Electronic Colloquium on Computational Complexity, Report No. 133 (2019)

More on $AC^0[\oplus]$ and Variants of the Majority Function

Nutan Limaye* Srikanth Srinivasan † Utkarsh Tripathi ‡

Abstract

In this paper we prove two results about $AC^0[\oplus]$ circuits.

- We show that for $d(N) = o(\sqrt{\log N / \log\log N})$ and $N \leq s(N) \leq 2^{dN^{1/d^2}}$ there is an explicit family of functions $\{f_N : \{0,1\}^N \to \{0,1\}\}$ such that
  - f_N has uniform AC^0 formulas of depth d and size at most s;
  - f_N does not have $AC^0[\oplus]$ formulas of depth d and size s^ε, where ε is a fixed absolute constant.

  This gives a quantitative improvement on the recent result of Limaye, Srinivasan, Sreenivasaiah, Tripathi, and Venkitesh, (STOC, 2019), which proved a similar *Fixed-Depth Size-Hierarchy theorem* but for $d \ll \log\log N$ and $s \ll \exp(N^{1/2^{\Omega(d)}})$.

  As in the previous result, we use the *Coin Problem* to prove our hierarchy theorem. Our main technical result is the construction of uniform size-optimal formulas for solving the coin problem with improved sample complexity $(1/\delta)^{d+4}$ (down from $(1/\delta)^{2^{O(d)}}$ in the previous result).
- In our second result, we show that randomness buys depth in the $AC^0[\oplus]$ setting. Formally, we show that for any fixed constant $d \geq 2$, there is a family of Boolean functions that has polynomial-sized randomized uniform AC^0 circuits of depth d but no polynomial-sized (deterministic) $AC^0[\oplus]$ circuits of depth d.

  Previously Viola (Computational Complexity, 2014) showed that an increase in depth (by at least 2) is essential to avoid superpolynomial blow-up while derandomizing randomized AC^0 circuits. We show that an increase in depth (by at least 1) is essential even for $AC^0[\oplus]$.

  As in Viola's result, the separating examples are promise variants of the Majority function on N inputs that accept inputs of weight at least $N/2 + N/(\log N)^{d-1}$ and reject inputs of weight at most $N/2 - N/(\log N)^{d-1}$.

1 Introduction

This paper addresses questions in the field of *Boolean Circuit complexity,* where we study the complexity of compuational problems, modeled as sequences of Boolean functions $f_N : \{0,1\}^N \to \{0,1\}$, in the combinatorially defined Boolean circuit model (see, e.g. [AB09] for an introduction).

Boolean circuit complexity is by now a classical research area in Computational complexity, with a large body of upper and lower bound results in many interesting circuit models. The questions we consider here are motivated by two of the most well-studied circuit models, namely AC^0 and $AC^0[\oplus]$. The circuit class AC^0 denotes the class of Boolean circuits of small-depth

*IIT Bombay, Department of Computer Science and Engineering, Mumbai, India. nutan@cse.iitb.ac.in
†IIT Bombay, Department of Mathematics, Mumbai, India. srikanth@math.iitb.ac.in
‡IIT Bombay, Department of Mathematics, Mumbai, India. utkarshtripathi.math@gmail.com

ISSN 1433-8092

made up of AND, OR and NOT gates, while $\mathrm{AC}^0[\oplus]$ denotes the circuit class that is also allowed the use of parity (addition modulo 2)[1] gates.[2]

Historically, AC^0 was among the first circuit classes to be studied and for which superpolynomial lower bounds were proved. Building on an influential line of work [Ajt83, FSS84, Yao85], Håstad [Has89] showed that any depth-d AC^0 circuit for the Parity function on N variables must have size $\exp(\Omega(N^{1/(d-1)}))$, hence proving an exponential lower bound for constant depths and superpolynomial lower bounds for all depths $d \ll \log N/\log\log N$. Researchers then considered the natural follow-up problem of proving lower bounds for $\mathrm{AC}^0[\oplus]$. Soon after, Razborov [Raz87] and Smolensky [Smo87, Smo93] showed a lower bound of $\exp(\Omega(N^{1/2(d-1)}))$ for computing the Majority function on N inputs, again obtaining an exponential lower bound for constant depths and superpolynomial lower bounds for all depths $d \ll \log N/\log\log N$.

Thus, we have strong lower bounds for both classes AC^0 and $\mathrm{AC}^0[\oplus]$. However, in many senses, $\mathrm{AC}^0[\oplus]$ remains a much more mysterious class than AC^0. There are many questions that we have been successfully able to answer about AC^0 but whose answers still evade us in the $\mathrm{AC}^0[\oplus]$ setting. This work is motivated by two such questions that we now describe.

Size Hierarchy Theorems. Size Hierarchy theorems are an analogue in the Boolean circuit complexity setting of the classical Time and Space hierarchy theorems for Turing Machines. Formally, the problem is to separate the power of circuits (from some class) of size s from that of circuits of size at most s^ε for some fixed $\varepsilon > 0$. As is usual in the setting of circuit complexity, we ask for *explicit* separations,[3] or equivalently, we ask that the separating sequence of functions be computed by a uniform family of circuits of size at most s.

The challenge here is to obtain explicit functions for which we can obtain *tight* (or near-tight) lower bounds, since we want the functions to have (uniform) circuits of size s but no circuits of size at most s^ε.

In the AC^0 setting, Håstad's theorem stated above immediately implies such a tight lower bound, since it is known (folklore) that the Parity function does have depth-d circuits of size $\exp(O(N^{1/d-1}))$ for every d. Varying the number of input variables to the Parity function suitably, this yields a Size Hierarchy theorem for the class of AC^0 circuits of depth d as long as $d \ll \log N/\log\log N$ and $s = \exp(o(N^{1/(d-1)}))$.

For $\mathrm{AC}^0[\oplus]$, however, this is not as clear, as explicit *tight* lower bounds are harder to prove. In particular, the lower bounds of Razborov [Raz87] and Smolensky [Smo87, Smo93] for the Majority function (and other symmetric functions) are not tight; indeed, the exact complexity of these functions in $\mathrm{AC}^0[\oplus]$ remains unknown [OSS19]. In a recent result, the authors along with Sreenivasaiah and Venkitesh [LSS+19] were able to show a size hierarchy theorem for $\mathrm{AC}^0[\oplus]$ *formulas*[4] for depths $d \ll \log\log N$ and size $s \ll \exp(N^{1/2^{\Omega(d)}})$. This is a weaker size hierarchy theorem than the one that follows from Håstad's theorem for AC^0, both in terms of the size parameter as well as the depths until which it holds. In this paper, we build upon the ideas in [LSS+19] and prove the following result that is stronger in both parameters.

[1] Though we state our results only for $\mathrm{AC}^0[\oplus]$, they extend in a straightforward way to $\mathrm{AC}^0[p]$, where we are allowed gates that add modulo p, for any fixed prime p.

[2] The formal definitions of AC^0 and $\mathrm{AC}^0[\oplus]$ only allow for polynomial-size circuits and constant depth. However, since some of our results apply to larger families of circuits, we will abuse notation and talk about AC^0 circuits of size $s(N)$ and depth $d(N)$ where s and d are growing functions of N.

[3] It is trivial to show a non-explicit separation by counting arguments.

[4] A formula is a circuit where the underlying undirected graph is a tree. For constant-depth, formulas and circuits are interchangeable with a polynomial blowup in depth. However, this is no longer true at superconstant depth [Ros08, RS17].

Theorem 1. *The following holds for some absolute constant $\varepsilon > 0$. Let N be a growing parameter and $d = d(N), s = s(N)$ be functions of N with $d = o\left(\sqrt{\frac{\log N}{\log\log N}}\right)$ and $N \leq s \leq 2^{dN^{1/d^2}}$. Then there is a family of functions $\{f_N\}$ such that f_N has uniform AC^0 formulas of depth d and size at most s but does not have any $\mathrm{AC}^0[\oplus]$ formulas of depth d and size at most s^{ε}.*

Randomized versus Deterministic circuits. The study of the relative power of randomized versus deterministic computation is an important theme in Computational complexity. In the setting of circuit complexity, it is known from a result of Adleman [Adl78] that unbounded-depth polynomial-sized *randomized* circuits[5] are no more powerful than polynomial-sized deterministic circuits.

However, the situation is somewhat more intriguing in the bounded-depth setting. Ajtai and Ben-Or [AB84] showed that for any randomized depth-d AC^0 circuit of size at most s, there is deterministic AC^0 circuit of depth $d+2$ and size at most $\mathrm{poly}(s)$ that computes the same function; a similar result also follows for $\mathrm{AC}^0[\oplus]$ with the deterministic circuit having depth $d+3$. This begs the question: is this increase in depth necessary?

For AC^0 circuits of constant depth, Viola [Vio14] gave an optimal answer to this question by showing that an increase of two in depth is necessary to avoid a superpolynomial blow-up in size. To the best of our knowledge, this problem has not been studied in the setting of $\mathrm{AC}^0[\oplus]$. In this paper, we show that an increase in depth (of at least one) is required even for $\mathrm{AC}^0[\oplus]$. More formally we prove the following theorem.

