Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 175 (2010)

Randomness buys depth for approximate counting

Emanuele Viola*

May 25, 2012

Abstract

We show that the promise problem of distinguishing n-bit strings of hamming weight $1/2+\Omega(1/\lg^{d-1} n)$ from strings of weight $1/2-\Omega(1/\lg^{d-1} n)$ can be solved by explicit, randomized (unbounded fan-in) poly(n)-size depth-d circuits with error $\leq 1/3$, but cannot be solved by deterministic poly(n)-size depth-$(d+1)$ circuits, for every $d \geq 2$; and the depth of both is tight. Our bounds match Ajtai's simulation of randomized depth-d circuits by deterministic depth-$(d+2)$ circuits (Ann. Pure Appl. Logic; '83), and provide an example where randomization buys resources.

To rule out deterministic circuits we combine Håstad's switching lemma with an earlier depth-3 lower bound by the author (Comp. Complexity 2009).

To exhibit randomized circuits we combine recent analyses by Amano (ICALP '09) and Brody and Verbin (FOCS '10) with derandomization. To make these circuits explicit we construct a new, simple pseudorandom generator that fools tests $A_1 \times A_2 \times \ldots \times A_{\lg n}$ for $A_i \subseteq [n], |A_i| = n/2$ with error $1/n$ and seed length $O(\lg n)$, improving on the seed length $\Omega(\lg n \lg\lg n)$ of previous constructions.

*Supported by NSF grant CCF-0845003. Email: viola@ccs.neu.edu

ISSN 1433-8092

1 Introduction

Approximate counting is the problem of computing the number of 1's in a (potentially very long) bit string with some error (which can be additive or relative). This is a central problem in complexity theory, studied in a number of contexts ranging from circuit complexity [Ajt83, ABO84, RW91, Ajt93, CR96] to parallel computation (cf. [CR96]), simulation of BPP in the polynomial-time hierarchy PH [Sip83, Lau83, Vio09], approximation algorithms for #P problems [Sto85], and AM protocols [GS86]. While some of these works explicitly study the complexity of approximate counting in the model of polynomial-size small-depth circuits (AC^0), *all* the above works can be instructively seen as giving various bounds on AC^0 circuits for this problem, where the circuits are possibly randomized. The ability of small AC^0 circuits to count approximately arguably remains one of the most surprising and useful tasks such circuits can accomplish.

Despite the importance of this problem, several basic questions remain open. In this work we focus on the trade-off between the approximation parameter and the depth of the polynomial-size circuits that count approximately. We obtain matching upper and lower bounds both for deterministic and randomized circuits.

Before stating formally our results we recall a few standard conventions. The size of a circuit is the number of its gates. A depth-d circuit consists of d alternating layers of unbounded fan-in And and Or gates, with wires only between adjacent layers; the circuit has access to both input bits and their negations. More liberal definitions of circuits are equivalent to the above one up to only a constant-factor increase in size, see, e.g., [Hås87]. A randomized circuit C is a circuit that takes two inputs x, r, and for every x it gives the correct answer with probability $\geq 2/3$ over the choice of r. A circuit on n bits is explicit (a.k.a. uniform) if it can be constructed in time polynomial in n.

Theorem 1.1 (Depth complexity of $1/2 \pm \epsilon$ approximate counting)**.** *For every integer $d \geq 2$: There are randomized* $\mathrm{poly}(n)$*-size depth-d circuits that distinguish n-bit strings of hamming weight $1/2+\epsilon$ from strings of weight $1/2-\epsilon$ if and only if $\epsilon = \Omega(1/\lg^{d-1} n)$. Moreover, when the circuits exist they are explicit.*

For every integer $d \geq 3$: There are (deterministic) $\mathrm{poly}(n)$*-size depth-d circuits that distinguish n-bit strings of hamming weight $1/2+\epsilon$ from strings of weight $1/2-\epsilon$ if and only if $\epsilon = \Omega(1/\lg^{d-3} n)$.*

To our knowledge, previously such a tight relationship was not known. In particular, note that the bounds in the above Theorem 1.1 distinguish between randomized and deterministic circuits. We are unaware of previous results distinguishing between the two types. Thus, besides settling an arguably natural question, our result has a qualitative consequence for the study of the power of randomness in computation. We elaborate on this next.

The power of randomness. Adleman showed [Adl78] that a randomized circuit $C(x, r)$ with error $1/3$ can be simulated by a (non-uniform) deterministic circuit $C'(x)$ with only a polynomial overhead. To prove this result one can use a Chernoff bound to exhibit a polyno-

mial number of choices $a_1, \ldots, a_{\mathrm{poly}(n)}$ for the coin tosses r of the circuit C such that for any n-bit input x the majority of the choices gives the correct answer. The deterministic simulation tries all choices and then computes majority: $C'(x) := \text{majority}\big(C(x, a_1), \ldots, C(x, a_{\mathrm{poly}(n)})\big)$.

The majority instances that arise in the above simulation have relative hamming weights bounded away from 1/2 by a constant. This allows one to carry through the simulation even in the restricted model AC^0 of constant-depth unbounded fan-in circuits, thanks to the result by Ajtai [Ajt83] (cf. [Vio09]) that majority on such instances can be computed by polynomial-size circuits of depth 3. Moreover, using the fact that Ajtai's circuit is monotone, one can collapse its bottom layer of gates with the output gates of the $\mathrm{poly}(n)$ copies of C (possibly after complementing circuits), to obtain: polynomial-size randomized circuits of depth d can be simulated by deterministic polynomial-size circuits of depth $d+2$. Henceforth we refer to this result as $(A\star)$. (For completeness we review Ajtai's construction in §4.)

A line of research in pseudorandomness has shown that, under complexity assumptions such as the existence of "hard" functions in E, the above simulations hold even in the uniform setting (though in the bounded-depth model, the depth increases by more than 2 as in $(A\star)$, see e.g. [IW97, Agr01, Vio04].

Such results all suggest that randomness can be removed with little overhead. But how much is this overhead? A proof that, say, randomization buys cubic time for a natural problem would be of significant interest regardless of how $P =^? BPP$ is resolved.

In this paper we prove a result showing that some overhead is necessary, in the AC^0 model. An immediate corollary to Theorem 1.1 is the existence of a promise problem [ESY84] that can be solved by poly-size *randomized* circuits of depth d, but cannot by solved by poly-size *deterministic* circuits of depth $d+1$. To our knowledge, this result was not known even with d instead of $d+1$. This weaker form would already give a separation between randomized and deterministic circuits. Our results go further and show that the depth-2 increase in the simulation $(A\star)$ is tight, at least for promise problems (the above simulations all hold for promise problems as well).

Corollary 1.2 (Randomness buys depth)**.** *For every $d \geq 2$ there is a promise problem Π (distinguishing n-bit strings of hamming weight $n(0.5 \pm \Omega(1/\lg^{d-1} n))$ such that:*

Π can be solved by explicit, randomized, $\mathrm{poly}(n)$-size circuits of depth d with error $1/3$;

Π cannot be solved by (deterministic) $\mathrm{poly}(n)$-size circuits of depth $d+1$.

Corollary 1.2 provides an example where randomness buys resources, in the well-studied model of small-depth circuits. As also hinted earlier, the power of randomness in this model has been studied extensively. We add that it is the main question addressed by Ajtai and Ben-Or in [ABO84].

For context, we point out next the simplest separation in the spirit of Corollary 1.2 we are aware of. It is not hard to show that a $\mathrm{poly}(n)$-size depth-2 circuit (e.g., a DNF) cannot distinguish n-bit strings with relative hamming weight $\geq 2/3$ from strings of weight $\leq 1/3$. On the other hand, a randomized $\mathrm{poly}(n)$-size depth-2 circuit $D(x, r)$, where $|r| = \lg n$, can distinguish them with probability $\geq 1/3$ simply by selecting r at random in the set $\{1, 2, \ldots, n\}$ (which we identify with $\{0,1\}^{\lg n}$) and by outputting the r-th input bit, as

follows

$$D(x,r) = 1 \Leftrightarrow \bigvee_{i \in \{0,1\}^{\lg n}} i = r \wedge x_i = 1 \tag{1}$$

(note that $i = r$ can be implemented with one And gate).

