On the Complexity of Hardness Amplification

Chi-Jen Lu[*] Shi-Chun Tsai[†] Hsin-Lung Wu[‡]

Abstract

We study the task of transforming a hard function f, with which any small circuit disagrees on $(1-\delta)/2$ fraction of the input, into a harder function f', with which any small circuit disagrees on $(1-\delta^k)/2$ fraction of the input, for $\delta \in (0,1)$ and $k \in \mathbb{N}$. We show that this process can not be carried out in a black-box way by a circuit of depth d and size $2^{o(k^{1/d})}$ or by a nondeterministic circuit of size $o(k/\log k)$ (and arbitrary depth). In particular, for $k = 2^{\Omega(n)}$, such hardness amplification can not be done in $\mathsf{ATIME}(O(1), 2^{o(n)})$. Therefore, hardness amplification in general requires a high complexity. Furthermore, we show that even without any restriction on the complexity of the amplification procedure, such a black-box hardness amplification must be inherently non-uniform in the following sense. Given as an oracle any algorithm which agrees with f' on $(1-\delta^k)/2$ fraction of the input, we still need an additional advice of length $\Omega(k\log(1/\delta))$ in order to compute f correctly on $(1-\delta)/2$ fraction of the input. Therefore, to guarantee the hardness, even against uniform machines, of the function f', one has to start with a function f which is hard against non-uniform circuits. Finally, we derive similar lower bounds for any black-box construction of pseudorandom generators from hard functions.

1 Introduction

1.1 Background

Understanding the power of randomness in computation is one of the central topics in theoretical computer science. A major open question is the BPP versus P question, asking whether or not all randomized polynomial-time algorithms can be converted into deterministic polynomial-time ones. A standard approach to derandomizing BPP relies on constructing the so-called pseudorandom generators (PRG), which stretch a short random seed into a long pseudorandom string that looks random to circuits of polynomial size. So far, all known constructions of PRG are based on unproven assumptions of the nature that certain functions are hard to compute. The idea of converting hardness into pseudorandomness first appeared implicitly in the work of Blum and Micali [2] and Yao [24]. This was made explicit by Nisan and Wigderson [14], who showed how to construct a PRG based on a Boolean function which is hard in an average-case sense. To get a stronger result,

[*]Institute of Information Science, Academia Sinica, Taipei, Taiwan. E-mail: `cjlu@iis.sinica.edu.tw`. This work is supported in part by the National Science Council of Taiwan under contract NSC93-2213-E-001-004.

[†]Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan. E-mail: `sctsai@csie.nctu.edu.tw`. The work was supported in part by the National Science Council of Taiwan under contract NSC-92-2213-E-009-035.

[‡]Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan. E-mail: `hsinlung@csie.nctu.edu.tw`

one would like to relax the hardness assumption, and a series of research [14, 1, 8] then worked on how to transform a function into a harder one. Finally, Impagliazzo and Wigderson [10] were able to convert a function in E that is hard in worst case into one that is hard in average case, both against circuits of exponential size. As a result, they obtained $\mathsf{BPP} = \mathsf{P}$ under the assumption that some function in E can not be computed by a circuit of sub-exponential size. Simpler proofs and better trade-offs have been obtained since then [18, 9, 17, 21].

Note that hardness amplification is the major step in derandomizing BPP in the research discussed above, as the step from an average-case hard function to a PRG is relatively simple and has low complexity. We say that a Boolean function f is α–hard (or has hardness α) against circuits of size s if any such circuit attempting to compute f must make errors on at least α fraction of the input. The error bound α is the main parameter characterizing the hardness; the size bound s also reflects the hardness, but it plays a lesser role in our study. Formally, the task of hardness amplification is to transform a function $f : \{0,1\}^n \to \{0,1\}$ which is α–hard against circuits of size $s(n)$ into a function $f' : \{0,1\}^m \to \{0,1\}$ which is α'–hard against circuits of size $s'(m)$, with $\alpha < \alpha'$ and $s'(m)$ close to (usually slightly smaller than) $s(n)$. Normally, one would like to have m as close to n as possible, preferably with $m = \text{poly}(n)$, so that one could have $s'(m)$ close to $s(m)$; otherwise, one would only be able to have the hardness of f' against much smaller circuits. Furthermore, one would like f' to stay in the same complexity class of f, so that one could establish the relation among hardness assumptions within the same complexity class.

Two issues come up from those works on hardness amplification. The first is on the complexity of the amplification procedure. All previous amplification procedures going from worst-case hardness ($\alpha = 2^{-n}$) to average-case hardness ($\alpha' = 1/2 - 2^{-\Omega(n)}$) need exponential time [1, 10, 18] (or slightly better, in linear space [12] or $\oplus\mathsf{ATIME}(O(1), n)$ [22]). As a result, such a hardness amplification is only known for functions in high complexity classes. Then a natural question is: can it be done for functions in lower complexity classes? For example, given a function in NP which is worst-case hard, can we transform it into another function in NP which is average-case hard? Only for some range of hardness (e.g. starting from mild hardness, with $\alpha = 1/\text{poly}(n)$) is this known to be possible [24, 14, 10, 15, 7].

The second issue is that hardness amplification typically involves non-uniformity in the sense that hardness is usually measured against *non-uniform* circuits. In fact, one usually needs to start from a function which is hard against non-uniform circuits, even if one only wants to produce a function which is hard against uniform Turing machines. This is why most results on derandomizing BPP are based on non-uniform assumptions (except [11, 20]).

1.2 Black-Box Hardness Amplification

In light of the discussion above, one would hope to show that some hardness amplification is indeed impossible. However, it is not clear what this means, especially given the possibility (in which many people believe) that average-case hard functions may indeed exist.

One important type of hardness amplification is called *black-box* hardness amplification. First, the initial function f is only given as a black-box to construct the new function f'. That is, there is an oracle Turing machine $\textsc{Amp}$ such that $f' = \textsc{Amp}^f$, so f' only uses f as an oracle and does not depend on the internal structure of f. Second, the hardness of the new function f' is proved in a black-box way. That is, there is an oracle Turing machine $\textsc{Dec}$, such that if some algorithm A computes f' correctly on α' fraction of the input, then $\textsc{Dec}$ using A as an oracle can compute f correctly on α fraction of the input. Again, $\textsc{Dec}$ only uses A as an oracle and does not depend on the internal structure of A. We call Amp the *encoding* procedure and Dec the *decoding* procedure. In fact, almost all previous constructions of hardness amplification (except [11, 20]) are done in a black-box way, so it is nice to establish impossibility results for such type of hardness amplification.

1.3 Previous Lower Bound Results

Viola [22] gave the first lower bound on the complexity required for black-box hardness amplification. He showed that to transform a worst-case hard function f into a mildly hard function f', both against circuits of size $2^{o(n)}$, the encoding procedure Amp can not possibly belong to the complexity class $\mathsf{ATIME}(O(1), 2^{o(n)}))$. This rules out the possibility of doing such hardness amplification in PH, which explains why previous such procedures all require a high computational complexity. He also showed a similar lower bound for black-box construction of PRG from a worst-case hard function.

Trevisan and Vadhan [20] observed that a black-box hardness amplification from worst-case hardness corresponds to an error-correcting code with some list-decoding property. Then results from coding theory can be used to show that for any such amplification from worst-case hardness to hardness $(1-\varepsilon)/2$, the decoding procedure Dec must need $\Omega(\log(1/\varepsilon))$ bits of advice in order to compute f. This explains why almost all previous hardness amplification results were done in a non-uniform setting, except [11, 20] which did not work in a black-box way.

There were also impossibility results on weaker types of hardness amplification, from worst-case hardness to average-case hardness. Bogdanov and Trevisan [3] considered hardness amplification for functions in NP in which the black-box requirement on the *encoding* procedure is dropped. They showed that the decoding procedure can not be computed non-adaptively in polynomial time unless PH collapses. Viola, in another recent paper [23], considered hardness amplification in which the black-box requirement on the *decoding* procedure is dropped. He showed that if the encoding procedure can be computed in PH, then there exists an average-case hard function in PH unconditionally. We will not consider such weaker types of hardness amplification in this paper, and hereafter when we refer to hardness amplification, we always mean the black-box one.

1.4 Our Results

Unlike previous results which only focus on specific range of hardness, we consider amplifying hardness in a much broader spectrum: from hardness $(1-\delta)/2$ to hardness $(1-\delta^k)/2$, for general $\delta \in (0,1)$ and $k \in \mathbb{N}$.

