On the Complexity of Hardness Amplification

Chi-Jen Lu, Shi-Chun Tsai, *Member, IEEE*, and Hsin-Lung Wu

***Abstract*—For $\delta \in (0,1)$ and $k, n \in \mathbb{N}$, we study the task of transforming a hard function $f : \{0,1\}^n \to \{0,1\}$, 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. First, we show that such hardness amplification, when carried out in some black-box way, must require a high complexity. In particular, it cannot be realized 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) for any $\delta \in (0,1)$. This extends the result of Viola, which only works when $(1-\delta)/2$ is small enough. Furthermore, we show that even without any restriction on the complexity of the amplification procedure, such a black-box hardness amplification must be inherently nonuniform in the following sense. To guarantee the hardness of the resulting function f', even against uniform machines, one has to start with a function f, which is hard against nonuniform algorithms with $\Omega(k\log(1/\delta))$ bits of advice. This extends the result of Trevisan and Vadhan, which only addresses the case with $(1-\delta)/2 = 2^{-n}$. Finally, we derive similar lower bounds for any black-box construction of a pseudorandom generator (PRG) from a hard function. To prove our results, we link the task of hardness amplifications and PRG constructions, respectively, to some type of error-reduction codes, and then we establish lower bounds for such codes, which we hope could find interest in both coding theory and complexity theory.**

***Index Terms*—Computational complexity, hardness amplification, list-decodable code, pseudorandom generator.**

I. Introduction

A. 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 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 in the work of Blum and Micali [2] and Yao [29], who showed how to obtain a PRG from a one-way function. Then, Nisan and Wigderson [18] showed that a PRG can be constructed from a Boolean function, which is hard in average case, and this initiated a series of works. To get a stronger result, one would like to weaken the hardness assumption, and [18], [1], [10] showed that, in fact, one can start from a (slightly) hard Boolean function and transform it into a much harder one, before using it to build a PRG. Finally, Impagliazzo and Wigderson [14] proved that one can transform 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 BPP = P under the assumption that some function in E cannot be computed by a circuit of subexponential size. Simpler proofs and better tradeoffs have been obtained since then [23], [13], [22], [26].

Note that hardness amplification is the major step in derandomizing BPP in the research discussed above, because 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 β–hard (or has hardness β) against circuits of size s if any such circuit attempting to compute f must make errors on at least β fraction of the input. The error bound β 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 $f : \{0,1\}^n \to \{0,1\}$ that is β–hard against circuits of size $s(n)$ into a function $f' : \{0,1\}^m \to \{0,1\}$ that is β'–hard against circuits of size $s'(m)$, with $\beta' > \beta$ 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 $m = \mathrm{poly}(n)$ or even $m = O(n)$, 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 ($\beta = 2^{-n}$) to average case hardness ($\beta' = 1/2 - 2^{-\Omega(m)}$) need exponential time [1], [14], [23] (or slightly better, in linear space [16] or $\oplus\mathsf{ATIME}(O(1), n)$ [27]). As a result, such a hardness amplification is only known for functions in high complexity classes. Then, a natural question is as follows: Can it be done for functions in lower complexity classes? For

Manuscript received March 7, 2007; revised July 9, 2008. Current version published September 17, 2008. The work of C.-J. Lu was supported in part by the National Science Council of Taiwan under Contract NSC93-2213-E-001-004. The work of S.-C. Tsai was supported in part by the National Science Council of Taiwan under Contract NSC-93-2213-E-009-035. The work of H.-L. Wu was supported in part by the National Science Council of Taiwan under Contract NSC-97-2218-E-305-001-MY2. The material in this paper was presented at the 20th Annual Computational Complexity Conference, San Jose, CA, June 2005.

C.-J. Lu is with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan (e-mail: cjlu@iis.sinica.edu.tw).

S.-C. Tsai is with the Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu 30050, Taiwan (e-mail: sctsai@csie.nctu.edu.tw).

H.-L. Wu is with the Department of Computer Science and Information Engineering, National Taipei University, Taipei, Taiwan (e-mail: hsinlung@mail.ntpu.edu.tw).

Communicated by A. Canteaut, Associate Editor for Complexity and Cryptography.

Digital Object Identifier 10.1109/TIT.2008.928988

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 $\beta = 1/\mathrm{poly}(n)$] is this known to be possible [19], [9].

The second issue is that hardness amplification typically involves nonuniformity in the sense that hardness is usually measured against *nonuniform* circuits. In fact, one usually needs to start from a function that is hard against nonuniform circuits, even if one only wants to produce a function that is hard against uniform Turing machines. This is why most results on hardness amplification are based on nonuniform assumptions.

B. 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 procedure AMP such that $f' = \mathrm{AMP}^f$, 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 procedure DEC, such that if some algorithm A disagrees with f' on less than β' fraction of the input, then DEC using A as an oracle disagrees with f on less than β fraction of the input. Again, DEC only uses A as an oracle and does not depend on the internal structure of A. We call AMP the *encoding* function and DEC the *decoding* function. In fact, almost all previous constructions of hardness amplification are done in such a black-box way, so it is nice to establish impossibility results for this type of approaches.

C. Previous Lower Bound Results

Viola [27] 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 function AMP cannot be realized in 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 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 [25] 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 such amplification from worst case hardness to hardness $(1-\varepsilon)/2$, the decoding function 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 nonuniform setting, except [15] and [25], 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* function is dropped. They showed that the decoding function cannot be computed nonadaptively in polynomial time unless PH collapses. Viola, in another recent paper [28], considered hardness amplification in which the black-box requirement on the *decoding* function is dropped. He showed that if the encoding function 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.

D. Our Results

Previous lower bound results only address hardness in a specific range. However, whether one can amplify hardness beyond this range is also a natural and interesting question. For example, it is known that a black-box hardness amplification from hardness $1/\mathrm{poly}(n)$ to average case hardness can be realized in polynomial time [29], [5], [10], [14]. Can such a hardness amplification be realized in a lower complexity class, such as AC^0? Can it start from hardness below $1/\mathrm{poly}(n)$ and still be realized in polynomial time? Can it be done in a uniform way (with a uniform decoding function)? In general, how does the quality of a hardness amplification (the amount of hardness increased) determine its inherent complexity or nonuniformity? All these questions will be addressed in this paper. We generalize previous results [27], [25] and consider hardness amplification in a much broader spectrum: from hardness $(1-\delta)/2$ to hardness $(1-\delta^k)/2$, for general $\delta \in (0,1)$ and $k \in \mathbb{N}$. We should stress that our work is mainly inspired by the seminal works of [27] and [25] and we follow closely their approaches; our main effort is to generalize their ideas and to get a more precise picture.

Following [28], we consider a more restricted model called *parallel* black-box hardness amplification, in which oracle queries by the encoding function are done in a nonadaptive way. More precisely, we say that a circuit class CKT realizes a parallel black-box hardness amplification if its encoding function AMP can be implemented in the following way. Given any input x, it first generates a circuit $T_x \in \mathsf{CKT}$ together with t query inputs $q_{x,1}, \ldots, q_{x,t}$, then queries f at those t inputs, and finally computes $T_x(f(q_{x,1}), \ldots, f(q_{x,t}))$ as its output. Note that here T_x and $q_{x,1}, \ldots, q_{x,t}$ only depend on x but not f. Although this is a more restricted model, almost all previous constructions of hardness amplification can be done in this way, so it would be nice to know its limitation. Furthermore, through a standard simulation [4], [8], negative results in this model can, in fact, be translated to those in the general black-box model.