However, it is not clear how to extend this to higher depth, nor how to get the tight separation of 2 in the depth. We now explain how we achieve that.

Constructing randomized circuits. We now explain how we construct randomized, polynomial-size circuits of depth d that tell n-bit strings of relative hamming weight $1/2 + \epsilon$ from strings of weight $1/2 - \epsilon$, for $\epsilon = \Omega(1/\lg^{d-1} n)$. First we mention a couple of natural ideas which do not work. The first is to sample a few input bits, and compute majority. However, to get constant error one needs to sample a number of bits which is quadratic in the "bias," i.e., $1/\epsilon^2 = \Omega(\lg^{2d-2} n)$ many bits. Computing their majority then requires circuits of depth $\geq 2d - 1$ [Hås87], which falls short of proving our upper bound. Another way to get circuits of depth $2d - \Theta(1)$ is to use [ABO84, Lemma 2]. Finally, the depth of the constructions by Ajtai [Ajt83] and Stockmeyer [Sto83] does not yield the bound in Theorem 1.1.

Instead, our starting points are recent works by Amano [Ama09] and by Brody and Verbin [BV10]. Using calculations similar to those in [Ajt83, ABO84], these works exhibit a deterministic circuit A of size $\text{poly}(n)$ and depth d which solves the related problem of distinguishing the following two distributions with error $1/3$: i.i.d. bits $X_1, \ldots, X_n$ such that $\Pr[X_i = 1] \geq 1/2 + \Omega(1/\lg^{d-1} n)$ or such that $\Pr[X_i = 1] \leq 1/2 - \Omega(1/\lg^{d-1} n)$.

In our setting, we do not have i.i.d. bits as inputs but we have to succeed w.h.p. on every fixed input, and this distinction is crucial to separate randomness from determinism. However, the probability gap $\Omega(1/\lg^{d-1} n)$ and the depth of the above circuit A correspond to what we are aiming for in Theorem 1.1. Hence we try to reduce an instance x of our promise problem to distinguishing these two distributions. A natural idea is to replace each input gate of the above circuit A with a randomized $\text{poly}(n)$-size depth-2 circuit which outputs a random bit in the input, as in Equation (1). By collapsing the output gates of these circuits with the bottom gates of A, one obtains depth $d + 1$ instead of d. (The bottom gates are those closest to the inputs.)

To get the tight result (depth d) we use the circuit A but with the layer closest to the input removed. This is a depth $d - 1$ circuit A' that can distinguish with error $1/3$ the following two distributions: i.i.d. bits $X_1, \ldots, X_n$ such that $\Pr[X_i = 1] \geq \frac{1}{n}(1 + \Omega(1/\lg^{d-2} n))$ or $\Pr[X_i = 1] \leq \frac{1}{n}(1 - \Omega(1/\lg^{d-2} n))$. The layer that we removed from A is a layer of disjoint And gates on $\lg n$ variables. Note that feeding A i.i.d. bits that equal 1 with probabilities $1/2 \pm \Theta(1/\lg^{d-1} n)$ is seen to have the same effect as feeding A' i.i.d. bits that equal 1 with probabilities $\frac{1}{n}(1 \pm \Theta(1/\lg^{d-2} n))$; and the decrease of the exponent of $\lg n$ is essential to obtain the tight result.

For our construction, an obvious idea is to replace each input gate in A' by a function D' that computes the And of $\lg n$ input bits that are selected at random using the random input bits.

A difficulty arises. To avoid increasing the depth of the circuit too much, it is necessary that this function D' be computable by a poly-size DNF. This allows the output Or gate of D' to be merged with the Or gates in A closest to the input. However, we do not see how to compute with these resources the naive implementation of D' that uses $\lg n$ bits of randomness for each bit to be selected. Instead, we reduce the randomness of D' from $\lg^2 n$ to $O(\lg n)$, which allows for the whole computation to be done by a poly(n)-size DNF. Specifically, rather than selecting the bits independently, we select them via a pseudorandom generator for combinatorial rectangles, and prove that the error can be tolerated in the analysis. A non-explicit construction is a straightforward application of the Chernoff bound. But previous explicit constructions are insufficient as we explain next.

A new pseudorandom generator to make the construction explicit. As hinted before, to make the upper bound explicit we use $O(\lg n)$ bits of randomness to select $\lg n$ input bits so that if the input $x \in \{0,1\}^n$ has weight $\alpha := 1/2 \pm \Theta(1/\lg^{d-1} n)$ then the And of the randomly selected $\lg n$ bits has probability of being one equal to

$$\frac{1}{n}(1 \pm \Theta(1/\lg^{d-2} n)) = \alpha^{\lg n} \pm o(1/n).$$

This amounts to constructing a pseudorandom generator for certain combinatorial rectangles. I.e., if $A \subseteq [n]$ is the set of bits of x that are 1, we need to fool the test

$$\underbrace{A \times A \times \ldots \times A}_{\lg n} \subseteq [n]^{\lg n}.$$

We stress that for the analysis we need both seed length $O(\lg n)$ and additive error $\leq 1/n$. This is not given by previous constructions: The generators for space-bounded computation [Nis92, NZ96, INW94] and the improvements for combinatorial rectangles [EGL+98, ASWZ96, Lu02] all use seed length $\geq \Omega(\lg n \lg\lg n)$ to achieve error $\leq 1/n$. Also, taking a random walk on a constant-degree expander [AKS87] (cf. [Gol10, §5.3]) one gets only error $\geq 1/n^{1-\epsilon}$.

However, we show how to get such a generator by a simple, recursive expander walk. Specifically, we start by using $O(\lg n)$ bits to perform a walk $v_1, v_2, \ldots, v_{\sqrt{\lg n}}$ of length $\sqrt{\lg n}$ on a poly(n)-size expander graph with degree $2^{O(\sqrt{\lg n})}$. Then we interpret each v_i as a random walk of length $\sqrt{\lg n}$ on an expander graph with n nodes and again degree $2^{O(\sqrt{\lg n})}$, and we output the $\lg n$ nodes. Picking expander graphs with second largest eigenvalue $\leq 2^{-\Omega(\sqrt{\lg n})}$ and using upper and lower bounds on the hitting probability of expander walks [Kah95, AFWZ95] we obtain the following theorem which allows us to make the construction explicit.

Theorem 1.3 (Rectangle generator)**.** *There is an explicit generator $G : \{0,1\}^{O(\lg n)} \to [n]^{\lg n}$ such that for any set $A \subseteq [n]$ of density $p := |A|/n \geq 0.001$, we have for all sufficiently large n, letting $G(x) = (y_1, \ldots, y_{\lg n})$:*

$$\left(p - 1/2^{\sqrt{\lg n}}\right)^{\lg n} \leq \Pr_{x \in \{0,1\}^{O(\lg n)}}[\forall i \leq \lg n : y_i \in A] \leq \left(p + 1/2^{\sqrt{\lg n}}\right)^{\lg n}.$$

In particular, if $p = |A|/n = 1/2$ *then*

$$\left| \Pr_{x \in \{0,1\}^{O(\lg n)}} [\forall i \leq \lg n : y_i \in A] - 1/n \right| \leq 1/n.$$

The generator can be generalized in a few ways, see §5.

This concludes the overview of our construction of randomized circuits.

Ruling out deterministic circuit. Our starting point for the lower bound for deterministic circuits is the result that depth-3 circuits with bottom fan-in $\leq 0.5 \lg n$ cannot tell n-bit strings of weight $\geq 2n/3$ from those of weight $\leq n/3$. This is Theorem 1 in [Vio09], and as discussed there the result can also be obtained using a switching lemma for small restrictions by Razborov [Raz03, Lemma 4.4]. We obtain the lower bound in Theorem 1.1 by combining this result with Håstad's switching lemma [Hås87].

