Constant-error pseudorandomness proofs from hardness require majority

Emanuele Viola*

February 5, 2019

Abstract

Research in the 80's and 90's showed how to construct a pseudorandom generator from a function that is hard to compute on more than 99% of the inputs. A more recent line of works showed however that if the generator has small error, then the proof of correctness cannot be implemented in subclasses of TC^0, and hence the construction cannot be applied to the known hardness results. This paper considers a typical class of pseudorandom generator constructions, and proves an analogous result for the case of large error.

The construction of *pseudorandom generators* from *hard functions* is a fundamental paradigm in complexity theory. Call a function $f : \{0,1\}^\ell \to \{0,1\}$ δ-hard if every "small circuit" fails to compute f on at least a δ fraction of the inputs. Perhaps the simplest example of the paradigm is this. Given f which is $(1/2-\epsilon)$-hard the *repetition* pseudorandom generator

$$PRG : \{0,1\}^{\ell\cdot s} \to \{0,1\}^{\ell\cdot s+s}$$
$$PRG(x_1, x_2, \dots, x_s) := (x_1, x_2, \dots, x_s) \circ f(x_1) \circ f(x_2) \circ \cdots \circ f(x_s)$$

has error $O(\epsilon s)$, by which we mean that small circuits cannot distinguish its output distribution from uniform with advantage more than $O(\epsilon s)$. The proof relies on the *hybrid argument*, see for example [FSUV13] and the discussion there. Hence, from hardness $1/2-\epsilon$ we can obtain $\Omega(1/\epsilon)$ bits of pseudorandomness, and this is tight for black-box reductions [FSUV13].

This repetition approach has poor output/input ratio less than two. The landmark paper by Nisan [Nis91] gave a better construction using a *design* $S_1, S_2, \dots, S_s \subseteq \{1, 2, \dots, m\}$. Here the generator gets an input σ and computes from it s inputs $x_1, x_2, \dots, x_s$ for f where x_i is the bits of σ indexed by S_i.

$$N : \{0,1\}^m \to \{0,1\}^{m+s}$$
$$N(\sigma) := \sigma \circ f(x_1) \circ f(x_2) \circ \cdots \circ f(x_s).$$

*Supported by NSF CCF award 1813930.

The repetition generator can be seen as a trivial design where the S_i are disjoint. Nisan allows us to take s much larger than m, and so the stretch is essentially unaffected if we drop σ from the output.

Jumping ahead, our result only applies if σ is revealed in the output. We note that most or all applications of such generators in the literature are unaffected if σ is revealed, while there are applications (for example [KvMS12]) which actually require that σ is revealed. Still, we view the case where σ is not revealed as an interesting open problem which we discuss more in Section 3.

Starting from less hard f. If the hardness of f is less, say if it is a constant bounded away from 1/2, then the above approaches give only a constant number of bits of pseudo-randomness. Since at least [NW94] the way around this has been to *amplify the hardness* of f to obtain another function f' with hardness $1/2 - \epsilon$ where $\epsilon \leq 1/s$, and then apply a generator construction to f'. The simplest and most well-known hardness amplification is the XOR lemma originating from oral presentations by Yao in the 80's, see [GNW95]. Here f' is simply the XOR of k independent copies of f:

$$f' : \{0,1\}^{\ell \cdot k} \to \{0,1\}$$
$$f'(x_1, x_2, \ldots, x_k) := \oplus_{j \leq k} f(x_j).$$

Here by $\oplus_{j \leq k}$ we mean the XOR over all $j \in \{1, 2, \ldots, k\}$.

For a concrete example, if we combine this hardness amplification with the repetition generator we obtain the following construction where $m = \ell k s$:

$$\begin{aligned} G &: \{0,1\}^m \to \{0,1\}^{m+s} \\ G(x_{ij})_{\substack{i \leq s \\ j \leq k}} &:= (x_{ij})_{\substack{i \leq s \\ j \leq k}} \circ \oplus_{j \leq k} f(x_{1j}) \circ \oplus_{j \leq k} f(x_{2j}) \circ \cdots \circ \oplus_{j \leq k} f(x_{sj}). \end{aligned} \tag{1}$$

If we start with f that has constant hardness then in Yao's XOR lemma for any given s we can take $k = O(\log s)$ to obtain hardness say $1/2 - 1/s^2$ which is sufficient to show that G has error $O(1/s)$.

The extension of Nisan's generator where we first apply Yao's XOR lemma appears in the work of Nisan and Wigderson [NW94]. It is as follows

$$\begin{aligned} NW &: \{0,1\}^m \to \{0,1\}^{m+s} \\ NW(\sigma) &:= \sigma \circ \oplus_{j \leq k} f(x_{1j}) \circ \oplus_{j \leq k} f(x_{2j}) \circ \cdots \circ \oplus_{j \leq k} f(x_{sj}), \end{aligned} \tag{2}$$

where now all the x_{ij} are computed from σ via a design as before.