Our first result addresses both the complexity issue and the nonuniformity issue in the same framework, showing how complexity constraints on the encoding function result in the inherent nonuniformity of the decoding function. Formally, we prove that if such a parallel black-box hardness amplification, from hardness $(1-\delta)/2$ to hardness $(1-\delta^k)/2$, is realized by circuits of depth d and size $2^{o(k^{1/d})}$, then the decoding function DEC must need an advice of length $2^{\Omega(n)}$. Translating this

to the general model, we obtain the same advice lower bound when such a (general) black-box hardness amplification is realized in $\mathsf{ATIME}(O(1), k^{o(1)})$. This implies that no such hardness amplification is possible if the hardness is measured against circuits of size $2^{o(n)}$.

Our lower bound is almost tight because the well-known XOR lemma [29], [5] gives a way to realize a parallel black-box hardness amplification by circuits of depth $O(d)$ and size $2^{O(k^{1/d})}$, with DEC using an advice of length $\mathrm{poly}(n/\delta^k)$. Note that Viola's result in [27] is a special case of ours, because he only addressed explicitly the specific case with $(1-\delta)/2 = 2^{-n}$ and $(1-\delta^k)/2 = 1/\mathrm{poly}(n)$ (or equivalently, $\delta = 1 - 2^{-n+1}$ and $k = 2^{\Omega(n)}$). Although it seems that his technique can be extended to show lower bounds when $(1-\delta)/2$ is small enough, but beyond that, say with $(1-\delta)/2 = \Omega(1)$, it fails to give a meaningful bound. We can, in fact, cover this case: our result implies that AC^0 circuits cannot realize a parallel black-box hardness amplification, say, from hardness $1/3$ to hardness $(1 - 2^{-\Omega(n)})/2$. On the other hand, our result when restricted to worst case to average case hardness amplification is incomparable to those of [3] and [28].[1] Finally, two interesting facts follow from our result. First, it is impossible to produce in a black-box way a function that is $(1-\delta^k)/2$–hard against a uniform low complexity class, say $\mathsf{DTIME}(O(1))$, even if we start from a function that is $(1-\delta)/2$–hard against a uniform but arbitrarily high complexity class equipped with an advice of length $2^{o(n)}$, say $\mathsf{DTIME}(2^{2^n})/2^{o(n)}$. On the other hand, it is easy to show that hard functions against $\mathsf{DTIME}(O(1))$ do exist.[2] This demonstrates one severe weakness of black-box hardness amplifications. Second, when amplifying hardness from $(1-\delta)/2$ to $(1-\delta^k)/2$, the complexity of such amplification is determined mainly by the parameter k; a larger value of k results in a higher complexity requirement, for typical values of δ. Thus, to determine the complexity needed for a hardness amplification process, one should express the initial and final hardness in the forms of $(1-\delta)/2$ and $(1-\delta^k)/2$, respectively.[3] This point was not clear from previous works.

Note that our first result becomes meaningless for $d = \Omega(\log k)$ as the circuit size becomes $2^{o(k^{1/d})} = O(1)$. Our second result takes care of this: we show that if a parallel black-box hardness amplification, from hardness $(1-\delta)/2$ to hardness $(1-\delta^k)/2$, is realized by nondeterministic circuits of size $o(k/\log k)$, even with arbitrary depth, then the decoding function DEC must need an advice of length $2^{\Omega(n)}$. For example, to amplify hardness from $\Omega(1)$ to $(1 - 2^{-\Omega(n)})/2$, our second result implies that it cannot be realized by nondeterministic circuits of size $o(n/\log n)$ in a parallel black-box way.

[1]In [3], the complexity lower bound is given on the decoding function instead, under the unproven (though widely believed) assumption that PH does not collapse. In [28], 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 function can be computed in PH, a hard function in PH exists unconditionally.

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

[3]Note that for a function with hardness $(1-\delta)/2$ against small circuits, the quantity δ is the maximum correlation of the function with such circuits. Therefore, our result shows that to reduce such correlation from δ to δ^k, the complexity is mainly determined by k.

Our third result shows that even without any complexity constraint on the encoding or decoding function, amplification between certain range of hardness is still inherently nonuniform. For the special case of amplifying hardness beyond 1/4, the need of nonuniformity can be shown using the Plotkin bound [21] from coding theory. We consider hardness amplification in a general range and obtain a quantitative bound on the amount of nonuniformity. More precisely, we show that to amplify hardness from $(1-\delta)/2$ to $(1-\varepsilon)/2$, the decoding function 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 nonuniform. Our result generalizes that of Trevisan and Vadhan [25].

Finally, we derive similar lower bounds on black-box constructions of PRG from hard functions.

E. 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 [25] and Viola [27]. A similar observation was also made by Trevisan [24]. Formally, a black-box amplification from hardness $(1-\delta)/2$ to hardness $(1-\varepsilon)/2$ induces a code with the following list decoding property, which is also known as approximate list decoding [11]. 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 results on such codes immediately give results on corresponding hardness amplifications.

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 that 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 cannot be used to compute such a code and the corresponding hardness amplification. This basically follows Viola's idea in [27], but because we consider hardness amplification in a much broader spectrum, a more involved analysis is required. For example, because Viola only considered the case with a small hardness, he only had to deal with noise of a small rate. With such a small noise rate, the output value will only be affected with a small probability, and small loss in his analysis does not matter too much. However, if a large hardness is considered, a high noise rate is needed and then the loss in his analysis will become intolerable, and his bound will become meaningless (see Remark 4 in Section II-D for details). To overcome this problem, we drive another upper bound on noise sensitivity, which works for any noise rate and thus can be used for hardness in a general range.

For the nonuniformity 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.

Finally, to prove lower bounds for black-box constructions of PRG from hard functions, we discover that there is also a connection between the error-reduction codes we just considered and such PRG constructions. Then, the results we obtain for such codes immediately yield results for such PRG constructions. Note that in [27], Viola used a connection between black-box PRG constructions and randomness extractors, and then he proved a separate lower bound for extractors, in addition to that for codes. Our connection, in fact, can be seen as a connection between extractors and codes, and with this connection, we no longer need a separate proof for PRG constructions.

F. Organization of This Paper

First, some preliminaries are given in Section II. Then, in Sections III and IV, we prove the impossibility results of hardness amplification by constant-depth circuits and nondeterministic circuits, respectively. In Section V, we show that hardness amplification, in general, is inherently nonuniform. Finally, we show the impossibility results for black-box PRG constructions from hard functions in Section VI.