For context, we mention that papers by Shaltiel and the author [SV10] and by Aaronson [Aar10] prove lower bounds for the problem of distinguishing i.i.d. input bits that equal 1 with probability α from i.i.d. input bits that equal 1 with probability β, for various α and β. The lower bound in this paper does not follow easily from those in [SV10, Aar10], and a qualitative difference is that the lower bounds in [SV10, Aar10] also apply to randomized circuits, while the one in this paper, like Theorem 1 in [Vio09] on which it is based, does not.

Organization. Because of Ajtai's simulation of randomized circuits of depth d by deterministic circuits of depth $d+2$ (cf. ($A\star$) in the section "The power of randomness"), to prove the two "if and only if" in Theorem 1.1 it is sufficient to prove Corollary 1.2. That is, it is sufficient to give a construction of randomized circuits and a lower bound for deterministic circuits. These are proved in § 2 and 3 respectively. In § 4 we review Ajtai's simulation for completeness. In §5 we generalize our generator for rectangles. Finally, in §6 we mention a few open problems.

2 Upper bound

In this section we prove the construction of randomized circuits in Theorem 1.1, restated next.

Theorem 2.1 (Randomized upper bound in Theorem 1.1)**.** *For every integer* $d \geq 2$ *and function* $g(n) := \Omega(1/\lg^{d-1} n)$ *there are explicit* $\mathrm{poly}(n)$*-size depth-*d *randomized circuits that tell* n*-bit strings of hamming weight* $\geq n(0.5 + g(n))$ *from strings of weight* $\leq n(0.5 - g(n))$ *with error* $\leq 1/3$*.*

The starting point is the following result which is essentially in [Ama09] and [BV10]. Later we use it with $k := d - 2$.

Lemma 2.2. *For every integer $k \geq 0$ there are* $\mathrm{poly}(n)$*-size circuits* $C : \{0,1\}^{\ell=\mathrm{poly}(n)} \to \{0,1\}$ *of depth* $k+1$ *s.t.:*
If $X^+ \in \{0,1\}^\ell$ *is a distribution of i.i.d. bits that equal* 1 *with probability* $p^+ \geq \frac{1}{n}(1 + 0.7 \cdot 10^k / \lg^k n)$*, then* $\Pr[C(X^+) = 1] \geq 2/3$*; while*
If $X^- \in \{0,1\}^\ell$ *is a distribution of i.i.d. bits that equal* 1 *with probability* $p^- \leq \frac{1}{n}(1 - 0.7 \cdot 10^k / \lg^k n)$*, then* $\Pr[C(X^-) = 1] \leq 1/3$*.*

For completeness, we prove Lemma 2.2 at the end of this section.

To prove Theorem 2.1 we need to show how given an n-bit input x of relative hamming weight $1/2 \pm \Theta(1/\lg^{d-1} n)$ we can generate independent bits with bias $\frac{1}{n}(1 \pm \Theta(1/\lg^{d-2} n))$ to feed to Lemma 2.2, using a $\mathrm{poly}(n)$-size depth-2 circuit.

To build intuition, first we give a construction that is not explicit. For simplicity, this construction will only work when the probability gap is $\geq C/\lg^{d-1} n$ for a certain constant C. Then we show an explicit construction. This explicit construction will work for any probability gap that is $\Omega(1/\lg^{d-1} n)$.

The proofs in the rest of this section use the following standard inequalities.

Fact 2.3.

$$\forall x \in R \qquad 1 + x \leq e^x; \tag{2}$$

$$\forall x \in [0,1] \qquad e^x \leq 1 + 2x \tag{3}$$

$$\forall x \in [0, 0.78] \qquad e^{-2x} \leq 1 - 1.01x \leq 1 - x; \tag{4}$$

$$\forall x \in [0, 1/2] \qquad 1 - x \geq e^{-x-x^2} = e^{-x(1+x)}; \tag{5}$$

$$\forall x \geq -1, r \geq 1 \qquad (1+x)^r \geq 1 + rx; \qquad \text{(Bernoulli's)} \tag{6}$$

$$\forall x \geq -1, r \in [0,1] \qquad (1+x)^r \leq 1 + rx; \tag{7}$$

$$\forall x, r \in R : x \cdot r \in [0,1] \qquad (1+x)^r \leq 1 + 2rx. \tag{8}$$

The parameters in Inequality (4) are not crucial for the proofs, but they are convenient and may be verified numerically. Inequality (8) follows by combining Inequalities (2) and (3).

In the remainder of the paper we sometimes write $e(x)$ for e^x. This is convenient when x is a long expression.

2.1 A non-explicit construction

We start with a standard lemma about non-explicit pseudorandom generators. For concreteness, we state it for combinatorial-rectangle tests.

Lemma 2.4 (Non-explicit generator for rectangles)**.** *For every integer $c \geq 4$ there is a collection C of n^c $\lg n$-tuples $S^1, \ldots, S^{n^c} \in [n]^{\lg n}$ such that for every set $A \subseteq [n]$ of relative hamming weight α we have:*

$$\left| \Pr_i[\forall j \leq \lg n : S^i_j \in A] - \alpha^{\lg n} \right| \leq 1/n^{c/3},$$

where S^i_j is the jth coordinate of the ith tuple.

Proof. Pick C uniformly at random. Fix any set A. Let X_i be the $0-1$ indicator variable of the event $\forall j \leq \lg n : S^i_j \in A$. Note that $\Pr_{S^i}[X_i = 1] = \alpha^{\lg n}$, and $\Pr_i[\forall j \leq \lg n : S^i_j \in A] = \sum_i X_i/n^c$. By a Chernoff bound,

$$\Pr_C\left[\left|\sum_i X_i/n^c - \alpha^{\lg n}\right| \geq \epsilon\right] \leq e(-\Omega(n^c\epsilon^2)).$$

Picking $\epsilon := 1/n^{c/3}$, the latter term becomes $e(-\Omega(n^{c/3}))$. In turn, for $c \geq 4$ and n large enough this is less than 2^{-n}. Since there are at most 2^n possible sets A, by a union bound there is a fixed choice for C that achieves the desired conclusion. □

We also use the next claim showing that shallow decision trees can be simulated in depth 2. Recall that a *decision tree* on m variables is a labeled binary tree where edges and leaves are labeled with 0 or 1, and internal nodes with variables. A decision tree computes a function in the intuitive way, starting at the root and following the path according to the values of the input variables, and outputting the value at the reached leaf. We note that a decision tree of depth s can be written as a DNF with $\leq 2^s$ terms and bottom fan-in $\leq s$, by including a term of size $\leq s$ for each of the $\leq 2^s$ paths in the tree. Analogously, it can be written as a CNF with the same parameters, by first complementing the leaves of the tree, writing that as a DNF, and then complementing the DNF to a CNF using De Morgan's law.

Claim 2.5. *Any function $f : \{0,1\}^n \to \{0,1\}$ computable by a decision tree of depth t is computable by a DNF or CNF of size $2^{O(t)}$. In particular, any function $f : \{0,1\}^n \times \{0,1\}^{O(\lg n)} \to \{0,1\}$ such that for any y the function $f(\cdot, y)$ depends only on $O(\lg n)$ bits of x is computable by a* poly(n)*-size DNF and CNF.*

Now we use the above lemmas to generate the distributions for Lemma 2.2.

Lemma 2.6. *There is depth-2 circuit $D(x, r)$ where $|x| = n, |r| = O(\lg n)$, of size* poly(n) *such that for any $1/n \leq \beta \leq 0.7/\lg n$ we have:*
$\forall x$ of hamming weight $\geq 1/2 + \beta$: $\Pr_r[D(x, r) = 1] \geq \frac{1}{n}(1 + \beta \lg n)$;
$\forall x$ of hamming weight $\leq 1/2 - \beta$: $\Pr_r[D(x, r) = 1] \leq \frac{1}{n}(1 - \beta \lg n)$.
The circuit can be written as a DNF or CNF.

