probability that F outputs z and $D_2(z)$ be the uniform distribution on codewords. Let $\max(z) = \max(D_1(z), D_2(z))$ and $\min(z)$ be the minimum. Let SD be the statistical distance between the two. Then $SD = 1/2 \sum_z (\max(z) - \min(z))$ and $1 = 1/2 \sum_z (\max(z) + \min(z))$ since both are probability distributions. Thus $1 - SD = \sum_z \min(z)$. $\min(z)$ is 0 unless it is a codeword, in which case it is the minimum of the fraction of preimages of z and $1/|\mathcal{C}|$. So for each codeword $z \in \mathcal{C}$ we can pick a $\min(z)$ fraction of preimages, and achieve $\Delta = 1 - SD$. No good set can have more than this number of preimages for any z, so this is the best achievable. □

In the sequel, the reader should generally think of τ as on the order of $2^{-\Omega(n^{1-\epsilon})}$, Δ on the order of $2^{n^{\varepsilon}}$, and h on the order of n^{ϵ}, where ϵ will be $1/O(d)$.

3.4 Bounds for balanced decision trees

Lemma 3.7 (Step 1)**.** For any forest of height h with all average significances at most β which has a (Δ, τ) set with respect to a good code,

$$\log \frac{1}{\Delta} = \Omega\left(h^{-4/3} \beta^{-1/3} \left(\frac{\log \frac{1}{\tau}}{n} \right)^{2/3} \right) ,$$

where the hidden constants depend only on the code.

As in [LV11], we use hypercontractivity to show that if Δ is too large, correlated inputs are likely to map to distinct codewords, which will contradict our concentration bound from Section 2.

Proof of Lemma 3.7. Let G be a (Δ, τ)-set of inputs for $\mathcal{F}$ as assumed.

Let x be a random vector, and y be chosen from the noise distribution μ_p. Applying Lemma 3.3 to G, we have

$$\Pr_{x,y}[x, x+y \in G] \geq \mu(G)^2 \ ,$$

and applying it to $G \cap \mathcal{F}^{-1}(c)$ for any $c \in \mathcal{C}$, we have

$$\Pr_{x,y}[x, x+y \in G \cap \mathcal{F}^{-1}(c)] \leq \mu(G \cap \mathcal{F}^{-1}(c))^{1+p} \ .$$

By a union bound,

$$\begin{aligned}
\Pr_{x,y}[x, x+y \in G, \mathcal{F}(x) \neq \mathcal{F}(x+y)] &\geq \mu(G)^2 - \sum_{c \in \mathcal{C}} \mu(G \cap \mathcal{F}^{-1}(c))^{1+p} \ , \\
&\geq \mu(G) \left(\mu(G) - \left(\max_c \mu(G \cap \mathcal{F}^{-1}(c)) \right)^p \right) \ , \\
&\geq \Delta \left(\Delta - \tau^p \right) \ ,
\end{aligned}$$

for any p. In particular,

$$\Pr_{x,y}[\mathrm{dist}(\mathcal{F}(x), \mathcal{F}(x+y)) = \Omega(n)] \geq (\Delta - \tau^p)^2 \ .$$

Each tree in $\mathcal{F}$ where the output differs between x and $x+y$ must have a bit i so that x_i is queried by the tree for assignment x and $y_i = 1$, because if not, the path that the tree follows with input x is still followed on input $x+y$. Suppose we fix some x such that $\mathrm{sig}_{\mathcal{F}}(x, x_i) \leq \gamma$ for all i. Then, each x_i is queried in at most γn trees on input x and so if y flips it, it can be responsible for at most γn changes in outcome. In particular we see that for any D,

$$\begin{aligned}
\Pr_{x,y}[\mathrm{dist}(\mathcal{F}(x), \mathcal{F}(x+y)) \geq D] \leq &\Pr_x \left[\max_i \mathrm{sig}_{\mathcal{F}}(x, x_i) > \gamma \right] \\
&+ \Pr_{x,y} \left[y \text{ flips at least } \frac{D}{\gamma n} \text{ variables queried on input } x \right] \ .
\end{aligned}$$