Hardness amplification proofs require majority. This hardness amplification step is problematic for *restricted computational models.* This is because most or all reductions that prove hardness amplification are *black-box,* and a line of research [Vio04, Vio06, LTW11, SV10, GR08, AS14, GSV18] has shown that these reductions cannot be implemented in a class less powerful than TC^0. The sense in which this holds is made precise below, but basically the reduction circuits that amplify hardness to $1/2 - \epsilon$ can be used to compute majority on $\Omega(1/\epsilon)$ bits. This means that when ϵ is polynomially small these reductions can only be used when starting from a lower bound against at least TC^0, a long-standing open problem which is believed to require new techniques [RR97, NR04]. In particular, they cannot be used for several classes for which we do have δ-hard functions for $\delta = 0.1$ but not even $\delta = 1/2 - 1/\sqrt{\ell}$ for functions on ℓ bits. Such classes include AC^0 circuits with parity gates [Raz87, Smo87] or with one majority gate [ABFR94]. We refer to [GSV18] for additional discussion and pointers.

Moreover, most proofs of pseudorandom generators $PRG : \{0,1\}^m \to \{0,1\}^{m+s}$ also yield error about $1/s$. For example in the above example of G if we amplify hardness to $1/s^2$ we obtain error $O(1/s)$. And the same limitations apply to a pseudorandomness proof with error ϵ as to a hardness amplification proof to hardness $1/2 - \epsilon$ [GSV18].

Constant-error pseudorandomness. However such limitations were not known for pseudorandomness proofs with *constant error*. Pseudorandom generators with constant error are important in their own right. For example they suffice for derandomization, and are not known for several frontier classes such as AC^0 with parity gates. In this paper we show that even the proofs of the constructions above *with constant error* require TC^0. For s bits of stretch we obtain the same limitations established in [GSV18] for amplifying hardness to $1/2 - 1/s$. We state a more general and abstract result, where the generator computes inputs x_{ij} from the seed σ via a function L, then evaluates f at those inputs and combines the output via a function H.

Let us prepare for the technical statement of the result. A *black-box proof of correctness* consists of a *reduction* showing that if the generator is broken then the function f is not hard. More formally, such a reduction is an oracle circuit C which receives an input x and oracle access to a distinguisher D for the generator, and outputs $f(x)$ correctly on a $1 - \delta$ fraction of inputs. If D has small size and depth, and C has small size and depth, one obtains a circuit $C'(x) = C^D(x)$ also of small size and depth that computes f well, thus contradicting the hardness of f. It is known that in this setup, the reduction circuit C must use an additional *advice string* α, which may arbitrarily depend on the function f and the distinguisher D. More precisely, the reduction circuit C fulfills the following guarantee: For every function f, and every distinguisher D, there exists an advice string α for which $\mathbb{P}_{x \in \{0,1\}^\ell}[C^D(x, \alpha) = f(s)] \geq 1 - \delta$. The reader is referred to [SV10, GSV18] for additional discussion. Following [Vio06, SV10], the theorem below shows that reduction circuits C of this type can be used to distinguish uniform bits from slightly biased bits. Specifically let us denote by $N^q_{1/2-\epsilon}$ a q-tuple of i.i.d. bits coming up 1 with probability $1/2 - \epsilon$. The theorem shows how to distinguish between $N^q_{1/2}$ and $N^q_{1/2-\Omega(1/s)}$ using roughly the same size and depth as that of the circuits C (with no oracle and using the same gates as in C plus gates for H and L). By arguments in [SV10] we can then show how to compute majority on $\Omega(s)$ bits, and moreover the circuits C must make many queries.

Theorem 1. *There is a constant c such that if* $\max\{s, a, q, k\} \leq g \leq 2^{\ell/c}$ *and* $2^{-\ell/c} \leq \delta \leq 1/3$ *the following holds.*

Hypothesis: *Let $G^f(\sigma) = \sigma \circ G^f_1(\sigma) \circ G^f_1(\sigma) \circ \cdots \circ G^f_s(\sigma)$ be a pseudorandom-generator construction mapping a seed $\sigma \in \{0,1\}^m$ and a function $f : \{0,1\}^\ell \to \{0,1\}$ to $m + s$ output bits defined by*

$$G^f_i(\sigma) = H(f(x_{i1}), f(x_{i2}), \ldots, f(x_{ik}))$$

where $H : \{0,1\}^k \to \{0,1\}$ is a function, and the $s \cdot k$ values x_{ij} for $i \leq s$ and $j \leq k$ are obtained from σ by the map $L : \{0,1\}^m \to \{0,1\}^{\ell \cdot s \cdot k}$. Suppose that with probability $1 - 1/1000$ over σ, the x_{ij} values $L(\sigma)$ are distinct.

Let $\{C^{\cdot}(., \alpha)\}_{\alpha \in \{0,1\}^a}$ be a family of 2^a oracle circuits such that for every α the circuit $C^{\cdot}(., \alpha)$ has size g, depth κ, makes q oracle queries, and uses gates $\Xi \supseteq \{And, Or, Not\}$.