Proof. Let C be a collection given by Lemma 2.4, for a sufficiently large constant c to be determined later. We let the input string r of $O(\lg n)$ bits index the tuple S^r. On input x, r, the circuit D outputs 1 if $\forall j \leq \lg n : S^r_j \in x$, i.e., if the S^r_jth bit of x is 1.

D can be implemented by a poly(n)-size DNF or CNF by Claim 2.5.

For the analysis, let x be a string of hamming weight $\geq 1/2 + \beta$. Then

$$\begin{aligned}
\Pr_r[D(x, r) = 1] &\geq (1/2 + \beta)^{\lg n} - 1/n^{c/3} && \text{(By Lemma 2.4)} \\
&= \frac{1}{n}(1 + 2\beta)^{\lg n} - 1/n^{c/3} \geq \frac{1}{n}(1 + 2\beta \lg n) - 1/n^{c/3} && \text{(By Ineq. 2.3-(6))} \\
&\geq \frac{1}{n}(1 + \beta \lg n). && \text{(Since } \beta \geq 1/n \text{ and letting } c \text{ be large enough)}
\end{aligned}$$

Conversely, let x be a string of hamming weight $\leq 1/2 - \beta$. Then

$$\begin{aligned}
\Pr_r[D(x,r)=1] &\leq (1/2-\beta)^{\lg n} + 1/n^{c/3} \qquad \text{(By Lemma 2.4)} \\
&= \frac{1}{n}(1-2\beta)^{\lg n} + 1/n^{c/3} \leq \frac{1}{n}e(-2\beta \lg n) + 1/n^{c/3} \qquad \text{(By Ineq. 2.3-(2))} \\
&\leq \frac{1}{n}(1-1.01\beta \lg n) + 1/n^{c/3} \qquad \text{(By Ineq.2.3-(4) since } \beta \lg n \leq 0.78) \\
&\leq \frac{1}{n}(1-\beta \lg n). \qquad \text{(Since } \beta \geq 1/n \text{ and letting } c \text{ be large enough).}
\end{aligned}$$

□

We can now prove the upper bound for non-explicit circuit and with slightly specialized probability gap.

Proof of Theorem 2.1 for non-explicit circuits when $g(n) := 0.7 \cdot 10^{d-2}/\lg^{d-1} n$. Combine the circuits from Lemma 2.2 with $k := d-2$ and Lemma 2.6 into a circuit of depth $(k+1)+1 = d$, by replacing inputs of the first with outputs of the latter. The transformation in Lemma 2.6 moves the biases to $\geq \frac{1}{n}(1 + 0.7 \cdot 10^{d-2}/\lg^{d-2} n)$ and $\leq \frac{1}{n}(1 - 0.7 \cdot 10^{d-2}/\lg^{d-2} n)$. This is what can be detected by the circuit of depth $d-1$ given by Lemma 2.2. □

2.2 Explicit construction

In this section we make the construction explicit. The main tool is the following new generator for certain combinatorial rectangle tests.

Theorem 1.3 (Rectangle generator)**.** (Restated.) *There is an explicit generator* $G : \{0,1\}^{O(\lg n)} \to [n]^{\lg n}$ *such that for any set* $A \subseteq [n]$ *of density* $p := |A|/n \geq 0.001$*, we have for all sufficiently large* n*, letting* $G(x) = (y_1, \ldots, y_{\lg n})$*:*

$$\left(p - 1/2^{\sqrt{\lg n}}\right)^{\lg n} \leq \Pr_{x \in \{0,1\}^{O(\lg n)}}[\forall i \leq \lg n : y_i \in A] \leq \left(p + 1/2^{\sqrt{\lg n}}\right)^{\lg n}.$$

In particular, if $p = |A|/n = 1/2$ *then*

$$\left| \Pr_{x \in \{0,1\}^{O(\lg n)}}[\forall i \leq \lg n : y_i \in A] - 1/n \right| \leq 1/n.$$

The generator in Theorem 1.3 is based on expander graphs. As is well-known, there are explicit expander graphs G_n on n nodes with second largest eigenvalue $\leq \lambda(n)$ and degree $(1/\lambda(n))^{O(1)}$, for any explicit function $\lambda(n)$. For example one can take powers of graphs with degree $O(1)$ and second largest eigenvalue $1 - \Omega(1)$.

We make use of the following standard hitting properties of walks on expander graphs.

Lemma 2.7 (Hitting properties of expander walk)**.** *Consider a regular graph such that all (normalized) eigenvalues are at most λ in absolute value, except the biggest one. Fix any subset A of the vertices that has density p. Let x be the probability that a random walk of (edge) length ℓ on the graph (started at a uniform vertex) stays inside A. We have:*

$$[Kah95] \qquad\qquad x \le p(p+(1-p)\lambda)^\ell;$$
$$[AFWZ95, Th.\ 4.2] \qquad p \ge 6\lambda \Rightarrow \qquad x \ge p(p-2\lambda)^\ell.$$

We are only going to use the weaker bounds

$$(p-2\lambda)^{\ell+1} \le x \le (p+2\lambda)^{\ell+1}$$

(the upper bound with 2 instead of $(1-p)$ is also in [AFWZ95], for large sets).

Proof of Theorem 1.3. Let $\ell := \lg n$, which we assume to be a square integer for simplicity. For a regular graph G, we write $\lambda(G)$ for the second largest eigenvalue in absolute value.

Consider an expander graph G_1 on n nodes with $\lambda(G_1) =: \alpha \le 1/2^{c\sqrt{\ell}}$ for a universal constant c to be determined later, and degree $d = (1/\alpha)^{O(1)} = 2^{O(c\sqrt{\ell})}$ which for simplicity we assume to be a power of 2. Walks of (vertex) length $\sqrt{\ell}$ are in $1-1$ correspondence with bit-strings of length $\ell + (\sqrt{\ell}-1)\cdot \lg(d) =: w = O(\ell)$, i.e. elements of $[2^w] =: W$. Note $|W| = \mathrm{poly}(n)$.

Now let $\gamma := p/2^c$ and consider another expander graph G_2 on $|W|$ nodes with $\lambda(G_2) \le \gamma^{\sqrt{\ell}}$ and degree $1/\gamma^{O(\sqrt{\ell})}$. Walks of length $\sqrt{\ell}$ on G_2 can be described with $\le w + \sqrt{\ell}\cdot \lg\left(1/\gamma^{O(\sqrt{\ell})}\right) = O(\ell)$ bits, for any fixed γ.

The generator G is defined as follows. Use the input to specify a walk of length $\sqrt{\ell}$ on G_2. Let $v_1, \ldots, v_{\sqrt{\ell}}$ be the steps in the walk. Now interpret each v_i as a walk of length $\sqrt{\ell}$ in G_1, and output the corresponding $\sqrt{\ell}\cdot\sqrt{\ell} = \ell$ elements in $[n]$.

Analysis. For any set $A \subseteq [n]$, let $B \subseteq W$ be the set of walks of length $\sqrt{\ell}$ in G_1 that stay inside A. Let $\beta := |B|/|W|$ and

$$m := \sqrt{\ell}.$$

By Lemma 2.7, recalling $\alpha = \lambda(G_1) \le 1/2^{c\sqrt{\ell}} \le p/6$, we have:

$$(p-2\alpha)^m \le \beta \le (p+2\alpha)^m.$$

Using the fact that p is bounded away from 0, that $\alpha \le 1/2^{cm}$, that m is sufficiently large, and Inequalities 2.3.(8) and 2.3.(6):

$$\beta \le (p+2\alpha)^m = p^m(1+2\alpha/p)^m \le p^m(1+4m\alpha/p) \le p^m(1+\sqrt{\alpha});$$
$$\beta \ge (p-2\alpha)^m = p^m(1-2\alpha/p)^m \ge p^m(1-2m\alpha/p) \ge p^m(1-\sqrt{\alpha}).$$

Now the probability τ that all outputs of G stay inside A is the probability that all steps of a random walk in G_2 stay inside B. By the upper bound in Lemma 2.7, using Inequality

2.3.(8) we have, recalling $\gamma := p/2^c$:

$$\begin{aligned}\tau &\le \left(p^m(1+\sqrt{\alpha}) + 2\gamma^m\right)^m \\ &\le p^\ell(1 + 1/2^{cm/2} + 2/2^{cm})^m \\ &\le p^\ell(1 + 3/2^{cm/2})^m \\ &\le p^\ell(1 + 6m/2^{cm/2}) \\ &\le p^\ell(1 + 1/2^{cm/4}).\end{aligned}$$

And this concludes the proof of the upper bound because c is arbitrary and for $q := 1/2^{cm/4}$ we have $p^\ell(1+q) \le p^\ell(1+q)^\ell \le (p+q)^\ell$.