So to get distinct codewords when x is such, at least $\Omega(1/\gamma)$ variables that are queried with input x need to be flipped by y. y is chosen independently of x, so if we prove a bound on this second probability for fixed x it holds for random x as well. There are at most nh variables total queried with input x, and since y flips each independently with probability p, the number of such variables is dominated by the binomial distribution $Bin(nh, p)$. For a value of $\gamma < 1$ to be chosen later, set $p = c/(nh\gamma)$, where c is at most $1/2$ the relative distance of the code, so that the expected number of such flips is at most c/γ. Applying standard Chernoff bounds we obtain:

$$\Pr_{x,y}[\mathrm{dist}(\mathcal{F}(x), \mathcal{F}(x+y)) = \Omega(n)] \leq \Pr_x \left[\max_i \mathrm{sig}_{\mathcal{F}}(x, x_i) > \gamma \right] + \exp\left(-\Omega(1/\gamma)\right) \ .$$

To bound the first probability, we do a change of variables in Corollary 2.4:

$$\Pr_x\left[\max_i \mathrm{sig}_{\mathcal{F}}(x,x_i) = \Omega\left(h\sqrt{\beta\log\frac{2h^5}{\beta\epsilon}}\right)\right] \le \epsilon \ ,$$

becomes

$$\Pr_x\left[\max_i \mathrm{sig}_{\mathcal{F}}(x,x_i) > \gamma\right] \le 2h^5\beta^{-1}\exp\left(-\Omega\left(\frac{\gamma^2}{h^2\beta}\right)\right) \ .$$

Thus we have overall

$$(\Delta-\tau^p)^2 \le 2h^5\beta^{-1}\exp\left(-\Omega\left(\frac{\gamma^2}{h^2\beta}\right)\right) + \exp\left(-\Omega\left(1/\gamma\right)\right) \ .$$

Taking log's and absorbing low order terms into the constants, we have

$$\log\frac{1}{\Delta} = \Omega\left(\min\left(\frac{1}{nh\gamma}\cdot\log\frac{1}{\tau} \ , \ \frac{\gamma^2}{h^2\beta} \ , \ \frac{1}{\gamma}\right)\right) \ .$$

As long as $\tau \ge 2^{-n}$ and $h \ge 1$, the last term is never the minimum, so we pick γ to balance the first two.

$$\gamma^3 = h\beta\frac{\log\frac{1}{\tau}}{n} \ .$$

This gives a bound on Δ of

$$\log\frac{1}{\Delta} = \Omega\left(h^{-4/3}\beta^{-1/3}\left(\frac{\log\frac{1}{\tau}}{n}\right)^{2/3}\right) \ ,$$

as claimed. □

(Note that, in particular, if $\tau = 1/|\mathcal{C}|$, then we get an exponentially small bound on Δ if $\beta < h^{-4}n^{-\epsilon}$.)

3.5 Making decision forests balanced

The following reduction lemma shows how we can construct balanced forests with a good set from arbitrary forests with a good set, at a small cost in parameters.

Lemma 3.8 (Step 2)**.** If there is an forest of height h with a (Δ,τ)-set for a good code $\mathcal{C}$, then for any ℓ,β with $\ell > 2h\beta^{-1}$, there is a forest of height h and all average significances at most β with a (Δ',τ')-set, where

$$\Delta' = \Delta - \exp\left(-\frac{\ell\beta^2}{8h^2}\right) \ ,$$
$$\tau' = 2^{\ell}\tau \ .$$

Together with Lemma 3.7, this gives us:

Corollary 3.9. [Step 2 Corollary] If there is a forest of height h with a (Δ, τ)-set for a good code $\mathcal{C}$, and $\log \frac{1}{\tau} = \Omega(n^{1/3}h^{5/6})$, then

$$\log \frac{1}{\Delta} = \Omega\left(\log^{5/7}\left(\frac{1}{\tau}\right) \cdot n^{-4/7}h^{-10/7}\right) ,$$

where the constants depend only on the quality of the code.