Suppose that for every $f : \{0,1\}^\ell \to \{0,1\}$ and $d : \{0,1\}^{m+s} \to \{0,1\}$ such that

$$\left|\mathbb{P}[d\left(G^f(\sigma)\right) = 1] - \mathbb{P}[d\left(U_{m+s}\right) = 1]\right| \geq \frac{1}{100},$$

where σ and U_{m+s} are uniform in $\{0,1\}^m$ and $\{0,1\}^{m+s}$ respectively, there exists $\alpha \in \{0,1\}^a$ such that

$$\mathbb{P}_{x \in \{0,1\}^\ell}[C^d(x, \alpha) = f(x)] \geq 1 - \delta.$$

Conclusion: *Then for every $\epsilon = \Omega(1/s)$ there is a circuit $t : \{0,1\}^q \to \{0,1\}$ such that*

$$\mathbb{P}[t(N^q_{1/2-\epsilon}) = 1] - \mathbb{P}[t(N^q_{1/2}) = 1] \geq 1 - O(\delta),$$

and moreover t has depth $O(\kappa)$, size $\mathrm{poly}(g, 1/\delta)$*, and only uses gates for H and L in addition to the gates Ξ.*

By the arguments in sections 5 and 6 of [SV10], the conclusion of the theorem has two corollaries:

(A) $q \geq \Omega(s^2 \log(1/\delta))$. For this the fact that t can be computed by small circuits is not required.

(B) there is a circuit with depth $O(\kappa)$, size $\mathrm{poly}(g, 1/\delta)$, and the same gates as t that computes majority on inputs of length $\Omega(s)$. In particular if H, L, C are computable in AC^0 or even in AC^0 with parity gates, the size of C must be exponential in $s^{\Omega(1)}$. By contrast, for unbounded-depth circuits the size can be polynomial. Note for this $\delta = 1/3$ is sufficient.

The function L is the identity map in the repetition PRG, and is a projection in NW. In both cases it is trivially computable in AC^0. The function H is simply parity on k bits in both cases, obviously computable in AC^0 with parity gates. In fact, as discussed before, in typical settings we have $k = O(\log s)$ and so this parity is even computable in AC^0.

The theorem assumes that with probability $1 - 1/1000$ over σ the x_{ij} are distinct. We now remark that for typical settings of both generators (1) and (2) this assumption holds. When each x_{ij} is the projection of σ on a set of ℓ bits, as long as the pairwise intersection of the sets is $\leq \ell/2$ then the probability that two x_{ij} will be equal is $\leq \binom{ks}{2} 2^{-\ell/2}$ by a union bound. In particular if $k, s = \ell^{O(1)}$ the assumption is satisfied for large enough ℓ. (The designs in [Nis91, NW94] have even smaller intersection guarantees.)

The corollaries (A) and (B) are the same obtained in [GSV18] for amplification to hardness $1/2 - 1/s$. As explained there, the bounds are tight for hardness amplification. Specifically, the bound on the number of queries matches [KS03], and the result about majority is tight because there exist proofs of Yao's XOR lemma which can be implemented in AC^0 with one majority gate [Kli01] (for a simplification of [Kli01], due to Klivans and Vadhan, see [Vio09]). We note that such tightness extends to the generators (1) and (2), because the pseudorandomness proof from a $1/2 - 1/s$ hard function can be implemented in AC^0 with only one query [Vio07].

Note that for constructions with small error ϵ the work [GSV18] rules out AC^0 reductions even for one bit of stretch, that is $s = 1$. In our setting of constant error ϵ we can't do that because it is possible to get $s = \operatorname{poly}\log g$ bits of stretch via an AC^0 reduction of size g, and our result shows that this is tight.

Theorem 1 would immediately follow from [GSV18] if there was a way to turn a constant-error generator into a $1/2 - 1/s$ hard function with a proof of correctness by a reduction that does not use majority. No such way is known, and in fact we suspect one can derive a black-box separation for this task along the lines of [GSV18].

Organization. After presenting the proof we conclude in Section 3 where we also discuss several open problems.

1 Proof discussion

We explain the basic idea of the proof in the case of the repetition generator (1). Take F to be a random function $F : \{0,1\}^\ell \to \{0,1\}$ and the distinguisher d as a distribution D defined as follows. Recall that the input to D is a string z of length $m + s$ which consists of sk inputs x_{ij} to F and s additional bits b_i. On input z the distinguisher D selects a uniform pointer $P(z) \in \{1, 2, \ldots, s\}$ and then answers one if and only if the bit b_i matches the output of the pseudorandom generator, that is, if $b_i = \oplus_{j \le k} F(x_{ij})$. Such a D breaks the pseudorandom generator, as it always outputs 1 when the input is pseudorandom, but if b_i is uniform then the equation $b_i = \oplus_{j \le k} F(x_{ij})$ is satisfied with probability 1/2.