For the lower bound, we start by noting that we can apply the lower bound in Lemma 2.7 because

$$\beta \ge p^m(1-\sqrt{\alpha}) \ge 0.5p^m \ge 6\gamma^m = 6(p/2^c)^m.$$

Hence, using that bound and Bernoulli's Inequality 2.3.(6) we derive

$$\begin{aligned}\tau &\ge \left(p^m(1-\sqrt{\alpha}) - 2\gamma^m\right)^m \\ &\ge p^\ell(1 - 1/2^{cm/2} - 2/2^{cm})^m \\ &\ge p^\ell(1 - 3/2^{cm/2})^m \\ &\ge p^\ell(1 - 3m/2^{cm/2}) \\ &\ge p^\ell(1 - 1/2^{cm/4}).\end{aligned}$$

And again this concludes the proof of the lower bound because c is arbitrary and for $q := 1/2^{cm/4}$ we have $p^\ell(1-q) \ge p^\ell(1-q)^\ell \ge (p-q)^\ell$. □

To get a construction for any probability gap $g(n) = \Omega(1/\lg^{d-1} n)$ (as opposed to $g(n) \ge C/\lg^{d-1} n$ for some large C as in §2.1), we also need to boost the probability gap by a constant. This is provided by the following lemma using recursive majorities.

Lemma 2.8 (Boosting gap by a constant)**.** *For any $c > 0$ there is $k > 0$ such that for all n:*
There is an explicit map $G : \{0,1\}^n \times \{0,1\}^{k \lg n} \to \{0,1\}$ such that
(1) For every y, $G(x,y)$ depends on $\le k$ bits of x;
(2) For every $x \in \{0,1\}^n$ such that the weight of x is $\ge 1/2 + \epsilon$ ($\le 1/2 - \epsilon$) for $\epsilon \le 1/k$ we have $\Pr_y[G(x,y) = 1] \ge 1/2 + c\epsilon$ ($\Pr_y[G(x,y) = 1] \le 1/2 - c\epsilon$).

Proof. We let $k := 3^t$ for a value t depending only on c and define G to pick k independent bits of x and output the recursive-majority-of-three of the bits. We just need to verify that majority-of-three "amplifies." Indeed, consider the majority of three i.i.d. bits x_1, x_2, x_3 coming up 1 with probability $1/2 + \beta$, where $\beta \in [-1/2, 1/2]$. We have:

$$\begin{aligned}\Pr[\mathrm{maj}(\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3) = 1] &= (1/2+\beta)^3 + 3(1/2+\beta)^2(1/2-\beta) \\ &= 1/2 + \beta(3/2 - \beta^2).\end{aligned}$$

Thus, when $|\beta|$ is small enough this results in multiplying β by a constant bigger than 1, say 1.1. So we can set $t := \lg_{1.1} c$ and for all sufficiently small β achieve (1) and (2). □

We can now state and prove the construction of explicit DNF (or CNF).

Lemma 2.9. *Let $\epsilon(n) := \alpha/\lg^b n$ for some $\alpha > 0$ and $b \geq 1$. There are explicit* poly(n)*-size DNF (or CNF) $D : \{0,1\}^n \times \{0,1\}^{O(\lg n)} \to \{0,1\}$ such that*

(1) for any string $x \in \{0,1\}^n$ of hamming weight $\geq 1/2 + \epsilon$ we have $\Pr_y[D(x,y) = 1] \geq \frac{1}{n}(1 + 0.7 \cdot 10^{b-1}/\lg^{b-1} n)$;

(2) for any string $x \in \{0,1\}^n$ of weight $\leq 1/2 - \epsilon$ we have $\Pr_y[D(x,y) = 1] \leq \frac{1}{n}(1 - 0.7 \cdot 10^{b-1}/\lg^{b-1} n)$.

Proof. Let $\ell := \lg n$. Consider the map G from Lemma 2.8 with a sufficiently large c depending on α, and let $k = O(1)$ be the associated constant guaranteed by the lemma. Note that the hypothesis $\epsilon \leq 1/k$ in that lemma is satisfied, because $\epsilon = o(1)$. Now consider the function $f : \{0,1\}^n \times \{0,1\}^{O(k\ell)=O(\ell)} \to \{0,1\}$ that on input (x,y) uses the generator from Theorem 1.3 on input y to select ℓ random choices $y_1, \ldots, y_\ell$ for G, and then outputs 1 iff for all $i \leq \ell$ we have $G(x, y_i) = 1$.

Note for every y this function just depends on $\ell \cdot c = O(\lg n)$ bits of x. Hence by Claim 2.5 f can be written as a DNF or CNF of size poly(n).

Now fix any x of weight $\geq 1/2 + \epsilon$. For large enough c, we have:

$$\begin{aligned}
\Pr_y[f(x,y) = 1] &\geq \left(1/2 + c\epsilon - 1/2^{\sqrt{\ell}}\right)^\ell \\
&\geq \left(1/2 + (c-1)\alpha/\ell^b\right)^\ell \\
&\geq \left(1/2 + 0.7 \cdot 10^{b-1}/\ell^b\right)^\ell \\
&\geq \frac{1}{n}(1 + 0.7 \cdot 10^{b-1}/\ell^{b-1}) \qquad \text{(By Inequality 2.3.(6))}.
\end{aligned}$$

Similarly, fix any x of weight $\leq 1/2 - \epsilon$. For large enough c, we have:

$$\begin{aligned}
\Pr_y[f(x,y) = 1] &\leq \left(1/2 - c\epsilon + 1/2^{\sqrt{\ell}}\right)^\ell \\
&\leq \left(1/2 - (c-1)\alpha/\ell^b\right)^\ell \\
&\leq \frac{1}{n}(1 - 2 \cdot 0.7 \cdot 10^{b-1}/\ell^b)^\ell \\
&\leq \frac{1}{n} e(-2 \cdot 0.7 \cdot 10^{b-1}/\ell^{b-1}).
\end{aligned}$$

Now, if $b > 1$ the argument of $e(\cdot)$ goes to 0 with n and so we conclude by Inequality 2.3.(4) that $\Pr_y[f(x,y) = 1] \leq \frac{1}{n}(1 - 0.7 \cdot 10^{b-1}/\ell^{b-1})$.

Otherwise, if $b = 1$ then we get $\Pr_y[f(x,y) = 1] \leq \frac{1}{n} e(-2 \cdot 0.7) = \frac{1}{n}(0.2465\ldots) < \frac{1}{n}(1 - 0.7)$. □

We can now prove the upper bound in the same way as we proved a special case at the end of the previous subsection.

Proof of Theorem 2.1. Combine the circuits from Lemma 2.2 with $k := d-2$ and Lemma 2.9 into a circuit of depth $(k+1)+1 = d$. The transformation in Lemma 2.9 moves the biases to $\geq \frac{1}{n}(1+0.7\cdot 10^{d-2}/\lg^{d-2} n)$ and $\leq \frac{1}{n}(1-0.7\cdot 10^{d-2}/\lg^{d-2} n)$. This is what can be detected by the circuit of depth $d-1$ given by Lemma 2.2. □

2.3 Proof of Lemma 2.2

For completeness in this section we give the proof of the following lemma which is essentially in [Ama09, BV10].

