Electronic Colloquium on Computational Complexity, Report No. 137 (2021)

Affine extractors and AC0-Parity

Xuangui Huang[*] Peter Ivanov[*] Emanuele Viola[*]

September 14, 2021

Abstract

We study a simple and general template for constructing affine extractors by composing a linear transformation with resilient functions. Using this we show that good affine extractors can be computed by non-explicit circuits of various types, including AC0-Xor circuits: AC0 circuits with a layer of parity gates at the input. We also show that one-sided extractor can be computed by small DNF-Xor circuits, and separate these circuits from other well-studied classes. As a further motivation for studying DNF-Xor circuits we show that if they can approximate inner product then small AC0-Xor circuits can compute it exactly – a long-standing open problem.

AC0 with parity gates is a frontier class in circuit complexity, essentially the strongest class for which we can prove strong lower bounds for explicit functions. These lower bounds however have been stuck since the classic results from the 80's [Raz87, Smo87]. In particular, unlike the case of AC0, we do not have (1) strong average-case lower bounds, (2) pseudorandom generators, or (3) hierarchy results for this class.

Remarkably, (1), (2), and (3) are not known even for the subclass AC0-Xor of AC0 circuits with a layer of parity gates (or their negations) next to the input level. (On the other hand, (1) and (2) are known for Xor-AC0 [Vio07].) In fact, (1) and (2) are not known even for Or-And-Xor circuits, a.k.a. DNF-Xor circuits. Hence these classes (AC0-Xor and DNF-Xor) have gained importance as prominent special cases of AC0 with parity gates which require new proof techniques.

A natural candidate for providing (1) is the inner product function, and the following question has been highlighted and studied in several works, including [SV12, CS16, ABG$^+$14, CGJ$^+$18, RDR21].

Problem 1. Is the Inner Product function $\mathrm{IP}(x,y) := \sum_i x_i y_i \mod 2$ computable by polynomial-size AC0-Xor circuits?

A number of works have solved special cases of the problem, proving lower bounds for computing IP when the circuit class is further restricted: [Juk06, CS16] prove exponential lower bounds for Or-And-Xor. [ABG$^+$14] prove a lower bound for small AC0-Xor circuits when the parity layer is "typical." [CGJ$^+$18] (cf. [LN90]) prove an $n^{2-o(1)}$ lower bound for And-Or-And-Xor circuits. For depth-d AC0-Xor circuits they prove an $n^{1+\Omega(1/4^d)}$ lower

[*]Supported by NSF grants CCF-1813930 and CCF-2114116.

ISSN 1433-8092

bound. The latter result is improved on in [BKT19] who obtain an $\Omega(n^{1+1/2^d})$ lower bound which holds even if the circuit computes IP on a $1/2 + n^{-\log n}$ fraction of inputs.

To summarize, to compute IP there are quadratic lower bounds for And-Or-And-Xor. These lower bounds hold in the worst case while average-case lower bounds are not known. Average-case lower bounds are not even known for polynomial-size DNF-Xor. For higher depth we have lower bounds for size which approaches linear exponentially fast with the depth, and these lower bounds hold even in the average case.

Extractors. An extractor for a class of distributions (a.k.a. source) is a function that is nearly unbiased when the input is chosen according to any distribution in the class. For various classes of distributions, extractors have been studied with remarkable intensity in the theoretical computer science literature for decades. A class of distributions which is important in many works including the present one is that of distributions which are uniform over *linear* or *affine vector subspaces* of $\{0,1\}^n$, which we simply call *affine.*

Definition 2. [Affine extractors] A function $f : \{0,1\}^n \to \{0,1\}$ is an *affine extractor* for dimension (a.k.a. entropy) k with error (a.k.a. bias) ϵ if for every k-dimensional affine space $A \subseteq \{0,1\}^n$ and for U_A the uniform distribution over A we have $|\mathbb{P}[f(A) = 1] - \mathbb{P}[f(A) = 0]| \leq \epsilon$.

We say that the extractor is *one-sided* if the conclusion is relaxed to $\mathbb{P}[f(A) = 1] \geq 1/2-\epsilon$, and f is nearly balanced: $|\mathbb{P}[f(U) = 1] - 1/2| \leq \epsilon$, where $U := U_{\{0,1\}^n}$.

Many papers have been devoted to constructing affine extractors. The latest [CGL21] works for nearly logarithmic dimension.

One motivation for studying affine extractors is that they arise naturally in the study of *circuit lower bounds.* For example, the method of *restrictions* in particular partitions the input in affine spaces, and so any function that becomes constant via a suitable restriction cannot be a good affine extractor. In particular, switching lemmas [FSS84, Ajt83, Hås87, IMP12, Hås14, HRST17] imply that small AC0 circuits cannot compute affine extractors. The same holds for models which shrink under restrictions, such as De Morgan formulas, see [Tal14] for the latest shrinkage bound and history. And the first numerical progress in more than 30 years on lower bounds for general circuits – [FGHK16] – holds for computing affine extractors. Finally, affine extractors also give *sampling lower bounds* [Vio14, Vio16, Vio20].

This also means that showing that a circuit class can compute good affine extractors indicates some of the difficulties that may arise when trying to prove lower bounds against that class. This direction has been pursued in a number of works, in fact going back to [Raz88] (cf. [Sav95]). More recently, the paper [CT15] (Theorem A.6) shows that affine extractors for dimension $k = O(\log n)$ can be computed by

1. polynomials mod 2 of degree $O(\log n)$,
2. Xor-And-Xor circuits of size $n^{2+o(1)}$,
3. De Morgan's formulas of size $n^{5+o(1)}$.

Their result also gives good dependence on ϵ, which we omit for simplicity.

It is a folklore result that IP is an affine extractor for dimension larger than $n/2$ (a proof can be found in [Vio16]). Moreover, some of the previous lower bounds hold for computing affine extractors. The worst-case $n^{1+c^{-d}}$ lower bound for depth d in [ABG$^+$14] holds for computing affine extractors, even with very weak parameters. The quadratic lower bound for And-Or-And-Xor [ABG+14] and the average-case lower bound for depth d [BKT19] hold for computing extractors if the error is exponentially small. We do not know if they can be generalized to affine extractors with constant error, but jumping ahead we give a simple proof of an $n^{1.5-o(1)}$ lower bound for computing constant-error extractors by And-Or-And-Xor circuits (Section 4).