Proof of Corollary 3.9. Apply Lemma 3.8 to the forest in question, setting $\ell = \log \frac{1}{\tau}$, so that $\log \frac{1}{\tau} = \Theta\left(\log \frac{1}{\tau'}\right)$ and $\log \frac{1}{\Delta} = \Omega\left(\min(\log \frac{1}{\Delta'}, \beta^2 h^{-2} \log \frac{1}{\tau})\right)$. By Lemma 3.7 applied to the resulting forest,

$$\log \frac{1}{\Delta'} = \Omega\left(h^{-4/3}\beta^{-1/3}\left(\frac{\log \frac{1}{\tau}}{n}\right)^{2/3}\right) .$$

Setting $\beta = n^{-2/7}h^{2/7}\log^{-1/7}\frac{1}{\tau}$ balances the two terms. Overall this yields

$$\log \frac{1}{\Delta} = \Omega\left(\log^{5/7}\left(\frac{1}{\tau}\right) \cdot n^{-4/7}h^{-10/7}\right) ,$$

as claimed. The constraint on τ is equivalent to $\ell > 2h\beta^{-1}$, as required by Lemma 3.8. □

Note: This immediately gives an exponentially strong sampling lower bound for small height forests. Setting $\tau = 1/|\mathcal{C}| = 2^{-\Omega(n)}$, for $h \leq n^{1/20}$, we get $\Delta \leq 2^{-\Omega(n^{1/14})}$.

Proof of Lemma 3.8. We'll look at a process that fixes the high significance variables, until none are left, and show that few variables are fixed with high probability.

Claim 3.10. Let $\mathcal{F}$ be an arbitrary forest of height h. Suppose we play ℓ rounds of the following game with an adversary. Each round, the adversary identifies a variable x_i with $\mathrm{sig}_{\mathcal{F}}(x_i) \geq \beta$, then x_i is restricted randomly and $\mathcal{F}$ is simplified. If there are no variables for the adversary to identify, he loses.

Then for any adversary strategy, the probability that the adversary does not lose after ℓ rounds is at most $\exp\left(-\frac{\ell\beta^2}{8h^2}\right)$, provided $\ell > 2h\beta^{-1}$.

Proof of Claim 3.10. Consider A_j, the average number of variables queried in a tree, averaging over both random inputs and trees in the forest, after j rounds. Each round of the game, the expectation of A_{j+1} over the settings of the variable found is at most $A_j - \beta$. $A_0 \leq h$, and $|A_j - A_{j+1}| \leq h$, since the decision trees never query more than h variables in the worst-case. Thus, the random variables $A_j + \beta j$ form a supermartingale of bounded differences. The probability that $A_\ell > 0$ is the probability that $A_\ell + \beta\ell > \beta\ell \geq A_0 + (\beta\ell - h) \geq A_0 + \beta\ell/2$. Applying Azuma's inequality, this is at most $\exp(-(\beta\ell/2h)^2/(2\ell)) = \exp\left(-\frac{\ell\beta^2}{8h^2}\right)$. □

Now, the lemma follows from the claim. We claim that for one of the restrictions in the above game, the restricted good set is a (Δ', τ')-set for the corresponding restricted forest. The original volume of the good set is the average over all restrictions in the above game of the restricted volume. Even if this is 1 in all paths where the game exceeds ℓ steps, this would total the failure probability above. So there must be a restriction of at most ℓ variables in the above game where the restricted volume is at least the difference of the original volume and the failure probability. For this restriction, the forest has all significances at most β by construction. The *size* of any intersection of the good set with the preimage of any code word has not increased after the restriction, but since we restricted at most ℓ variables, its relative measure in the subcube corresponding to the restriction is larger by at most a factor of 2^ℓ. Thus the restricted good set is a (Δ', τ')-set for this forest as claimed. □

3.6 Lower bound for constant depth circuits

We reduce sampling bounds for constant depth circuits to that for decision trees by taking a random restriction. With very high probability, the circuits all become small depth decision trees. The main thing we need to show is that, with high probability, no one code word becomes too likely after the restriction. Here, we use hypercontractivity again. (This step, while developed independently, is similar to the idea of Lemma 1.7 in [Vio11].)

Lemma 3.11. Let $\mathcal{C}$ be a good code, and let F be a function with a $(\Delta, 1/|\mathcal{C}|)$ good set. Let ρ be a random restriction with probability p of leaving a variable unset.

Then with probability at least $\Delta/4$, $F|_\rho$ has a $(\Delta/2, 2|\mathcal{C}|^{-p/4})$-set.