Our first two results address both the complexity issue and the non-uniformity issue in the same framework, showing how complexity constraints on the encoding procedure result in the inherent non-uniformity of the decoding procedure. Formally, we prove that if the encoding procedure Amp for such a hardness amplification is computed by a circuit of depth d and size $2^{o(k^{1/d})}$ or by a nondeterministic circuit of size $o(k/\log k)$ (and arbitrary depth), the decoding procedure Dec must need an advice of length $2^{\Omega(n)}$. As a result, with Amp $\in \mathsf{PH}$ when $k = n^{\omega(1)}$ or with Amp $\in \mathsf{ATIME}(O(1), 2^{o(n)})$ when $k = 2^{\Omega(n)}$, such a hardness amplification is impossible if the hardness is measured against circuits of size $2^{o(n)}$. In fact in either case, it is impossible to produce a function which is $(1-\delta^k)/2$–hard against a uniform low complexity class, say $\mathsf{DTIME}(O(1))$, even if we start from a function which is $(1-\delta)/2$–hard against a uniform but arbitrarily high complexity class equipped with an advice of length $2^{o(n)}$, say $\mathsf{DTIME}(2^{2^n})/2^{o(n)}$.[1] This demonstrates one severe

[1]Note that hard functions against $\mathsf{DTIME}(O(1))$ do exist. For example, the parity function is $(1/2 - 2^{-\Omega(n)})$–hard against $\mathsf{DTIME}(O(1))$, but according to our result, its hardness cannot be shown in such a black-box way.

weakness of black-box hardness amplifications. Another interesting fact from our results is the following. When amplifying hardness from $(1-\delta)/2$ to $(1-\delta^k)/2$, what determines the complexity of such amplification is the parameter k; a large k forces a high complexity requirement, for typical values of δ. This has Viola's result in [22] as a special case, which addresses only the specific case with $(1-\delta)/2 = 2^{-n}$ and $(1-\delta^k)/2 = 1/\text{poly}(n)$ (or equivalently, $\delta = 1 - 2^{-n+1}$ and $k = 2^{\Omega(n)}$). Our lower bound is tight since the well known XOR lemma [24] indeed can achieve such a hardness amplification. Note that our result, when restricted to worst-case to average-case hardness amplification, is incomparable to those of [3] and [23].[2]

Our third result shows that even without any complexity constraint on the encoding or decoding procedure, amplification between certain range of hardness is still inherently non-uniform. One can derive this for the case of amplifying hardness beyond 1/4, using Plotkin Bound [16] from coding theory. We obtain a quantitative bound on the non-uniformity for a general range of hardness amplification. We show that to amplify from hardness $(1-\delta)/2$ to hardness $(1-\varepsilon)/2$, the decoding procedure DEC must need an advice of $\Omega(\log(\delta^2/\varepsilon))$ bits. Thus, when $\varepsilon = \delta^k$, an advice of length $\Omega(k \log(1/\delta))$ is necessary, and when $\varepsilon \leq c\delta^2$ for some constant c, such hardness amplification must be inherently non-uniform. Our result generalizes that of Trevisan and Vadhan [20].

Finally, we derive similar lower bounds on black-box constructions of PRG from any function with a given hardness.

1.5 Our Techniques

Our results are obtained via a connection between black-box hardness amplifications and some type of "error-reduction" codes, which generalizes the connection given by Trevisan and Vadhan [20] and Viola [22]. A similar observation was also made by Trevisan [19]. Formally, a black-box amplification from hardness $(1-\delta)/2$ to hardness $(1-\varepsilon)/2$ induces a code with the following list-decoding property. Given a corrupted codeword with a fraction of less than $(1-\varepsilon)/2$ errors, we can always find a small list of candidate messages such that one of them is close to the original message, with their relative Hamming distance less than $(1-\delta)/2$. Therefore, we can focus our attention on such codes, as lower bounds on such codes immediately translate to lower bounds on the corresponding hardness amplification.

Our first two results are based on the following idea. A code with such a list-decoding property can only have a small number of codewords close to any codeword, so a random perturbation on an input message is unlikely to result in a close codeword. On the other hand, if such a code is computed by an algorithm which is insensitive to noise on the input, then a random perturbation on an input message is likely to result in a close codeword, and we reach a contradiction. Circuits of small size, or circuits of small depth and moderate size can be shown to be insensitive to noise on their input. Thus, they can not be used to compute such a code and the corresponding hardness amplification. This basically follows Viola's idea in [22], but since we consider hardness amplification in a much broader spectrum, a more involved analysis is required. For example, we need to prove a new upper bound on noise sensitivity, because Viola's bound does not work well for us when we consider amplifying hardness from $(1-\delta)/2$ to $(1-\delta^k)/2$, for general $\delta \in (0,1)$ and $k \in \mathbb{N}$.

[2] In [3], the complexity lower bound is placed on the decoding procedure instead, under the unproven (though widely believed) assumption that PH does not collapse. In [23], a more general type of hardness amplification than ours is considered, but the possibility of such hardness amplification is not ruled out as we do; instead, it was shown that if the encoding procedure can be computed in PH, a hard function in PH exists unconditionally.

For the non-uniformity of hardness amplification, we show that given a corrupted codeword with a high fraction $(1-\varepsilon)/2$ (for a small ε) of errors, one may need a long list of candidate messages in order to have one of them within a small relative distance $(1-\delta)/2$ (for a large δ) to the original message. To show this, we would like to find a set of messages such that some ball of relative radius $(1-\varepsilon)/2$ in the codeword space contains many of their corresponding codewords, but any ball of relative radius $(1-\delta)/2$ in the message space contains only a small number of messages from that set. We choose these messages randomly and show that they have some chance of satisfying the condition above when $(1-\varepsilon)/2$ is larger than $(1-\delta)/2$ to some extent.

The methods described above can be easily modified to prove the corresponding lower bounds for black-box constructions of PRG from hard functions. Note that lower bounds for PRG in fact imply lower bounds for hardness amplification. However, we choose to present the proofs for hardness amplification in detail and then indicate the changes needed for PRG. Our reason is that the proofs for hardness amplification are simpler so it is easier to get a good picture. Furthermore, there is a natural connection between hardness amplification and some list-decodable code as discussed before, and establishing such lower bounds for these codes may have interests of their own, so it may still be nice to have more direct proofs for them.

1.6 Organization of this paper

First, some preliminaries are given in Section 2. Then in Section 3 and Section 4, we prove the impossibility results of hardness amplification by constant-depth circuits and non-deterministic circuits respectively. In Section 5, we show that hardness amplification in general is inherently non-uniform. Finally, we show the impossibility results for black-box construction of pseudorandom generators from hard functions in Section 6.

2 Preliminaries

For any $n \in \mathbb{N}$, let $[n]$ denote the set $\{1, 2, \ldots, n\}$ and let U_n denote the uniform distribution over the set $\{0,1\}^n$. For a string z, let z_i denote the i'th bit of z. All the logarithms in this paper will have base two. Define the binary entropy function $H(x) = -x \log x - (1-x) \log(1-x)$.

We need some standard complexity classes. Let $\mathsf{ATIME}(d,t)$ denote the class of functions computed by alternating Turing machines in time t with d alternations. Let PH denote the polynomial-time hierarchy, which is $\mathsf{ATIME}(O(1), \mathrm{poly}(n))$. Let $\mathsf{NTIME}(t)$ denote the class of functions computed by nondeterministic Turing machines in time t. For our convenience, we also introduce some slightly unconventional complexity classes defined in terms of Boolean circuits. The circuits we consider consist of AND/OR/NOT gates, allowing unbounded fan-in for AND/OR gates. The size of a circuit is the number of gates it has and the depth of circuit is the number of gates on the longest path from an input bit to the output gate. We call such circuits AC circuits.

Definition 1. *Let $\mathsf{AC}(s)$ be the class of functions computed by AC circuits of size s. Let $\mathsf{AC}(d,s)$ denote the class of functions computed by AC circuits of depth d and size s.*

Note that the standard complexity class AC^0 corresponds to our class $\mathsf{AC}(O(1), \mathrm{poly}(n))$. We also introduce the nondeterministic version of AC circuits. An NAC circuit C has two parts of inputs: the real input x and the witness input y. The Boolean function f computed by such a circuit C is defined as $f(x) = 1$ if and only if there exists a y such that $C(x,y) = 1$.

Definition 2. *Let* $\mathsf{NAC}(s)$ *be the class of functions computed by* NAC *circuits of size* s*.*

A function with more than one output bits is said to be computed by some type of circuits (e.g. $\mathsf{AC}(d, s)$ or $\mathsf{NAC}(s)$) if each output bit can be computed by one such circuit.

2.1 Black-Box Hardness Amplification and Pseudorandom Generators

Informally speaking, a function is hard if any algorithm without enough complexity must make some mistakes.

Definition 3. *We say that a function* $f : \{0,1\}^n \to \{0,1\}$ *has hardness* δ *against circuits of size* s *if for any circuit* $C : \{0,1\}^n \to \{0,1\}$ *of size* s*,* $\Pr_{x \in U_n}[f(x) \neq C(x)] \geq \delta$*.*

Note that we use the error bound δ to characterize the hardness of a function, and we pay less (sometimes no) attention to the size bound s. For hardness amplification, we want to transform a function $f : \{0,1\}^n \to \{0,1\}$ with a smaller hardness δ into a function $f' : \{0,1\}^m \to \{0,1\}$ with a larger hardness ε. Next, we define what we mean by a black-box hardness amplification.