Lemma 2.2. *For every integer $k \geq 0$ there are* poly(n)*-size circuits $C : \{0,1\}^{\ell=\mathrm{poly}(n)} \to \{0,1\}$ of depth $k+1$ s.t.:*
If $X^+ \in \{0,1\}^\ell$ is a distribution of i.i.d. bits that equal 1 with probability $p^+ \geq \frac{1}{n}(1+0.7\cdot 10^k/\lg^k n)$, then $\Pr[C(X^+)=1] \geq 2/3$; while
If $X^- \in \{0,1\}^\ell$ is a distribution of i.i.d. bits that equal 1 with probability $p^- \leq \frac{1}{n}(1-0.7\cdot 10^k/\lg^k n)$, then $\Pr[C(X^-)=1] \leq 1/3$.

Proof of Lemma 2.2. We proceed by induction on k.

Base case $k=0$. C consists of one And gate on the complement of n bits. Then

$$\Pr[C(X^+)=1] = (1-p^+)^n \leq (1-\frac{1}{n}(1+0.7))^n \leq e(-(1+0.7)) \leq 1/3; \qquad \text{(By Ineq. 2.3-(2))}$$
$$\Pr[C(X^-)=1] = (1-p^-)^n \geq (1-\frac{1}{n}(1-0.7))^n \geq 1-(1-0.7) = 0.7 \geq 2/3. \qquad \text{(By Ineq. 2.3-(6))}$$

Complementing the circuit concludes the proof of this case.

Induction step $k > 1$. C consists of the circuit C' for $k-1$ where each input gate is replaced with the circuit, denoted by D, which is one And gate on the complement of $\lfloor n \lg_e n \rfloor$ bits. The inputs to these gates are disjoint. All we need to verify is that D amplifies the bias to the value that can be detected by C'. Let $c := 0.7\cdot 10^k$. We have

$$\begin{aligned}
\Pr[D(X^-)=1] &= (1-p^-)^{\lfloor n\lg_e n\rfloor} \geq (1-\frac{1}{n}(1-c/\lg^k n))^{n\lg_e n} \\
&\geq e\left(-n\lg_e n\left[\frac{1}{n}(1-c/\lg^k n)\right]\left[1+\frac{1}{n}(1-c/\lg^k n)\right]\right) \qquad \text{(By Ineq. 2.3-(5))} \\
&\geq e\left(-\lg_e n(1-c/\lg^k n)\left[1+\frac{1}{n}\right]\right) \\
&\geq e(-\lg_e n(1-0.5c/\lg^k n)) \\
&\geq \frac{1}{n}e(0.5c\lg_e n/\lg^k n) \\
&\geq \frac{1}{n}(1+0.5c/(\lg_2 e\lg^{k-1} n)) \qquad \text{(By Ineq. 2.3-(2))} \\
&\geq \frac{1}{n}(1+0.7\cdot 10^{k-1}/\lg^{k-1} n).
\end{aligned}$$

Conversely:

$$\begin{aligned}
\Pr[D(X^+)=1] &= (1-p^+)^{\lfloor n\lg_e n\rfloor} \le (1-\frac{1}{n}(1+c/\lg^k n))^{\lfloor n\lg_e n\rfloor} \\
&\le e\left(-(\lfloor n\lg_e n\rfloor/n)(1+c/\lg^k n)\right) \qquad \text{(By Ineq. 2.3-2)} \\
&\le e\left(-(\lg_e n - 1/n)(1+c/\lg^k n)\right) \\
&\le e(-\lg_e n - c\lg_e n/\lg^k n + 2/n) \\
&\le \frac{1}{n}e(-c\lg_e n/\lg^k n)\cdot e(2/n) \\
&\le \frac{1}{n}e(-c\lg_e 2/\lg^{k-1} n)(1+4/n).
\end{aligned}$$

If $k=1$ then we get $\Pr[D(X^+)=1] \le \frac{1}{n}e(-0.7\cdot 10\cdot \lg_e 2)(1+4/n) \le \frac{1}{n}(1-0.7)$.

If $k>1$ then the argument to $e(\cdot)$ is small and we can use Inequality 2.3-(4) to get

$$\begin{aligned}
\Pr[D(X^+)=1] &\le \frac{1}{n}(1-0.5\cdot c\cdot \lg_e 2/\lg^{k-1} n)(1+4/n) \\
&\le \frac{1}{n}(1-0.49\cdot 0.7\cdot 10^k\cdot \lg_e 2/\lg^{k-1} n) \\
&\le \frac{1}{n}(1-0.7\cdot 10^{k-1}/\lg^{k-1} n).
\end{aligned}$$

Either way, $\Pr[D(X^+)=1] \le \frac{1}{n}(1-0.7\cdot 10^{k-1}/\lg^{k-1} n)$.

This achieves the desired bias amplification, except that X^+ and X^- are "swapped." Complementing the circuit C fixes this. Finally, inspection reveals that for any k the size of the circuits (and in particular their input length) is polynomial in the parameter n. □

3 Lower bound

In this section we prove the negative result for deterministic circuits in Theorem 1.1:

Theorem 3.1. *For any $d\ge 3, c\ge 1$ there is $\epsilon=\epsilon(d,c)$ such that for sufficiently large n: Depth-d size-n^c circuits cannot tell n-bit strings of hamming weight $\ge n(0.5+\epsilon/\lg^{d-3} n)$ from strings of weight $\le n(0.5-\epsilon/\lg^{d-3} n)$.*

To get a sense of the parameters, we note that for $d=3$ the dependence of ϵ on c is necessary, as can be verified using the arguments in [Ajt83] or [Vio09].

Theorem 3.1 implies that for any $d\ge 3$ and any function $g(n)=o(1/\lg^{d-3} n)$, poly(n)-size circuits of depth d cannot tell n-bit strings of hamming weight $\ge n(0.5+g(n))$ from strings of weight $\le n(0.5-g(n))$.

Recall that a *restriction* on m variables $x_1,x_2,\ldots,x_m$ is a map $\rho:\{x_1,x_2,\ldots,x_m\}\to\{0,1,*\}$. For a function $f:\{0,1\}^m\to\{0,1\}$ we denote by $f|_\rho$ the function we get by doing the substitutions prescribed by ρ. $f|_\rho$ will be a function of the variables that were given the value * by ρ. Let $R_m^{\delta\cdot m}$ denote the uniform distribution on restrictions on m variables assigning exactly δm variables to $*$, and assigning random values to the others.

Lemma 3.2 (Switching lemma [Hås87, Bea94])**.** *Let φ be a DNF or a CNF formula in m variables with bottom fan-in at most r. For every $s \geq 0, p < 1/7$, the probability over $\rho \in R_m^{p \cdot m}$ that the function computed by $\varphi|_\rho$ is not computable by a decision tree of height strictly less than s is less than $(7pr)^s$.*

We prove Theorem 3.1 in two stages. First we prove it under the additional assumption that the circuits have bottom fan-in $\leq 0.5 \lg n$. This is the next lemma. Then we get rid of the assumption on the bottom fan-in.

Lemma 3.3. *For any $d \geq 3, c \geq 1$ there is $\epsilon = \epsilon(d, c)$ such that for sufficiently large n: Depth-d size-n^c circuits with bottom fan-in $\leq 0.5 \lg n$ cannot tell n-bit strings of hamming weight $\geq n(0.5 + \epsilon/\lg^{d-3} n)$ from strings of weight $\leq n(0.5 - \epsilon/\lg^{d-3} n)$.*

Proof of Lemma 3.3.

Induction on d.

For $d = 3$ and any c, one can take $\epsilon = 1/6$ [Vio09, Theorem 1].

Fix any $d > 3$ and any c. Let C be a circuit of depth d and size n^c. Let ρ be a random restriction on n bits assigning exactly pn bits to $*$, where

$$p := 1/(4^{c+2} \lg n),$$

which we assume to be even. We want to show that simultaneously two things happen:
(1) $C|_\rho$ is a circuit of depth $d-1$, size $\leq n^{c+1}$, and bottom fan-in $\leq 0.5 \lg n$, and
(2) the bits fixed by ρ are balanced, i.e., exactly $(n - pn)/2$ bits are set to 1 by ρ. The probability that (2) happens is $\geq \Omega(1/\sqrt{n - pn}) \geq \Omega(1/\sqrt{n})$ (standard approximation of the central binomial coefficients).