Our first main result is that small (non-explicit) AC0-Xor circuits can compute very good affine extractors. In fact, And-Or-And-Xor circuits of size $n^2 \log^{O(1)} n$ suffice, matching the depth and – up to logarithmic factors – the size lower bound in [CGJ+18].

Theorem 3. *There exists an And-Or-And-Xor circuit C of size $n^2 \log^{O(1)} n$ that computes an affine extractor for dimension $k \geq \log^c n$ with error $1/\Omega(\log n)$, were $c > 0$ is an absolute constant.*

The proof is in Section 1. We actually give a general template for constructing affine extractors, and obtain constructions in other models as well. In particular, we show that De Morgan formulas of size $n^{4+o(1)}$ can compute affine extractors (see Theorem 15), improving the $n^{5+o(1)}$ bound from [CT15] (Item 3 above).

The only obstacle to an explicit construction is the layer of parity gates. Should an explicit construction for that be found, the affine extractor would be simpler than previous explicit constructions for comparable entropy (see [CGL21] and references therein).

It is natural to ask if the depth of the circuit in Theorem 3 is tight, that is if Or-And-Xor (a.k.a. DNF-Xor) circuits can compute good affine extractors. We note that an And-Xor circuit computes (the characteristic function of) an affine space, and so a DNF-Xor circuit of size s computes the *union of s affine* spaces. Understanding the power of union of affine spaces seems interesting from a mathematical perspective as well, and a natural next step towards more general models after affine spaces.

It is easy to show that DNF-Xor circuits require exponential size to compute good affine extractors, and a proof can be found in [CS16]. However, we show next that they can compute *one-sided* extractors. (Note that the DNF-Xor sub-circuits in the construction in Theorem 3 are not balanced and so do not compute one-sided affine extractors.)

Theorem 4. *There exists a $O(n \log^2 n)$ size DNF-Xor circuit that computes a one-sided affine extractor for dimension $k \geq c \log^3 n$ with error $1/\log^{1.9} n$, were $c > 0$ is an absolute constant.*

We apply this theorem to separate DNF-Xor circuits from a number of other classes: *(i) disjoint* unions of affine spaces, (ii) parity decision trees, and (iii) AC0-Xor circuits with n parity gates. These separations hold in the average case too, and we show tightness with respect to several parameters. These results point to the strength of the model and to the techniques we can (not) use for lower bounds.

Let us elaborate on the separation from parity decision trees (PDTs). For comparison, recall that any polynomials-size DNF on n bits can be approximated by a decision tree (DT) of depth $n - \Omega(n/\log n)$. (Proof sketch: We can ignore terms of size $\omega(\log n)$. Then a switching lemma shows that we can fix all but $\Omega(n/\log n)$ variables and the DNF collapses to a decision tree of depth $O(\log n)$.) It is natural to ask if a corresponding switching lemma or simulation exist for DNF-Xor in terms of parity decision trees (PDT). We show that the answer is negative:

Corollary 5. *There exists a DNF-Xor circuit $f : \{0,1\}^n \to \{0,1\}$ of size $n \cdot \operatorname{poly}\log n$ such that for any depth $n - \log^{2+o(1)} n$ PDT $T : \{0,1\}^n \to \{0,1\}$ we have $\mathbb{P}[f(U) = T(U)] \leq 1/2 + 1/\Omega(\log n)$.*

Note that the "depth deficiency" (i.e., n minus the depth of the tree) decreases exponentially from the $\Omega(n/\log n)$ in the simulation of DNFs by DTs to $\log^{2+o(1)} n$ in the simulation of DNF-Xors by PDTs. We summarize this finding informally as follows:

— PDTs are *not* to DNF-Xor what DTs are to DNFs —

This finding stands in contrast with our extensions of other simulations of AC0 circuits by DTs to the setting of AC0-Xor circuits and PDTs. This includes simulations given by the switching lemma, and simulations that exploit various restrictions on fan-in, see Section 2.

The study of DNF-Xor circuits is also motivated by our next result, which shows that if IP can be approximated by small such circuits, then in fact IP can be computed (exactly) by small AC0-Xor circuits.

Theorem 6. *Suppose there is $c > 0$ and a DNF-Xor circuit of size n^c that computes IP correctly on a $1/2 + 1/\log^c n$ fraction of the inputs. Then there are polynomial-size, constant-depth AC0-Xor circuits that compute IP.*

The proof is in Section 3.

A concurrent work [RDR21] shows that if small DNF-Xor circuits compute IP on a $5/6+\epsilon$ fraction of the inputs, then there are efficient data-streaming and communication protocols for low-degree polynomials. The conclusions in [RDR21] and the present work thus concern different models. The hypotheses are also different. Whereas [RDR21] requires the circuit to compute IP on $5/6 + \epsilon$, in our application $1/2 + 1/\operatorname{poly}\log$ suffices. Also, a partial converse to Theorem 6 is given by the so-called discriminator lemma [HMP+93]: if a size-s And-Or-And-Xor circuit computes IP, then a size-s DNF-Xor circuit computes IP on $1/2 + 1/s$ fraction of the inputs.

We note that the hypothesis in Theorem 6 is related to extractors. Indeed, let C be a DNF-Xor circuit of size s. Let $S := \{x : C(x) = 1\}$ and let $|S|/2^n =: p$. Suppose that IP is biased on S, that is $|\mathbb{P}[IP(S) = 1] - \mathbb{P}[IP(S) = 0]| \geq \epsilon$. Then either C or the negation of C computes IP correctly on a $1/2 + p\epsilon$ fraction of inputs. In other words, if IP is not an extractor, then we can approximate IP, and by Theorem 6 we can compute it with small AC0-Xor circuits. To avoid the latter, IP should have bias $\leq 1/\log^c n$ on any set S as above of size $\geq 1/\log^c$n, for any c.

Problem 7. Does IP extract randomness from unions of polynomially many affine spaces?