Now imagine, for some fixed α, a reduction circuit $C^D(x^*, \alpha)$ that is trying to compute $F(x^*)$ by querying D. Equivalently, the circuit is trying to distinguish $F_0 := F|F(x^*) = 0$ from $F_1 := F|F(x^*) = 1$. We can assume that in any query z to D the inputs x_{ij} are all different (by instructing D to answer 0 otherwise). Then the answers to each query in F_0 and F_1 are close: they are random bits whose statistical distance is $O(1/s)$. When s is large enough, this difference is too small to be detected by a constant-depth circuit.

Formalizing this idea presents several difficulties. First, the above definition of D actually does not work. This is because, using advice, the reduction could ask queries where all the values $F(x_{ij})$ are 0 except possibly $F(x^*)$. In such a situation the answers of D would be always 0 under F_0, and non-zero with probability $\ge 1/s$ under F_1. This *can* be detected in constant-depth (by a simple Or). The fix is to pad the bits b_i with a balanced string, so that the answer bits always come up 1 with probability bounded away from 0 and 1.

To handle advice, we use the *fixed-set lemma* from [GSV18], which basically says that conditioning on a specific α can be thought of as fixing some values of F and D. As in [GSV18], this lemma appears to greatly simplify the argument.

We note that although we use this lemma from [GSV18], and other results from [SV10], our proof presents several differences. The main one is in the choice of the oracle. In previous works this choice is straightforward: the oracle is simply obtained by perturbing the encoding of f with suitable noise, and the main focus is on the analysis. In our case the choice is less obvious. This difference then propagates to several other parts of the proof, for example in how we handle adaptive queries.

2 Formal proof

Let $F : \{0,1\}^\ell \to \{0,1\}$ be a uniform function. We define the distinguisher D as follows. Let $P : \{0,1\}^{m+s} \to [3s]$ be a uniform function. On input $z = \sigma \circ b_1 \circ \cdots \circ b_s \in \{0,1\}^{m+s}$, compute

$$L(\sigma) = \{x_{ij}\}_{\substack{i \le s \\ j \le k}}.$$

For simplicity we think of the output of $L(\sigma)$ as a multiset. If the x_{ij} are not all distinct then output zero. Otherwise compute the string $v(z) \in \{0,1\}^s$ where the bit i is the indicator of whether b_i is the correct output of the hard function, i.e., it is the indicator of $H(F(x_{i1}), F(x_{i2}), \ldots, F(x_{ik})) = b_i$. Return the bit $P(z)$ of the string $v(z)0^s1^s \in \{0,1\}^{3s}$.

Claim 2. For every F, with probability $\Omega(1)$ over the choice of P, D distinguishes the PRG from uniform with advantage $\Omega(1)$.

Proof. We first prove the claim ignoring the fact that D outputs 0 if the x_{ij} are not all distinct.

On any input z from the PRG, D outputs 1 unless $P(z)$ lands on 0^s. The latter happens with probability 1/3 independently. Hence with probability say $1 - 1/100$ over P, assuming s is large enough, D outputs 1 with probability $\ge 2/3 - 1/100$ over a uniform output of the PRG. Note we can assume that s is large enough for else the conclusion of the theorem is trivial.

We now analyze the behavior for a uniform input u. First note that at least say a 0.99 fraction of the strings $u \in \{0,1\}^{m+s}$ are such that $v(u)$ has Hamming weight $\le 0.51s$, again using that s is large enough. (Note that this statement only depends on F, which is fixed.) For any such string u, the distinguisher outputs 1 with probability $\le (1 + 0.51)/3$ over the choice of $P(u)$. Hence with probability $1 - 1/100$ over P the distinguisher outputs 1 on at most a $(1+0.52)/3$ fraction of those strings, and hence overall on at most a $1/100+(1+0.52)/3$ fraction of the strings in $\{0,1\}^{m+s}$.

By a union bound, with probability $\ge 1 - 1/50$ over P, D has an advantage of

$$\ge (2/3 - 1/100) - (1/100 + (1 + 0.52)/3) = 0.14.$$

Finally we take into account the probability that the x_{ij} are not all distinct. This happens with probability $\leq 1/1000$ by assumption. Hence we still have advantage $\geq \Omega(1)$. $\square$

By the above claim, averaging over F, and our assumption, there is an advice $\alpha \in \{0,1\}^a$ such that with probability $\geq \Omega(2^{-a})$ over F and P the event $A :=$ "$\mathbb{P}_{x\in\{0,1\}^\ell}[C^D(x,\alpha) = F(x)] \geq 1-\delta$" happens. We now use the *fixed-set lemma* from [GSV18], restated next.

Lemma 3. *Let N, a, q' be integers. Let $Y = (Y_1, \ldots, Y_N)$ be independent random variables, each uniform over some finite set Σ. Let $A \subseteq \Sigma^N$ be an event such that $\mathbb{P}[Y \in A] \geq 2^{-a}$, and let $X = (Y|Y \in A)$. For every $\eta > 0$, there exists a set $B \subseteq [N]$ of size* $\mathrm{poly}(a, q', \eta^{-1})$*, and $\mu \in \{0,1\}^B$ such that for $Y' := (Y|Y_B = \mu)$ and $X' := (X|X_B = \mu) = (Y|Y_B = \mu, Y \in A)$, and every q'-query decision tree t, $|\mathbb{P}[t(Y') = 1] - \mathbb{P}[t(X') = 1]| \leq \eta$.*

The notation X_B means the variables X_i where $i \in B$. We apply the lemma to the $2^\ell + 2^{m+s}$ random variables $F(x), P(z)$. We set $q' = O(qk)$ and $\eta = \delta$.