Definition 4. *We say that an oracle algorithm* $\textsc{Amp}^{(\cdot)} : \{0,1\}^m \to \{0,1\}$ *realizes a black-box* $(n, \delta, \varepsilon, \ell)$ *hardness amplification if there exists a (non-uniform) oracle Turing machine* $\textsc{Dec}$ *with* ℓ *bits of advice satisfying the following. For any* $f : \{0,1\}^n \to \{0,1\}$ *and any* $A : \{0,1\}^m \to \{0,1\}$*, if* $\Pr_{z \in U_m}[A(z) \neq \textsc{Amp}^f(z)] < \varepsilon$*, then* $\Pr_{x \in U_n}[\textsc{Dec}^{A,\alpha}(x) \neq f(x)] < \delta$ *for some* ℓ*-bit advice* α*.*

Here, the transformation of the initial function f into a harder function is done in a black-box way, as the harder function $\textsc{Amp}^f$ only uses f as an oracle. Moreover, the hardness of the new function $\textsc{Amp}^f$ is also guaranteed in a black-box way. Namely, any algorithm A breaking the hardness condition of $\textsc{Amp}^f$ can be used as an oracle for a machine $\textsc{Dec}$ to break the hardness condition of f. Note that neither of the hardness refers to circuit size, and no constraint is placed on the complexity of the procedure $\textsc{Dec}$ and the complexity of the function A. This freedom makes our impossibility results stronger. The parameter ℓ characterizes the amount of non-uniformity associated with this process. When $\ell \geq 1$, we say the hardness amplification is non-uniform.

Similarly, we can define the notion of black-box construction of pseudo-random generators from hard functions.

Definition 5. *We say that an oracle algorithm* $G^{(\cdot)} : \{0,1\}^r \to \{0,1\}^M$ *realizes a black-box* $(n, \delta, \varepsilon, \ell)$*-PRG if there exists a (non-uniform) oracle Turing machine* $\textsc{Dec}$ *with* ℓ *bits of advice satisfying the following. For any* $f : \{0,1\}^n \to \{0,1\}$ *and any* $D : \{0,1\}^M \to \{0,1\}$*, if* $|\Pr_{x \in U_r}[D(G^f(x)) = 1] - \Pr_{y \in U_M}[D(y) = 1]| > \varepsilon$*, then* $\Pr_{x \in U_n}[\textsc{Dec}^{D,\alpha}(x) \neq f(x)] < \delta$*, for some* ℓ*-bit advice* α*.*

2.2 Codes and Correspondence to Hardness Amplification

The distance between two strings we adopt is their relative Hamming distance.

Definition 6. *For* $u, v \in \{0,1\}^M$*, define their distance* $\triangle(u, v)$ *as their relative Hamming distance, namely* $\triangle(u, v) = \frac{1}{M}|\{i \in [M] : u_i \neq v_i\}|$*.*

According to this distance, we define open balls of radius δ in the space $\{0,1\}^N$.

Definition 7. *For any $N \in \mathbb{N}$, $\delta \in (0,1)$, and $x \in \{0,1\}^N$, let $\textsc{Ball}_x(\delta, N) = \{x' \in \{0,1\}^N : \triangle(x,x') < \delta\}$, which is the open ball in $\{0,1\}^N$ of radius δ centered at x. Let $\textsc{Ball}(\delta, N)$ denote the set containing all such balls.*

The following fact gives an upper bound on the size of such a Hamming ball.

Fact 1. *The size of any ball in $\textsc{Ball}(\delta, N)$ is at most $2^{H(\delta)N}$.*

We borrow the notion of list-decodable codes, but we extend it in a way that leads to some natural correspondence with black-box hardness amplifications.

Definition 8. *We call $C : \{0,1\}^N \to \{0,1\}^M$ a (δ, ε, L)-list code if for any $z \in \{0,1\}^M$, there are L balls from $\textsc{Ball}(\delta, N)$ such that if a codeword $C(x)$ is contained in $\textsc{Ball}_z(\varepsilon, M)$, then x is contained in one of those L balls.*

A (δ, ε, L)-list code is related to a standard list-decodable code in the way that each ball in $\textsc{Ball}(\varepsilon, M)$ contains at most $L \cdot 2^{H(\delta)N}$ codewords. Next, we show how such a code arises naturally from a black-box hardness amplification. Let $N = 2^n$ and $M = 2^m$. Given any oracle algorithm $\textsc{Amp}^{(\cdot)} : \{0,1\}^m \to \{0,1\}$, let us define the corresponding code $C : \{0,1\}^N \to \{0,1\}^M$ as $C(f) = \textsc{Amp}^f$. That is, seeing any function $f : \{0,1\}^n \to \{0,1\}$ as a vector in $\{0,1\}^N$, $C(f)$ produces as output the function $\textsc{Amp}^f$, which is seen as a vector in $\{0,1\}^M$. The following is a simple generalization of an observation by Viola [22].

Lemma 1. *If $\textsc{Amp}^{(\cdot)} : \{0,1\}^m \to \{0,1\}$ realizes a black-box $(n, \delta, \varepsilon, \ell)$ hardness amplification, then $C : \{0,1\}^N \to \{0,1\}^M$, defined as $C(f) = \textsc{Amp}^f$, is a $(\delta, \varepsilon, 2^\ell)$-list code.*

Proof. Suppose $\textsc{Amp}$ realizes a black-box $(n, \delta, \varepsilon, \ell)$ hardness amplification and $\textsc{Dec}$ is the corresponding decoding procedure, which is an oracle Turing machine with an ℓ-bit advice. Consider any $A \in \{0,1\}^M$, seen as $A : \{0,1\}^m \to \{0,1\}$. For any codeword $C(f)$ with $\triangle(A, C(f)) = \Pr_x[A(x) \neq \textsc{Amp}^f(x)] < \varepsilon$, by Definition 4, there exists an $\alpha \in \{0,1\}^\ell$ such that $\triangle(\textsc{Dec}^{A,\alpha}, f) = \Pr_y[\textsc{Dec}^{A,\alpha}(y) \neq f(y)] < \delta$. That is, if $C(f)$ is in $\textsc{Ball}_A(\varepsilon, M)$, then f is contained in one of the 2^ℓ balls of radius δ centered at $\textsc{Dec}^{A,\alpha}$ for $\alpha \in \{0,1\}^\ell$. Therefore, C is a $(\delta, \varepsilon, 2^\ell)$-list code.

□

2.3 Noise Sensitivity

Following [15, 22], we apply Fourier analysis on Boolean functions. For any $g : \{0,1\}^N \to \{0,1\}$ and for any $J \subseteq [N]$, let $\hat{g}(J) = \mathrm{E}_y\left[(-1)^{g(y)} \cdot \prod_{i \in J} (-1)^{y_i}\right]$. Here is a simple fact.

Fact 2. *For any $g : \{0,1\}^N \to \{0,1\}$, $\sum_{J \subseteq [N]} \hat{g}(J)^2 = 1$.*

It is known that for AC circuits of small depths, the main contribution to the above sum comes from the low-order terms.

Lemma 2. *[13] For any $g : \{0,1\}^N \to \{0,1\} \in \mathsf{AC}(d, s)$ and for any $t \in [N]$, $\sum_{|J| > t} \hat{g}(J)^2 \leq s \cdot 2^{-\Omega(t^{1/d})}$.*

This can be used to show that AC circuits of small depth are insensitive to noise on their input. We will need the following more precise relation between the noise sensitivity of a Boolean function and its Fourier coefficients.

Lemma 3. *Suppose x is sampled from the uniform distribution over $\{0,1\}^N$ and $\tilde{x}$ is obtained by flipping each bit of x independently with probability $\frac{1-\alpha}{2}$. Then for any $g : \{0,1\}^N \to \{0,1\}$ and for any $t \in \mathbb{N}$, $\Pr_{x,\tilde{x}}[g(x) \neq g(\tilde{x})] \leq \frac{1}{2}(1 - \alpha^t(1 - \sum_{|J|>t} \hat{g}(J)^2))$.*

Proof. We know from [15] that $\Pr_{x,\tilde{x}}[g(x) \neq g(\tilde{x})] = \frac{1}{2}(1 - \sum_{J \subseteq [N]} \alpha^{|J|} \hat{g}(J)^2)$. Note that

$$\sum_{J \subseteq [N]} \alpha^{|J|} \hat{g}(J)^2 \geq \sum_{|J| \leq t} \alpha^{|J|} \hat{g}(J)^2 \geq \alpha^t \sum_{|J| \leq t} \hat{g}(J)^2.$$

Then the lemma follows from Fact 2. □

3 Impossibility of Amplification by Small-Depth Circuits

We will show that any AC circuit of small depth realizing a black-box hardness amplification requires a large size. Let $N = 2^n$ and $M = 2^m$. Recall from Lemma 1 that any $\text{Amp}^{(\cdot)} : \{0,1\}^m \to \{0,1\}$ realizing a black-box $(n, \frac{1-\delta}{2}, \frac{1-\delta^k}{2}, \ell)$ hardness amplification induces a $(\frac{1-\delta}{2}, \frac{1-\delta^k}{2}, 2^\ell)$-list code $C : \{0,1\}^N \to \{0,1\}^M$. Suppose Amp is realized by an $\mathsf{AC}(d,s)$ circuit, with oracle answers as part of the input. Then for each $i \in [M]$, the i'th output bit of C can be seen as a function $C(\cdot)_i : \{0,1\}^N \to \{0,1\}$ computable by an $\mathsf{AC}(d,s)$ circuit. So it suffices to prove that no $\mathsf{AC}(d,s)$ circuits with small d and s can compute such a code.