Proof. For any codeword $c \in \mathcal{C}$, let S_c be the subset of the good set mapping to the codeword. Consider picking ρ and then two inputs x and x' consistent with ρ and otherwise random and independent. $x + x'$ is distributed as a random noise vector with probability $p/2$ of noise, since each bit position is unset with probability p, and they are equally likely to agree and disagree if it is unset. Then $\mathrm{E}_\rho[\mu(S_c|_\rho)^2] = \Pr_{\rho,x,x'}[x, x' \in S_c] \leq (\mu(S_c))^{1+p/2}$, by Lemma 3.3. By Markov's inequality, the probability that $\mu(S_c|_\rho) \geq 2|\mathcal{C}|^{-p/4}$ is at most $\mu(S_c)^{1+p/2} \cdot |\mathcal{C}|^{p/2}/4$. So the probability that there exists such a codeword c is at most

$$\left(\sum_c \mu(S_c)^{1+p/2} \right) |\mathcal{C}|^{p/2}/4 \leq \Delta/4 \ .$$

On the other hand, a simple averaging argument shows that with probability at least $\Delta/2$, the volume of the restricted good set is at least $\Delta/2$. So the probability over ρ that both the restricted good set has size $\Delta/2$ and no codeword has probability greater than $2|\mathcal{C}|^{-p/4}$ is at least $\Delta/4$. □

Combining this with the switching lemma gives:

Lemma 3.12. Assume there is a size S depth d circuit family C computing a function with statistical distance $1 - \Delta$ from a good code $\mathcal{C}$. Let h be such that $|S|2^{-h} < \Delta/4$. Then there is a family of height h decision trees with a $(\Delta/2, 2|\mathcal{C}|^{-\frac{1}{4}(14h)^{-d}})$-set for $\mathcal{C}$.

Proof. For $p = (14h)^{-d}$, consider $C|_\rho$. By the previous lemma, the probability that $C|_\rho$ has a good set of the given size is at least $\Delta/4$. On the other hand, by the Switching Lemma, and the assumption on Δ, the fraction of ρ so that C_ρ is not computable by depth h decision trees is less than $\Delta/4$. So there exists a ρ so that $C|_\rho$ is equivalent to a family of height h decision trees, and has a good set of the claimed parameters. □

We can now prove our main theorem:

Proof of Theorem 3.2. Let c be a small constant determined later.
Let $h = c \cdot \min(n^{1/(5d+17)}, n^{4/(6d+5)})$, and assume C is a circuit family of size at most $S = 2^{h/2}$ with distance $1 - \Delta$ from a good code $\mathcal{C}$. If $\Delta < 4|S|2^{-h}$, we are done. Otherwise, by Lemma 3.12 there is a family of height h decision trees with a $(\Delta/4, \tau')$ good set, where $\tau' = |\mathcal{C}|^{-\frac{1}{4}(14h)^{-d}}$. Now apply Lemma 3.8. We meet the condition as long as we have $\log 1/\tau' = \Omega(n^{1/3}h^{5/6})$, which holds as long as $n^{2/3} = \Omega(h^{5/6+d})$,or $h = O(n^{4/(6d+5)})$, which is true for small enough c. Thus the lemma implies $\log 1/\Delta > \Omega\left((\log 1/\tau')^{5/7} n^{-4/7} h^{-10/7} \geq \Omega\left((n/(14h)^d\right)^{5/7} n^{-4/7} h^{-10/7} = \Omega(n^{1/7} h^{(-5d-10)/7})\right.$. Since we weren't done before, $\log 1/\Delta = O(h)$, so $h^{(5d+17)/7} = \Omega(n^{1/7})$, a contradiction if c is chosen small enough. □

4 Discussion and Open Questions

Up to constants, it seems unlikely that Theorem 3.2 can be improved without a major breakthrough in our understanding of $\mathbf{AC^0}$-circuits, since getting a size lower bound better than $2^{n^{\Omega(1/d)}}$, or getting improved correlation bounds, are longstanding open questions.

Open Question 1. Other applications for Theorem 2.3 and Corollary 2.4?

Open Question 2. Is something like Theorem 2.3 true under the weaker assumption of small average influences rather than small average significances? Do small AC^0 circuits with small average influences satisfy concentration?