Theorem 2. *Fix any constant $d \geq 2$. There is a family of Boolean functions that has polynomial-sized randomized uniform AC^0 circuits of depth d but no polynomial-sized (deterministic) $\mathrm{AC}^0[\oplus]$ circuits of depth d.*

1.1 Proof Ideas

The proofs of both theorems are based on analyzing the complexity of Boolean functions that are closely related to the Majority function.

Size-Hierarchy Theorem. To prove the size hierarchy theorem for constant-depth $\mathrm{AC}^0[\oplus]$ formulas, [LSS+19] studied the $\mathrm{AC}^0[\oplus]$ complexity of the *δ-coin problem* [BV10], which is the problem of distinguishing between a coin that is either heads with probability $(1+\delta)/2$ or is heads with probability $(1-\delta)/2$, given a sequence of a large number of independent tosses of this coin. This problem has been studied in a variety of computational models [SV10, BV10, CGR14, GII+19]. It is known [OW07, Ama09] that this problem can be solved by AC^0 formulas of depth d and size $\exp(O(d(1/\delta)^{1/(d-1)}))$ and further [OW07, SV10, LSS+19] that this upper bound is tight up to the constant in the exponent even for $\mathrm{AC}^0[\oplus]$ formulas of depth d. This gives a family of functions for which we have tight lower bounds for $\mathrm{AC}^0[\oplus]$ formulas.

Based on this, [LSS+19] noted that to prove $\mathrm{AC}^0[\oplus]$ size-hierarchy theorems for size $s(N)$ and depth $d(N)$, it suffices to construct a uniform sequence of formulas of size s and depth d solving the coin problem optimally (i.e. for δ such that $s = \exp(O(d(1/\delta)^{1/(d-1)}))$) using at most N samples. Before [LSS+19], all known size-optimal formula constructions for solving the δ-coin problem used $N = s = \exp(O(d(1/\delta)^{1/(d-1)}))$ many samples. The work of [LSS+19] brought the number of samples down to $N = (1/\delta)^{2^{O(d)}}$. Our main technical result here is an explicit size-optimal formula for solving the δ-coin problem using only $(1/\delta)^{d+4}$ samples. Plugging this into the framework from [LSS+19], we immediately get the improved size-hierarchy theorem.

[5] A *randomized Boolean circuit* for a Boolean function $f(x)$ is a Boolean circuit C that takes as input variables x and r such that for each setting of x and uniformly random r, $C(x) = f(x)$ with probability at least $3/4$.

While the reason for this improvement is rather technical, we try to give a high-level outline here. It was shown by O'Donnell and Wimmer [OW07] and Amano [Ama09] that the δ-coin problem is solved by read-once AC^0 formulas of depth d with gates of prescribed fan-ins. While the size s of these formulas is optimal, the number of samples is $N = s$, which is too big for our purposes. In [LSS+19], this number is brought down by distributing a smaller number of variables across the formula in a pseudorandom way (specifically using a Nisan-Wigderson design). The challenge now is to show that the formula still solves the δ-coin problem: the reason this is challenging is that various subformulas now share variables and hence the events that they accept or reject are no longer independent. However [LSS+19] note that *Janson's inequality* [Jan90], a tool from probabilistic combinatorics, can be used to argue that if the variables are spread out in a suitably "random"-like fashion, then various subformulas at a certain depth may, for our intents and purposes, be treated as "nearly" independent.

This "distance" from independence is determined by a parameter Δ that goes into the statement of Janson's inequality, and hence let us call it the *Janson parameter*. In [LSS+19], this parameter was measured in a very brute-force way, forcing us to square the number of samples every time the depth of the formula increased by 1. This leads to a sample complexity of $(1/\delta)^{2^{O(d)}}$. Here, however, we give a different way of bounding the Janson parameter via a recursive analysis, which works as long as the number of variables grows by a factor of $(1/\delta)$ for each additional depth. This gives the improvement in our construction.

Randomized versus Deterministic circuits. For his separation of deterministic and randomized AC^0 circuits, Viola [Vio14] used the *k-Promise-Majority* functions [6] which are Boolean functions that accept inputs with at least $N/2 + k$ many 1s and reject inputs with at most $N/2 - k$ many 0s. Building on work of [AB84, OW07, Ama09], Viola [Vio14] showed that for $k = N/(\log N)^{d-1}$, there are k-Promise-Majorities that have uniform polynomial-sized *randomized* depth-d AC^0 circuits. On the other hand, he also showed that the same problem has no deterministic circuit of depth d (and in fact even $d + 1$).

The challenge in proving such a lower bound is that if a Boolean function has a randomized circuit of depth d and size s, then it immediately follows that there is also a deterministic circuit of the same depth and size *approximating* the same Boolean function (i.e. computing it correctly on most inputs). In particular, the lower bound technique must be able to distinguish circuits that are computing the function exactly (since this is hard) from circuits that are merely approximating it (as this is easy). Viola overcomes this hurdle in the case of AC^0 with a clever argument for depth-3 circuits and an inductive use of the Håstad Switching lemma for higher depths. Neither of these techniques is available for $\mathrm{AC}^0[\oplus]$ circuits. In fact, the standard techniques for proving lower bounds against $\mathrm{AC}^0[\oplus]$ involve approximating the circuits to constant error using low-degree polynomials from $\mathbb{F}_2[x_1, \ldots, x_N]$. Note that this immediately runs into the obstacle mentioned above since we can then no longer distinguish between circuits that are exactly correct and those that are approximately correct.

The way we get around this argument is to use a recent result of Oliveira, Santhanam and the second author [OSS19] where it is observed that the standard construction of approximating polynomials for $\mathrm{AC}^0[\oplus]$ actually gives polynomials that approximate the given circuit C to very small error on either the zero or the one inputs of C. They are able to use this to improve known $\mathrm{AC}^0[\oplus]$ lower bounds for the Majority function. Our main observation is that this stronger lower bound is actually able to distinguish between circuits that approximate the Majority function to constant error (say from [OW07, Ama09]) and those that compute it exactly, thus overcoming

[6] These are called *Approximate Majorities* in a lot of the earlier literature ,including in Viola's work. We avoid this name, since Approximate Majorities are also used for functions more closely related to the coin problem [OW07], and in our opinion, the name "Promise Majorities" better describes these functions.

the barrier we mentioned above. We then note that their proof can also be made to work for k-Promise-Majorities. This yields the separation.

2 Size hierarchy theorem for $\mathrm{AC}^0[\oplus]$

Definition 3 (The δ-Coin Problem)**.** *Let $\delta \in (0,1)$ be a parameter. Given an $N \in \mathbb{N}$, we define the probability distributions $\mu_{\delta,0}^N$ and $\mu_{\delta,1}^N$ to be the product distributions where each bit is set to 1 with probability $(1-\delta)/2$ and $(1+\delta)/2$ respectively. We omit the δ in the subscript and N in the superscript when these are clear from context.*

Given a function $g : \{0,1\}^N \to \{0,1\}$, we say that g solves the δ-coin problem *if*

$$\Pr_{\boldsymbol{x}\sim\mu_0^N}[g(\boldsymbol{x}) = 1] \leq 0.1 \text{ and } \Pr_{\boldsymbol{x}\sim\mu_1^N}[g(\boldsymbol{x}) = 1] \geq 0.9. \tag{1}$$

We say that the sample complexity *of g is N.*

Parameters Let m, d be growing parameters such that $d = o(m/\log m)$. Let $1/\delta = (m \ln 2)^{d-1}/C_1$, where C_1 is a fixed large constant, to be specified below. Let $M = \lceil m \cdot 2^m \cdot \ln 2 \rceil$ and let $M_1 = 2^m$.

Theorem 4. *For large enough absolute constant C_1, the following holds. For parameters m, δ, d as above and for $d \geq 2$, there is an explicit depth-d AC^0 formula of size $\exp(O(dm)) = \exp(O(d(1/\delta)^{1/d-1}))$ and sample complexity $(1/\delta)^{d+o(d)}$ that solves the δ-coin problem.*

2.1 Proof of Theorem 1

We use Theorem 4 for a suitable choice of parameters to define the explicit function.

Let $m = \lfloor (\alpha \log s)/d \rfloor$ for some absolute constant $\alpha < 1$ that we fix below. It can be checked that as $s \geq N$ and $d = o(\sqrt{\log N / \log\log N})$, we have $d = o(m/\log m)$. Define δ as above and note that $(1/\delta)^{d+o(d)} \leq m^{2d^2} \leq (\log s)^{2d^2} \leq N$, where the final inequality uses the given upper bounds on d and s.

We set f_N to be the Boolean function computed by the formula F_d constructed above on the first $(1/\delta)^{d+4}$ of the N input variables. By Theorem 4, the size of F_d is $\exp(O(dm)) \leq s$ for a small enough absolute constant α and F_d solves the δ-coin problem. Moreover, it was shown in [LSS+19] that any depth-d $\mathrm{AC}^0[\oplus]$ formula solving the δ-coin problem must have size $\exp(\Omega(d(1/\delta)^{1/(d-1)})) = \exp(\Omega(md)) = s^{\varepsilon}$ for some absolute constant $\varepsilon > 0$. This proves the theorem.