This work raises several other questions. Besides the question of explicitness, an obvious question is matching lower bounds and affine-extractor constructions. In particular, it would be interesting to know if one can compute affine extractors by depth-d AC0-Xor circuits of size $n^{1+c^{-d}}$. This would follow if one can show a *size-depth tradeoff* for r-wise resilient functions (defined later), which we also raise as a question. In general, we raise the question of understanding the complexity of computing r-wise resilient functions in various models of computation. For example, can they be computed by linear-size circuits? From the side of lower bounds, it would be interesting to strengthen our lower bound for computing affine extractors to quadratic.

1 Constructing affine extractors

The proof of Theorem 3 builds on ideas developed in the literature on extractors. At the high level, we use an approach from [GVW15], Section 5.3, of combining a suitable linear transformation with a *resilient function* (defined below). [GVW15] aims to construct extractors for *bit-fixing* sources (a special case of affine sources) of *large* entropy ($n/\mathrm{poly}\log$) and computable in *AC0*. They pick a sparse linear transformation, which guarantees that the extractor is computable in AC0, and which is sufficient because the entropy is close to n. By contrast, we aim to extract from the more general affine sources, and even with polylogarithmic entropy. On the other hand, we can pick a non-sparse linear transformation thanks to the layer of parity gates. [GVW15] shows that the output of the linear transformation is uniform except for few bits; instead we can only guarantee that it is r-wise independent except for few bits.

Definition 8. [Vio14] A distribution D over $\{0,1\}^m$ is *r-wise uniform but for b bits* if there is a set $S \subseteq [m]$ of size $m-b$ such that for any r elements in S the projection of D onto the corresponding bits is uniform over $\{0,1\}^r$. If $b=0$ we simply say r-wise uniform.

A main and simple result in this paper is that applying a suitable linear transformation one can turn an affine source into a distribution of the type above. The corresponding linear transformations seem interesting to study, so we give a definition.

Definition 9. An $m \times n$ matrix T is k-affine to r-wise uniform but for b-bits if for any distribution U_A uniform over an affine space $A \subseteq \{0,1\}^n$ of dimension $\geq k$ the distribution TU_A is r-wise uniform but for b bits.

We raise the question of understanding the complexity of computing such matrices efficiently. For example, in particular we ask if these transformations (with good parameters as below) can be computed by linear-size circuits, local maps, etc. For starters, we prove that such matrices exist via the probabilistic method.

Lemma 10. *A matrix T as in Definition 9 exists for any $b > 3n$ and $k \geq 2r \log m$.*

Proof. It suffices to prove the lemma for any *linear space* (rather than affine). To verify this, write the uniform distribution over an affine space A as $SX+s$ where S is a full-rank $n \times k$ matrix, $X \in \{0,1\}^k$ is uniform, and $s \in \{0,1\}^n$ is a fixed shift. Consider $T(SX+s) = TSX + Ts$. Since SX is a linear space, TSX is r-wise uniform but for b bits. This property is unaffected by adding the fixed shift Ts.

Recall that for an $r \times k$ matrix M the distribution MU where $U \in \{0,1\}^k$ is uniform if and only if the rows of M are linearly independent. Hence, for our goal it suffices to construct matrices with the latter property. Pick T uniformly at random. Fix a full-rank $n \times k$ matrix S and note that TS is a uniform $m \times k$ matrix M. We bound the probability that there exists a bad set $B \subseteq [m]$ of b bits such that each row (with index) in B is a linear combination of $\leq r$ rows not in B. If such bad sets do not exist then the proof is completed as follows. Greedily pick rows of TS that are not a linear combination of $\leq r$ rows already picked. One can pick $\geq m - b$ rows, otherwise a bad set of size b exists.

Fix B, and fix arbitrarily the rows of M not in B. Let H be the set of vectors that can be obtained as a linear combination of $\leq r$ rows not in B. We have

$$|H| \leq \binom{m}{0} + \binom{m}{1} + \cdots + \binom{m}{r} \leq 2^{r \log m}.$$

The probability that each row in B falls in H is then

$$\left(\frac{|H|}{2^k}\right)^b = 2^{b(r \log m - k)}.$$

When $k \geq 2r \log m$ this probability is

$$\leq 2^{-kb/2}.$$

Hence the probability that there exists a bad set of size b is

$$\leq \binom{m}{b} 2^{-kb/2} \leq 2^{b(\log m - k/2)} \leq 2^{-bk/3}.$$

Finally, there are at most 2^{kn} linear spaces of dimension k. Hence the probability that there exists such a space with a bad set as above is $\leq 2^{k(n-b/3)}$. Setting $b > 3n$ this probability is less than one and the desired matrix T exists. □

Given this lemma, it remains to extract from distributions over $\{0,1\}^m$ which are r-wise uniform but for b bits.

Definition 11. A function $f : \{0,1\}^m \to \{0,1\}$ is r-wise (b, ϵ)-resilient if for any r-wise uniform distribution X we have $|\mathbb{P}[f(X) = 1] - 1/2| \leq \epsilon$, and the probability over X that changing at most b bits of X changes the value of f is $\leq \epsilon$ (and in particular the bias that one can obtain changing b bits is at most $1/2 + 2\epsilon$). Note that this is equivalent to saying that f is an extractor with error 2ϵ for distributions which are r-wise uniform but for b bits.

The paper [Vio14] showed that the majority function is resilient over r-wise uniform distributions, relying on the Central Limit Theorem for r-wise uniform distributions from [DGJ$^+$10].

Lemma 12. *[Vio14] The Majority function is r-wise $(m^{0.499}, 1/100)$-resilient for all sufficiently large r.*

Using this, we can show that Maj-Xor circuits can compute affine extractors with optimal dependence on dimension, up to constant factors.

Theorem 13. *There is a non-explicit Maj-Xor circuit of polynomial size that computes an affine extractor for dimension $O(\log n)$ with error $1/100$.*

Proof. Apply Lemma 10 with $m = n^{2.1}$ and $b = 4n$. Let V be an affine space of dimension $k \geq 2r \log m = O(\log n)$. Let U_V be the uniform distribution over V. By Lemma 10, TU_V is r-wise independent except for b bits. We conclude by Lemma 12. □

For Theorem 3 we need extractors computable in AC0 however. The seminal work [AL93] (cf. [Wel20] for a streamlined exposition of a slighly weaker result) showed through the probabilistic method the existence of functions on m bits that are m-wise $(\Omega(\alpha m/\log^2 m), O(\alpha))$-resilient for any α. We observe that their construction is also resilient over $\operatorname{poly}\log m$-wise distributions. This can be shown using the fact that polylog-wise uniformity fools AC0 circuits [Baz09, Raz09, Bra10], and for completeness we include a proof in Section A.