Remark 4. The fixed-set lemma is only stated for variables with the same range Σ, whereas $F(x)$ and $P(z)$ have different range. However we can for example rewrite $F(x)$ and $P(z)$ as fixed functions of random variables with the same range Σ (for example $|\Sigma| = 2 \cdot 3s$ will do). So we can apply it in our setting as well.

The lemma gives a set B of size $\mathrm{poly}(a, q', 1/\eta) = \mathrm{poly}(g, 1/\delta)$ (using $a, q, k \leq g$) and a string μ such that for every x

$$|\mathbb{P}_{F,P}[C^D(x,\alpha) = F(x)|A, (FP)_B = \mu] - \mathbb{P}_{F,P}[C^D(x,\alpha) = F(x)|(FP)_B = \mu]| \leq \delta, \qquad (3)$$

where note in the second probability there is no conditioning on A. Here we are using that checking if $C^D(x,\alpha) = F(x)$ can be computed by a decision tree with access to the variables $F(x)$ and $P(z)$ making $\leq 1 + q + qk = O(qk)$ queries. The tree first queries $F(x)$, then it simulates $C^D(x,\alpha)$, answering each oracle query $D(z)$ by querying $P(z)$ and then the k corresponding values of F.

Recall that for every F, P such that A holds we have by definition of A that

$$\mathbb{P}_x[C^D(x,\alpha) = F(x)] \geq 1-\delta.$$

Hence the same holds if we average over F, P and further condition:

$$\mathbb{P}_{x,F,P}[C^D(x,\alpha) = F(x)|A, (FP)_B = \mu] \geq 1-\delta.$$

By Equation 3 we can drop the conditioning over A and have

$$\mathbb{P}_{x,F,P}[C^D(x,\alpha) = F(x)|(FP)_B = \mu] \geq 1-2\delta.$$

We now want to fix a "good" x such that $F(x)$ is uniform and C does not get too much "information" about $F(x)$ from the oracle. The first condition simply means $x \notin B$. The second is more complicated. We need to avoid that C makes a query to a z such that $P(z)$ is fixed to a bit that depends on $F(x)$; as that could give away the value $F(x)$ (for example if the answer is $F(x) \oplus 0 \oplus 0 \oplus \cdots \oplus 0$). For simplicity, we shall fix F at all values $x \in \{0,1\}^\ell$ that appear in any $z \in B$. We can enlarge B to include such problematic x:

$$B' := B \bigcup \{x \in \{0,1\}^\ell : x \in L(\sigma) \text{ for some } \sigma \circ b \in B\}.$$

Note that $|B'| \leq |B| + |B| \cdot k \cdot s \leq \text{poly}(g, 1/\delta)$ (by the above bound on $|B|$ and using $k, s \leq g$). By averaging there exists a corresponding fixing μ' such that again

$$\mathbb{P}_{x,F,P}[C^D(x,\alpha) = F(x) | (FP)_{B'} = \mu'] \geq 1 - 2\delta.$$

Hence we have that there exists a fixed $x^* \notin B'$ such that

$$\mathbb{P}_{F,P}[C^D(x^*,\alpha) = F(x^*) | (FP)_{B'} = \mu'] \geq 1 - 2\delta - |B'|/2^\ell \geq 1 - 3\delta,$$

using that $\delta \geq |B'|/2^\ell$, which is implied by $2^{-\ell/c} \geq \text{poly}(2^{\ell/c})/2^\ell$ because $\delta \geq 2^{-\ell/c}$ and $g \leq 2^{\ell/c}$, and hence true for a large enough c.

Now let us hardwire x^* and α and write C_0 for $C(x^*, \alpha)$. So we have

$$\mathbb{P}_{F,P}[C_0^D = F(x^*) | (FP)_{B'} = \mu'] \geq 1 - 3\delta.$$

Next we produce a series of circuits C_i to arrive to the desired t. Each circuit will have depth $O(\kappa)$, size $\text{poly}(g, 1/\delta)$, will not increase the number of oracle queries, and will use the same gates Ξ used by C_0, and also gates for H and L.

Our first task is to get rid of the queries $z \in B$. If $z \in B$ then $P(z)$ is fixed. Let the corresponding evaluation be $H = H(F(x_1), F(x_2), \ldots, F(x_k))$. All the x_i belong to B', hence the corresponding values of F are fixed and so $D(z)$ is fixed if $z \in B$. Construct a circuit C_1 which has all the values $D(z)$ for $z \in B$ stored in a table, and answers a query $z \in B$ with that fixed value. So we have

$$\mathbb{P}_{F,P}[{C_1}^D = F(x^*) | (FP)_{B'} = \mu'] \geq 1 - 3\delta,$$

and we now know that C_1 only makes queries z where $P(z)$ is uniform. Note that because $|B| = \text{poly}(g, 1/\delta)$ the size of C_1 is again $\text{poly}(g, 1/\delta)$.