Open Question 3. As mentioned earlier [LV11] gives a result for (n, k, d) codes with $dk = \Omega(n^{1+\epsilon})$. Although stated only for good codes, our proof generalizes to give an exponential improvement in the range $d^4k^5 = \Omega(n^{8+\epsilon})$. Can this improvement be obtained $dk = \Omega(n^{1+\epsilon})$?

A *circuit source* is a random string computed by a circuit whose input bits are uniformly random. Trevisan and Vadhan [TV00] pointed out that obtaining weak seedless extractors for circuit sources is equivalent to proving weak sampling lower bounds for those circuits.

Open Question 4. Viola [Vio11] gave a seedless extractor which yields, for any $\gamma > 0$, $k(k/n^{1+\gamma})^{O(1)}$ truly random bits from any n-bit polynomial size $\mathbf{AC^0}$-circuit source of min-entropy k, with superpolynomially small error. First, this was reduced to the task of extracting from small height decision tree sources. Since a decision tree of height h depends on at most 2^h bits, it is in particular a 2^h-local source. Viola then showed that Rao's extractor [Rao09] for low-weight affine sources also extracts with some loss from local sources. Is there a better seedless extractor for decision tree sources or for $\mathbf{AC^0}$ sources?

References

[Ajt83] Miklós Ajtai. Σ_1^1-formulae on finite structures. *Ann. Pure Appl. Logic*, 24(1):1–48, 1983.

[ASE92] Noga Alon, Joel H. Spencer, and Paul Erdős. *The Probabilistic Method*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley and Sons, Inc., 1992.

[Bea94] Paul Beame. A switching lemma primer. Technical Report UW-CSE-95-07-01, Department of Computer Science and Engineering, University of Washington, November 1994. Available from http://www.cs.washington.edu/homes/beame/.

[FSS84] Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.

[GGN10] Oded Goldreich, Shafi Goldwasser, and Asaf Nussboim. On the implementation of huge random objects. *SIAM J. Comput.*, 39(7):2761–2822, 2010.

[Hås86] Johan Håstad. Almost optimal lower bounds for small depth circuits. In Juris Hartmanis, editor, *STOC*, pages 6–20. ACM, 1986.

[JS89] Mark Jerrum and Alistair Sinclair. Approximating the permanent. *SIAM J. Comput.*, 18(6):1149–1178, 1989.

[JVV86] Marc R. Jerrum, Leslie G. Valiant, and Vijay V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43(2–3):169–188, 1986.

[KV00] Jeong Han Kim and Van H. Vu. Concentration of multivariate polynomials and its applications. *Combinatorica*, 20(3):417–434, 2000.

[LMN93] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, Fourier transform, and learnability. *J. Assoc. Comput. Mach.*, 40(3):607–620, 1993.

[LV11] Shachar Lovett and Emanuele Viola. Bounded-depth circuits cannot sample good codes. In *Conference on Computational Complexity (CCC)*, 2011.

[OSSS05] Ryan O'Donnell, Michael E. Saks, Oded Schramm, and Rocco A. Servedio. Every decision tree has an influential variable. In *Symposium on Foundations of Computer Science (FOCS)*, pages 31–39. IEEE, 2005.

[Rao09] Anup Rao. Extractors for low-weight affine sources. In *Conference on Computational Complexity (CCC)*, pages 95–101. IEEE, 2009.

[TV00] L. Trevisan and S. Vadhan. Extracting randomness from samplable distributions. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, FOCS '00, pages 32–, Washington, DC, USA, 2000. IEEE Computer Society.

[Vio09] Emanuele Viola. Bit-probe lower bounds for succinct data structures. In *41th Symposium on the Theory of Computing (STOC)*, pages 475–482. ACM, 2009.

[Vio10] Emanuele Viola. The complexity of distributions. In *51th Symposium on Foundations of Computer Science (FOCS)*, pages 202–211. IEEE, 2010.

[Vio11] Emanuele Viola. Extractors for circuit sources. In Rafail Ostrovsky, editor, *FOCS*, pages 220–229. IEEE, 2011.

[Yao85] Andrew Yao. Separating the polynomial-time hierarchy by oracles. In *26th Symposium on Foundations of Computer Science (FOCS)*, pages 1–10. IEEE, 1985.