The probability that (1) does not happen is at most the probability that some of the $\leq n^c$ gates at distance two from the input cannot be written as a decision tree of depth $\leq 0.5 \lg n$ (cf. the comment after Lemma 3.2 about writing such a decision tree as a CNF or a DNF with bottom fan-in $\leq 0.5 \lg n$ and size $\leq \sqrt{n}$). By Lemma 3.2, this probability is at most

$$n^c (7p 0.5 \lg n)^{0.5 \lg n} \leq n^c \left(4 \frac{\lg n}{4^{c+2} \lg n} \right)^{0.5 \lg n} \leq 1/n.$$

So there exists ρ satisfying both (1) and (2). Fix such a ρ, and let $C' := C|_\rho$. C' is a circuit of depth $d-1$ on

$$m := pn \geq n^{0.9}$$

input bits of size $\leq n^{c+1} \leq n^{2c} \leq m^{3c}$. By induction hypothesis, there is a constant $\epsilon(d-1, 3c)$ such that for sufficiently large m, C' cannot distinguish m-bit strings of hamming weight $\geq m(0.5 + \epsilon(d-1, 3c)/\lg^{d-1-3} m)$ from strings of weight $\leq m(0.5 - \epsilon(d-1, 3c)/\lg^{d-1-3} m)$. Without loss of generality, consider an m-bit string of hamming weight $\geq m(0.5 + \epsilon(d - 1, 3c)/\lg^{d-1-3} m)$ on which C' mistakenly outputs 0. Since the restriction ρ sets to 1 exactly

$(n-pn)/2 = (n-m)/2$ bits, note that this output of C' is the same as the output of the original circuit C on an n-bit input of hamming weight at least

$$(n-m)/2 + m(0.5 + \epsilon(d-1,3c)/\lg^{d-1-3} m) \geq n\left(0.5 + \frac{p\epsilon(d-1,3c)}{\lg^{d-1-3} n}\right) \qquad \text{(since } n \geq m\text{)}$$
$$= n\left(0.5 + \frac{\epsilon(d-1,3c)}{4^{c+2}\lg^{d-3} n}\right).$$

This concludes the proof for $\epsilon(d,c) := \epsilon(d-1,3c)/4^{c+2}$. □

Proof of Theorem 3.1. Fix any $d \geq 3$ and any c, and let C be a circuit of depth d and size n^c, which we view as a circuit of depth $d+1$ with bottom fan-in 1. Let ρ be a random restriction on n bits assigning exactly $m := pn$ bits to $*$, where

$$p := 1/(7 \cdot 4^{c+1}).$$

As in the previous proof, with non-zero probability both the bits fixed by ρ are balanced, and $C|_\rho$ can be written as a circuit of depth d, size $\leq n^{c+1}$, and bottom fan-in $\leq 0.5 \lg n$.

By the previous proof, for some ϵ' which depends on d and c, for large enough m the circuit $C' := C|_\rho$ cannot distinguish m-bit strings of hamming weight $\geq m(0.5+\epsilon(d,c+1)/\lg^{d-3} m)$ from strings of weight $\leq m(0.5 - \epsilon(d,c+1)/\lg^{d-3} m)$. Without loss of generality, suppose C' gives the wrong answer on a string of weight $\geq m(0.5 + \epsilon(d,c+1)/\lg^{d-3} m) \geq m(0.5 + \epsilon(d,c+1)/\lg^{d-3} n)$.

Therefore C gives the wrong answer on a string of weight

$$0.5(n-m) + m(0.5 + \epsilon'/\lg^{d-3} n) = n(0.5 + p\epsilon'/\lg^{d-3} n).$$

This concludes the proof for $\epsilon(d,c) := p\epsilon' = \epsilon'/(7 \cdot 4^{c+1})$. □

4 Ajtai's construction

In this section for completeness we include a proof of the following result by Ajtai.

Theorem 4.1 ([Ajt83])**.** *For any $n, \epsilon > 0$, there is a deterministic, monotone, And-Or-And circuit of size $n^{O(1/\epsilon)}$ that distinguishes n-bit strings with hamming weight $\geq 1/2 + \epsilon$ from weight $\leq 1/2 - \epsilon$.*

In particular, any function $f : \{0,1\}^n \to \{0,1\}$ computable by randomized, poly(n)*-size circuits of depth d with error $1/3$ can be computed by (deterministic)* poly(n)*-size circuits of depth $d+2$.*

Proof. We give a probabilistic construction that for every fixed input x has error probability $< 2^{-n}$. One can then apply a union bound.

Pick the And close to the input at random over $c\lg(n)/\epsilon$ variables, for a large enough c. Then the middle Or has fan-in $n^{c/\epsilon}$, and the output And has fan-in n. We write $A(x), OA(x)$,

and $AOA(x)$ for the output of a gate at distance $1, 2$, and 3 from the input x. We also make use of the inequalities in Fact 2.3.

Let x be any input with weight $\geq 1/2 + \epsilon$:

$$\begin{aligned}
\Pr[A(x) = 1] &= (1/2 + \epsilon)^{c \lg(n)/\epsilon} = n^{-c/\epsilon}(1 + 2\epsilon)^{c \lg(n)/\epsilon} \geq n^{-c/\epsilon} e^{c \lg n}; \\
\Pr[OA(x) = 1] &\geq 1 - \left(1 - n^{-c/\epsilon} e^{c \lg n}\right)^{n^{c/\epsilon}} \geq 1 - \left(e^{-e^{c \lg n}}\right); \\
\Pr[AOA(x) = 1] &\geq \left(1 - e^{-e^{c \lg n}}\right)^n \geq 1 - n e^{-e^{c \lg n}} \geq 1 - 2^{-n}
\end{aligned}$$

for c, n sufficiently large.

Conversely, let x be any input with weight $\leq 1/2 - \epsilon$:

$$\begin{aligned}
\Pr[A(x) = 1] &= (1/2 - \epsilon)^{c \lg(n)/\epsilon} = n^{-c/\epsilon}(1 - 2\epsilon)^{c \lg(n)/\epsilon} \leq n^{-c/\epsilon} e^{-2c \lg n}; \\
\Pr[OA(x) = 1] &\leq 1 - \left(1 - n^{-c/\epsilon} e^{-2c \lg n}\right)^{n^{c/\epsilon}} \leq e^{-2c \lg n}; \\
\Pr[AOA(x) = 1] &\leq \left(e^{-2c \lg n}\right)^n \leq 2^{-n}.
\end{aligned}$$

The "in particular" part is obtained as follows. Run in parallel bn copies of the circuit with independent choices for the random coins, for a large enough universal constant b. By a Chernoff bound, you can fix the randomness so that for any input at least $2/3$ of the outputs equal $f(x)$. Thus feeding these $O(n)$ outputs to the circuit constructed in the first part of the theorem we compute f correctly. By collapsing adjacent layers of gates, the final circuit has depth $d+2$ in case each output was and And gate. If not, complement the circuit, apply this construction, and complement again. □

5 A more general generator

In this section we prove the following theorem.

Theorem 5.1. *There is an explicit algorithm that given m, d, and $p \leq 1/2$, uses a seed x of length $O(\lg m + \lg d + pd)$ to produce $(y_1, \ldots, y_d) \in [m]^d$ such that for any d sets $A_1, \ldots, A_d \subseteq [m]$ of density $1 - p = |A_i|/m$ each, we have*

$$\Pr_x[\forall i \leq d : y_i \in A_i] = (1 - p)^d (1 \pm 100^{-(pd)^{2/3}}).$$

We recover the "in particular" part of Theorem 1.3 by setting $m = n$, $d = \lg n$, and $p = 1/2$. However now we can also have, say, $m = n$, $d = n \lg n$, and $p = 1/n$, and still use seed length $O(\lg n)$ to obtain an error $(1 \pm o(1))$ which is multiplicative in $(1-p)^d \leq e^{-\lg n} \leq 1/n$ and hence, in additive terms, exponentially small in the seed length.