At this point the conditioning on $P_{B'} = \mu'$ is immaterial, hence we drop it. Let us pause a moment to discuss the meaning of the notation used. We defined above X_B to mean the variables X_i where $i \in B$. Typically the number of such variables is $|B|$, but we can also use this notation when B is a superset of the index set of the X_i. In this case the number of the variables X_B may be smaller than $|B|$. This is what we mean with the notation $P_{B'} = \mu'$. So we can write

$$\mathbb{P}_{F,P}[{C_1}^D = F(x^*) | F_{B'} = \mu'] \geq 1 - 3\delta.$$

Recall that $F(x^*)$ is still uniform conditioned on $F_{B'} = \mu'$ because $x^\star \notin B'$. Hence the previous equation can be rewritten as

$$\frac{1}{2}(\mathbb{P}_{F,P}[{C_1}^D = 1 | F_{B'} = \mu', F(x^*) = 1] + \mathbb{P}_{F,P}[{C_1}^D = 0 | F_{B'} = \mu', F(x^*) = 0]) \geq 1 - 3\delta.$$

We multiply by 2 this inequality and then complement the second probability to rewrite the inequality as

$$\mathbb{P}_{F,P}[C_1^D = 1 | F_{B'} = \mu', F(x^*) = 1] - \mathbb{P}_{F,P}[{C_1}^D = 1 | F_{B'} = \mu', F(x^*) = 0] \geq 1 - 6\delta. \qquad (4)$$

Now for $b \in \{0,1\}$ define the oracle E_b as follows. The input is a string $w \in \{0, 1, ?, 1-?\}^s$ with exactly one occurrence of ? (possibly as $1-?$). The output is obtained by replacing ? with b (and $1-?$ with $1-b$) to obtain $w_b \in \{0,1\}^s$ and then outputting a uniformly selected bit of $w_b 0^s 1^s$.

Lemma 5. *There is a distribution on circuits C_2 such that for every $b \in \{0,1\}$ we have*

$$\mathbb{P}[{C_2}^{E_b} = 1] = \mathbb{P}_{F,P}[{C_1}^D = 1 | F_{B'} = \mu', F(x^*) = b].$$

Proof. The high-level idea is that C_2 fills the unfixed values of $F(x)$ on the fly, for every $x \neq x^*$, while keeping a table of its choices. More precisely, initialize the table with the values $F_{B'} = \mu'$. Then arrange the oracle gates of C_1 in levels $1, 2, \ldots$ with level 1 being closest to the input. To perform the queries

$$\begin{aligned} z_1 &= \sigma_1 b_{11} \cdots b_{1s} \\ z_2 &= \sigma_2 b_{21} \cdots b_{2s} \\ &\cdots \\ z_t &= \sigma_t b_{t1} \cdots b_{ts} \end{aligned}$$

for $t \leq q$, at some level i, C_2 first computes $L' := L(\sigma_1) \bigcup L(\sigma_2) \bigcup \ldots \bigcup L(\sigma_t)$. For every $x \in L'$ that is in the table, it fetches the corresponding value of F. For the others, except for x^*, it tosses a coin, and stores the value in the table. From these values it computes t strings $w'_1, w'_2, \ldots, w'_t \in \{0, 1, ?, 1-?\}^s$ by computing H. If an evaluation of H depends on the unknown value $F(x^*)$ then C_2 writes ? or $1-?$ depending on what this dependency is. (If H syntactically depends on $F(x^*)$ but actually the other input bits of H already determine the value of H, C_2 writes a value in $\{0,1\}$. This obviously never happens when $H =$Parity.)

Recall from above that there is at most one occurrence of x^* in each $L(\sigma_i)$ (for else the oracle always outputs 0 and the query isn't made). Hence each of the strings w'_i has ≤ 1 occurrence of ?. Then the strings $w_1, w_2, \ldots, w_t$ are constructed by performing bit-by-bit equality with the b_{ij}. More formally the j bit of w_i is $w_{ij} := 1 \oplus w'_{ij} \oplus b_{ij}$, where $\oplus$ is bit-wise xor and the expression is the indicator of the equality function (here $(1-?) \oplus 1 =?$ etc.). For those strings w_i which happen to be in $\{0,1\}^s$, the oracle query is simply answered by the circuit by returning a uniform bit of $w_i 0^s 1^s$. For the other strings, C_2 queries E_b. □

Now we further simplify the oracle to zoom in on strings which are nearly balanced. Earlier we had $|w| = s$ now we will have $|w| = 1$. Let E'_b be the oracle that takes as input $w \in \{?, 1-?\}$, substitutes b for ? to obtain $w_b \in \{0,1\}$ and then outputs a uniform bit of $w_b 0^s 1^s$.