Let x be sampled from the uniform distribution over $\{0,1\}^N$ and let $\tilde{x}$ be the random variable obtained by flipping each bit of x independently with some probability $\frac{1-\alpha}{2}$. We set $\alpha = \delta^{1.1}$ so that $\frac{1-\alpha}{2}$ is only slightly larger than $\frac{1-\delta}{2}$.[3] We call any two codewords *close* if their (relative) distance is less than $\frac{1-\delta^k}{2}$. The next lemma gives a lower bound on the probability that $C(\tilde{x})$ is close to $C(x)$. The idea is that an AC circuit of small depth and small size is insensitive to noise on the input, so a random perturbation on an input message is likely to result in a close codeword.

Lemma 4. *For some constant c, for any $t, d \in \mathbb{N}$ with $\alpha^t \leq 1 - 2^{-ct^{1/d}}$, and for any $C \in \mathsf{AC}(d, 2^{ct^{1/d}})$, $\Pr_{x,\tilde{x}}[C(x)$ is close to $C(\tilde{x})] \geq \alpha^{2t} - \delta^k$.*

Proof. From Lemma 2 and Lemma 3, we know that for each $i \in [M]$,

$$\Pr_{x,\tilde{x}}[C(x)_i \neq C(\tilde{x})_i] \leq \frac{1}{2}\left(1 - \alpha^t\left(1 - 2^{ct^{1/d}} \cdot 2^{-\Omega(t^{1/d})}\right)\right) \leq \frac{1-\alpha^{2t}}{2},$$

for some constant c. Therefore, $\mathrm{E}_{x,\tilde{x}}[\triangle(C(x), C(\tilde{x}))] \leq \frac{1-\alpha^{2t}}{2}$, and from Markov's inequality, $\Pr_{x,\tilde{x}}[C(x)$ is not close to $C(\tilde{x})] \leq \frac{1-\alpha^{2t}}{1-\delta^k}$. Thus

$$\Pr_{x,\tilde{x}}[C(x) \text{ is close to } C(\tilde{x})] \geq 1 - \frac{1-\alpha^{2t}}{1-\delta^k} \geq \frac{\alpha^{2t} - \delta^k}{1-\delta^k} \geq \alpha^{2t} - \delta^k.$$

□

Next, we give an upper bound on this probability. The idea is that if C is a code with each codeword only close to a small number of other codewords, then a random perturbation on an input message is unlikely to result in a close codeword.

[3]We do not attempt to optimize parameters here, and in fact it suffices to set $\alpha = \delta(1 - o(1))$.

Lemma 5. *For any $(\frac{1-\delta}{2}, \frac{1-\delta^k}{2}, 2^\ell)$-list code C, $\Pr_{x,\tilde{x}}[C(x)$ is close to $C(\tilde{x})] \le 2^\ell \cdot 2^{-\Omega(\delta^2 N)}$.*

Proof. Consider any fixed $x \in \{0,1\}^N$. Since C is a $(\frac{1-\delta}{2}, \frac{1-\delta^k}{2}, 2^\ell)$-list code, there are at most $2^{\ell + H(\frac{1-\delta}{2})N}$ different y's such that $C(y)$ is close to $C(x)$. The lemma would follow easily if each such y had a very small probability to occur. However, this may not be the case in general. We will show that although some y's may occur with higher probability, there are not too many of them, so their overall contribution is still tolerable.

For any $y \in \{0,1\}^N$, $\Pr_{\tilde{x}}[\tilde{x} = y] = \left(\frac{1-\alpha}{2}\right)^{\triangle(x,y)N}\left(\frac{1+\alpha}{2}\right)^{(1-\triangle(x,y))N}$, which decreases as $\triangle(x,y)$ increases. Let $\beta = \alpha^{0.91} = \delta^{1.001}$.[4] Call $y \in \{0,1\}^N$ *good* for x if $\triangle(x,y) \ge \frac{1-\beta}{2}$ and call y *bad* for x otherwise. Note that for any y which is good for x,

$$\begin{aligned}
\Pr_{\tilde{x}}[\tilde{x} = y] &\le \left(\frac{1-\alpha}{2}\right)^{\frac{1-\beta}{2}N}\left(\frac{1+\alpha}{2}\right)^{\frac{1+\beta}{2}N} \\
&= 2^{\left(\frac{1-\beta}{2}\log\frac{1-\alpha}{2} + \frac{1+\beta}{2}\log\frac{1+\alpha}{2}\right)N} \\
&\le 2^{-H\left(\frac{1-\beta}{2}\right)N}.
\end{aligned}$$

On the other hand, $\tilde{x}$ is only bad for x with a small probability. This is because $\tilde{x}$ is obtained by flipping each bit of x independently with probability $\frac{1-\alpha}{2}$, so $\mathrm{E}_{\tilde{x}}[\triangle(x,\tilde{x})] = \frac{1-\alpha}{2}$, and by Chernoff bound,

$$\Pr_{\tilde{x}}[\tilde{x} \text{ is bad for } x] = \Pr_{\tilde{x}}\left[\triangle(x,\tilde{x}) < \frac{1-\beta}{2}\right] \le 2^{-\Omega(\beta^2 N)}.$$

Thus, $\Pr_{\tilde{x}}[C(\tilde{x})$ is close to $C(x)]$ is at most

$$\begin{aligned}
&\Pr_{\tilde{x}}[C(\tilde{x}) \text{ is close to } C(x) \wedge \tilde{x} \text{ is good for } x] + \Pr_{\tilde{x}}[\tilde{x} \text{ is bad for } x] \\
\le\; & 2^{\ell + H\left(\frac{1-\delta}{2}\right)N} \cdot 2^{-H\left(\frac{1-\beta}{2}\right)N} + 2^{-\Omega(\beta^2 N)} \\
=\; & 2^\ell \cdot 2^{(1-\Theta(\delta^2))N} \cdot 2^{-(1-\Theta(\beta^2))N} + 2^{-\Omega(\beta^2 N)} \\
=\; & 2^\ell \cdot 2^{-\Omega(\delta^2 N)}.
\end{aligned}$$

Since this holds for every x, the lemma follows. □

Combing Lemma 4 and Lemma 5, we have

$$\alpha^{2t} - \delta^k \;\le\; \Pr_{x,\tilde{x}}[C(x) \text{ is close to } C(\tilde{x})] \;\le\; 2^\ell \cdot 2^{-\Omega(\delta^2 N)}.$$

Choose $t = \Theta(k)$ such that $\alpha^{2t} - \delta^k = \delta^{O(k)}$, and we have the following.

Lemma 6. *For some constant c, for any $\delta \in (0,1)$, and for any $d, k \in \mathbb{N}$ with $\delta^k \le 1 - 2^{-ck^{1/d}}$, if $C : \{0,1\}^N \to \{0,1\}^M$ is a $(\frac{1-\delta}{2}, \frac{1-\delta^k}{2}, L)$-list code computable by an $\mathsf{AC}(d, 2^{ck^{1/d}})$ circuit, then $L = 2^{\Omega(\delta^2 N)}\delta^{O(k)}$.*

From Lemma 1, we obtain the following impossibility result for hardness amplification.

[4] Again, we make no attempt on optimizing the parameter here. In fact it suffices to set $\beta = \alpha(1 + o(1))$ while still maintaining $\beta = \delta(1 - o(1))$.

Theorem 1. *Suppose $2^{-c_0 n} \leq \delta < 1$ and $2^{-2^{c_1 n}} \leq \delta^k \leq 1 - 2^{-c_2 k^{1/d}}$, for some suitable constants c_0, c_1, c_2. Then any black-box $(n, \frac{1-\delta}{2}, \frac{1-\delta^k}{2}, \ell)$ hardness amplification in $\mathsf{AC}(d, 2^{c_2 k^{1/d}})$ must have $\ell = 2^{\Omega(n)}$.*

The condition on δ and δ^k in the theorem above is natural in the following sense. When $\delta \leq 2^{-\Omega(n)}$, the initial function is already hard enough, so hardness amplification is usually not needed. When $\delta^k \geq 1 - 2^{-\Omega(k^{1/d})}$, the resulting function only has a very small hardness, which is rarely what hardness amplification is used to achieve. Finally, as discussed in the introduction, hardness amplifications normally have m close to n (preferably with $m = \text{poly}(n)$), therefore δ^k, which is at least 2^{-m}, would be much larger that $2^{-2^{\Omega(n)}}$.