II. Preliminaries

For any $n \in \mathbb{N}$, let $[n]$ denote the set $\{1,2,\ldots,n\}$ and let $\mathcal{U}_n$ denote the uniform distribution over the set $\{0,1\}^n$. When we sample from a finite set, the default distribution is the uniform one. For a string z, let z_i denote the ith 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 at most d alternations, and let $\mathsf{ATIME}(t)$ denote $\mathsf{ATIME}(t,t)$. 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. More information about complexity classes can be found in standard textbooks, such as [20]. The circuits we consider here consist of AND/OR/NOT gates, allowing unbounded fan-in for AND/OR gates. The size of a circuit is the number of noninput 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 $\mathsf{AC}(d,s)$ 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 witness 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.

A. Black-Box Hardness Amplification and Pseudorandom Generators

Informally speaking, a function is hard if any algorithm without enough complexity must make some mistakes. Formally, we define the hardness of a function as follows.

Definition 3: We say that a function $f:\{0,1\}^n \to \{0,1\}$ has hardness β against circuits of size s if for any circuit $C:\{0,1\}^n \to \{0,1\}$ of size s

$$\Pr_{x\in U_n}[f(x)\neq C(x)] \geq \beta.$$

Note that we use the error bound β 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 β into a function $f':\{0,1\}^m \to \{0,1\}$ with a larger hardness β'. We will focus on a special type of hardness amplification called black-box hardness amplification, defined next, which consists of two oracle procedures $\textsc{Amp}$ and $\textsc{Dec}$. We allow $\textsc{Dec}$ to be a nonuniform oracle Turing machine, and we write $\textsc{Dec}^{A,\nu}$ to denote $\textsc{Dec}$ taking an oracle A and an advice string ν.

Definition 4: A black-box (n,β,β',ℓ) hardness amplification consists of an oracle procedure $\textsc{Amp}^{(\cdot)}:\{0,1\}^m \to \{0,1\}$ (called encoding function) and a nonuniform oracle Turing machine $\textsc{Dec}^{(\cdot)}:\{0,1\}^n \to \{0,1\}$ (called decoding function) with the following property. For any $f:\{0,1\}^n \to \{0,1\}$, if a function $A:\{0,1\}^m \to \{0,1\}$ satisfies

$$\Pr_{z\in U_m}[A(z)\neq \textsc{Amp}^f(z)] < \beta'$$

then there exists an advice string $\nu=\nu(f,A)\in\{0,1\}^\ell$ such that

$$\Pr_{x\in U_n}[\textsc{Dec}^{A,\nu}(x)\neq f(x)] < \beta.$$

For a complexity class C, we say that the black-box hardness amplification can be realized in C if for any oracle f, the procedure $\textsc{Amp}^f$ can be computed in C^f.

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 $\textsc{Dec}$. This freedom makes our impossibility

results stronger. The parameter ℓ characterizes the amount of nonuniformity associated with this process. When $\ell \geq 1$, we say the hardness amplification is nonuniform.

Remark 1: One can also use the notion of "advantage" to characterize the hardness of a Boolean function. We say that any circuit of size s has advantage at most δ for computing f if for any such a circuit C, $\Pr_x[f(x) = C(x)] - \Pr_x[f(x) \neq C(x)] \leq \delta$. Clearly, the advantage δ is related to the hardness β in the form $\beta = \frac{1-\delta}{2}$. We will focus on the task of amplifying hardness from $\frac{1-\delta}{2}$ to $\frac{1-\delta^k}{2}$, or equivalently, reducing the advantage from δ to δ^k. We choose to present our results in terms of hardness instead of advantage for the following two reasons. First, when talking about hardness amplification, it seems more natural and less confusing to use hardness instead of advantage. Second, as we will see, there is some nice connection between hardness amplifications and error-correcting codes, in which hardness of functions corresponds naturally to distance in codes. However, the drawback of using hardness instead of advantage is that our notation sometimes looks more cumbersome.

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

Definition 5: A black-box $(n, \beta, \varepsilon, \ell)$ PRG construction consists of an oracle procedure $\mathrm{G}^{(\cdot)} : \{0,1\}^m \to \{0,1\}^r$ (called encoding function) and a nonuniform oracle Turing machine $\mathrm{DEC}^{(\cdot)} : \{0,1\}^n \to \{0,1\}$ (called decoding function) with the following property. For any $f : \{0,1\}^n \to \{0,1\}$, if a function $D : \{0,1\}^r \to \{0,1\}$ satisfies

$$\left| \Pr_{u \in U_m}[D(\mathrm{G}^f(u)) = 1] - \Pr_{w \in U_r}[D(w) = 1] \right| > \varepsilon$$

then there exists an advice string $\nu = \nu(f, D) \in \{0,1\}^\ell$ such that

$$\Pr_{x \in U_n}[\mathrm{DEC}^{D,\nu}(x) \neq f(x)] < \beta.$$

For a complexity class C, we say that the black-box PRG construction can be realized in C if for any oracle f, the procedure G^f can be computed in C^f.

Remark 2: When talking about a black-box hardness amplification or PRG construction, we usually mean a sequence of them, parameterized by the parameter $n \in \mathbb{N}$. Other parameters such as $m, \beta, \beta', \ell, r, \varepsilon, k$ are, in fact, allowed to be functions of n.

In this general model of black-box hardness amplification or PRG construction, we do not put any restriction on how the oracle f is queried by the encoding function (AMP or G). On the other hand, we will also consider the following more restricted model, first introduced in [28], in which the oracle f can only be queried in a nonadaptive way. We call such model a *parallel* black-box hardness amplification or PRG construction. More precisely, we define the following.

Definition 6: Let CKT be a class of circuits, such as $\mathsf{AC}(d, s)$ or $\mathsf{NAC}(s)$. We say that CKT realizes a parallel black-box hardness amplification, if we have a black-box hardness amplification in which the encoding function $\mathrm{AMP}^{(\cdot)}$ can be implemented in the following way. Given any oracle $f : \{0,1\}^n \to \{0,1\}$ and any input $x \in \{0,1\}^m$, it first generates a circuit $T_x \in \mathsf{CKT}$ together with t query inputs $q_{x,1}, \ldots, q_{x,t} \in \{0,1\}^n$, then queries f at those t inputs, and finally outputs $T_x(f(q_{x,1}), \ldots, f(q_{x,t}))$. The case of parallel black-box PRG construction is defined similarly.

Note that T_x and $q_{x,1}, \ldots, q_{x,t}$ are produced before the oracle f is actually queried, so they depend on x but not on the oracle f. This restriction makes it easier to obtain negative (or lower bound) results in such a parallel model. Nevertheless, the following lemma provides a way to translate such results to those in the general black-box model.

Lemma 1: If a black-box (n, β, β', ℓ) hardness amplification (or PRG construction) can be realized in $\mathsf{ATIME}(d, t)$, then a parallel black-box (n, β, β', ℓ) hardness amplification (or PRG construction) can be realized in $\mathsf{AC}(d + 2, 2^{O(t)})$.