Lemma 6. *There is a distribution on circuits C_3 of size polynomial in that of C_2 and constant depth such that for every $b \in \{0,1\}$ we have*

$$\mathbb{P}[C_3^{E'_b} = 1] = \mathbb{P}[{C_2}^{E_b} = 1].$$

Proof. Think of answering an oracle query to E_b as follows. On input $w \in \{0, 1, ?, 1-?\}^s$, write $w = w'w''$ where $w'' \in \{?, 1-?\}$. First toss a *selection coin* to decide if the answer will be a uniform bit from w' or from $w''_b 0^s 1^s$. In the former case we can answer the query without invoking the oracle. In the latter case the oracle query is answered using E'_b. □

Now we want to replace the oracles E'_0, E'_1 with inputs $N^q_{1/2}, N^q_{1/2-\epsilon}$. Note E'_0 on input ? outputs $N_{s/(2s+1)}$ and on input $1-?$ outputs $N_{(s+1)/(2s+1)}$. E'_1 does the same but with ? and $1-?$ swapped. We use the following lemma to map pairs of distributions.

Lemma 7. *Let p, γ, ϵ be such that the following quantities are in $[0, 1]$: $1/2 - \epsilon, p + \gamma, p - \gamma, p + \gamma + \gamma/2\epsilon, p + \gamma - \gamma/2\epsilon$. There is a distribution on functions $M : \{0, 1\} \to \{0, 1\}$ such that $M(N_{1/2}) \equiv N_{p+\gamma}$ and $M(N_{1/2-\epsilon}) \equiv N_{p-\gamma}$, where $\equiv$ denotes equivalence as distributions.*

Proof. $M(0)$ outputs 1 with probability $\alpha := p + \gamma - \gamma/2\epsilon$, $M(1)$ outputs 1 with probability $\beta := p + \gamma + \gamma/2\epsilon$. Then

$$\begin{aligned} \mathbb{P}[M(N_{1/2}) = 1] &= (1/2)\beta + (1/2)\alpha = p + \gamma, \\ \mathbb{P}[M(N_{1/2-\epsilon}) = 1] &= (1/2 - \epsilon)\beta + (1/2 + \epsilon)\alpha = p + \gamma + \epsilon(-\beta + \alpha) = p - \gamma. \end{aligned}$$

□

Consider the circuit C_4 that on input a string $x \in \{0, 1\}^q$ simulates C_3. Oracle query at gate i on input ? is answered applying Lemma 7 to bit x_i of x with $p = 1/2$ and $\gamma = -1/2(2s + 1)$. If x_i is sampled according to $N_{1/2}$ then we get $N_{1/2-1/2(2s+1)} = N_{s/(2s+1)}$, and if x_i is sampled according to $N_{1/2-\epsilon}$ then we get $N_{1/2+1/2(2s+1)} = N_{(s+1)/(2s+1)}$, as desired. On input $1-?$ we instead pick $\gamma = +1/2(2s + 1)$. In both cases the hypotheses of the lemma hold for $\epsilon = \Omega(1/s)$. This shows that

$$\begin{aligned} \mathbb{P}[C_4(N^q_{1/2}) = 1] &= \mathbb{P}[C_3^{E'_0} = 1], \\ \mathbb{P}[C_4(N^q_{1/2-\epsilon}) = 1] &= \mathbb{P}[C_3^{E'_1} = 1]. \end{aligned}$$

Combining this with the above lemmas and Equation 4 gives that

$$\mathbb{P}[C_4(N^q_{1/2-\epsilon}) = 1] - \mathbb{P}[C_4(N^q_{1/2}) = 1] \geq 1 - 6\delta.$$

By averaging we can fix C_4 to a circuit t, and this concludes the proof.

3 Discussion and open problems

This work points to several open problems. First, can we prove similar limitations when the seed σ is not part of the output of the generator? Our proof strategy immediately encounters a problem because it is not clear anymore how to define the distinguisher D. On input z, which values of F should D use to answer? Here is a candidate definition of D. On input z, consider the string z' that is closest to z in Hamming distance and that can be output by the generator. Pick a uniform index i and answer $z_i \oplus z'_i$ (suitably padded). Then we would need to understand how changing one value of F affects the output of D. But here technical difficulties arise that we have not yet been able to overcome.

Problem 8. Prove Theorem 1 when the seed σ is not part of the output (for $s \gg m$).

Second, what can we prove for fixed hard functions f? The result is actually false! More in detail, recall that our proof is black-box in both the use of the distinguisher d and of the hard function f. For proofs that are only black-box in d but that can be tailored to specific functions f the result is false: [FSUV13] showed with an AC^0 reduction that the repetition generator is pseudorandom *with no error loss* when starting with any "resamplable function," such as Parity. Using this they constructed the best-known generators for classes such as AC^0 with modular gates. However it is not known how to push their techniques to Nisan-style pseudorandom generators with much better stretch. A natural proof strategy would require sampling the output distribution of the generator, which is impossible [LV12]. So it may be interesting to understand if the techniques in this paper can be extended to handle a fixed f like Parity. We remark that for this one needs a proof that further exploits that the reduction's queries are a low-complexity function of the input x, or a different distinguisher. This is because a reduction on input x could make for example the query $(x_1 0 x_2 0 \ldots x_t 0)$ where each x_i has the same parity as x. Then a distinguisher such as ours would allow to compute the parity of x. And this is exactly how the proof in [FSUV13] works. Intuitively we cannot compute such correlated x_i for a random function f, and our proof formalizes the intuition.