2.2 Proof of Theorem 4

Construction There exist integers Q, D, such that Q is a prime power, $M \leq Q^D \leq 2M$ and $m^4/\delta \leq Q \leq (m^4/\delta)^{1+o(1)}$. Let $\mathbb{F}$ be a finite field with Q elements and $A \subseteq \mathbb{F}$ be a set of size m. For completeness, we outline how such Q, D and $\mathbb{F}$ can be constructed in time $\mathrm{poly}(m)$ in the Appendix (Section A).

Let $\mathcal{P}_D$ be the lexicographically first M univariate polynomials over $\mathbb{F}$ of degree less than D. Similarly, let $\mathcal{P}'_D$ be the lexicographically first M_1 univariate polynomials over $\mathbb{F}$ of degree less than D.

We now describe the construction of our formula. The variables in the formula correspond to the points in the set $A \times \mathbb{F}^{d-1}$. i.e. for each $(a, c_1, \ldots, c_{d-1}) \in A \times \mathbb{F}^{d-1}$, we have a variable $x(a, c_1, \ldots, c_{d-1})$. We have $m \cdot Q^{d-1}$ many variables. We use N to denote this number.

For each $i \in [d-1]$ and $\bar{P} = (P_i, \ldots, P_{d-1}) \in \mathcal{P}_D^{d-i}$, define a depth-$i$ formula $\mathcal{C}_{(P_i,\ldots,P_{d-1})}$ inductively as follows.

$$\mathcal{C}_{(P_1,\ldots,P_{d-1})} = \bigwedge_{a \in A} x(a, P_1(a), \ldots, P_{d-1}(a))$$
$$\mathcal{C}_{(P_2,\ldots,P_{d-1})} = \bigvee_{R_1 \in \mathcal{P}_D} \mathcal{C}_{(R_1,P_2,\ldots,P_{d-1})}$$
$$\mathcal{C}_{(P_3,\ldots,P_{d-1})} = \bigwedge_{R_2 \in \mathcal{P}_D} \mathcal{C}_{(R_2,P_3,\ldots,P_{d-1})}$$

and so on, with the gates alternately repeating between **AND** and **OR**. Finally, $\mathcal{C}_{(\emptyset)}$ is the output of the formula. If the depth of the formula is odd (even) then $\mathcal{C}_{(\emptyset)}$ is an **AND** gate (**OR** gate resp.), where the **AND** (**OR** resp.) is over $\mathcal{C}(R)$, where $R \in \mathcal{P}'_D$.

$$\mathcal{C}_{(\emptyset)} = \begin{cases} \bigwedge_{R \in \mathcal{P}'_D} \mathcal{C}(R) & \text{if } i \text{ is odd} \\ \bigvee_{R \in \mathcal{P}'_D} \mathcal{C}(R) & \text{if } i \text{ is even} \end{cases}$$

This finishes the description of our formula. We use $F_d = \mathcal{C}_{(\emptyset)}$ to denote this formula.

Analysis of the construction Here we present the details regarding the analysis of our construction presented above, which will be used to prove Theorem 4. We will start with some definitions, notations and some useful inequalities.

Definition 5. *For $1 \leq i \leq d-1$, we define the following terms.*

1. *For $\bar{P} = (P_i, \ldots, P_{d-1}) \in \mathcal{P}_D^{d-i}$ and $b \in \{0,1\}$, let*

$$Acc_{\bar{P},b} := \Pr_{\mu_b}[\mathcal{C}_{(P_i,\ldots,P_{d-1})} \text{ accepts}] \text{ and}$$

$$Rej_{\bar{P},b} := \Pr_{\mu_b}[\mathcal{C}_{(P_i,\ldots,P_{d-1})} \text{ rejects}].$$

   Let $q_{\bar{P},b} = min\ \{Acc_{\bar{P},b}, Rej_{\bar{P},b}\}$.

2. *For $\bar{P} = (P_i, \ldots, P_{d-1}), \bar{P}' = (P'_i, \ldots, P'_{d-1}) \in \mathcal{P}_D^{d-i}$, we say that $\bar{P} \sim \bar{P}'$ when $\mathcal{C}_{\bar{P}}$ and $\mathcal{C}_{\bar{P}'}$ are distinct gates, which share a common input variable.*

3. *For $\bar{P} = (P_{i+1}, \ldots, P_{d-1}), \bar{P}' = (P'_{i+1}, \ldots, P'_{d-1}) \in \mathcal{P}_D^{d-i-1}$, $b \in \{0,1\}$,*

$$\Delta_{\bar{P},\bar{P}',b} = \begin{cases} \sum_{\substack{R_i, R'_i \in \mathcal{P}_D \\ (R_i,\bar{P}) \sim (R'_i,\bar{P}')}} \mathbf{Pr}_{\mathcal{D}_\lambda}[\mathcal{C}_{(R_i,\bar{P})} = 0 \ \textbf{\textit{AND}}\ \mathcal{C}_{(R'_i,\bar{P}')} = 0] & \text{if } \mathcal{C}_{\bar{P}} \text{ and } \mathcal{C}_{\bar{P}'} \text{ are AND gates} \\ \sum_{\substack{R_i, R'_i \in \mathcal{P}_D \\ (R_i,\bar{P}) \sim (R'_i,\bar{P}')}} \mathbf{Pr}_{\mathcal{D}_\lambda}[\mathcal{C}_{(R_i,\bar{P})} = 1 \ \textbf{\textit{AND}}\ \mathcal{C}_{(R'_i,\bar{P}')} = 1] & \text{if } \mathcal{C}_{\bar{P}} \text{ and } \mathcal{C}_{\bar{P}'} \text{ are OR gates} \end{cases}$$

A useful tool in our analysis of the circuit is the Janson's inequality stated here in the language of Boolean circuits.

Theorem 6 (Janson's inequality)**.** *Let $\mathcal{C}_1, \ldots, \mathcal{C}_M$ be any monotone Boolean circuits over inputs $x_1, \ldots, x_n$ and let $\mathcal{C}$ denote $\bigvee_{i\in[M]} \mathcal{C}_i$. For each distinct $i, j \in [M]$, we use $i \sim j$ to denote the fact that $\mathcal{C}_i$ and $\mathcal{C}_j$ share a common variable. Assume each x_j ($j \in [M]$) is chosen independently to be 1 with probability $p_j \in [0,1]$, and that under this distribution, we have $\mathbf{max}_{i\in[M]}\{\mathbf{Pr}_x[\mathcal{C}_i(x) = 1]\} \leq 1/2$. Then we have*

$$\prod_{i\in[M]} \mathbf{Pr}_x[\mathcal{C}_i(x) = 0] \;\leq\; \mathbf{Pr}_x[\mathcal{C}(x) = 0] \;\leq\; \left(\prod_{i\in[M]} \mathbf{Pr}_x[\mathcal{C}_i(x) = 0]\right) \cdot \exp(\Delta) \tag{2}$$

where $\Delta := \sum_{i\sim j} \mathbf{Pr}_x[(\mathcal{C}_i(x) = 1) \wedge (\mathcal{C}_j(x) = 1)]$.

Throughout, we use $\log(\cdot)$ to denote logarithm to the base 2 and $\ln(\cdot)$ for the natural logarithm. We use $\exp(x)$ to denote e^x.

Fact 7. *Assume that $x \in [-1/2, 1/2]$. Then we have the following chain of inequalities.*

$$\exp(x - (|x|/2)) \underset{\text{(a)}}{\leq} \exp(x - x^2) \underset{\text{(b)}}{\leq} 1 + x \underset{\text{(c)}}{\leq} \exp(x) \underset{\text{(d)}}{\leq} 1 + x + x^2 \underset{\text{(e)}}{\leq} 1 + x + (|x|/2) \tag{3}$$

We define a few parameters, which will be useful in the main technical lemma that helps in proving Theorem 4.

For $i \in [d-1]$, let $\alpha_i = m^i \cdot (\ln 2)^{i-1} \cdot \delta$. Let also for $i \in [d-2]$, $\beta_i = \beta_{i-1} + 2\alpha_i + \frac{2}{m^i (\ln 2)^{i-1}}$

Observation 8. *For all $i \in [d-2]$, $\beta_i \leq O(1/m)$.*

Lemma 9. *Assume $d \geq 3$ and $q_{\bar{P},b}$ and formula $\mathcal{C}_{(\emptyset)}$ defined as before. We have the following properties.*

1. *For $b \in \{0,1\}$, $i \in [d-2]$ such that $i \equiv b \pmod 2$,*

$$\frac{1}{2^m} \cdot (1 + \alpha_i \exp(-\beta_i)) \leq q_{\bar{P},b} \leq \frac{1}{2^m} \cdot (1 + \alpha_i \exp(\beta_i))$$
$$\frac{1}{2^m} \cdot (1 - \alpha_i \exp(\beta_i)) \leq q_{\bar{P},(1-b)} \leq \frac{1}{2^m} \cdot (1 - \alpha_i \exp(-\beta_i))$$

2. *Say $d - 1 \equiv b \pmod 2$. Then*

$$q_{\bar{P},b} \geq \frac{1}{2^m} \cdot \exp(\alpha_{d-1}/4) \text{ and } q_{\bar{P},1-b} \leq \frac{1}{2^m} \cdot \exp(-\alpha_{d-1}/4)$$

3. *For all $i \in [d-1]$, $b \in \{0,1\}$ and $\bar{P}, \bar{P}' \in \mathcal{P}_D^{d-i-1}$, $\Delta_{\bar{P},\bar{P}',b} < \delta$.*

Assuming that the above lemma holds, we will prove Theorem 4.