Proof: Consider any black-box hardness amplification (the case of PRG construction is similar) with the encoding function AMP such that for any oracle f, AMP^f belongs to $\mathsf{ATIME}^f(d, t)$. It is known from [4] and [8] that by adding two alternations (an existential one for guessing the oracle answers along a computational branch and a universal one for verifying the guessed answers), one can transform AMP into another procedure AMP' that only queries f once in each branch of its computation. Then, by a standard simulation of alternating Turing machines by circuits [4], [8], we know that for any input x, the value of $\mathrm{AMP}'^f(x)$ can be computed by a circuit in $\mathsf{AC}(d + 2, 2^{O(t)})$ with the answers to the corresponding oracle queries given as part of the input. Note that the circuit and the oracle queries depend only on the input x but not the oracle f. Thus, we have a parallel black-box hardness amplification realized in $\mathsf{AC}(d + 2, 2^{O(t)})$. □

B. Codes and Correspondence to Hardness Amplification

We measure the distance between two strings by their relative Hamming distance.

Definition 7: 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 β in the space $\{0,1\}^N$.

Definition 8: For any $N \in \mathbb{N}$, $\beta \in (0,1)$, and $x \in \{0,1\}^N$, let $\mathrm{BALL}_x(\beta, N) = \{x' \in \{0,1\}^N : \triangle(x, x') < \beta\}$, which is the open ball in $\{0,1\}^N$ of radius β centered at x. Let $\mathrm{BALL}(\beta, N)$ denote the set consisting of all such balls.

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

Fact 1: The size of any ball in $\mathrm{BALL}(\beta, N)$ is at most $2^{H(\beta)N}$.

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 9: We call $C : \{0,1\}^N \to \{0,1\}^M$ a (β, β', L)-list code if for any $z \in \{0,1\}^M$, there are L balls from $\mathrm{BALL}(\beta, N)$ such that if a codeword $C(x)$ is contained in $\mathrm{BALL}_z(\beta', M)$, then x is contained in one of those L balls.

A (β, β', L)-list code is related to a standard list-decodable code in the way that each ball in $\mathrm{BALL}(\beta', M)$ contains at most $L \cdot 2^{H(\beta)N}$ 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 $\mathrm{AMP}^{(\cdot)} : \{0,1\}^m \to \{0,1\}$, let us define the corresponding code $C : \{0,1\}^N \to \{0,1\}^M$ as $C(f) = \mathrm{AMP}^f$. 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 AMP^f, which is seen as a vector in $\{0,1\}^M$. The following is a simple generalization of an observation by Viola [27].

Lemma 2: Let $\mathrm{AMP}^{(\cdot)} : \{0,1\}^m \to \{0,1\}$ be the encoding function of a black-box (n, β, β', ℓ) hardness amplification. Then, $C : \{0,1\}^N \to \{0,1\}^M$, defined as $C(f) = \mathrm{AMP}^f$, is a $(\beta, \beta', 2^\ell)$-list code.

Proof: Let AMP be the encoding function of a black-box (n, β, β', ℓ) hardness amplification, and let DEC be the corresponding decoding function that 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 $\triangle(A, C(f)) = \Pr_z[A(z) \neq \mathrm{AMP}^f(z)] < \beta'$, by Definition 4, there exists an $\nu \in \{0,1\}^\ell$ such that $\triangle(\mathrm{DEC}^{A,\nu}, f) = \Pr_x[\mathrm{DEC}^{A,\nu}(x) \neq f(x)] < \beta$. That is, if $C(f)$ is in $\mathrm{BALL}_A(\beta', M)$, then f is contained in one of the 2^ℓ balls of radius β centered at $\mathrm{DEC}^{A,\nu}$ for $\nu \in \{0,1\}^\ell$. Therefore, C is a $(\beta, \beta', 2^\ell)$-list code. □

Remark 3: Note that if a circuit class CKT can realize a parallel hardness amplification, then every output bit of the corresponding code C can be computed by a circuit in CKT. This is because for any input $f \in \{0,1\}^N$, the xth output bit of $C(f)$ equals $\mathrm{AMP}^f(x) = T_x(f(q_{x,1}), \ldots, f(q_{x,t}))$, which is computed by some circuit $T_x \in \mathsf{CKT}$ on some t bits of f.

In Section VI, we will show that there also exists a natural correspondence between black-box PRG constructions and such list-decodable codes.

C. Noise Sensitivity

Following [19] and [27], we will 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 well-known 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 3 [17]: 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 4: 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 $t \in [N]$, $\Pr_{x,\tilde{x}}[g(x) \neq g(\tilde{x})] \leq \frac{1}{2}(1 - \alpha^t(1 - \sum_{|J|>t} \hat{g}(J)^2))$.

Proof: We know from [19, Prop. 9] 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. □

Combining Lemmas 3 and 4, we immediately have the following.

Corollary 1: Suppose x and $\tilde{x}$ are sampled as in Lemma 4. Then, for any $g : \{0,1\}^N \to \{0,1\} \in \mathsf{AC}(d, s)$ and for any $t \in [N]$, $\Pr_{x,\tilde{x}}[g(x) \neq g(\tilde{x})] \leq \frac{1}{2}(1 - \alpha^t(1 - s \cdot 2^{-\Omega(t^{1/d})}))$.

Remark 4: In [27], Viola derived a weaker bound $\Pr_{x,\tilde{x}}[g(x) \neq g(\tilde{x})] \leq O(\beta \log^d s)$, with $\beta = \frac{1-\alpha}{2}$, which becomes vacuous when β is not small enough. This prevents him from having a meaningful bound when the hardness is not small enough. The main loss in his derivation comes from his use of the inequality $\frac{1}{2}(1 - \sum_{J \subseteq [N]} \alpha^{|J|} \hat{g}(J)^2) \leq \frac{1}{2}(1 - \sum_{J \subseteq [N]}(1-\alpha)|J|\hat{g}(J)^2)$. Our Lemma 4 uses a different inequality to avoid this problem.

III. Impossibility of Amplification by Small-Depth Circuits

In this section, we will show that any parallel black-box $(n, \frac{1-\delta}{2}, \frac{1-\delta^k}{2}, \ell)$ hardness amplification realized in $\mathsf{AC}(d, s)$ with small d and s must be highly nonuniform. More precisely, we will prove the following.

Theorem 1: There exist constants c_0, c_1, c_2, c_3 such that for any $\delta \in (0,1)$ and any $d, k \in \mathbb{N}$ with $2^{-c_0 n} \leq \delta < 1$ and $2^{-2^{c_1 n}} \leq \delta^k \leq 1 - 2^{-c_2 k^{1/d}}$, any parallel black-box $(n, \frac{1-\delta}{2}, \frac{1-\delta^k}{2}, \ell)$ hardness amplification realized in $\mathsf{AC}(d, 2^{c_3 k^{1/d}})$ must have $\ell = 2^{\Omega(n)}$.

Before giving the proof, let us take a closer look at the theorem itself and discuss some of its consequences. First, note that the conditions on the ranges of δ and δ^k are 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. Also, as discussed in the Introduction, hardness amplifications normally have m close to n (preferably with $m = \mathrm{poly}(n)$), therefore δ^k, which is at least 2^{-m}, would be much larger than $2^{-2^{\Omega(n)}}$.

Although Theorem 1 is on the more restricted parallel model, it in fact implies the following result on the general model of hardness amplification, according to Lemma 1.