Proof. Let $\ell := pd$. Divide the d dimensions in $\ell^{1/3}$ blocks of $d/\ell^{1/3}$ dimensions each. In each block, use Lu's generator [Lu02] with (additive) error $c^{-\ell^{2/3}}$ for a sufficiently large constant c to be determined later. This generator uses seed length $h = O(\lg m + \lg d + \ell)$.

Now produce the $\ell^{1/3}$ seeds for this generator by an expander walk over the space $\{0,1\}^h$ of seeds, of length $\ell^{1/3}$ and with largest second eigenvalue $\leq c_2^{-\ell^{2/3}}$, for a universal constant c_2 to be determined later. This takes seed length $h + \ell^{1/3}O(\ell^{2/3}) = O(\lg m + \lg d + \ell)$, as desired.

The set to hit has measure $(1-p)^{d/\ell^{1/3}} \pm c^{-\ell^{2/3}} = e^{-\Theta(\ell^{2/3})}$ (using $p \leq 1/2$), which is at least six times the second largest eigenvalue by setting c_2 appropriately. Hence by Lemma 2.7 we can bound the hitting probability as

$$\left((1-p)^{d/\ell^{1/3}} \pm c^{-\ell^{2/3}} \pm 2c_2^{-\ell^{2/3}}\right)^{\ell^{1/3}} = (1-p)^d(1 \pm c_3^{-\ell^{2/3}})^{\ell^{1/3}} = (1-p)^d(1 \pm c_4^{-\ell^{2/3}}),$$

for appropriate constants c_3, c_4 which can be chosen arbitrarily large by increasing c and c_2. In the first equality we used again that $(1-p)^{d/\ell^{1/3}} = e^{-\Theta(\ell^{2/3})}$ because $p \leq 1/2$. □

6 Open problems

We now list a few open problems.

(1) Make the deterministic circuits explicit. This is only known for depth 3 [Vio09].

(2) Prove that small depth-3 circuits that tell weights $1/2 \pm \epsilon$ require bottom fan-in $\Omega(\lg(n)/\epsilon)$. This seems to require new techniques (neither the switching lemma nor [Vio09] gives this) and should imply a tight size lower bound in the superpolynomial regime.

(3) Obtain a generator like Theorem 1.3 but with additive error $1/n^2$ instead of $1/n$, or with the same error $1/n$ but to fool sets of relative size $1 - 1/\lg n$ instead of $1/2$ (so that the rectangle has constant relative size).

(4) Obtain a separation like Corollary 1.2 for non-promise problems.

Acknowledgments. We thank Kord Eickmeyer for helpful initial discussions on this problem, and Joshua Brody and Elad Verbin for sending us a preliminary version of [BV10]. We are also grateful to Oded Goldreich and the anonymous referees for helpful comments on the write-up.

References

[Aar10] Scott Aaronson. BQP and the polynomial hierarchy. In *42nd ACM Symp. on the Theory of Computing (STOC)*, pages 141–150. ACM, 2010.

[ABO84] Miklós Ajtai and Michael Ben-Or. A theorem on probabilistic constant depth computation. In *16th ACM Symp. on the Theory of Computing (STOC)*, pages 471–474, 1984.

[Adl78] Leonard Adleman. Two theorems on random polynomial time. In *19th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 75–83. 1978.

[AFWZ95] Noga Alon, Uriel Feige, Avi Wigderson, and David Zuckerman. Derandomized graph products. *Computational Complexity*, 5(1):60–75, 1995.

[Agr01] Manindra Agrawal. Hard sets and pseudo-random generators for constant depth circuits. In *21st Foundations of Software Technology and Theoretical Computer Science*, pages 58–69. Springer-Verlag, 2001.

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

[Ajt93] Miklós Ajtai. Approximate counting with uniform constant-depth circuits. In *Advances in computational complexity theory*, pages 1–20. Amer. Math. Soc., Providence, RI, 1993.

[AKS87] Miklós Ajtai, János Komlós, and Endre Szemerédi. Deterministic simulation in logspace. In *19th ACM Symp. on the Theory of Computing (STOC)*, pages 132–140, 1987.

[Ama09] Kazuyuki Amano. Bounds on the size of small depth circuits for approximating majority. In *36th Coll. on Automata, Languages and Programming (ICALP)*, pages 59–70. Springer, 2009.

[ASWZ96] Roy Armoni, Michael E. Saks, Avi Wigderson, and Shiyu Zhou. Discrepancy sets and pseudorandom generators for combinatorial rectangles. In *37th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 412–421, 1996.

[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/.

[BV10] Joshua Brody and Elad Verbin. The coin problem, and pseudorandomness for branching programs. In *51th IEEE Symp. on Foundations of Computer Science (FOCS)*, 2010.

[CR96] Shiva Chaudhuri and Jaikumar Radhakrishnan. Deterministic restrictions in circuit complexity. In *28th ACM Symp. on the Theory of Computing (STOC)*, pages 30–36, 1996.

[EGL+98] Guy Even, Oded Goldreich, Michael Luby, Noam Nisan, and Boban Velickovic. Efficient approximation of product distributions. *Random Struct. Algorithms*, 13(1):1–16, 1998.

[ESY84] Shimon Even, Alan L. Selman, and Yacov Yacobi. The complexity of promise problems with applications to public-key cryptography. *Inform. and Control*, 61(2):159–173, 1984.

[Gol10] Oded Goldreich. *Pseudorandom Generators: A Primer*, volume 55 of *University Lecture Series*. AMS, 2010.

[GS86] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *18th ACM Symposium on Theory of Computing (STOC)*, pages 59–68, 1986.

[Hås87] Johan Håstad. *Computational limitations of small-depth circuits*. MIT Press, 1987.

[INW94] Russell Impagliazzo, Noam Nisan, and Avi Wigderson. Pseudorandomness for network algorithms. In *26th ACM Symp. on the Theory of Computing (STOC)*, pages 356–364, 1994.

[IW97] Russell Impagliazzo and Avi Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *29th ACM Symp. on the Theory of Computing (STOC)*, pages 220–229. ACM, 1997.

[Kah95] Nabil Kahale. Eigenvalues and expansion of regular graphs. *J. of the ACM*, 42(5):1091–1106, 1995.

[Lau83] Clemens Lautemann. BPP and the polynomial hierarchy. *Information Processing Letters*, 17(4):215–217, 1983.

[Lu02] Chi-Jen Lu. Improved pseudorandom generators for combinatorial rectangles. *Combinatorica*, 22(3):417–433, 2002.

[Nis92] Noam Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.

[NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *J. of Computer and System Sciences*, 52(1):43–52, February 1996.

[Raz03] Alexander Razborov. Pseudorandom generators hard for k-dnf resolution and polynomial calculus resolution, 2002-2003. Manuscript. Available from http://www.mi.ras.ru/∼razborov/.

[RW91] Prabhakar Ragde and Avi Wigderson. Linear-size constant-depth polylog-treshold circuits. *Inf. Process. Lett.*, 39(3):143–146, 1991.

[Sip83] Michael Sipser. A complexity theoretic approach to randomness. In *15th ACM Symposium on Theory of Computing*, pages 330–335, Boston, Massachusetts, 25–27 April 1983.

[Sto83] Larry Stockmeyer. The complexity of approximate counting. In *15th Symposium on Theory of Computing (STOC)*, pages 118–126. ACM, 1983.

[Sto85] Larry Stockmeyer. On approximation algorithms for #P. *SIAM J. on Computing*, 14(4):849–861, November 1985.

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

[Vio04] Emanuele Viola. The complexity of constructing pseudorandom generators from hard functions. *Computational Complexity*, 13(3-4):147–188, 2004.

[Vio09] Emanuele Viola. On approximate majority and probabilistic time. *Computational Complexity*, 18(3):337–375, 2009.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de