Corollary 1. *Under the same condition as in Theorem 1, no black-box $(n, \frac{1-\delta}{2}, \frac{1-\delta^k}{2}, 2^{o(n)})$ hardness amplification can be realized in $\mathsf{ATIME}(O(1), k^{o(1)})$. In particular, no such hardness amplification is possible in PH when $k = n^{\omega(1)}$ or in $\mathsf{ATIME}(O(1), 2^{o(n)})$ when $k = 2^{\Omega(n)}$.*

Proof. It is known (e.g. from [4]) that any $\mathsf{ATIME}(d, t)$ computation with an oracle can be simulated by an $\mathsf{AC}(d + 1, 2^{O(dt)})$ circuit with oracle answers as part of its input. Then the corollary follows from Theorem 1. $\square$

Our bound is tight as it can be achieved by the well-known XOR lemma [24].

Theorem 2. *For any $\delta \in (0, 1)$ and any $k, d \in \mathbb{N}$, a black-box $(n, \frac{1-\delta}{2}, \frac{1-\delta^k}{2}, \ell)$ hardness amplification can be realized in $\mathsf{AC}(d, 2^{O(k^{1/d})})$ with $\ell = k \cdot \text{poly}\left(\frac{n}{\delta}\right)$.*

Proof. The XOR of k bits can be computed by an $\mathsf{AC}(d, 2^{O(k^{1/d})})$ circuit (c.f. [6]), and note that the proof of XOR lemma in [5] shows that $\ell \leq k \cdot \text{poly}\left(\frac{n}{\delta}\right)$ suffices. $\square$

4 Impossibility of Amplification by Nondeterministic Circuits

The bound in the previous section vanishes when $d = \Omega(\log k)$. In this section, we show that even without any restriction on the circuit depth, there still exists a lower bound on the circuit size.

We use the method of random restriction. A restriction on a set of variables $V = \{x_i : i \in [N]\}$ is a mapping $\rho : V \to \{0, 1, \star\}$, which either fixes the value of a variable x_i with $\rho(x_i) \in \{0, 1\}$ or leaves x_i free with $\rho(x_i) = \star$. For $p \in (0, 1)$, let R_p denote the distribution on such restrictions such that each variable x_i is mapped independently with $\Pr_{\rho \in R_p}[\rho(x_i) = \star] = p$ and $\Pr_{\rho \in R_p}[\rho(x_i) = 0] = \Pr_{\rho \in R_p}[\rho(x_i) = 1] = (1 - p)/2$. For a Boolean function g and a restriction ρ, let g_ρ denote the function obtained from g by applying the restriction ρ to its variables. That is, $g_\rho(x_1, \ldots, x_N) = g(y_1, \ldots, y_N)$ with $y_i = x_i$ if $\rho(x_i) = \star$ and $y_i = \rho(x_i)$ otherwise.

Define the degree of a function g as $deg(g) = \max_J\{|J| : \hat{g}(J) \neq 0\}$. It is not hard to see that a constant function has degree 0 and a function depending on only t input bits has degree at most t. We need the following lemma which bounds the contribution of higher-order Fourier coefficients.

Lemma 7. *[13] Let t be an integer and p with $pt > 8$. For any Boolean function g, $\sum_{|J|>t} \hat{g}(J)^2 \leq 2 \cdot \Pr_{\rho \in R_p}[deg(g_\rho) \geq pt/2]$.*

The following is the key lemma in this section, which gives a concrete bound on the sum above for NAC circuits.

Lemma 8. *For any $g : \{0,1\}^N \to \{0,1\} \in \mathsf{NAC}(s)$, $\sum_{|J|>t} \hat{g}(J)^2 \leq s \cdot 2^{-\Omega(t/s)}$, when $9 \leq t \leq N$.*

Proof. Suppose g is computed by an NAC circuit of size s, which divides its input into the real part and the witness part. Let $\mathcal{B}$ be the set of gates which receive real input variables directly. Consider applying a random restriction $\rho \in R_p$ on the real input variables. We say a gate in $\mathcal{B}$ is *killed* if it is an AND gate and receives a real input variable which is fixed to 0 by ρ, or if it is an OR gate and receives a real input variable which is fixed to 1 by ρ. For a gate $A \in \mathcal{B}$, let $\#(A)$ denote the number of real input variables it receives. For a restriction ρ, let $\#(A_\rho)$ denote the the number of remaining real input variables it receives if A is not killed by ρ, and let $\#(A_\rho) = 0$ otherwise. Set p to be any constant in $(0,1)$ so that $pt > 8$. Then

$$\begin{aligned} \Pr_{\rho \in R_p}[deg(g_\rho) \geq pt/2] &\leq \Pr_{\rho \in R_p}[\exists A \in \mathcal{B} : \#(A_\rho) \geq pt/(2s)] \\ &\leq s \cdot \max_{A \in \mathcal{B}} \Pr_{\rho \in R_p}[\#(A_\rho) \geq pt/(2s)]. \end{aligned}$$

Any $A \in \mathcal{B}$ with $\#(A) < pt/(2s)$ clearly has $\Pr_{\rho \in R_p}[\#(A_\rho) \geq pt/(2s)] = 0$. On the other hand, any $A \in \mathcal{B}$ with $\#(A) \geq pt/(2s)$ is likely to be killed, so that

$$\Pr_{\rho \in R_p}[\#(A_\rho) \geq pt/(2s)] \leq \Pr_{\rho \in R_p}[A \text{ is not killed by } \rho] = (1-(1-p)/2)^{pt/(2s)} = 2^{-\Omega(t/s)}.$$

From Lemma 7, we have $\sum_{|J|>t} \hat{g}(J)^2 \leq 2s \cdot 2^{-\Omega(t/s)} = s \cdot 2^{-\Omega(t/s)}$. □

Then the rest follows the same line of arguments in the previous section. Suppose $9 \leq t \leq N$ and $C \in \mathsf{NAC}(\frac{t}{c \log t})$ for some large enough constant c. Lemma 8 implies that for each $i \in [M]$,

$$\Pr_{x,\tilde{x}}[C(x)_i \neq C(\tilde{x})_i] \leq \frac{1}{2}\left(1 - \alpha^t \left(1 - \frac{t}{c \log t} \cdot 2^{-\Omega(c \log t)}\right)\right) \leq \frac{1-\alpha^{2t}}{2},$$

when $\alpha^t \leq 1 - t^{-c'}$ for some constant c'. Note that larger c gives larger c'. As in Lemma 4, one can then show that $\Pr_{x,\tilde{x}}[C(x) \text{ is close to } C(\tilde{x})] \geq \alpha^{2t} - \delta^k$. This combined with Lemma 5 gives the following.

Lemma 9. *For some constant c_0, c_1, for any $\delta \in (0,1)$, and for any $k \in \mathbb{N}$ with $\delta^k \leq 1 - k^{-c_0}$, if $C : \{0,1\}^N \to \{0,1\}^M$ is a $(\frac{1-\delta}{2}, \frac{1-\delta^k}{2}, L)$-list code computable by an $\mathsf{NAC}(\frac{k}{c_1 \log k})$ circuit, then $L = 2^{\Omega(\delta^2 N)} \delta^{O(k)}$.*

Then as in the previous section, we get the following impossibility result.

Theorem 3. *Suppose $2^{-c_0 n} \leq \delta < 1$ and $2^{-2^{c_1 n}} \leq \delta^k \leq 1 - k^{-c_2}$, for some suitable constants c_0, c_1, c_2. Then any black-box $(n, \frac{1-\delta}{2}, \frac{1-\delta^k}{2}, \ell)$ hardness amplification in $\mathsf{NAC}(\frac{k}{c_3 \log k})$ or $\mathsf{NTIME}(\frac{k}{c_3 \log^2 k})$, for some constant c_3, must have $\ell = 2^{\Omega(n)}$.*

5 Inherent Non-uniformity of Hardness Amplification

In the previous two sections, we have shown that any black-box hardness amplification must be very non-uniform when the computational complexity of the amplification procedure Amp is bounded in certain ways. In this section, we show that even without any such complexity bound, there still exists some inherent non-uniformity. This reduces to the coding-theoretical question: for which values of α and β do we have a $(\alpha, \beta, 1)$-list code?

We call $C : \{0,1\}^N \to \{0,1\}^M$ an $[N, M, \alpha]$ code if the (relative) distance of any two codewords is at least α. We need the following good code, which can be constructed using, say, the concatenation of Reed-Solomon code with Hadamard code.

Fact 3. $[N, O((\frac{N}{\gamma})^2), \frac{1-\gamma}{2}]$ *codes exist for any* $\gamma \in (0,1)$.

This says that unique decoding is possible if the fraction of error is slightly smaller than $\frac{1}{4}$. On the other hand, according to the following Plotkin bound, unique decoding is basically impossible if the fraction of error grows beyond $\frac{1}{4}$.

Fact 4. *(Plotkin Bound [16]) An* $[N, M, \alpha]$ *code with* $\alpha \geq \frac{1}{2}$ *must have* $N \leq \log(2M)$.

Combining the two facts above, we have the following impossibility result.

Lemma 10. *For some constant* c *and for any* $\gamma \in (0,1)$, *any* $(\frac{1-\gamma}{4}, \frac{1}{4}, L)$*-list code* $C : \{0,1\}^N \to \{0,1\}^M$ *with* $c\gamma\sqrt{N} > \log(2M)$ *must have* $L \geq 2$.