Lemma 14. *There exists $c > 0$ and a function $f : \{0,1\}^m \to \{0,1\}$ that is $\log^c m$-wise $(\Omega(\alpha m/\log^2 m), O(\alpha))$-resilient, for any $\alpha \geq 1/m$. Moreover, f is computable by depth-3 circuits of size $O(m^2/\log m)$.*

We note that explicit constructions of $\operatorname{poly}\log$-wise resilient functions appear in [CZ16] and [Mek17]. One can use either [CZ16] or [Mek17] to obtain affine extractors with our approach. However, some of the parameters would be a little worse than what we claimed. For example, the circuit size would be n^c for $c > 2$.

Proof. [Theorem 3]. The parity gates compute the linear transformation T in Lemma 10 with the parameters $m = O(n \log^3 n)$ and $b = m/\log^3 m > 3n$. By the assumption on k we have $r = \log^{c'} m$ for a constant c' as large as desired. The distribution TU is r-wise uniform but for b bits. We feed its output to the function f in Lemma 14 for $\alpha = \Theta(1/\log m)$. Then f is $(m/\log^3 m, O(1/\log m))$-resilient, and the result follows. □

Finally, we obtain a construction for De Morgan formulas.

Theorem 15. *The affine extractor in Theorem 3 can be computed by De Morgan formulas of size $n^4 \log^{O(1)} n$, or by formulas over the full binary base B_2 of size $n^3 \log^{O(1)} n$.*

Proof. From the proof of Theorem 3 we know that the fan-ins of the And-Or-And-Xor circuit, starting from the output And, are $n \log^{O(1)} n, n \log^{O(1)} n, \log^{O(1)} n, n$. Note that an And or Or on t bits can be computed by De Morgan formulas of size $O(t)$, while Parity on t bits can be computed by such formulas of size $O(t^2)$ and B_2 formulas of size $O(t)$. The result follows. □

2 DNF-Xor

Our construction proving Theorem 4 is similar to our affine-extractor construction. We show that the so-called Tribes function is "one-sided resilient," so composing it with the layer of Xor gates from Lemma 10 yields a one-sided extractor.

Definition 16. [BL85] (cf. [O'D14], Proposition 4.12) $\text{Tribes}_w : \{0,1\}^m \to \{0,1\}$ is the read-once DNF where every term has size w and $|\mathbb{P}[Tribes(U) = 1] - 1/2| = O(\log m)/m$. This makes $w = \log m - \log\log m + O(1)$.

We need the following lemma.

Lemma 17. *Let D be a $w \log(1/\epsilon)$-wise uniform distribution on n bits. Let $f : \{0,1\}^n \to \{0,1\}$ be a read-once DNF with terms of size $\leq w$. Let U be the uniform distribution over $\{0,1\}^n$. Then $|\mathbb{P}[f(D) = 1] - \mathbb{P}[f(U) = 1]| \leq \epsilon$.*

This claim follows by noting that the input distribution to the Or in the DNF is $\log(1/\epsilon)$-wise *independent* (and not necessarily uniform). We can then apply the corresponding fundamental result in pseudorandomness from [EGL+92]; see [Vio17], Lecture 1, for an exposition.

Proof. [Theorem 4] We need a slight extension of Lemma 10. We claim that the matrix T constructed there has the additional property $(\star)$ that any linear combination of $\leq r$ rows of T is linearly independent. This can be established with essentially the same proof, because the probability that a uniform $m \times k$ matrix does not satisfy this is $\leq 2^{O(r \log m) - k}$ which is less than $1/2$ by our choice of parameter (and the proof of the lemma shows that a uniformly selected T satisfies the lemma with probability $> 1/2$).

Hence, consider the matrix T from Lemma 10 with $m = O(n \log^2 n)$ and $b = 4n$, and further take it to satisfy $(\star)$. Feed this distribution into the Tribes function on m bits. First, note that by $(\star)$ we have that TU is r-wise uniform where $r = k/2 \log m$. Hence, the output distribution of the And gates is $r/\log m$-wise independent. By our assumption that $k \geq c \log^3 n$, it will be $c \log m$-wise independent for a c large enough so that by Lemma 17 the probability that Tribes outputs 1 on TU is within $1/m$ of the probability it outputs 1 over the uniform distribution, and so still within $O(\log m)/m$ of $1/2$.

This proves that our function is indeed nearly balanced. There remains to prove that it is 1 with high probability over any large affine space S. We have that TSU is $k/4 \log m$-wise uniform but for $b = 4n$ bits. Now we basically show that the good bits suffice to make the function 1 with probability about $1/2$. The bad bits touch $\leq b$ terms. Hence there are $m/w - b$ good terms, defined as those terms that do not take any bad bit as input. By Lemma 17 as above, the probability that the Or of the good terms is 0 over TSU is within $1/m$ of the probability that it is 0 over U. The latter probability is

$$(1 - 2^{-w})^{m/w - b} \leq (1/2 + O(\log m)/m)(1 - 2^{-w})^{-b},$$

where the first term in the right-hand side is from the definition of Tribes. For the second term note that

$$(1 - 2^{-w})^{-b} \leq e^{b/2^w}.$$

We have $2^w = \Theta(m/\log m) = \Theta(n \log^2 n / \log\log n)$ and so $b/2^w \leq 1/\log^{1.9} n$ and $e^{b/2^w} \leq 1 + O(1)/\log^{1.9} n$, and the result follows. □

We now use the above result to give separations.

Definition 18. We say $g : \{0,1\}^n \to \{0,1\}$ is a *k-affine-partition* if g can be expressed as $g(x) = \sum_{i=1}^{t} \alpha_i \mathbf{1}_{V_i}$ where $V_1, V_2, \ldots, V_t$ are *disjoint* affine subspaces of dimension $\geq k$ that form a partition of $\mathbb{F}_2^n$ and for each i, $\alpha_i \in \{0,1\}$.