Corollary 2: 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 $\mathsf{ATIME}(O(1), k^{o(1)})$.

Note that Viola's result [27] is a special case of ours, with initial hardness $\frac{1-\delta}{2} = 2^{-n}$ (amplifying from worst case hardness). A closer look at his technique shows that it, in fact, can be extended to cases with small initial hardness. For example, with $\frac{1-\delta}{2} = n^{-\omega(1)}$, his technique can be modified to show the impossibility in PH to amplify the hardness to $\frac{1-\delta^k}{2}$ with $k = n^{\omega(1)}$, which also follows from our corollary above. However, as discussed in Remark 4, when the initial hardness grows beyond a certain point, say to $\frac{1-\delta}{2} = \Omega(1)$, his technique fails to give a meaningful bound. Moreover, our lower bound almost matches the upper bound given by the well-known XOR lemma [29], [5], while the technique in [27] does not yield such a bound.

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

Proof: The encoding function is $\textsc{Amp}^f : (\{0,1\}^n)^t \to \{0,1\}$, with $t = O(k)$, defined as

$$\textsc{Amp}^f(x_1, \ldots, x_t) = f(x_1) \oplus \cdots \oplus f(x_t).$$

It is known that the parity of t bits can be computed by an $\mathsf{AC}(d+1, 2^{O(t^{1/d})})$ circuit (cf., [8]), and note that this circuit and those t query inputs do not depend on the oracle f. Furthermore, using Levin's proof for the XOR lemma given in [5], one can construct a decoding function that uses an advice of length $\ell \le \mathrm{poly}(\frac{n}{\delta^k})$. Thus, we have the theorem. □

Now we proceed to prove Theorem 1.

Proof (of Theorem 1): Consider any parallel black-box $(n, \frac{1-\delta}{2}, \frac{1-\delta^k}{2}, \ell)$ hardness amplification realized in $\mathsf{AC}(d, s)$, with $s = 2^{c_3 k^{1/d}}$ for a small enough positive constant c_3. Let $N = 2^n$ and $M = 2^m$. Recall from Lemma 2 that such a hardness amplification induces a $(\frac{1-\delta}{2}, \frac{1-\delta^k}{2}, 2^\ell)$-list code $C : \{0,1\}^N \to \{0,1\}^M$. Then, from Remark 3, it suffices to show that any such code C computed by an $\mathsf{AC}(d, s)$ circuit must have $\ell = 2^{\Omega(n)}$.

The basic idea behind the proof is the following. Suppose C has only a small number of codewords close to any codeword. Then, a random perturbation on an input message is unlikely to result in a close codeword. On the other hand, if C is computed by an $\mathsf{AC}(d, s)$ circuit with small d and s, 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.

Now we give the details. 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 $\frac{1-\delta}{2}$.[4] 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 $C(x)$, which relies on the fact that such an AC circuit is insensitive to noise on the input.

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

Lemma 5: There exist constants c_2, c_3, c_4 such that for any $\delta \in (0,1)$ and any $k, d \in \mathbb{N}$ with $\delta^k \le 1 - 2^{-c_2 k^{1/d}}$, if $C \in \mathsf{AC}(d, 2^{c_3 k^{1/d}})$, then $\Pr_{x,\tilde{x}}[C(x) \text{ is close to } C(\tilde{x})] \ge \delta^{c_4 k}$.

Proof: Suppose $C \in \mathsf{AC}(d, 2^{c_3 k^{1/d}})$ for a small enough constant c_3. Suppose $\delta^k \le 1 - 2^{-c_2 k^{1/d}}$ for some constant c_2 such that $\delta^{0.5k} \le 1 - 2^{-c_3 k^{1/d}}$. Then, using Corollary 1 with $t = k/3$, we have that for each $i \in [M]$

$$\begin{aligned}
&\Pr_{x,\tilde{x}}[C(x)_i \ne C(\tilde{x})_i] \\
&\quad \le \frac{1}{2}\left(1 - \alpha^t\left(1 - 2^{c_3 k^{1/d}} \cdot 2^{-\Omega(t^{1/d})}\right)\right) \\
&\quad \le \frac{1}{2}\left(1 - \delta^{0.4k}\left(1 - 2^{-c_3 k^{1/d}}\right)\right) \\
&\quad \le \frac{1}{2}\left(1 - \delta^{0.9k}\right).
\end{aligned}$$

Therefore, $\mathrm{E}_{x,\tilde{x}}[\triangle(C(x), C(\tilde{x}))] \le \frac{1}{2}(1-\delta^{0.9k})$, which implies that $\Pr_{x,\tilde{x}}[C(x) \text{ is not close to } C(\tilde{x})] \le \frac{1-\delta^{0.9k}}{1-\delta^k}$ by Markov inequality. Thus

$$\begin{aligned}
\Pr_{x,\tilde{x}}[C(x) \text{ is close to } C(\tilde{x})] &\ge 1 - \frac{1-\delta^{0.9k}}{1-\delta^k} \\
&\ge \frac{\delta^{0.9k} - \delta^k}{1-\delta^k} \\
&\ge \delta^{0.9k} - \delta^k \\
&\ge \delta^{c_4 k}
\end{aligned}$$

for some constant c_4. □

Next, we give an upper bound on the probability that $C(\tilde{x})$ is close to $C(x)$, which relies on the fact that each codeword is only close to a small number of other codewords. This requires a more careful analysis than that in [27], in order to get the tighter bound we need.

Lemma 6: For any $(\frac{1-\delta}{2}, \frac{1-\delta^k}{2}, 2^\ell)$-list code C, $\Pr_{x,\tilde{x}}[C(x) \text{ is close to } C(\tilde{x})] \le 2^\ell \cdot 2^{-\Omega(\delta^2 N)}$.

Proof: Consider any fixed $x \in \{0,1\}^N$. Because 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 $\beta = \alpha^{0.91} = \delta^{1.001}$.[5] 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 that is good for x

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

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

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

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

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

Suppose $2^{-c_0 n} \le \delta < 1$ and $2^{-2^{c_1 n}} \le \delta^k \le 1 - 2^{-c_2 k^{1/d}}$ for suitable constants c_0, c_1, c_2. Then, from Lemmas 5 and 6, we get

$$\delta^{c_4 k} \le \Pr_{x,\tilde{x}}[C(x) \text{ is close to } C(\tilde{x})] \le 2^{\ell} \cdot 2^{-\Omega(\delta^2 N)}$$

which implies that

$$2^{\ell} \ge \delta^{c_4 k} \cdot 2^{\Omega(\delta^2 N)} \ge 2^{2^{\Omega(n)}}.$$

Thus, we have the following.

Lemma 7: There exist constants c_0, c_1, c_2, c_3 such that for any $\delta \in (0,1)$ and any $d, k \in \mathbb{N}$ with $2^{-c_0 n} \le \delta < 1$ and $2^{-2^{c_1 n}} \le \delta^k \le 1 - 2^{-c_2 k^{1/d}}$, if $C : \{0,1\}^{2^n} \to \{0,1\}^M$ is a $(\frac{1-\delta}{2}, \frac{1-\delta^k}{2}, 2^{\ell})$-list code computable by an $\mathsf{AC}(d, 2^{c_3 k^{1/d}})$ circuit, then $2^{\ell} = 2^{2^{\Omega(n)}}$.