Proof. From Fact 3, there exists an $[K, N, \frac{1-\gamma}{2}]$ code C' with $K \geq c\gamma\sqrt{N}$ for some constant c. Suppose that C is a $(\frac{1-\gamma}{4}, \frac{1}{4}, L)$-list code with $c\gamma\sqrt{N} > \log(2M)$. If $L = 1$, then $C \circ C' : \{0,1\}^K \to \{0,1\}^M$ is a $[K, M, \frac{1}{2}]$ code with $K > \log(2M)$, which is impossible according to Fact 4. □

Then from Lemma 1, we have the following.

Theorem 4. *For some constant* c *and for any* $\gamma \in (0,1)$, *no oracle algorithm* $\textsc{Amp}^{(\cdot)} : \{0,1\}^m \to \{0,1\}$ *can realize a black-box* $(n, \frac{1-\gamma}{4}, \frac{1}{4}, 0)$ *hardness amplification with* $c\gamma 2^{n/2} > m + 1$.

As discussed in the introduction, hardness amplifications normally have $m = \text{poly}(n)$. Thus, the theorem basically says that amplifying hardness beyond $\frac{1}{4}$ must introduce non-uniformity in general. However, the theorem does not provide a quantitative bound on the non-uniformity. This will be addressed next.

5.1 Lower Bounds on Non-uniformity

Next, we will show that any black-box $(n, \frac{1-\delta}{2}, \frac{1-\delta^k}{2}, \ell)$ hardness-amplification must have $\ell = \Omega(k \log \frac{1}{\delta})$. Consider an arbitrary $(\frac{1-\delta}{2}, \frac{1-\varepsilon/c}{2}, L)$-list code $C : \{0,1\}^N \to \{0,1\}^M$, for some suitable constant c. We would like to find $z \in \{0,1\}^M$ and a large enough set $S \subseteq \{0,1\}^N$ such that:

- for every $x \in S$, $C(x)$ is contained in the ball $\textsc{Ball}_z(\frac{1-\varepsilon/c}{2}, M)$, and
- S needs many balls in $\textsc{Ball}(\frac{1-\delta}{2}, N)$ to cover with.

Choose $x^1, \ldots, x^t$ uniformly and independently from $\{0,1\}^N$ to form the set R, for some $t = O(\frac{1}{\varepsilon^2})$. Call the set R δ-good if $|R| = t$ (i.e. $x^i \neq x^j$ for any $i \neq j$) and any ball in $\textsc{Ball}(\frac{1-\delta}{2}, N)$ contains $O(\frac{1}{\delta^2})$ elements of R. Later, we will derive the set S from a δ-good R.

Lemma 11. *When* $N = \omega(\frac{1}{\delta^2} \log \frac{1}{\varepsilon})$, R *is* δ*-good with probability* $1 - 2^{-\Omega(N)}$.

Proof. Clearly, the probability that $x^i = x^j$ for some $i \neq j$ is at most $\binom{t}{2} \cdot 2^{-N} \leq 2^{2\log t - N}$. Also, the probability that some ball in $\textsc{Ball}(\frac{1-\delta}{2}, N)$ contains r elements of R is at most $2^N \cdot \binom{t}{r} \cdot 2^{(H(\frac{1-\delta}{2})-1)Nr} \leq 2^{N + r\log t - \Omega(\delta^2)rN}$. For some $r = O(\frac{1}{\delta^2})$, both probabilities above are $2^{-\Omega(N)}$ when $N = \omega(\frac{1}{\delta^2}\log t)$. $\square$

We want to choose a string $z \in \{0,1\}^M$ such that the ball $\textsc{Ball}_z(\frac{1-\varepsilon}{2}, M)$ contains a lot of codewords coming from a δ-good R. We will fix some of z's bits first.

Definition 9. *For each $y \in [M]$, let b_y be the bit such that $\Pr_{x\in\{0,1\}^N}[C(x)_y \neq b_y] \leq \frac{1}{2}$. Call R (δ,ε)-good for y if R is δ-good and $\Pr_{x\in R}[C(x)_y \neq b_y] \leq \frac{1-\varepsilon}{2}$.*

Lemma 12. *Suppose $N = \omega(\frac{1}{\delta^2}\log\frac{1}{\varepsilon})$. Then for any $y \in [M]$, R is (δ,ε)-good for y with probability $\Omega(1)$.*

Proof. From Lemma 11, R is not δ-good with probability $2^{-\Omega(N)}$. Now fix any $y \in [M]$. Let I_i, for $i \in [t]$, be the random variable such that $I_i = 1$ if $C(x^i)_y \neq b_y$ and $I_i = 0$ otherwise. Note that $I_1, \ldots, I_t$ form a sequence of i.i.d., with $\mathrm{E}[I_i] \leq \frac{1}{2}$ for each i. Then

$$
\begin{aligned}
\Pr_R\left[\Pr_{x\in R}[C(x)_y \neq b_y] \leq \frac{1-\varepsilon}{2}\right] &= \Pr_{x^1,\ldots,x^t}\left[\frac{1}{t}\sum_{i\in[t]} I_i \leq \frac{1-\varepsilon}{2}\right] \\
&\geq \sum_{\frac{1-2\varepsilon}{2}t \leq j \leq \frac{1-\varepsilon}{2}t} \Pr_{x^1,\ldots,x^t}\left[\sum_{i\in[t]} I_i = j\right] \\
&\geq \frac{\varepsilon t}{2} \cdot \binom{t}{\frac{1-2\varepsilon}{2}t} \cdot 2^{-t} \\
&= \frac{\varepsilon t}{O(\sqrt{t})} \cdot 2^{H(\frac{1-2\varepsilon}{2})t - t} \\
&= \Omega(\varepsilon\sqrt{t}) \cdot 2^{-O(\varepsilon^2 t)} \\
&= \Omega(1),
\end{aligned}
$$

as $t = O(\frac{1}{\varepsilon^2})$. Then R is (δ,ε)-good for y with probability at least $\Omega(1) - 2^{-\Omega(N)} = \Omega(1)$. $\square$

An averaging argument immediately gives the following.

Corollary 2. *Suppose $N = \omega(\frac{1}{\delta^2}\log\frac{1}{\varepsilon})$. Then there exist a set $R \subseteq \{0,1\}^N$ with $|R| = \Omega(\frac{1}{\varepsilon^2})$ and a set $A \subseteq [M]$ with $|A| = \Omega(M)$ such that for any $y \in A$, R is (δ,ε)-good for y.*

Let us fix the sets R and A guaranteed by the corollary above. Next, we want to show that many x's from R satisfy the property that the codeword $C(x)$ has enough agreement with the vector b on those dimensions in A.

Lemma 13. *There exists $R' \subseteq R$ with $|R'| = \Omega(\frac{1}{\varepsilon})$ such that for any $x \in R'$, $\Pr_{y\in A}[C(x)_y \neq b_y] < \frac{1-\varepsilon/2}{2}$.*

Proof. For any $y \in A$, R is (δ, ε)-good for y, so

$$\mathop{\mathrm{E}}_{x\in R}\left[\Pr_{y\in A}\left[C(x)_y \neq b_y\right]\right] = \mathop{\mathrm{E}}_{y\in A}\left[\Pr_{x\in R}\left[C(x)_y \neq b_y\right]\right] \leq \frac{1-\varepsilon}{2}.$$

By Markov's inequality,

$$\Pr_{x\in R}\left[\Pr_{y\in A}\left[C(x)_y \neq b_y\right] > \frac{1-\varepsilon/2}{2}\right] < \frac{\frac{1-\varepsilon}{2}}{\frac{1-\varepsilon/2}{2}} \leq 1 - \frac{\varepsilon}{2}.$$

Thus, there exists $R' \subseteq R$ of size $\frac{\varepsilon}{2}|R| = \Omega(\frac{1}{\varepsilon})$ such that for any $x \in R'$, $\Pr_{y\in A}\left[C(x)_y \neq b_y\right] \leq \frac{1-\varepsilon/2}{2}$. □

We let the vector z inherit from the vector b those bits indexed by A, and it remains to set the values for the remaining bits. It is easy to show that there exist $v \in \{0,1\}^M$ (in fact, v can be either 0^M or 1^M) and $S \subseteq R'$ with $|S| \geq \frac{1}{2}|R'|$ such that for any $x \in S$, $\Pr_{y\notin A}\left[C(x)_y \neq v_y\right] \leq \frac{1}{2}$. So we just define $z \in \{0,1\}^M$ as $z_y = b_y$ if $y \in A$ and $z_y = v_y$ otherwise. Then, for any $x \in S$,

$$\begin{aligned}\triangle(C(x), z) &= \Pr_{y\in[M]}[y \in A] \cdot \Pr_{y\in A}\left[C(x)_y \neq b_y\right] + \Pr_{y\in[M]}[y \notin A] \cdot \Pr_{y\notin A}\left[C(x)_y \neq v_y\right] \\ &< \frac{|A|}{M} \cdot \frac{1-\varepsilon/2}{2} + \frac{M-|A|}{M} \cdot \frac{1}{2} \\ &= \frac{1}{2}\left(1 - \frac{|A|(\varepsilon/2)}{M}\right) \\ &\leq \frac{1-\varepsilon/c}{2},\end{aligned}$$

for some constant c.