We next show one-sided extractors for dimension k cannot even be approximated by k-affine partitions.

Claim 19. Let $f : \{0,1\}^n \to \{0,1\}$ be a one-sided extractor for dimension k with error ϵ, and let $g : \{0,1\}^n \to \{0,1\}$ be a k-affine partition. Then

$$\mathbb{P}[f(U) = g(U)] \leq \frac{1}{2} + 3\epsilon.$$

Proof. Let $G_0 := \{x : g(x) = 0\}$, $p := |G_0|/2^n$ and $G_1 := \{x : g(x) = 1\}$. Let $\alpha := \mathbb{P}_{x \in G_0}[f(x) = 0]$ and $\beta := \mathbb{P}_{x \in G_1}[f(x) = 0]$. We have

$$\mathbb{P}[f(U) \neq g(U)] = p(1-\alpha) + (1-p)\beta.$$

Because f is nearly balanced, we have $p\alpha + (1-p)\beta \geq 1/2 - \epsilon$, and so $(1-p)\beta \geq 1/2 - \epsilon - p\alpha$. Plugging this above we get

$$\mathbb{P}[f(U) \neq g(U)] \geq 1/2 - \epsilon - p\alpha + p(1-\alpha) = 1/2 - \epsilon + p(1-2\alpha).$$

Also by the extractor property we have $\alpha \leq 1/2 + \epsilon$. (Since G_0 is the disjoint union of spaces on which f outputs 0 on at most a $1/2 + \epsilon$ fraction of the elements.) Hence $1 - 2\alpha \geq -2\epsilon$. Combining with above yields

$$\mathbb{P}[f(U) \neq g(U)] \geq 1/2 - \epsilon - 2p\epsilon \geq 1/2 - 3\epsilon.$$

□

We showed in Theorem 4 that small DNF-Xor circuits can compute such extractors. In fact, the circuits are of the type $\text{Or}_{n \log^{O(1)} n}$-$\text{And}_{O(\log n)}$-Xor; subscripts indicate fan-ins. Again, this is equivalent to a nearly-linear collection of spaces of very large dimension ($n - O(\log n)$) that cannot be approximated by *disjoint* spaces, even if the dimensions of the latter spaces is as small as polylogarithmic. Note that a parity decision tree (PDT) on n bits with depth $n - k$ gives a k-affine partition, and this proves Corollary 5.

It is natural to ask if this separation (between DNF-Xor and PDT) is tight. We show that indeed it is, in three different settings.

2.1 Setting 1: The number of parity gates

We consider AC0-Xor circuits where the Xor gates correspond to a basis.

Definition 20. AC0-Xor-B circuits on n bits are AC0-Xor circuits where the number of Xor gates is n and the corresponding vectors form a basis.

We show that small AC0-Xor-B circuits *can* be approximated by moderate-depth PDTs, showing that in Corollary 5 it is essential that the number of Xor gates is larger than n, even we allow general AC0 post-process (instead of DNF).

The proof amounts to observing that switching lemmas for AC0 apply as stated to AC0-Xor-B, except that they yield PDTs rather than DTs. Specifically, it follows for example from the switching lemma in [Hås14] (see Corollary 11 in [Vio] for an explicit statement about AC0) that for $h := 2^{o(n/2^d \log^{d-1} n)}$ an AC0 circuit of size $\leq h$ and depth d can be approximated by a DT of depth $n - \Omega(n/\log^{d-1} n)$ except with error $1/h$. The corresponding statement applies to AC0-Xor-B.

Claim 21. For $h := 2^{o(n/2^d \log^{d-1} n)}$, an AC0-Xor-B circuit of size $\leq h$ and depth d can be approximated by a PDT of depth $n - \Omega(n/\log^{d-1} n)$ except with error $1/h$.

To prove this, apply the result for AC0 mentioned above to the AC0 part of the circuit. Querying one input bit to the AC0 part can be simulated by querying a parity of the input bits to the AC0-Xor-B circuit, resulting into a PDT. A straightforward combination of the above results also yields a separation between small DNF-Xor and AC0-Xor-B circuits.

2.2 Setting 2: The fan-in of the And gates

Next we show that the fan-in of the And gates in the sepration (between DNF-Xor and PDTs) are tight up to a $O(\log \log n)$ factor: We show that any Or-And-Xor circuit where the And fan-in is at most $\log n - 2 \log \log n$ can be approximated by a depth $O(n/\log n)$ PDT with at most constant error. This follows from the following lemma, which is a "PDT version" of the corresponding result for DNF and decision trees, see [AW89, Tre04]. We follow the exposition in [Vio].

Lemma 22. *For every Or-And$_w$-Xor circuit $C : \{0,1\}^n \to \{0,1\}$, there exists a PDT T of depth $\leq 2w2^w \log(1/\epsilon)$ with range $\{0,1,?\}$ such that:*

1. $\Pr_{x \in \{0,1\}^n}[T(x) =?] \leq \epsilon$.

2. *For all $x \in \{0,1\}^n$, $T(x) \neq ? \Rightarrow T(x) = C(x)$.*

Proof. We are going to define $T : \{0,1\}^n \to \{0,1,?\}$ recursively. If C is a constant then T is a constant. Otherwise, let $C = \vee_{i=1}^m C_i$ where each subcircuit C_i is an And of a set of at most w parities, denoted by P_i. We can assume w.l.o.g. that for each i, P_i is linearly independent. We greedily construct an index set $I \subseteq [m]$ as follows: we look at each P_i one-by-one, and add i into I if $P_i \cup \bigcup_{j \in I} P_j$ is linearly independent. There are two cases:

1. If $|I| \geq 2^w \log(1/\epsilon)$, we let T query all the parities in P_i for the first $2^w \log(1/\epsilon)$ indices in I, which decides the values for the corresponding subcircuits C_i. If any of the subcircuits is True, then T outputs 1, otherwise it outputs ?.