Combining this lemma with Lemma 2, we obtain Theorem 1. □

IV. Impossibility of Amplification by Nondeterministic Circuits

Note that the result in the previous section becomes meaningless for $d = \Omega(\log k)$, as it only rules out circuits in $\mathsf{AC}(d,s)$ with $s = 2^{O(k^{1/d})} = O(1)$. In this section, we show that even without any restriction on the circuit depth, a meaningful lower bound on the circuit size can still be derived. Formally, we have the following theorem.

Theorem 3: There exist constants c_0, c_1, c_2, c_3 such that for any $\delta \in (0,1)$ and any $k \in \mathbb{N}$ with $2^{-c_0 n} \le \delta < 1$ and $2^{-2^{c_1 n}} \le \delta^k \le 1 - k^{-c_2}$, any parallel black-box $(n, \frac{1-\delta}{2}, \frac{1-\delta^k}{2}, \ell)$ hardness amplification realized in $\mathsf{NAC}(\frac{k}{c_3 \log k})$ must have $\ell = 2^{\Omega(n)}$.

To the best of our knowledge, no such result has been shown for NAC circuits. From Lemma 1, this implies the following impossibility result on general black-box hardness amplification.

Corollary 3: Under the same condition as in Theorem 3, no black-box $(n, \frac{1-\delta}{2}, \frac{1-\delta^k}{2}, 2^{o(n)})$ hardness amplification can be realized in $\mathsf{ATIME}(c \log k)$, for some constant $c > 0$.

Now we prove the theorem.

Proof (of Theorem 3): The basic proof idea is similar to that for Theorem 1. The only difference is to replace Lemma 5 by an analogous one for NAC circuits. Here 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 $\mathcal{R}_p$ denote the distribution on such restrictions such that each variable x_i is mapped independently with $\Pr_{\rho \in \mathcal{R}_p}[\rho(x_i) = \star] = p$ and $\Pr_{\rho \in \mathcal{R}_p}[\rho(x_i) = 0] = \Pr_{\rho \in \mathcal{R}_p}[\rho(x_i) = 1] = (1-p)/2$. 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 $\deg(g) = \max_J\{|J| : \hat{g}(J) \ne 0\}$. It is not hard to verify 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 that bounds the contribution of higher order Fourier coefficients.

Lemma 8 [17]: Let $p \in (0,1)$ and $t \in \mathbb{N}$ with $pt > 8$. Then, for any Boolean function g, $\sum_{|J|>t} \hat{g}(J)^2 \le 2 \cdot \Pr_{\rho \in \mathcal{R}_p}[\deg(g_\rho) \ge pt/2]$.

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

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

Proof: Suppose g is computed by an NAC circuit of size s, which divides its input into the real input part and the witness part. Let $\mathcal{B}$ be the set of gates that receive some real input variables directly. Consider applying a random restriction $\rho \in \mathcal{R}_p$ 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 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

$$\begin{aligned}
&\Pr_{\rho \in \mathcal{R}_p}[\deg(g_\rho) \ge pt/2] \\
&\quad \le \Pr_{\rho \in \mathcal{R}_p}[\exists A \in \mathcal{B} : \#(A_\rho) \ge pt/(2s)] \\
&\quad \le s \cdot \max_{A \in \mathcal{B}} \Pr_{\rho \in \mathcal{R}_p}[\#(A_\rho) \ge pt/(2s)]
\end{aligned}$$

where the first inequality holds because if no gate A exists, then g_ρ must depend on fewer than $pt/2$ variables, and therefore, must have degree less than $pt/2$.

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

$$\begin{aligned}&\Pr_{\rho\in\mathcal{R}_p}[\#(A_\rho) \geq pt/(2s)]\\ &\quad\leq \Pr_{\rho\in\mathcal{R}_p}[A \text{ is not killed by } \rho]\\ &\quad\leq (1-(1-p)/2)^{pt/(2s)}\\ &\quad= 2^{-\Omega(t/s)}.\end{aligned}$$

From Lemma 8, we have $\sum_{|J|>t}\hat{g}(J)^2 \leq 2s\cdot 2^{-\Omega(t/s)} = s\cdot 2^{-\Omega(t/s)}$. □

Then, analogously to Lemma 5 (in the previous section), we have the following.

Lemma 10: There exist constants c_2, c_3, c_4 such that for any $\delta \in (0,1)$ and any $k \in \mathbb{N}$ with $\delta^k \leq 1-k^{-c_2}$, if $C \in \mathsf{NAC}(\frac{k}{c_3\log k})$, then $\Pr_{x,\tilde{x}}[C(x) \text{ is close to } C(\tilde{x})] \geq \delta^{c_4 k}$.

Proof: Suppose $C:\{0,1\}^N \to \{0,1\}^M \in \mathsf{NAC}(\frac{k}{c_3\log k})$, for some large enough constant c_3. Using Lemmas 4 and 9 with $t = k/3$, we have that for each $i \in [M]$

$$\begin{aligned}&\Pr_{x,\tilde{x}}[C(x)_i \neq C(\tilde{x})_i]\\ &\quad\leq \frac{1}{2}\left(1-\alpha^t\left(1-\frac{k}{c_3\log k}\cdot 2^{-\Omega(c_3\log k)}\right)\right)\\ &\quad\leq \frac{1}{2}\left(1-\delta^{0.4k}\left(1-k^{-\Omega(1)}\right)\right)\\ &\quad\leq \frac{1}{2}\left(1-\delta^{0.9k}\right)\end{aligned}$$

when $\delta^k \leq 1-k^{-c_2}$ for some suitable constant c_2. Then, the rest is the same as that for Lemma 5, and we can have $\Pr_{x,\tilde{x}}[C(x) \text{ is close to } C(\tilde{x})] \geq \delta^{c_4 k}$ for some constant c_4. □

Suppose $2^{-c_0 n} \leq \delta < 1$ and $2^{-2^{c_1 n}} \leq \delta^k \leq 1-k^{-c_2}$, for suitable constants c_0, c_1, c_2. By combining Lemma 10 with Lemma 6, we get $2^\ell \geq \delta^{c_4 k}\cdot 2^{\Omega(\delta^2 N)} \geq 2^{2^{\Omega(n)}}$, which gives the following.

Lemma 11: There exist constant c_0, c_1, c_2, c_3 such that for any $\delta \in (0,1)$ and any $k \in \mathbb{N}$ with $2^{-c_0 n} \leq \delta < 1$ and $2^{-2^{c_1 n}} \leq \delta^k \leq 1-k^{-c_2}$, if $C:\{0,1\}^{2^n} \to \{0,1\}^M$ is a $(\frac{1-\delta}{2}, \frac{1-\delta^k}{2}, 2^\ell)$-list code computable by $\mathsf{NAC}(\frac{k}{c_3\log k})$, then $2^\ell = 2^{2^{\Omega(n)}}$.