Furthermore, as $S \subseteq R$ and R is δ-good, any ball in $\mathrm{BALL}(\frac{1-\delta}{2}, N)$ contains only $O(1/\delta^2)$ elements of S. Thus, S must need $\frac{|S|}{O(1/\delta^2)} = \Omega(\frac{\delta^2}{\varepsilon})$ such balls to cover with. Replace the parameter ε/c by ε, and we have the following.

Lemma 14. *Suppose $\varepsilon < \frac{1}{c}$ for some suitable constant c, and suppose $N = \omega(\frac{1}{\delta^2}\log\frac{1}{\varepsilon})$. Then any $(\frac{1-\delta}{2}, \frac{1-\varepsilon}{2}, L)$-list code must have $L = \Omega(\frac{\delta^2}{\varepsilon})$.*

This, combined with Lemma 1, implies the following.

Theorem 5. *Suppose $\varepsilon < \frac{1}{c}$ for some suitable constant c, and suppose $2^n = \omega(\frac{1}{\delta^2}\log\frac{1}{\varepsilon})$. Then any black-box $(n, \frac{1-\delta}{2}, \frac{1-\varepsilon}{2}, \ell)$ hardness amplification must have $\ell = \Omega(\log\frac{\delta^2}{\varepsilon})$.*

Thus, any such hardness amplification, even without any complexity constraint, must be inherently non-uniform, with $\ell \geq 1$ when $\varepsilon \leq c'\delta^2$ for some constant c', or with $\ell = \Omega(k\log\frac{1}{\delta})$ when $\varepsilon = \delta^k$.

6 Impossibility Results on Constructing PRG

In this section we modify the methods developed in previous sections to prove impossibility results for black-box constructions of pseudo-random generators from hard functions.

6.1 Impossibility of Constructing PRG by Constant-Depth Circuits

Consider an oracle algorithm $G^{(\cdot)} : \{0,1\}^r \to \{0,1\}^M$ which realizes a black-box $(n, \frac{1-\delta}{2}, \frac{\delta^k}{4}, \ell)$-PRG. We write $G \in \mathsf{AC}(d,s)$ to mean that for each $u \in \{0,1\}^r$ and each $i \in [M]$, the function from x to $G^x(u)_i$ is computed by an $\mathsf{AC}(d,s)$ circuit. Assume all the convention in Section 3. Now we call two generators G^x and G^y *close* if $\mathrm{E}_u[\triangle(G^x(u), G^y(u))] < \frac{1-\delta^k}{2}$.

Lemma 15. *For some constant c, for any $t, d \in \mathbb{N}$ with $\alpha^t \leq 1 - 2^{-ct^{1/d}}$, and for any $G \in \mathsf{AC}(d, 2^{ct^{1/d}})$, $\Pr_{x,\tilde{x}}[G^x$ is close to $G^{\tilde{x}}] \geq \alpha^{2t} - \delta^k$.*

Proof. As in Lemma 4, $\mathrm{E}_{x,\tilde{x},u}[\triangle(G^x(u), G^{\tilde{x}}(u))] \leq \frac{1-\alpha^{2t}}{2}$, and $\Pr_{x,\tilde{x}}[G^x$ is not close to $G^{\tilde{x}}] \leq \frac{1-\alpha^{2t}}{1-\delta^k}$ by Markov's inequality. Thus $\Pr_{x,\tilde{x}}[G^x$ is close to $G^{\tilde{x}}] \geq 1 - \frac{1-\alpha^{2t}}{1-\delta^k} \geq \alpha^{2t} - \delta^k$. □

Lemma 16. *For any black-box $(n, \frac{1-\delta}{2}, \frac{\delta^k}{4}, \ell)$-PRG G, $\Pr_{x,\tilde{x}}[G^x$ is close to $G^{\tilde{x}}] \leq 2^\ell \cdot 2^{-\Omega(\delta^2 N)}$, if $\delta^k \geq 2^{2+r-c\delta^{2k}M}$ for some large enough constant c.*

Proof. Consider any fixed $x \in \{0,1\}^N$. Now it suffices to bound the number of y's such that G^y is close to G^x. Note that for any such y, $\mathrm{E}_u[\triangle(G^x(u), G^y(u))] < \frac{1-\delta^k}{2}$, so by Markov's inequality, $\Pr_u[\triangle(G^x(u), G^y(u)) \geq \frac{1-\delta^k/2}{2}] < 1 - \frac{\delta^k}{2}$.

Define the distinguisher $D^x : \{0,1\}^M \to \{0,1\}$ as $D^x(z) = 1$ if and only if $\triangle(G^x(u), z) < \frac{1-\delta^k/2}{2}$ for some $u \in \{0,1\}^r$. Clearly, $\Pr_{z \in U_M}[D^x(z) = 1] \leq 2^r \cdot 2^{-\Omega(\delta^{2k})M}$. On the other hand, for any y such that G^y is close to G^x, $\Pr_u[D^x(G^y(u)) = 1] \geq \Pr_u[\triangle(G^x(u), G^y(u)) < \frac{1-\delta^k/2}{2}] > \frac{\delta^k}{2}$. So for any such y,

$$\left| \Pr_{u \in U_r}[D^x(G^y(u)) = 1] - \Pr_{z \in U_M}[D^x(z) = 1] \right| > \frac{\delta^k}{2} - 2^{r-\Omega(\delta^{2k}M)} \geq \frac{\delta^k}{4}.$$

Because G is a black-box $(n, \frac{1-\delta}{2}, \frac{\delta^k}{4}, \ell)$-PRG, there are only $2^{\ell + H(\frac{1-\delta}{2})N}$ such y's. Then the rest of the proof follows that of Lemma 5. □

Choose $t = \Theta(k)$ such that $\alpha^{2t} - \delta^k = \delta^{O(k)}$, and we have the following.

Lemma 17. *For some constants c_0, c_1, for any $\delta \in (0,1)$, and for any $d, k \in \mathbb{N}$ with $2^{2+r-c_0\delta^{2k}M} \leq \delta^k \leq 1 - 2^{-c_1 k^{1/d}}$, if $G^{(\cdot)} : \{0,1\}^r \to \{0,1\}^M \in \mathsf{AC}(d, 2^{c_1 k^{1/d}})$ realizes a black-box $(n, \frac{1-\delta}{2}, \frac{\delta^k}{4}, \ell)$-PRG, then $2^\ell = 2^{\Omega(\delta^2 N)}\delta^{O(k)}$.*

Theorem 6. *Suppose $2^{-c_0 n} \leq \delta < 1$ and $2^{-c_1 r} \leq \delta^k \leq 1 - 2^{-c_2 k^{1/d}}$, for some suitable constants c_0, c_1, c_2. Then any black-box $(n, \frac{1-\delta}{2}, \frac{\delta^k}{4}, \ell)$-PRG $G^{(\cdot)} : \{0,1\}^r \to \{0,1\}^M$ realized in $\mathsf{AC}(d, 2^{c_2 k^{1/d}})$ must have $\ell = 2^{\Omega(n)}$ or $M = O(\frac{r}{\delta^{2k}})$.*

The theorem says that when realizing a black-box PRG using such an AC circuit, either a large amount of non-uniformity must be introduced or the output of the generator cannot be too long.

Corollary 3. *Under the same condition as in Theorem 6, no black-box $(n, \frac{1-\delta}{2}, \frac{\delta^k}{4}, 2^{o(n)})$-PRG $G^{(\cdot)} : \{0,1\}^r \to \{0,1\}^M$ with $M = \omega(\frac{r}{\delta^{2k}})$ can be realized in $\mathsf{ATIME}(O(1), k^{o(1)})$. In particular, no such PRG is possible in PH when $k = n^{\omega(1)}$ or in $\mathsf{ATIME}(O(1), 2^{o(n)})$ when $k = 2^{\Omega(n)}$.*

6.2 Impossibility of Constructing PRG by Nondeterministic Circuits

The previous argument can also be applied to NAC circuits to get the following.

Theorem 7. *Suppose $2^{-c_0 n} \leq \delta < 1$ and $2^{-c_1 r} \leq \delta^k \leq 1 - k^{-c_2}$, for some suitable constants c_0, c_1, c_2. Then any black-box $(n, \frac{1-\delta}{2}, \frac{\delta^k}{4}, \ell)$-PRG $G^{(\cdot)} : \{0,1\}^r \to \{0,1\}^M$ in $\mathsf{NAC}(\frac{k}{c_3 \log k})$ or $\mathsf{NTIME}(\frac{k}{c_3 \log^2 k})$, for some constant c_3, must have $\ell = 2^{\Omega(n)}$ or $M = O(\frac{r}{\delta^{2k}})$.*

6.3 Inherent Non-uniformity of Constructing PRG

Consider an arbitrary black-box $(n, \frac{1-\delta}{2}, \frac{\varepsilon}{c}, \ell)$-PRG $G^{(\cdot)} : \{0,1\}^r \to \{0,1\}^M$, for some suitable constant c. Assume all the convention in Section 5.1 unless stated otherwise. Now we would like to find a distinguisher $D : \{0,1\}^M \to \{0,1\}$ and a large enough set $S \subseteq \{0,1\}^N$ such that:

- for every $x \in S$, $|\Pr_u[D(G^x(u)) = 1] - \Pr_z[D(z) = 1]| > \frac{\varepsilon}{c}$, and
- S needs many balls in $\textsc{Ball}(\frac{1-\delta}{2}, N)$ to cover with.