2. Otherwise $|I| < 2^w \log(1/\epsilon)$, then the size of $\bigcup_{i \in I} P_i$ is at most $w2^w \log(1/\epsilon)$. Moreover, for any P_j with $j \notin I$, there must exists a parity $p_j \in P_j$ such that $p_j \in \mathit{span}(\bigcup_{i \in I} P_i \cup (P_j \setminus \{p_j\}))$. The tree T first queries every parity in $\bigcup_{i \in I} P_i$. After that, we know that for each$j \notin I$, $p_j \in \mathit{span}(P_j \setminus \{p_j\})$. As the subcircuit C_j is an And of the parities in

P_j, if setting all the parities in $P_j \setminus \{p_j\}$ to be 1 forces p_j to be 0, we can just ignore this subcircuit. If it forces p_j to be 1, we can safely remove p_j to get an *And* of $\leq w-1$ parities. Therefore what we get is an Or-And-Xor circuit C' where the fan-in of each And is $\leq w-1$ (which might depend on the results of the queries), and we recurse on C' to get a parity decision tree T'.

The depth of t_C is $\leq w2^w \log(1/\epsilon) + (w-1)2^{w-1}\log(1/\epsilon) + \cdots \leq 2w2^w \log(1/\epsilon)$. Item 2 is evident by definition. For Item 1, note that T outputs ? only if none of the first $2^w \log(1/\epsilon)$ subcircuits in I is True. Each P_i are linearly independent, and by construction of I the outputs of these subcircuits are independent, so the probability can be bounded by $(1-2^{-w})^{2^w \log(1/\epsilon)} \leq \epsilon$. □

2.3 Setting 3: The fan-in of the Or gate

We show that $\mathrm{Or}_{o(n/\log n)}$-And-Xor circuits can be approximated by moderate-depth PDTs.

Claim 23. Let $C : \{0,1\}^n \to \{0,1\}$ be an $\mathrm{Or}_{o(n/\log n)}$-And-Xor circuit. There exists a PDT T of depth $(1-\Omega(1))n$ such that $\mathbb{P}[C(U) = T(U)] \geq 1/2 + \Omega(1)$.

Proof. Let C' denote the circuit obtained from C by deleting all the And gates with fan-in greater than $\log n - \log\log n - O(1)$. By Lemma 22 a PDT with depth $(1-\Omega(1))n$ can approximate C' with constant error. Now we argue that the removal of the And gates does not introduce too much error. Any And gate removes evaluates to True under uniform input with probability $\leq 2^{-(\log n - \log\log n - O(1))} = O(\log n/n)$. Since the number of removed And gates is $o(n/\log n)$, by a union bound the total error introduced is $o(1)$. □

3 Proof of Theorem 6

We begin by observing that IP can be "randomly self-reduced" very efficiently: the overhead is just computing a parity. More formally:

$$IP(x+a) = IP(x) + L_a(x)$$

for any $x, a \in \{0,1\}^n$, and where $L_a : \{0,1\}^n \to \{0,1\}$ is an affine transformation that depends on a only. To verify this just consider a monomial and note that $(x_1+a_1)(x_2+a_2) = x_1x_2 + a_1x_2 + a_2x_1 + a_1a_2 = x_1x_2 + L_a(x)$. The same fact is used for example in pseudorandom generators for low-degree polynomials [BV10], and in [RDR21]. In general the proof is also similar to the simplified average-case lower bounds for parity [Vio09].

Let $C : \{0,1\}^n \to \{0,1\}$ be a circuit that computes IP on $1/2+\epsilon$ fraction of the inputs. Consider the random circuit C_A for uniform A which on input x outputs

$$C_A(x) := C(x+A) + L_A(x).$$

Note that for every x, $\mathbb{P}_A[C_A(x) = IP(x)] \geq \mathbb{P}_A[C(x+A) = IP(x+A)] \geq 1/2+\epsilon$. Moreover, for any fixed A the circuit C_A is an AC-Xor circuit of polynomial size. To verify this, note that we can compute $L_A(x)$ using parities at the input level, and the output Xor is on two bits and can be computed in AC0. Also, adding the "shift" A to x can be absorbed in the parity gates at the input level.

There remains to boost the probability. Consider the random circuit D which computes $t = O(n/\epsilon^2)$ copies of C_A with independent A, and then computes approximate majority. Specifically, it outputs 1 if at least $(1/2 + \epsilon/2)t$ copies output 1, and it outputs 0 if at least $(1/2 + \epsilon/2)t$ copies output 0. By a Chernoff bound, on any input this circuit has error probability $< 2^{-n}$. Hence we can fix the randomness so that it computes correctly every input. Moreover, the approximate majority computation can be done by polynomial-size AC0 circuits for $\epsilon = 1/\log^{O(1)} n$ [Ajt83, Ajt93].

4 A lower bound for computing affine extractors

In this section we prove the following almost $n^{1.5}$ lower bound for computing affine extractors, even if the error is constant (when the error is exponentially small, a quadratic lower bound can be inferred from the techniques in [CGJ$^+$18]). While the bound is weaker, the proof appears more elementary than the one in [CGJ$^+$18].

Theorem 24. *Let $C : \{0,1\}^n \to \{0,1\}$ be an And-Or-And-Xor circuit that computes an affine extractor for dimension $n/2$ with error $1/4$. Then C has size $\Omega\left(n^{1.5}/\log n\right)$.*

Lemma 25. *Let C be as in Theorem 24 but with the additional restriction that the fan-in of the middle And gates is t. Then C has size $\Omega\left(n^2/(t \log n)\right)$.*

Proof. [Lemma 25] We assume that a circuit of size $o(n^2/t \log n)$ exists, and reach a contradiction. Let R denote the set of Or gates in C, and let A denote the set of And gates in C, excluding the output And gate. Draw a bipartite graph $G = (R \cup A, E)$ between R and A. Each Or gate must have at least $n/16$ edges, otherwise we can set C to 0 using $n/16$ linear restrictions (corresponding to a vector space of dimension $n - n/16$).

Hence there exists some And gate that is connected to at least a $\frac{n}{16|A|}$ fraction of Or gates in R. We set a to 1 using at most t restrictions, eliminate the adjacent Or gates, and consider G on the resulting affine subspace. We repeat this process k times for $k = n/16t$. Note that we can always find an Or gate with fan-in $\geq n/16$, for else we can set the circuit to 0 by setting $kt + n/16 \leq n/8$ parities.