Proof of Theorem 4. We start by bounding the size of $F_d = \mathcal{C}_{(\emptyset)}$. As per our construction, the gates at level 1 are **AND** gates with fan-in m each. For all $2 \leq i \leq d-1$, the fan-in of each gate on level i is $M = \lceil m \cdot 2^m \cdot \ln 2 \rceil$ and the top fan-in is $M_1 = 2^m$. Therefore, the total number of gates in the formula is $m \cdot M^{d-2} \cdot M_1$. We can trivially bound this by $M^d = O(m^d 2^{dm})$. As $d = o(m/\log m)$,

we get that the size is bounded by $\exp(O(dm))$. Recall that $1/\delta = (m \ln 2)^{d-1}/C_1$, where C_1 is an appropriately chosen constant. Hence $\exp(O(dm)) = \exp(O(d(1/\delta)^{1/(d-1)}))$.

We will now bound the number of variables, N, used by the formula. As mentioned above, $N = m \cdot Q^{d-1}$. As Q is chosen such that $(m^4/\delta) \leq Q \leq (m^4/\delta)^{1+o(1)}$, we have $N \leq Q^d = (1/\delta)^{d+o(d)}$ as claimed.

Finally, we will show that the formula solves the δ-coin problem. Let us assume that d is even. In that case, the output gate $\mathcal{C}_{(\emptyset)}$ is an **OR** gate. (When it is an **AND** gate, the analysis is very similar.) We bound the probabilities $\Pr_{a\in\mu_0}[F_d(a) = 1]$ and $\Pr_{a\in\mu_1}[F_d(a) = 0]$ by $1/10$ each.

$$\begin{aligned}
\Pr_{a\in\mu_0}[F_d(a) = 1] &\leq \sum_{\bar{P}\in\mathcal{P}'_D} \Pr_{a\in\mu_0}[\mathcal{C}_{(\bar{P})}(a) = 1] && \text{Using a Union bound} \\
&\leq 2^m \cdot \frac{1}{2^m} \cdot \exp(-\alpha_{d-1}/4) && |\mathcal{P}'_D| = 2^m, \text{ using Lemma 9, (2)} \\
&\leq \exp(-\Omega(C_1)) && \text{Using the value of } \alpha_{d-1} \\
&\leq 1/10 && \text{for large enough } C_1
\end{aligned}$$

$$\begin{aligned}
\Pr_{a\in\mu_1}[F_d(a) = 0] &\leq \prod_{\bar{P}\in\mathcal{P}'_D} \Pr_{a\in\mu_1}[\mathcal{C}_{(\bar{P})}(a) = 0] \cdot \exp(\delta) && \text{Using Janson's inequality and Lemma 9, (3)} \\
&\leq \prod_{\bar{P}\in\mathcal{P}'_D} (1 - \Pr_{a\in\mu_1}[\mathcal{C}_{(\bar{P})}(a) = 1]) \cdot \exp(\delta) \\
&\leq (1 - \frac{1}{2^m} \cdot \exp(\alpha_{d-1}/4))^{2^m} \cdot \exp(\delta) && |\mathcal{P}'_D| = 2^m, \text{ using Lemma 9, (2)} \\
&\leq \exp\left(-\frac{2^m}{2^m} \cdot \exp(\alpha_{d-1}/4)\right) \cdot 2 && \text{As } \exp(\delta) \leq 2 \\
&\leq 1/10 && \text{Using the value of } \alpha_{d-1} \\
& && \text{and for large enough } C_1
\end{aligned}$$

This finishes the proof of Theorem 4 assuming Lemma 9. □

We now give the proof of Lemma 9. The proof is by induction on the depth of the circuit.

Proof of Lemma 9. The lemma has three parts. As mentioned above, we proceed by induction on the depth.

Base case ($i = 1$): Here let us first assume that we are working with μ_1^N. We start with part (1). We wish to bound $q_{\bar{P},1}$. From the construction of our formula, we know that the formula has **AND** gates at layer 1 and the inputs to these are independent. Therefore, $q_{\bar{P},1} = \left(\frac{1+\delta}{2}\right)^m$. We will upper and lower bound this quantity.

$$\begin{aligned}
\left(\frac{1+\delta}{2}\right)^m &= \frac{1}{2^m} \cdot (1+\delta)^m \geq \frac{1}{2^m} \cdot \exp(\delta m - \delta^2 m) && \text{Fact 7 (b)} \\
&\geq \frac{1}{2^m} \cdot (1 + \delta m) \\
&= \frac{1}{2^m} \cdot (1 + \alpha_1 \cdot \exp(\beta_1)) && \text{As } \alpha_1 = \delta m, \beta_1 = 2\alpha_1
\end{aligned}$$

$$\begin{aligned}
\left(\frac{1+\delta}{2}\right)^m &= \frac{1}{2^m}\cdot(1+\delta)^m \le \frac{1}{2^m}\cdot\exp(\delta m) && \text{Fact 7 (c)}\\
&\le \frac{1}{2^m}\cdot(1+\delta m+(\delta m)^2) && \text{Fact 7 (d)}\\
&\le \frac{1}{2^m}\cdot(1+\delta m\cdot\exp(2\delta m)) && \text{Fact 7 (c)}\\
&= \frac{1}{2^m}\cdot(1+\alpha_1\cdot\exp(\beta_1)) && \text{As } \alpha_1=\delta m, \beta_1=2\alpha_1
\end{aligned}$$

In the case of μ_0^N, we get $q_{\bar{P},0} = \left(\frac{1-\delta}{2}\right)^m$ and a very similar computation can be used to upper and lower bound this quantity.

There is nothing to prove for part (2) in the base case. We now prove the base case for part (3). Let $\bar{P} = (P_2, \ldots, P_{d-1}), \bar{P'} = (P'_2, \ldots, P'_{d-1}) \in \mathcal{P}_D^{d-2}$. We will analyse $\Delta_{\bar{P},\bar{P'},1}$ here. The analysis for $\Delta_{\bar{P},\bar{P'},0}$ is very similar. Let λ denote $(1+\delta)/2$.

$$\begin{aligned}
\Delta_{\bar{P},\bar{P'},1} &= \sum_{\substack{R,R'\\(R,\bar{P})\sim(R',\bar{P'})}} \mathbf{Pr}_{\mathcal{D}_\lambda}[C_{(R,\bar{P})}=1 \ \mathbf{AND}\ C_{(R',\bar{P'})}=1]\\
&= \sum_{\substack{R,R'\\(R,\bar{P})\sim(R',\bar{P'})}} \lambda^{|\mathbf{Var}(C_{(R,\bar{P})})\cup\mathbf{Var}(C_{(R',\bar{P'})})|}\\
&= \sum_{1\le k<D} \lambda^{2m-k} \sum_{\substack{R,R'\\(R,\bar{P})\sim(R',\bar{P'})\\|\mathbf{Var}(C_{(R,\bar{P})})\cap\mathbf{Var}(C_{(R',\bar{P'})})|=k}} 1\\
&= \sum_{1\le k<D} \lambda^{2m-k} Q^D \sum_{\substack{R\\(R,\bar{P})\sim(0,\bar{P'})\\|\mathbf{Var}(C_{(R,\bar{P})})\cap\mathbf{Var}(C_{(0,\bar{P'})})|=k}} 1 \qquad (4)
\end{aligned}$$

We analyse the second summation term in (4) by considering two cases.

(a) If $\bar{P} = \bar{P'}$, $(R,\bar{P}) \sim (0,\bar{P'})$ if and only if $R \neq 0$ and R has a zero in A. Furthermore $|\mathbf{Var}(C_{(R,\bar{P})}) \cap \mathbf{Var}(C_{(0,\bar{P'})})| = |\mathcal{Z}(R) \cap A|$. Therefore in this case, the inner sum in the last line is exactly the number of non-zero polynomials of degree $< D$ with exactly k zeros in A which is bounded from above by $\binom{m}{k}Q^{D-k}$. Therefore in this case,

$$
\begin{aligned}
\Delta_{\bar{P},\bar{P}',1} &\leq \sum_{1\leq k<D} \lambda^{2m-k} Q^D \binom{m}{k} Q^{D-k} \\
&= (\lambda^m Q^D)^2 \sum_{1\leq k<D} \binom{m}{k} \lambda^{-k} Q^{-k} \qquad (5)\\
&\leq (\lambda^m Q^D)^2((1+\frac{1}{\lambda Q})^m - 1) = \\
&= (\lambda^m \cdot 2 \cdot M)^2 \cdot \frac{2m}{\lambda Q} && \text{As } Q^D \leq 2M \text{ and Fact 7 (b), (c)} \\
&\leq (2(1+\delta)^m m \cdot \ln 2)^2 \frac{2m}{Q(1+\delta)} && \text{As } M = m \cdot 2^m \cdot \ln 2 \text{ and } \lambda = (1+\delta)/2 \\
&= 16(\ln 2)^2 \cdot \frac{m^3}{Q}(1+\delta)^{m-1} \\
&\leq \frac{32m^3}{Q} < \delta/2 < \delta && \text{As } (1+\delta)^{m-1} \leq 2 \text{ and } \delta > m^4/Q.
\end{aligned}
$$