As in Section 5.1, we sample $x^1, \ldots, x^t$ from $\{0,1\}^N$ to form the set R, for some $t = O(\frac{1}{\varepsilon^2})$.

Definition 10. *For $(y,u) \in [M] \times \{0,1\}^r$, let $B(u)_y$ be the bit with $\Pr_{x \in \{0,1\}^N}[G^x(u)_y \neq B(u)_y] \leq \frac{1}{2}$. Call R (δ,ε)-good for (y,u) if R is δ-good and $\Pr_{x \in R}[G^x(u)_y \neq B(u)_y] \leq \frac{1-\varepsilon}{2}$.*

Similar to Lemma 12, we have the following.

Lemma 18. *Suppose $N = \omega(\frac{1}{\delta^2} \log \frac{1}{\varepsilon})$. Then for any $(y,u) \in [M] \times \{0,1\}^r$, R is (δ,ε)-good for (y,u) with probability $\Omega(1)$.*

An averaging argument immediately gives the following.

Corollary 4. *Suppose $N = \omega(\frac{1}{\delta^2} \log \frac{1}{\varepsilon})$. Then there exist a set $R \subseteq \{0,1\}^N$ with $|R| = \Omega(\frac{1}{\varepsilon^2})$ and a set $A \subseteq [M] \times \{0,1\}^r$ with $|A| = \Omega(M2^r)$ such that for any $(y,u) \in A$, R is (δ,ε)-good for (y,u).*

Let us fix the sets R and A guaranteed by the corollary above. As in Lemma 13, we have the following.

Lemma 19. *There exists a subset $R' \subseteq R$ with $|R'| = \Omega(\frac{1}{\varepsilon})$ such that for any $x \in R'$, $\Pr_{(y,u) \in A}[G^x(u)_y \neq B(u)_y] < \frac{1-\varepsilon/2}{2}$.*

It is easy to show that there exists $V : \{0,1\}^r \to \{0,1\}^M$ (in fact, for any u, $V(u)$ can be either 0^M or 1^M) and $S \subseteq R'$ with $|S| \geq \frac{1}{2}|R'|$ such that for any $x \in S$, $\Pr_{(y,u) \notin A}[G^x(u)_y \neq V(u)_y] \leq \frac{1}{2}$. Now, for any $u \in \{0,1\}^r$, define $Z(u) \in \{0,1\}^M$ as $Z(u)_y = B(u)_y$ if $(y,u) \in A$ and $Z(u)_y = V(u)_y$ otherwise. Then as in Section 5.1, one can show that for any $x \in S$,

$$\mathop{\mathrm{E}}_u[\triangle(G^x(u), Z(u))] < \frac{1 - \Omega(\varepsilon)}{2}.$$

Thus, by Markov's inequality, there exists some constant c such that

$$\Pr_u\left[\triangle(G^x(u), Z(u)) < \frac{1 - 2\varepsilon/c}{2}\right] > \frac{2\varepsilon}{c}.$$

Define the distinguisher $D : \{0,1\}^M \to \{0,1\}$ as $D(z) = 1$ if and only if $\triangle(z, Z(u)) < \frac{1-2\varepsilon/c}{2}$. Then we have $\Pr_u[D(G^x(u)) = 1] > \frac{2\varepsilon}{c}$. On the other hand, a union bound gives $\Pr_z[D(z) = 1] \le 2^{r+(H(\frac{1-c\varepsilon}{2})-1)M} = 2^{r-\Omega(\varepsilon^2)M}$. This is at most $\frac{\varepsilon}{c}$ when $M = \omega(\frac{r+\log(1/\varepsilon)}{\varepsilon^2}) = \omega(\frac{r}{\varepsilon^2})$.[5] In this case, for any $x \in S$,

$$\left|\Pr_u[D(G^x(u)) = 1] - \Pr_z[D(z) = 1]\right| > \frac{\varepsilon}{c}.$$

Furthermore, as $S \subseteq R$ and R is δ-good, any ball in $\textsc{Ball}(\frac{1-\delta}{2}, N)$ contains only $O(1/\delta^2)$ elements of S. Thus, S must need $\frac{|S|}{O(1/\delta^2)} = \Omega(\frac{\delta^2}{\varepsilon})$ such balls to cover with. Replacing $\frac{\varepsilon}{c}$ by ε, we have the following.

Theorem 8. *Suppose $\varepsilon < \frac{1}{c}$ for some suitable constant c, and suppose $N = \omega(\frac{1}{\delta^2}\log\frac{1}{\varepsilon})$. Then any black-box $(n, \frac{1-\delta}{2}, \varepsilon, \ell)$ pseudo-random generator $G^{(\cdot)} : \{0,1\}^r \to \{0,1\}^M$ with $M = \omega(\frac{r}{\varepsilon^2})$ must have $\ell = \Omega(\log\frac{\delta^2}{\varepsilon})$.*

Therefore, any such construction of pseudo-random generators, even without any complexity constraint, must be inherently non-uniform, with $\ell \ge 1$ when $\varepsilon \le c'\delta^2$ for some constant c', or with $\ell = \Omega(k\log\frac{1}{\delta})$ when $\varepsilon = \delta^k$.

Acknowledgment

The authors would like to thank Emanuele Viola for many helpful discussions and anonymous referees for their useful comments.

References

[1] László Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3(4), pages 307–318, 1993.

[2] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo random bits. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 112–117, 1982.

[3] Andrej Bogdanov and Luca Trevisan. On worst-case to average-case reductions for NP problems. In *44th Annual Symposium on Foundations of Computer Science*, Cambridge, Massachusetts, pages 11-14, October 2003.

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

[5] Oded Goldreich, Noam Nisan, and Avi Wigderson. On Yao's XOR lemma. Technical Report TR95–050, Electronic Colloquium on Computational Complexity, 1995.

[5]For a pseudo-random generator $G : \{0,1\}^r \to \{0,1\}^M$, one can only expect $\varepsilon \ge 2^{-r} - 2^{-M} = 2^{-O(r)}$, or equivalently $\log(1/\varepsilon) = O(r)$, because this can be achieved by a simple distinguisher T defined as $T(z) = 1$ if and only if $z = G(0^r)$.

[6] Johan Håstad. *Computational limitations for small depth circuits.* PhD thesis, MIT Press, 1986.

[7] Alexander Healy, Salil P. Vadhan, and Emanuele Viola. Using nondeterminism to amplify hardness. In *Proceedings of the 36th ACM Symposium on Theory of Computing*, pages 192–201, 2004.

[8] Russel Impagliazzo. Hard-core distributions for somewhat hard problems. In *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science*, pages 538–545, 1995.

[9] Russell Impagliazzo, Ronen Shaltiel, and Avi Wigderson. Extractors and pseudo-random generators with optimal seed length. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 1–10, 2000.

[10] Russel Impagliazzo and Avi Wigderson. P=BPP if E requires exponential circuits: derandomizing the XOR lemma. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 220–229, 1997.

[11] Russel Impagliazzo and Avi Wigderson. Randomness vs. time: de-randomization under a uniform assumption. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 734–743, 1998.

[12] Adam Klivans and Dieter van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM Journal on Computing*, 31(5), pages 1501–1526, 2002.

[13] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, Fourier transform, and learnability. *Journal of the ACM*, 40(3), pages 607–620, 1993.

[14] Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of Computing System Science*, 49(2), pages 149–167, 1994.

[15] Ryan O'Donnell. Hardness amplification within NP. In *Proceedings of the 34th ACM Symposium on Theory of Computing*, pages 751–760, 2002.

[16] M. Plotkin. Binary codes with specified minimum distance. *IEEE Transactions on Information Theory*, 6, pages 445–450, 1960.

[17] Ronen Shaltiel and Christopher Umans. Simple extractors for all min-entropies and a new pseudo-random generator. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 648–657, 2001.

[18] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62(2), pages 236–266, 2001.

[19] Luca Trevisan. List decoding using the XOR lemma. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 126–135, 2003.

[20] Luca Trevisan and Salil Vadhan. Pseudorandomness and average-case complexity via uniform reductions In *Proceedings of the 17th Computational Complexity Conference*, pages 129–138, IEEE, 2002.

[21] Christopher Umans. Pseudo-random generators for all hardnesses. *Journal of Computer and System Sciences.* 67(2), pages 419–440, 2003.

[22] Emanuele Viola. The Complexity of Constructing Pseudorandom Generators from Hard Functions. To appear in *Computational Complexity.*

[23] Emanuele Viola. On parallel pseudorndom generators. To appear in *Proceedings of the 20th Computational Complexity Conference*, IEEE, 2005.

[24] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 80–91, 1982.