At the end of the process, the number of Or gates is

$$|R|\left(1 - \frac{n}{16|A|}\right)^k \leq |R|\left(1 - \frac{100t\log n}{n}\right)^{n/16t} \leq n^2/n^3 < 1.$$

This means that the circuit is fixed, which is a contradiction. □

Lemma 26. *Let $C : \{0,1\}^n \to \{0,1\}$ be an And-Or-And-Xor circuit that computes an affine-extractor for dimension $n/2$ with error $1/4$. Either the size of C is $\Omega\left(nt/\log n\right)$ or there exists an affine subspace H of dimension $\geq 7n/8$ such that $C|_H$ is an And-Or-And$_t$-Xor circuit.*

Proof. [Lemma 26] Let A_t denote the set of And gates of fan-in greater than t, excluding the output, and let X_t denote the set of Xor gates connected toA_t. Draw the bipartite graph $G = (A_t \cup X_t, E)$ connecting A_t, X_t. There is some gate $x \in X_t$ connected to at least a $\frac{t}{|X_t|}$ fraction of nodes in A_t as long as $|A_t| \geq 1$. After $k = n/16$ iterations of setting the XOR gate with the highest degree in G to 0, if $|A_t| \geq 1$ we have at most

$$|A_t|\left(1 - \frac{t}{|X_H|}\right)^k \leq |A_t|e^{-\frac{nt}{16|X_t|}}$$

And gates left in G. If $|X_t| \geq nt/16\log n$ we are done, since obviously C has size $\geq |X_t|$. Otherwise,

$$|A_t|e^{-\frac{nt}{16|X_t|}} \leq |A_t|\frac{1}{n} \leq \frac{n}{16}.$$

The last inequality follows because $|A_t| \leq n^2/16$. So after making $n/16$ restrictions, we are left with at most $n/16$ And gates in A_t. We can make at most $n/16$ additional restrictions setting them to 0, so that there are no more And gates in A_t (we might set some to 1 during this process, but that only helps us). We have made at most $n/16 + n/16 = n/8$ restrictions to reach a subspace H where $C|_H$ has no And gates of fan-in $\geq t$. □

Proof. [Theorem 24] We combine Lemmas 25 and 26 with the threshold $t = \sqrt{n}$. By Lemma 26, either $C = \Omega\left(n^{3/2}/\log n\right)$ or there is some affine subspace H of dimension $7n/8$ on which $C|_H$ has middle And gates of fan-in $\leq \sqrt{n}$. In the first case we are done. In the second case, let $n' = 7n/8$. Then we can think of $C|_H$ as a $(4n'/7, 1/4)$ affine extractor on n' variables and apply Lemma 25. □

References

[ABG+14] Adi Akavia, Andrej Bogdanov, Siyao Guo, Akshay Kamath, and Alon Rosen. Candidate weak pseudorandom functions in AC0 MOD 2. In Moni Naor, editor, *ACM Innovations in Theoretical Computer Science conf. (ITCS)*, pages 251–260. ACM, 2014.

[Ajt83] Miklós Ajtai. Σ^1_1-formulae on finite structures. *Annals of Pure and Applied 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.

[AL93] Miklos Ajtai and Nathan Linial. The influence of large coalitions. *Combinatorica*, 13:129–145, 1993.

[AW89] Miklos Ajtai and Avi Wigderson. Deterministic simulation of probabilistic constant-depth circuits. *Advances in Computing Research - Randomness and Computation*, 5:199–223, 1989.

[Baz09] Louay M. J. Bazzi. Polylogarithmic independence can fool DNF formulas. *SIAM J. Comput.*, 38(6):2220–2272, 2009.

[BKT19] Mark Bun, Robin Kothari, and Justin Thaler. Quantum algorithms and approximating polynomials for composed functions with shared inputs. In Timothy M.

Chan, editor, *ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 662–678. SIAM, 2019.

[BL85] Michael Ben-Or and Nathan Linial. Collective coin flipping, robust voting schemes and minima of Banzhaf values. In *26th Symposium on Foundations of Computer Science*, pages 408–416, Portland, Oregon, 21–23 October 1985. IEEE.

[Bra10] Mark Braverman. Polylogarithmic independence fools AC^0 circuits. *J. of the ACM*, 57(5), 2010.

[BV10] Andrej Bogdanov and Emanuele Viola. Pseudorandom bits for polynomials. *SIAM J. on Computing*, 39(6):2464–2486, 2010.

[CGJ+18] Mahdi Cheraghchi, Elena Grigorescu, Brendan Juba, Karl Wimmer, and Ning Xie. $\mathrm{AC}^0 \circ \mathrm{mod}_2$ lower bounds for the boolean inner product. *J. Comput. Syst. Sci.*, 97:45–59, 2018.

[CGL21] Eshan Chattopadhyay, Jesse Goodman, and Jyun-Jie Liao. Affine extractors for almost logarithmic entropy. *Electron. Colloquium Comput. Complex.*, 28:75, 2021.

[CS16] Gil Cohen and Igor Shinkar. The complexity of DNF of parities. In *ACM Innovations in Theoretical Computer Science conf. (ITCS)*, pages 47–58, 2016.

[CT15] Gil Cohen and Avishay Tal. Two structural results for low degree polynomials and applications. In Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015, August 24-26, 2015, Princeton, NJ, USA*, volume 40 of *LIPIcs*, pages 680–709. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015.

[CZ16] Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. In *ACM Symp. on the Theory of Computing (STOC)*, pages 670–683, 2016.

[DGJ+10] Ilias Diakonikolas, Parikshit Gopalan, Ragesh Jaiswal, Rocco A. Servedio, and Emanuele Viola. Bounded independence fools halfspaces. *SIAM J. on Computing*, 39(8):3441–3462, 2010.

[EGL+92] Guy Even, Oded Goldreich, Michael Luby, Noam Nisan, and Boban Velickovic. Approximations of general independent distributions. In *ACM Symp. on the Theory of Computing (STOC)*, pages 10–16, 1992.

[FGHK16] Magnus Gausdal Find, Alexander Golovnev, Edward A. Hirsch, and Alexander S. Kulikov. A better-than-3n lower bound for the circuit complexity of an explicit function. In Irit Dinur, editor, *IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 89–98. IEEE Computer Society, 2016.