(b) If $\bar{P} \neq \bar{P}'$, $(R,\bar{P}) \sim (0,\bar{P}')$ if and only if $R = 0$ or R has a zero in A. Furthermore $|\mathbf{Var}(C_{(R,\bar{P})}) \cap \mathbf{Var}(C_{(0,\bar{P}')})| \leq |\mathcal{Z}(R) \cap A|$. Therefore in this case,

$$
\begin{aligned}
\Delta_{\bar{P},\bar{P}',1} &\leq \lambda^{2m} Q^D \sum_{\substack{R \\ (R,\bar{P})\sim(0,\bar{P}')}} (\frac{1}{\lambda})^{(|\mathbf{Var}(C_{(R,\bar{P})}) \cap \mathbf{Var}(C_{(0,\bar{P}')})|)} \\
&\leq \lambda^{2m} Q^D (\frac{1}{\lambda})^D + \lambda^{2m} Q^D \sum_{\substack{R\neq 0 \\ (R,\bar{P})\sim(0,\bar{P}')}} (\frac{1}{\lambda})^{|\mathcal{Z}(R)\cap A|} \\
&\leq \lambda^{2m} Q^D \lambda^{-D} + \lambda^{2m} Q^D \sum_{\substack{R\neq 0 \\ R \text{ has a zero in } A}} (\frac{1}{\lambda})^{|\mathcal{Z}(R)\cap A|} \\
&\leq (\lambda^m 2M)^2 \cdot \frac{3^D}{M} + \lambda^{2m} Q^D \sum_{1\leq k<D} \lambda^{-k} \sum_{\substack{R\neq 0 \\ R \text{ has a zero in } A \text{ and } \deg(R)<D}} 1 \qquad (6)\\
&\leq (\lambda^m 2M)^2 \cdot \frac{3^D}{M} + \lambda^{2m} Q^D \sum_{1\leq k<D} \lambda^{-k} \binom{m}{k} Q^{D-k} \\
&\leq 4 \cdot \left(\frac{(1+\delta)^m}{2^m} \cdot m \cdot 2^m \cdot \ln 2\right)^2 \cdot \frac{3^D}{m \cdot 2^m \cdot \ln 2} \qquad (7)\\
&\qquad + (\lambda^m Q^D)^2 \sum_{1\leq k<D} \binom{m}{k} \lambda^{-k} Q^{-k} \\
&\leq O\left(\frac{m3^D}{2^m}\right) + \delta/2 \qquad (8)\\
&\leq \frac{1}{2^{\Omega(m)}} + \delta/2 < \delta
\end{aligned}
$$

In the above computation Step (6) follows because $Q^D \leq 2M$ and $(1/\lambda) \leq 3$. Step (7) is obtained by substituting the the values of λ and M. Note that the second summation term

in Step (7) is exactly the same as in the previous computation and hence, the upper bound of $\delta/2$ on that term in Step (8) follows. Finally as $D = o(m)$ and because $1/2^{\Omega(m)} < \delta/2$, we get the final inequality.

To see that $\Delta_{\bar{P},\bar{P}',0}$ is also upper bounded by δ, a similar analysis can be used. We omit the details. This finishes the base case.

Inductive case: We now start the proof of the inductive case. Here again, let us handle the case of μ_1^N distribution. We assume that we are at an **AND** layer (the **OR** layer is similar). By using Janson's inequality to analyse $q_{\bar{P},1}$, we get the following upper and lower bounds on this quantity.

$$\prod_{R\in\mathcal{P}_D}(1-q_{(R,\bar{P}),0}) \leq q_{\bar{P},1} \leq \prod_{R\in\mathcal{P}_D}(1-q_{(R,\bar{P}),0})\cdot\exp(\delta) \tag{9}$$

Here, the factor $\exp(\delta)$ in the upper bound comes from the application of the Janson's inequality (Theorem 6) and the induction hypothesis for Lemma 9 Part (3). Let us use (9) in order to bound $q_{\bar{P},1}$.

To lower bound $q_{\bar{P},1}$, we can upper bound $q_{(R,\bar{P}),0}$ for each $R \in \mathcal{P}_D$. We can use induction hypothesis to do that. By observing that for each $R \in \mathcal{P}_D$ we get the same bound, we will get the following lower bound on $q_{\bar{P},1}$.

$$\begin{aligned}
q_{\bar{P},1} &\geq \left(1-\frac{1}{2^m}\cdot(1-\alpha_{i-1}\exp(-\beta_{i-1}))\right)^M \\
&\geq \exp\left(-\frac{M}{2^m}\cdot(1-\alpha_{i-1}\exp(-\beta_{i-1}) - M\cdot\left(\frac{1}{2^m}\cdot(1-\alpha_{i-1}\exp(-\beta_{i-1}))\right)^2\right) \\
&\geq \exp\left(-(m\ln 2)\cdot(1-\alpha_{i-1}\exp(-\beta_{i-1}) - O\left(\frac{m}{2^m}\right)\right) \\
&\geq \frac{1}{2^m}\cdot\exp\left(m\cdot\ln 2\cdot\alpha_{i-1}\exp(-\beta_{i-1}) - O\left(\frac{m}{2^m}\right)\right) \\
&\geq \frac{1}{2^m}\cdot\exp\left(\alpha_i\exp(-\beta_{i-1}) - O\left(\frac{m}{2^m}\right)\right) \quad (\text{As } \alpha_i = m\cdot\ln 2\cdot\alpha_{i-1}) \\
&\geq \frac{1}{2^m}\cdot\exp\left(\alpha_i\exp(-\beta_{i-1}) - \frac{\alpha_i^2}{10}\right) \\
&\geq \frac{1}{2^m}\cdot\exp\left(\alpha_i(\exp(-\beta_{i-1}) - \frac{\alpha_i}{10})\right) \\
&\geq \frac{1}{2^m}\cdot\left(1+\alpha_i\exp\left(-\beta_{i-1}-\frac{\alpha_i}{10}\right)\right) \\
&\geq \frac{1}{2^m}\cdot(1+\alpha_i\exp(-\beta_i))
\end{aligned} \tag{10}$$

Here, the second inequality comes from Fact 7 (b). The third inequality uses the value of M. It is easy to see that $O(m/2^m)$ is bounded from above by $\alpha_i^2/10$ by our choice of α_i, which gives us (10). The second last inequality follows from Fact 7 (c). Finally, the last inequality follows from the choice of β_i. This finishes the proof of the lower bound in the case of $b = 1$.

We now obtain an upper bound on $q_{\bar{P},1}$.

$$
\begin{aligned}
q_{\bar{P},1} &\leq \left(1 - \frac{1}{2^m} \cdot (1 - \alpha_{i-1} \exp(\beta_{i-1}))\right)^M \cdot \exp(\delta) \\
&\leq \exp\left(-\frac{M}{2^m} \cdot (1 - \alpha_{i-1} \exp(\beta_{i-1}))\right) \cdot \exp(\delta) \\
&\leq \exp\left(-m \ln 2 \cdot (1 - \alpha_{i-1} \exp(\beta_{i-1}))\right) \cdot \exp(\delta) \\
&\leq \frac{1}{2^m} \cdot \exp\left(m \ln 2 \cdot \alpha_{i-1} \exp(\beta_{i-1})\right) \cdot \exp(\delta) \\
&\leq \frac{1}{2^m} \cdot \exp\left(\alpha_i \exp(\beta_{i-1}) + \delta\right) \qquad (11) \\
&\leq \frac{1}{2^m} \cdot \left(1 + (\alpha_i \exp(\beta_{i-1}) + \delta) + (\alpha_i \exp(\beta_{i-1}) + \delta)^2\right) \\
&\leq \frac{1}{2^m} \cdot \left(1 + \left(\alpha_i \exp(\beta_{i-1}) + \frac{\alpha_i}{m^i (\ln 2)^{i-1}}\right) + 2 \cdot \alpha_i^2\right) \\
&\leq \frac{1}{2^m} \cdot \left(1 + \alpha_i (\exp(\beta_{i-1}) + 2\alpha_i + \frac{2}{m^i (\ln 2)^{i-1}})\right) \\
&\leq \frac{1}{2^m} \cdot (1 + \alpha_i (\exp(\beta_i)))
\end{aligned}
$$

Here the first inequality comes from the induction hypothesis. The second inequality comes from Fact 7 (c). The third inequality is obtained by substituting the value of M. We obtain (11) by substituting $\alpha_i = m \cdot \ln 2 \cdot \alpha_{i-1}$. The inequality following that uses Fact 7 (d). We use the value of α_i in obtaining the next inequality. The final inequality follows from the choice of β_i.

This finishes the proof of Part 1 when $b = 1$ and i is odd. The other cases can be handled in the exactly same way.