Because Nisan's generator applied to the parity function is just the uniform distribution over a vector space, we can ask:

Problem 9. Understand the pseudorandomness properties of vector spaces $V \subseteq \{0,1\}^n$ with bit-wise addition modulo 2. In particular, for how small m there exists a space of dimension m that fools AC^0 with mod 3 gates? What about a single mod 3 gate? What can be proved using reductions that are black-box in the use of the distinguisher?

We remark that some of the earlier works did hold for fixed f. For example Artemenko and Shaltiel proved [AS14] a query lower bound even for fixed f (in the setting of hardness amplification with small error).

Finally, for the NW generator (where L is projection and H is parity) the circuits t in the conclusion of Theorem 1 are AC^0 with parity gates. It is not clear to us how to obtain AC^0 circuits (unless k is small). It may be useful to think about this as a stepping stone towards handling arbitrary H.

Acknowledgments. In a previous version, the conclusion of Theorem 1 had the incorrect size bound poly(g) instead of poly($g, 1/\delta$). Moreover, in the proof η was set incorrectly, whereas it suffices to set $\eta = \delta$. We thank a referee for pointing this out. The change in the size bound from poly(g) to poly($g, 1/\delta$) does not affect the corollaries mentioned after Theorem 1, because there either δ is a constant or the size bound is irrelevant. Another referee, whom we thank as well, pointed out that in the conclusion of Theorem 1 we can take t to be a fixed circuit, as opposed to a distribution over circuits as also stated in a previous version. We are also grateful to the referees for several other detailed comments which improved the presentation.

References

[ABFR94] James Aspnes, Richard Beigel, Merrick Furst, and Steven Rudich. The expressive power of voting polynomials. *Combinatorica. An Journal on Combinatorics and the Theory of Computing*, 14(2):135–148, 1994.

[AS14] Sergei Artemenko and Ronen Shaltiel. Lower bounds on the query complexity of non-uniform and adaptive reductions showing hardness amplification. *Computational Complexity*, 23(1):43–83, 2014.

[FSUV13] Bill Fefferman, Ronen Shaltiel, Christopher Umans, and Emanuele Viola. On beating the hybrid argument. *Theory of Computing*, 9:809–843, 2013.

[GNW95] Oded Goldreich, Noam Nisan, and Avi Wigderson. On Yao's XOR lemma. Technical Report TR95–050, *Electronic Colloquium on Computational Complexity*, March 1995. www.eccc.uni-trier.de/.

[GR08] Dan Gutfreund and Guy Rothblum. The complexity of local list decoding. In *12th Intl. Workshop on Randomization and Computation (RANDOM)*, 2008.

[GSV18] Aryeh Grinberg, Ronen Shaltiel, and Emanuele Viola. Indistinguishability by adaptive procedures with advice, and lower bounds on hardness amplification proofs. In *IEEE Symp. on Foundations of Computer Science (FOCS)*, 2018. Available at http://www.ccs.neu.edu/home/viola/.

[Kli01] Adam R. Klivans. On the derandomization of constant depth circuits. In *Workshop on Randomization and Computation (RANDOM)*. Springer, 2001.

[KS03] Adam Klivans and Rocco A. Servedio. Boosting and hard-core sets. *Machine Learning*, 53(3):217–238, 2003.

[KvMS12] Jeff Kinne, Dieter van Melkebeek, and Ronen Shaltiel. Pseudorandom generators, typically-correct derandomization, and circuit lower bounds. *Computational Complexity*, 21(1):3–61, 2012.

[LTW11] Chi-Jen Lu, Shi-Chun Tsai, and Hsin-Lung Wu. Complexity of hard-core set proofs. *Computational Complexity*, 20(1):145–171, 2011.

[LV12] Shachar Lovett and Emanuele Viola. Bounded-depth circuits cannot sample good codes. *Computational Complexity*, 21(2):245–266, 2012.

[Nis91] Noam Nisan. Pseudorandom bits for constant depth circuits. *Combinatorica. An Journal on Combinatorics and the Theory of Computing*, 11(1):63–70, 1991.

[NR04] Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. *J. of the ACM*, 51(2):231–262, 2004.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. of Computer and System Sciences*, 49(2):149–167, 1994.

[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.

[RR97] Alexander Razborov and Steven Rudich. Natural proofs. *J. of Computer and System Sciences*, 55(1):24–35, August 1997.

[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.

[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.

[Vio06] Emanuele Viola. *The Complexity of Hardness Amplification and Derandomization.* PhD thesis, Harvard University, 2006.

[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.