Combining this with Lemma 2, we obtain Theorem 3. □

V. Inherent Nonuniformity of Hardness Amplification

In the previous two sections, we have proven that any black-box hardness amplification must be very nonuniform when the computational complexity of the amplification procedure Amp is bounded in certain ways. In this section, we prove that even without any such complexity bound, there still exists some inherent nonuniformity.

First, we state the following simple result that seems to be a folklore. For completeness, we include the proof in the Appendix.

Theorem 4: For some constant c and for any $\gamma \in (0,1)$, no oracle algorithm $\mathrm{Amp}^{(\cdot)}:\{0,1\}^m \to \{0,1\}$ 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 $m = \mathrm{poly}(n)$. Thus, the theorem basically says that amplifying hardness beyond $\frac{1}{4}$ must introduce nonuniformity in general. However, the theorem does not provide a quantitative bound on the nonuniformity. This is addressed by our next theorem.

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 nonuniform, with $\ell \geq 1$ when $\varepsilon \leq c'\delta^2$ for some constant c', or with $\ell = \Omega(k\log\frac{1}{\delta})$ when $\varepsilon = \delta^k$. Note that our lower bound generalizes that of Trevisan and Vadhan [25]: they only considered the case with $\delta = 1-2^{-n+1}$ (or equivalently $\frac{1-\delta}{2} = 2^{-n}$) and obtained the lower bound $\ell = \Omega(\log\frac{1}{\varepsilon})$, while we consider general δ and obtain the lower bound $\ell = \Omega(\log\frac{\delta^2}{\varepsilon})$.

Now we proceed to the proof of Theorem 5.

Proof (of Theorem 5): Consider an arbitrary code $C:\{0,1\}^N \to \{0,1\}^M$. We would like to show that for some constant c to be determined later, one can find a string $z \in \{0,1\}^M$ and a set $S \subseteq \{0,1\}^N$ such that the following two conditions hold.

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

For this, we first choose $x^1,\ldots,x^t$ uniformly and independently from $\{0,1\}^N$ to form the set R, for some $t = \Theta(\frac{1}{\varepsilon^2})$. 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 $\mathrm{Ball}(\frac{1-\delta}{2}, N)$ contains $O(\frac{1}{\delta^2})$ elements of R. Later, we will derive the set S from a δ-good R.

Lemma 12: When $N = \omega(\frac{1}{\delta^2}\log\frac{1}{\varepsilon})$, R is δ-good with probability $1-2^{-\Omega(N)}$.

Proof: First, the probability that $x^i = x^j$ for some $i \neq j$ is at most $\binom{t}{2}\cdot 2^{-N} \leq 2^{2\log t - N}$. Next, the probability that some ball in $\mathrm{Ball}(\frac{1-\delta}{2}, N)$ 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 $r = \Theta(\frac{1}{\delta^2})$, both probabilities above are $2^{-\Omega(N)}$ when $N = \omega(\frac{1}{\delta^2}\log t)$. This proves the lemma. □

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

Definition 10: 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 13: 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 12, 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 indicator random variable such that $I_i = 1$ if $C(x^i)_y \neq b_y$ and $I_i = 0$ otherwise. Then

$$\begin{aligned}\Pr_R\left[\Pr_{x\in R}[C(x)_y \neq b_y] \le \frac{1-\varepsilon}{2}\right] &= \Pr_{x^1,\ldots,x^t}\left[\frac{1}{t}\left|\{i\in[t] : C(x^i)_y \neq b_y\}\right| \le \frac{1-\varepsilon}{2}\right] \\ &= \Pr_{x^1,\ldots,x^t}\left[\frac{1}{t}\sum_{i\in[t]} I_i \le \frac{1-\varepsilon}{2}\right].\end{aligned}$$

Note that $I_1,\ldots,I_t$ form a sequence of independent identically distributed (i.i.d.) random variables, with $\mathrm{E}[I_i] \le \frac{1}{2}$ for each i. Let $J_1,\ldots,J_t$ be the sequence of i.i.d. binary random variables with $\mathrm{E}[J_i] = \frac{1}{2}$ for each i. Then

$$\Pr\left[\frac{1}{t}\sum_{i\in[t]} I_i \le \frac{1-\varepsilon}{2}\right] \ge \Pr\left[\frac{1}{t}\sum_{i\in[t]} J_i \le \frac{1-\varepsilon}{2}\right].$$

Therefore, we have

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

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

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 $A \subseteq [M]$ with $|A| = \Omega(M)$ such that for every $y \in A$, 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 (with each bit b_y defined in Definition 10) on those dimensions in A.

Lemma 14: 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

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

By Markov's inequality

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

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

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 chosen from $\{0^M, 1^M\}$) and $S \subseteq R'$ with $|S| \ge \frac{1}{2}|R'|$ such that for any $x \in S$, $\Pr_{y\notin A}[C(x)_y \neq v_y] \le \frac{1}{2}$, 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 $x \in S$

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

for any large enough constant c.

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

Lemma 15: Suppose $\varepsilon < \frac{1}{c}$ for some suitable constant c, and suppose $N = \omega(\frac{1}{\delta^2}\log\frac{1}{\varepsilon})$. Then, any $(\frac{1-\delta}{2}, \frac{1-\varepsilon}{2}, 2^\ell)$-list code must have $2^\ell = \Omega(\frac{\delta^2}{\varepsilon})$.

This, combined with Lemma 2, proves the theorem. □

Remark 5: Recently (after the conference version of our paper), Guruswami and Vadhan [7] used a more involved argument to proved that any $(2^{-n}, \frac{1-\varepsilon}{2}, 2^\ell)$-list code must have $2^\ell = \Omega(\frac{1}{\varepsilon^2})$, which is tight as a matching upper bound (within a constant factor) is known to exist [6], [12]. The proof in [7], in fact, can be extended to show that any $(\frac{1-\delta}{2}, \frac{1-\varepsilon}{2}, 2^\ell)$-list code must have $2^\ell = \Omega(\frac{\delta^2}{\varepsilon^2})$. Therefore, any such black-box hardness amplification with $\delta \ge c_0\varepsilon$, for some constant c_0, must be inherently nonuniform.

VI. Impossibility Results on PRG Constructions

In this section, we prove lower bound (impossibility) results for black-box PRG constructions from hard functions. For this,

we establish a connection between black-box PRG constructions and codes. Then, using those lower bound results for codes in previous sections, we obtain lower bound results for black-box PRG constructions.

Consider any black-box PRG construction with an encoding function $\mathrm{G}^{(\cdot)} : \{0,1\}^m \to \{0,1\}^r$. We call the ratio $\frac{r}{m}$ as the stretch factor of the PRG construction. Let $N = 2^n$ and $M = r2^m$, and define the corresponding code $C : \{0,1\}^N \to \{0,1\}^M$ as $C(f) = \mathrm{G}^f$. 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 G^f, which is seen as a vector in $(\{0,1\}^r)^{2^m} = \{0,1\}^M$ (the concatenation of $\mathrm{G}^f(u)$'s over $u \in \{0,1\}^m$). Analogously to Lemma 2, we have the following connection between PRG constructions and codes.