We now turn to Part2 of the lemma, which is relevant only when $i = d - 1$. Assume that $b = 1$ (the other case is similar). Assuming that $d - 1$ is odd (i.e. we are dealing with an **AND** layer), it suffices to consider the inequality (10) to obtain the lower bound (note that $\beta_{i-1} = o(1)$ by Observation 8). Similarly, when $d - 1$ is even, we use the the inequality analogous[7] to (11) to obtain an upper bound $2^{-m} \cdot \exp(-\alpha_i \exp(-\beta_{i-1}) + \delta) \leq 2^{-m} \exp(-\alpha_i/4)$. The case $b = 0$ can be worked out similarly.

Finally, we prove the inductive statement about $\Delta_{\bar{P},\bar{P}',1}$ in the case that i is odd. The proof for other cases can be worked out similarly. Fix any $\bar{P}, \bar{P}' \in \mathcal{P}^{d-i+1}$ (in the case that $i = d - 1$, we will have $\bar{P} = \bar{P}' = (\emptyset)$). The computation goes as follows. (The crucial step is the second equality, where we interpret each term in the sum as the probability that a depth $i - 1$ circuit takes the value 0, which can be bounded using Janson's inequality and the induction hypothesis.)

[7]Note that when i is even, we need to do a different analysis to obtain an upper bound for $q_{\bar{P},1}$. However, since the analysis is very similar to the case when i is odd, we have omitted it.

$$
\begin{aligned}
\Delta_{\bar{P},\bar{P}',1} &= \sum_{\substack{R,R' \\ (R,\bar{P})\sim(R',\bar{P}')}} \Pr_{\mu_1^N}[C_{R,\bar{P}} = 0 \textbf{ AND } C_{R',\bar{P}'} = 0] \\
&= \sum_{\substack{R,R' \\ (R,\bar{P})\sim(R',\bar{P}')}} \Pr_{\mu_1^N}[\bigvee_S C_{S,R,\bar{P}} \vee \bigvee_{S'} C_{S',R',\bar{P}'} = 0] \\
&\leq \sum_{\substack{R,R' \\ (R,\bar{P})\sim(R',\bar{P}')}} \prod_S \Pr[C_{S,R,\bar{P}} = 0] \cdot \prod_{S'} \Pr[C_{S',R',\bar{P}'} = 0] \cdot \exp(\delta) \\
&\leq \exp(\delta) \cdot \sum_{\substack{R,R' \\ (R,\bar{P})\sim(R',\bar{P}')}} \left(1 - \frac{(1-2\alpha_{i-1})}{2^m}\right)^{2M} \\
&= \exp(\delta) \left(1 - \frac{(1-2\alpha_{i-1})}{2^m}\right)^{2M} \cdot \sum_{\substack{R,R' \\ (R,\bar{P})\sim(R',\bar{P}')}} 1 \\
&\leq 2\exp(-m\ln 2 + O(\alpha_i)) \cdot \sum_{\substack{R,R' \\ (R,\bar{P})\sim(R',\bar{P}')}} 1 \\
&= O\left(\frac{1}{2^{2m}}\right) \cdot \sum_{\substack{R,R' \\ (R,\bar{P})\sim(R',\bar{P}')}} 1
\end{aligned}
$$

where the first inequality is just Janson's inequality applied to the formula $\bigvee_S C_{S,R,\bar{P}} \vee \bigvee_{S'} C_{S',R',\bar{P}'}$; the second inequality follows from the induction hypothesis applied to level $i-1 \leq d-2$ (we have used a slightly weaker bound that is applicable also to other cases such as when $b = 0$); and the last inequality follows from our choice of M and the fact that $\alpha_i = \alpha_{i-1} \cdot (m \ln 2)$. The sum in the final term may be bounded by $Q^{2D} \cdot O(m/Q)$ almost exactly as in the base case (we omit the computation). We thus get

$$
\Delta_{\bar{P},\bar{P}',1} \leq O\left(\frac{Q^{2D}}{2^m}\right) \cdot \frac{m}{Q} = O\left(\frac{M^2}{2^m}\right) \cdot \frac{m}{Q} \leq \frac{O(m^3)}{Q} < \delta
$$

as $Q \geq m^4/\delta$. This finishes the analysis of $\Delta_{\bar{P},\bar{P}',1}$.

□

3 Deterministic $\mathrm{AC}^0[\oplus]$ circuits

For $a \in \{0,1\}^n$, let $|a|$ denote the Hamming weight of a, i.e. the number of 1s in a.

Definition 10. *Let $k, \ell \leq n/2$. The Promise Majority problem, $\mathsf{PrMaj}^n_{k,\ell}$, is a promise problem of distiguishing n-bit strings of Hamming weight less than $n/2 - k$ from those with Hamming weight more than $n/2 + \ell$. Formally,*

$$
\mathsf{PrMaj}^n_{k,\ell}(a) = \begin{cases} 0 & \text{if } |a| < (\frac{n}{2} - k) \\ 1 & \text{if } |a| \geq (\frac{n}{2} + \ell) \end{cases}
$$

If the length of the input is clear from the context then we drop the superscript n. If $k = 0$ then we denote $\mathsf{PrMaj}_{0,\ell}$ by $\mathsf{LowPrMaj}_\ell$. Similarly, $\ell = 0$ then we denote $\mathsf{PrMaj}_{k,0}$ by $\mathsf{UpPrMaj}_k$.

When both k, ℓ are zero, $\mathsf{PrMaj}_{0,0}$ is the Majority function. If $k = \ell$ then we use PrMaj_k to denote $\mathsf{PrMaj}_{k,k}$.

Let $\mathsf{Yes}^n_\ell, \mathsf{No}^n_k$ denote the yes and no instances of $\mathsf{PrMaj}^n_{k,\ell}$. That is, $\mathsf{Yes}^n_\ell = \{a \in \{0,1\}^n \mid |a| \geq n/2 + \ell\}$ and $\mathsf{No}^n_k = \{a \in \{0,1\}^n \mid |a| < n/2 - k\}$. In [Vio14], the following theorem was proved.

Theorem 11 (Theorem 1.2 [Vio14])**.** *For any $d \geq 2$ and $k(N) = \Omega(N/(\log N)^{d-1})$, there is a randomized AC^0 circuit of depth d computing $\mathsf{PrMaj}^N_{k(N)}$, which has size $\mathrm{poly}(N)$.*

Here, we prove the following theorem.

Theorem 12. *For any $d \geq 2$, say $\mathcal{C}$ is a (deterministic) $\mathrm{AC}^0[\oplus]$ circuit of depth d computing $\mathsf{PrMaj}^N_{N/2\cdot(\log N)^{d-1}}$, then $\mathcal{C}$ must have size $N^{\omega(1)}$.*

It is easy to see that using Theorem 11 and Theorem 12, we immediately get Theorem 2. In order to prove Theorem 12 we need the following claim. This is our main technical claim.

Claim 13. *Let $n \in \mathbb{N}$ and let $k = \Theta(n/(\log n)^c)$. Let $p \in \mathbb{F}[x_1, \ldots, x_n]$ be a (deterministic) polynomial such that it satisfies one of the following two conditions*

$$\textit{either} \quad \Pr_{a \in \mathsf{No}^n_k}[p(a) = 1] \leq 1/n, \qquad \Pr_{a \in \mathsf{Yes}^n_0}[p(a) = 0] \leq 1/10 \tag{12}$$

$$\textit{or} \quad \Pr_{a \in \mathsf{No}^n_0}[p(a) = 1] \leq 1/10 \qquad \Pr_{a \in \mathsf{Yes}^n_k}[p(a) = 0] \leq 1/n \tag{13}$$

Then $\deg(p) = \Omega(\log^{c+1} n)$.

Proof of Theorem 12 using Claim 13. We will first show that Theorem 12 follows from the above claim. We will do this using the following two step argument.

(I) Let us assume for now that $\mathcal{C}$ is a circuit of size s and depth d with either OR gate or $\oplus$ gate as its output gate. Let us call the output gate G_{out}. We will show that if $\mathcal{C}$ computes PrMaj^N_k then we have a circuit $\mathcal{C}'$ of size s, depth d and with output gate G_{out}, such that it computes $\mathsf{UpPrMaj}^n_{2k}$, where $n = N - 2k$.[8].

(II) We will then show that any depth d circuit with OR or $\oplus$ output gate computing $\mathsf{UpPrMaj}^n_{2k}$ must have size $n^{\omega(1)}$.

As we will invoke this for $k = N/2(\log N)^c$, which is $o(N)$, an $n^{\omega(1)}$ lower bound on $\mathsf{UpPrMaj}^n_{2k}$ will imply a $N^{\omega(1)}$ lower bound on PrMaj^N_k, thereby proving the theorem.

Here, (I) can be shown by simply fixing some of the input bits to the constant 1. Specifically, let us set $2k$ bits out of the N bits to 1s. Let $n = N - 2k$. It is easy to see that if $x \in \{0,1\}^n$ has Hamming weight at least $n/2$, then in fact $y = x \cdot 1^{2k}$ has $N/2 + k$ many 1s. Similarly, if $x \in \{0,1\}^n$ has Hamming weight at most $n/2 - 2k$ then the Hamming weight of $y = x \cdot 1^{2k}$ is at most $N/2 - k$.