[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.

[GVW15] Oded Goldreich, Emanuele Viola, and Avi Wigderson. On randomness extraction in AC0. In *IEEE Conf. on Computational Complexity (CCC)*, 2015.

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

[Hås14] Johan Håstad. On the correlation of parity and small-depth circuits. *SIAM J. on*

Computing, 43(5):1699–1708, 2014.

[HMP+93] András Hajnal, Wolfgang Maass, Pavel Pudlák, Márió Szegedy, and György Turán. Threshold circuits of bounded depth. *J. of Computer and System Sciences*, 46(2):129–154, 1993.

[HRST17] Johan Håstad, Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. An average-case depth hierarchy theorem for boolean circuits. *J. of the ACM*, 64(5):35:1–35:27, 2017.

[IMP12] Russell Impagliazzo, William Matthews, and Ramamohan Paturi. A satisfiability algorithm for AC^0. In *ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 961–972, 2012.

[Juk06] Stasys Jukna. On graph complexity. *Comb. Probab. Comput.*, 15(6):855–876, 2006.

[LN90] Nathan Linial and Noam Nisan. Approximate inclusion-exclusion. *Combinatorica*, 10(4):349–365, 1990.

[Mek17] Raghu Meka. Explicit resilient functions matching ajtai-linial. In *ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 1132–1148, 2017.

[O'D14] Ryan O'Donnell. *Analysis of Boolean Functions.* Cambridge University Press, 2014.

[Raz87] Alexander Razborov. Lower bounds on the dimension of schemes of bounded depth in a complete basis containing the logical addition function. *Akademiya Nauk SSSR. Matematicheskie Zametki*, 41(4):598–607, 1987. English translation in Mathematical Notes of the Academy of Sci. of the USSR, 41(4):333-338, 1987.

[Raz88] A. A. Razborov. Formulas of bounded depth in the basis $\{\&, \oplus\}$ and some combinatorial problems. *Voprosy Kibernet. (Moscow)*, (134):149–166, 1988.

[Raz09] Alexander A. Razborov. A simple proof of Bazzi's theorem. *ACM Transactions on Computation Theory (TOCT)*, 1(1), 2009.

[RDR21] Michael Ezra Ron D. Rothblum. Small circuits imply efficient arthur-merlin protocols. *Electronic Coll. on Computational Complexity (ECCC)*, 2021.

[Sav95] Petr Savický. Improved boolean formulas for the ramsey graphs. *Random Struct. Algorithms*, 6(4):407–416, 1995.

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *19th ACM Symp. on the Theory of Computing (STOC)*, pages 77–82. ACM, 1987.

[SV12] Rocco A. Servedio and Emanuele Viola. On a special case of rigidity. Available at http://www.ccs.neu.edu/home/viola/, 2012.

[Tal14] Avishay Tal. Shrinkage of de morgan formulae by spectral techniques. In *IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 551–560. IEEE Computer Society, 2014.

[Tre04] Luca Trevisan. Some applications of coding theory in computational complexity. In *Complexity of computations and proofs*, volume 13 of *Quad. Mat.*, pages 347–424. Dept. Math., Seconda Univ. Napoli, Caserta, 2004.

[Vio] Emanuele Viola. AC0 unpredictability. *ACM Trans. Computation Theory.* Available at http://www.ccs.neu.edu/home/viola/.

[Vio07] Emanuele Viola. Pseudorandom bits for constant-depth circuits with few arbitrary symmetric gates. *SIAM J. on Computing*, 36(5):1387–1403, 2007.

[Vio09] Emanuele Viola. On the power of small-depth computation. *Foundations and Trends in Theoretical Computer Science*, 5(1):1–72, 2009.

[Vio14] Emanuele Viola. Extractors for circuit sources. *SIAM J. on Computing*, 43(2):355–972, 2014.

[Vio16] Emanuele Viola. Quadratic maps are hard to sample. *ACM Trans. Computation Theory*, 8(4), 2016.

[Vio17] Emanuele Viola. Special topics in complexity theory. ECCC lecture notes. Also available at http://www.ccs.neu.edu/home/viola/classes/spepf17.html, 2017.

[Vio20] Emanuele Viola. Sampling lower bounds: boolean average-case and permutations. *SIAM J. on Computing*, 49(1), 2020. Available at http://www.ccs.neu.edu/home/viola/.

[Wel20] Jake Wellens. *Assorted results in boolean function complexity, uniform sampling and clique partitions of graphs.* PhD thesis, Massachusetts Institute of Technology, 2020.

A Proof of Lemma 14

Let $f : \{0,1\}^m \to \{0,1\}$ be the function [AL93, Theorem 5.1] which is m-wise $(\Omega(\alpha m/\log^2 m), O(\alpha))$-resilient. f is the And of m read-once DNFs, so it is a depth-3 circuit of size $O(m^2/\log m)$. We need to show that over any r-wise distribution D:

1. The bias of f is $\leq O(\alpha)$.

2. The probability of changing the value of f by changing at most $\alpha m/\log^2 m$ bits of D is $O(\alpha)$.

[AL93, Theorem 5.1] proves 1. and 2. for $r = m$. The fact that 1. holds for $r = \text{poly}\log m$ then follows by [Bra10], using that $\alpha \geq 1/m$. For the second point we reason as follows. Fix some set Q of $\alpha m/\log^2 m$ bad bits. Let $e(y) : \{0,1\}^{|\overline{Q}|} \to \{0,1\}$ denote the indicator function of f not being fixed after assigning y to the good bits $\overline{Q}$. Now we show that $e(y)$ is computable by an AC0 circuit so we can again apply [Bra10] and reduce to the known resilience under the uniform distribution from [AL93, Theorem 5.1]. For some partial assignment y to $\overline{Q}$,f is not fixed if and only if at least one DNF function is not fixed, and no DNF outputs 0. What remains to show is that for each DNF function, the corresponding indicator $e'(y)$ can be expressed as an AC0 function. To verify this, note that the DNF is not fixed by y if every And term that does not intersect with Q has a bit set to 0, and there is at least one And term intersecting with Q such that all possible bits set by y are 1. This computation can be written as a polynomial-size AC0 circuit.