To show (II) requires a little more work. In particular, to show (II), we use a result from [OSS19] about degree of polynomials approximating $\mathrm{AC}^0[\oplus]$ circuits. To state their result, we will introduce some notation.

[8] As PrMaj is a self-dual function and $\mathsf{UpPrMaj}$ and $\mathsf{LowPrMaj}$ are duals of each other, we can assume that the output gate of $\mathcal{C}$ is OR or $\oplus$ without loss of generality.

Definition 14. *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. For any parameters $\varepsilon_0, \varepsilon_1$, $(\varepsilon_0, \varepsilon_1)$-error probabilistic polynomial for f is a random multilinear polynomial P chosen from $\mathbb{F}_2[x_1, \ldots, x_n]$, such that for any $b \in \{0,1\}$ and any $a \in f^{-1}(b)$, $\Pr[P(a) \neq f(a)] \leq \varepsilon_b$.*

A probabilistic polynomial is said to have degree at most d if the underlying distribution is supported on monomials of degree at most d.

We define the $(\varepsilon_0, \varepsilon_1)$-error probabilistic polynomial degree of a Boolean function f, denoted as $\mathsf{pdeg}_{\varepsilon_0,\varepsilon_1}(f)$, to be the smallest d such that there is an $(\varepsilon_0, \varepsilon_1)$-error probabilistic polynomial of degree d for f.

Lemma 15 (Corollary 15 [OSS19])**.** *Let C be a size s, depth d circuit with OR or $\oplus$ as its output gate. Then there is a probabilistic polynomial $\overline{p}$ approximating C such that $\mathsf{pdeg}_{1/n^2,1/100}(\overline{p})$ is at most $O(\log s)^{d-1}$.*

Remark 16. *Let $\mathcal{C}$ be a circuit of size s and depth d (with any output gate). It is known that if $\overline{p}$ is a probabilistic polynomial for $\mathcal{C}$ such that $\varepsilon_0 = \varepsilon_1 = 1/s^{O(1)}$, then $\mathsf{pdeg}_{1/s^{O(1)},1/s^{O(1)}}(\overline{p})$ is $O(\log s)^d$. The above lemma says that if we need only constant error on one of sides, i.e. say if either ε_0 or ε_1 is $\Omega(1)$, then we can get a better degree upper bound. Instead of having d in the exponent, we get $d-1$ in the exponent. This is crucial for us.*

Note that, if the output gate of $\mathcal{C}$ is OR (AND) then we can ensure that $\varepsilon_0 = 1/n^2$ ($\varepsilon_1 = 1/n^2$, resp.). If it is a $\oplus$ gate, then either can be ensured.

Suppose there is an $\mathrm{AC}^0[\oplus]$ circuit $\mathcal{C}$ of size $s = n^t$ and depth d with top gate OR or $\oplus$ and computing $\mathsf{UpPrMaj}_{2k}$.

Applying Lemma 15 and by standard averaging arguments we can show that there is a fixed polynomial $P \in F[X]$ that satisfies conditions (12) for $c = d-1$ and has the same degree as the degree of $\overline{p}$. Therefore on the one hand, we know that $\deg(P)$ is less than or equal to $O(t \log n)^{d-1}$, while on the other hand using Claim 13 we get that $\deg(P)$ is at least $\Omega(\log n)^d$. (As $N/(\log N)^c = \Theta(n/(\log n)^c)$, Claim 13 is applicable.) Thus, $O(t \log n)^{d-1} \geq \Omega(\log n)^d$ and hence we get $t \geq \Omega(\log n)^{1/d-1}$. Therefore we get (II). This finishes the proof of Theorem 12. □

We now proceed with the proof of Claim 13. We will use the following fact in the proof of Claim 13.

Fact 17. *Say $R \in \mathbb{F}[X]$ is a non-zero polynomial that vanishes on No^n_k, then degree of R is at least $n/2 - k$.*

Proof of Claim 13. We will show that if a deterministic polynomial $p \in \mathbb{F}[X]$ satisfies condition (12), then it has degree $C \cdot \log^{c+1} n$ for some constant C. The proof for the lower bound on the degree of p assuming condition (13) is similar. For simplicity we will work out the proof when $k = n/(\log n)^c$. The proof is similar when $k = \Theta(n/(\log n)^c)$.

Let us use D to denote $C \cdot \log^{c+1} n$. Consider a polynomial p satisfying condition (12). Let $\mathcal{E}_0$ and $\mathcal{E}_1$ be error sets of this polynomial on no and yes instances respectively, i.e. $\mathcal{E}_0 = \{a \in \mathsf{No}^n_{n/(\log n)^c} \mid p(a) = 1\}$ and $\mathcal{E}_1 = \{a \in \mathsf{Yes}^n_0 \mid p(a) = 0\}$. From condition (12) we have a bound on the cardinalities of $\mathcal{E}_0, \mathcal{E}_1$.

We will first observe that in order to prove the claim, we need to show the existence of a polynomial $Q \in \mathbb{F}[X]$ with the following three properties.

(a) $Q(a) = 0$ for all $a \in \mathcal{E}_0$.

(b) $Q \cdot p \neq 0$.

(c) $\deg(Q) \leq r - D$, where $r = n/2 - n/(\log n)^c$ and D is as defined above.

Suppose we have such a Q then let $R = Q \cdot p$. Now R is a polynomial that vanishes on $\mathsf{No}^n_{n/(\log n)^c}$. This is because either p vanishes on $\mathsf{No}^n_{n/(\log n)^c} \setminus \mathcal{E}_0$ or Q vanishes on $\mathcal{E}_0$. Due to property (b), R is also a non-zero polynomial. Therefore using Fact 17, we know that it has degree at least r. Now assuming property (c) we get that p must have degree at least D, thereby proving the claim.

We will prove the existence of a polynomial Q with the above properties. In order to prove that such a Q exists, we proceed as follows.

Let $\mathcal{P}$ be a class of polynomials of degree at most $r - D$ that vanish on $\mathcal{E}_0$. Let $\mathsf{cl}_{r-D}(\mathcal{E}_0)$ denote the set of all the points in $\{0,1\}^n$ such that for each point in the set some polynomial from $\mathcal{P}$ vanishes on it. Formally,

$$\mathsf{cl}_{r-D}(\mathcal{E}_0) = \{a \in \{0,1\}^n \mid \forall q \in \mathcal{Q}, q(a) = 0\},$$

This is also called the $(r-D)$-closure of $\mathcal{E}_0$. Suppose the cardinality of this closure set is strictly smaller than $2^n/10$, i.e suppose the following holds:

$$|\mathsf{cl}_{r-D}(\mathcal{E}_0)| < 2^n/10. \tag{14}$$

Then we know that there is a point $a_0 \in \mathsf{Yes}^n \setminus \mathcal{E}_1$ and a polynomial Q in $\mathcal{P}$ such that $Q(a_0) \neq 0$. It is easy to see that this polynomial has the desired properties; it vanishes on $\mathcal{E}_0$, $Q \cdot p$ is non-zero (this is because $Q(a_0) \neq 0$ and $p(a_0) \neq 0$), and the degree of Q is at most $r - D$. Therefore, assuming (14) we are done. □

Proof of the bound (14). To bound the cardinality of the $(r-D)$-closure of $\mathcal{E}_0$, we will use the following theorem of Nie and Wang [NW15].

Theorem 18. *Let $E \subseteq \mathbb{F}_2^n$ then*

$$\frac{|cl_{r-D}(E)|}{2^n} \leq \frac{|E|}{N_{r-D}},$$

where N_t stands for the number of multilinear monomials of degree at most t.

We will apply this theorem for $E = \mathcal{E}_0$. We know that $N_{r-D} = \binom{n}{\leq r-D}$. We also know that $|\mathcal{E}_0| \leq \frac{1}{n} \cdot \binom{n}{\leq r}$, as it is $1/n$ fraction of the cardinality of $\mathsf{No}^n_{n/(\log n)^c}$. In order to prove (14), it suffices to prove the following.

Subclaim 19. *As long as $r = n/2 - k$, where $k = n/(\log n)^c$ and $D = C \cdot \log^{c+1} n$, where $C \leq 1/100$, we have the following.*

$$\frac{\binom{n}{\leq r}}{n \cdot \binom{n}{\leq r-D}} < 1/10$$

It is clear that proving the subclaim will prove (14). Therefore, assuming the subclaim we are done with the proof of (14). □

Proof of Subclaim 19. As $\binom{n}{\leq r} = \binom{n}{\leq r-D} + \sum_{j=1}^{D} \binom{n}{r-D+j}$ and as $\binom{n}{r-D+j} \leq \binom{n}{r-D+j'}$, where $j' \geq j$, we know that

$$\binom{n}{\leq r} \leq \binom{n}{\leq r-D} + D \cdot \binom{n}{r}.$$

Therefore, to prove the subclaim, it suffices to show that

$$2 \cdot D \cdot \frac{\binom{n}{r}}{n \cdot \binom{n}{\leq r-D}} < 1/10$$

As $\binom{n}{\leq r-D} \geq \binom{n}{r-D}$, in fact it suffices to show that

$$2 \cdot D \cdot \frac{\binom{n}{r}}{n \cdot \binom{n}{r-D}} < 1/10 \tag{15}$$

Now,

$$\frac{\binom{n}{r}}{\binom{n}{r-D}} = \frac{\binom{n}{r}}{\binom{n}{r-1}} \times \frac{\binom{n}{r-1}}{\binom{n}{r-2}} \times \ldots \times \frac{\binom{n}{r-D+1}}{\binom{n}{r-D}} \tag{16}$$

It is easy to see that

$$\begin{aligned}
\frac{\binom{n}{r}}{\binom{n}{r-1}} &\leq \frac{n-r}{r} \cdot (1+o(1)) \\
&= \frac{n/2+k}{n/2-k} \cdot (1+o(1)) && \text{(assuming } r = n/2-k\text{)} \\
&\leq 1 + \frac{8k}{n} \leq e^{8k/n} && \text{(as } 1+x \leq e^x\text{)}
\end{aligned}$$

By using a similar argument to bound $\frac{\binom{n}{r-j}}{\binom{n}{r-j-1}}$ for $j \in \{0, \ldots, D-1\}$, we get that each term in (16) is bounded by $e^{8k/n}$. Therefore we get that $\frac{\binom{n}{r}}{\binom{n}{r-D}} \leq e^{8kD/n}$. As $k = n/(\log n)^c$ and $D = C \cdot log^{c+1} n$, as long as $C \leq 1/100$, say, we get $\frac{\binom{n}{r}}{\binom{n}{r-D}} \leq e^{8kD/n} = o(n)$. This shows (15), thereby proving the subclaim. □

Remark 20. *Note that in Claim 13 we have assumed that $k = \Theta(n/(\log n)^c)$, while we proved it for the specific value of $k = n/(\log n)^c$. To prove it in its full generality we need to simply observe that instead of setting C to some constant less than or equal to say $1/100$, as we did in the above proof, we could set it to an appropriately small constant depending on the constants hidden in $\Theta(\cdot)$.*

References

[AB84] Miklós Ajtai and Michael Ben-Or. A theorem on probabilistic constant depth computations. In *STOC*, pages 471–474. ACM, 1984.

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity : a modern approach*. Cambridge University Press, 2009.

[Adl78] Leonard M. Adleman. Two theorems on random polynomial time. In *19th Annual Symposium on Foundations of Computer Science, Ann Arbor, Michigan, USA, 16-18 October 1978*, pages 75–83, 1978.

[Ajt83] Miklós Ajtai. $\sum^1_1$-formulae on finite structures. *Ann. Pure Appl. Logic*, 24(1):1–48, 1983.

[Ama09] Kazuyuki Amano. Bounds on the size of small depth circuits for approximating majority. In *ICALP (1)*, Lecture Notes in Computer Science, pages 59–70. Springer, 2009.

[BV10] Joshua Brody and Elad Verbin. The coin problem and pseudorandomness for branching programs. In *FOCS*, pages 30–39. IEEE Computer Society, 2010.

[CGR14] Gil Cohen, Anat Ganor, and Ran Raz. Two sides of the coin problem. In *APPROX-RANDOM*, volume 28 of *LIPIcs*, pages 618–629. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2014.

[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.

[GII+19] Alexander Golovnev, Rahul Ilango, Russell Impagliazzo, Valentine Kabanets, Antonina Kolokolova, and Avishay Tal. Ac^0[p] lower bounds against MCSP via the coin problem. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece.*, pages 66:1–66:15, 2019.

[Has89] John Hastad. Almost optimal lower bounds for small depth circuits. *Advances in Computing Research*, 5:143–170, 1989.

[Jan90] Svante Janson. Poisson approximation for large deviations. *Random Struct. Algorithms*, 1(2):221–230, 1990.

[LSS+19] Nutan Limaye, Karteek Sreenivasaiah, Srikanth Srinivasan, Utkarsh Tripathi, and S. Venkitesh. A fixed-depth size-hierarchy theorem for $\mathrm{AC}^0[\oplus]$ via the coin problem. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019.*, pages 442–453, 2019.

[NW15] Zipei Nie and Anthony Y. Wang. Hilbert functions and the finite degree zariski closure in finite field combinatorial geometry. *J. Comb. Theory Ser. A*, 134(C):196–220, August 2015.

[OSS19] Igor Carboni Oliveira, Rahul Santhanam, and Srikanth Srinivasan. Parity helps to compute majority. In *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA.*, pages 23:1–23:17, 2019.

[OW07] Ryan O'Donnell and Karl Wimmer. Approximation by DNF: examples and counterexamples. In *ICALP*, volume 4596 of *Lecture Notes in Computer Science*, pages 195–206. Springer, 2007.

[Raz87] Alexander A. Razborov. Lower bounds on the size of constant-depth networks over a complete basis with logical addition. *Mathematicheskie Zametki*, 41(4):598–607, 1987.

[Ros08] Benjamin Rossman. On the constant-depth complexity of k-clique. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 721–730, 2008.

[RS17] Benjamin Rossman and Srikanth Srinivasan. Separation of $\mathrm{AC}^0[\oplus]$ formulas and circuits. In *ICALP*, volume 80 of *LIPIcs*, pages 50:1–50:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[Sho90] Victor Shoup. New algorithms for finding irreducible polynomials over finite fields. *Mathematics of Computation*, 54(189):435–447, 1990.

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *STOC*, pages 77–82. ACM, 1987.

[Smo93] Roman Smolensky. On representations by low-degree polynomials. In *FOCS*, pages 130–138. IEEE Computer Society, 1993.

[SV10] Ronen Shaltiel and Emanuele Viola. Hardness amplification proofs require majority. *SIAM J. Comput.*, 39(7):3122–3154, 2010.

[Vio14] Emanuele Viola. Randomness buys depth for approximate counting. *Computational Complexity*, 23(3):479–508, 2014.

[Yao85] Andrew Chi-Chih Yao. Separating the polynomial-time hierarchy by oracles (preliminary version). In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 1–10, 1985.

A Finding Q, D and a representation for $\mathbb{F}$ efficiently

We show how to find a prime power Q and a positive D such that $M \leq Q^D \leq 2M$ and $m^4/\delta \leq Q \leq (m^4/\delta)^{1+o(1)}$ in time $\mathrm{poly}(m)$. We will ensure that Q is a power of a prime p such that $p = m^{O(1)}$. By a result of Shoup [Sho90], it then follows that an irreducible univariate polynomial $P(X)$ of degree D over the field $\mathbb{F}_p$ can be found in time $\mathrm{poly}(m, D) = \mathrm{poly}(m)$. We take the field $\mathbb{F}$ to be the field $\mathbb{F}_p[X]/(P(X))$. Note that field arithmetic over $\mathbb{F}$ now takes only $\mathrm{poly}(m)$ time and it follows that the formula F_d constructed in Section 2 is fully explicit: that is, there is a $\mathrm{poly}(m)$-time deterministic algorithm which, when given the descriptions of two gates in F_d outputs the labels of the gates and whether the first gate is a child of the second gate or not.

To show that Q, D as above can be computed, we first need to show they exist. To see this, we first choose D to be the largest positive integer so that $x_0 := M^{1/D} \geq m^4/\delta$. Note that we have $M^{1/D} \geq m^4/\delta$ and $M^{1/(D+1)} < m^4/\delta$, which implies that $D = \Theta(m/(\log m + \log(1/\delta))$. Using the fact that $1/\delta \leq m^{O(d)} \leq 2^{o(m)}$, we have $D = \omega(1)$. In particular, we also get $x_0 = M^{1/D} \leq \left(M^{1/D+1}\right)^{1+1/D} \leq (m^4/\delta)^{1+o(1)}$. Finally, note that D can be computed in $\mathrm{poly}(m)$ time.

Note that $x_0 \geq m^4/\delta \geq m^5$. Define $y_0 = x_0^{1/k}$ where k is the largest positive integer so that $x_0^{1/k} \geq m^5$. It follows that $k = o(m)$ and $y_0 \leq (m^5)^{1+1/k} \leq m^{10}$.

Let p denote the least prime greater than y_0. A theorem of Baker, Harman and Pintz [**?**] shows that $p \leq y_0 + y_0^{0.53} \leq y_0(1 + 1/m^2)$. We take $Q = p^k$. We thus have

$$\frac{m^4}{\delta} \leq x_0 = y_0^k \leq Q \leq x_0 \cdot \left(1 + \frac{1}{m^2}\right)^k \leq x_0 \cdot \left(1 + \frac{1}{m}\right) \leq \left(\frac{m^4}{\delta}\right)^{1+o(1)}.$$

Finally, we also have

$$M = x_0^D \leq Q^D \leq x_0^D \cdot \left(1 + \frac{1}{m}\right)^D = M \cdot (1 + o(1)) \leq 2M.$$

This shows that p, Q, D as specified above exist. As noted above, D can be computed in $\mathrm{poly}(m)$ time. Similarly, we can also compute x_0, y_0 and k. The prime p can be found in $\mathrm{poly}(m)$ time by brute force search in the required range. It thus follows that Q can also be computed in $\mathrm{poly}(m)$ time.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il