Lemma 16: Suppose $\mathrm{G}^{(\cdot)} : \{0,1\}^m \to \{0,1\}^r$ is the encoding function of a black-box $(n, \beta, \frac{\varepsilon}{2}, \ell)$ PRG construction with a stretch factor $\frac{r}{m} = \omega(\frac{1}{\varepsilon^2})$. Then, $C : \{0,1\}^N \to \{0,1\}^M$, defined as $C(f) = \mathrm{G}^f$, is a $(\beta, \frac{1-\varepsilon}{2}, 2^\ell)$-list code.

Proof: Suppose G is the encoding function of a black-box $(n, \beta, \frac{\varepsilon}{2}, \ell)$ PRG construction, and DEC is the decoding function, which is an oracle Turing machine with an ℓ-bit advice. Consider any string $A \in \{0,1\}^M$, which can be seen as a function $A : \{0,1\}^m \to \{0,1\}^r$. We want to show that not many codewords are close to A. For this, we show that there exists a distinguisher D_A such that if any $C(f)$ is close to A, then D_A can distinguish G^f from random.

Define the distinguisher $D_A : \{0,1\}^r \to \{0,1\}$ as $D_A(w) = 1$ if and only if $\exists u \in \{0,1\}^m : \triangle(w, A(u)) \le \frac{1-\varepsilon/4}{2}$. Suppose $r = \omega(\frac{m}{\varepsilon^2})$, and assume without loss of generality that $2^{-\omega(m)} \le \frac{\varepsilon}{4}$.[6] Then

$$\begin{aligned}
&\Pr_{w \in \mathcal{U}_r}[D_A(w) = 1] \\
&\quad \le \sum_{u \in \{0,1\}^m} \Pr_{w \in \mathcal{U}_r}\left[\triangle(w, A(u)) \le \frac{1-\varepsilon/4}{2}\right] \\
&\quad \le 2^m \cdot 2^{-\Omega(\varepsilon^2 r)} \\
&\quad \le 2^{-\omega(m)} \\
&\quad \le \frac{\varepsilon}{4}.
\end{aligned}$$

Consider any codeword $C(f)$ with $\triangle(A, C(f)) < \frac{1-\varepsilon}{2}$. Now as $\mathrm{E}_{u \in \mathcal{U}_m}[\triangle(A(u), \mathrm{G}^f(u))] = \triangle(A, C(f))$, by Markov inequality, we have $\Pr_{u \in \mathcal{U}_m}[\triangle(A(u), \mathrm{G}^f(u)) > \frac{1-\varepsilon/4}{2}] < \frac{1-\varepsilon}{1-\varepsilon/4} \le 1 - \frac{3\varepsilon}{4}$. Thus

$$\begin{aligned}
&\Pr_{u \in \mathcal{U}_m}\left[D_A(\mathrm{G}^f(u)) = 1\right] \\
&\quad \ge \Pr_{u \in \mathcal{U}_m}\left[\triangle(\mathrm{G}^f(u), A(u)) \le \frac{1-\varepsilon/4}{2}\right] \\
&\quad > \frac{3\varepsilon}{4}.
\end{aligned}$$

[6]For a PRG $G : \{0,1\}^m \to \{0,1\}^r$, one can only expect $\varepsilon \ge 2^{-m} - 2^{-r}$, because this can be achieved by a simple distinguisher T defined as $T(z) = 1$ if and only if $z = G(0^r)$. Because G is a PRG, $r \ge m + 1$, $\varepsilon \ge 2^{-m} - 2^{-(m+1)} = 2^{-(m+1)}$, and we have $2^{-\omega(m)} \le \frac{\varepsilon}{4}$.

Therefore, we have

$$\begin{aligned}
&\left|\Pr_{u \in \mathcal{U}_m}\left[D_A(\mathrm{G}^f(u)) = 1\right] - \Pr_{w \in \mathcal{U}_r}\left[D_A(w) = 1\right]\right| \\
&> \frac{3\varepsilon}{4} - \frac{\varepsilon}{4} \\
&= \frac{\varepsilon}{2}.
\end{aligned}$$

From Definition 5, this implies that there exists an $\nu \in \{0,1\}^\ell$ such that $\triangle(\mathrm{DEC}^{D_A,\nu}, f) = \Pr_x[\mathrm{DEC}^{D_A,\nu}(x) \ne f(x)] < \beta$.

We have shown that if $C(f)$ is in $\mathrm{BALL}_A(\frac{1-\varepsilon}{2}, M)$, then f is contained in one of the 2^ℓ balls of radius β centered at $\mathrm{DEC}^{D_A,\nu}$ for $\nu \in \{0,1\}^\ell$. This implies that C is a $(\beta, \frac{1-\varepsilon}{2}, 2^\ell)$-list code. □

With the help of this lemma, lower bound results on codes in previous sections now immediately yield results on black-box constructions of PRG.

First, observe that if the PRG construction has a parallel realization in a circuit class, then every output bit of C can be computed by a circuit in the class. Then, by combining Lemma 16 with Lemma 7, we have the following theorem on parallel black-box PRG constructions realized by small-depth AC circuits.

Theorem 6: There exist constants c_0, c_1, c_2, c_3 such that for any $\delta \in (0,1)$ and any $d, k \in \mathbb{N}$ with $2^{-c_0 n} \le \delta < 1$ and $2^{-2^{c_1 n}} \le \delta^k \le 1 - 2^{-c_2 k^{1/d}}$, any parallel black-box $(n, \frac{1-\delta}{2}, \frac{\delta^k}{2}, \ell)$ realized in $\mathsf{AC}(d, 2^{c_3 k^{1/d}})$ with a stretch factor $\omega(\frac{1}{\delta^{2k}})$ must have $\ell = 2^{\Omega(n)}$.

Next, by combining Lemma 16 with Lemma 11, we immediately have the following theorem on parallel black-box PRG constructions realized by NAC circuits.

Theorem 7: There exist constants c_0, c_1, c_2, c_3 such that for any $\delta \in (0,1)$ and any $k \in \mathbb{N}$ with $2^{-c_0 n} \le \delta < 1$ and $2^{-2^{c_1 n}} \le \delta^k \le 1 - k^{-c_2}$, any parallel black-box $(n, \frac{1-\delta}{2}, \frac{\delta^k}{2}, \ell)$ PRG construction realized in $\mathsf{NAC}(\frac{k}{c_3 \log k})$ with a stretch factor $\omega(\frac{1}{\delta^{2k}})$ must have $\ell = 2^{\Omega(n)}$.

Similar to those in Sections III and IV, the two theorems above on the parallel model immediately imply impossibility results on general black-box PRG constructions, via Lemma 1.

Finally, by combining Lemma 16 with Lemma 15, we have the following theorem on the inherent nonuniformity of black-box PRG constructions.

Theorem 8: 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 $(n, \frac{1-\delta}{2}, \varepsilon, \ell)$ PRG construction with a stretch factor $\omega(\frac{1}{\varepsilon^2})$ must have $\ell = \Omega(\log \frac{\delta^2}{\varepsilon})$.

APPENDIX
PROOF OF THEOREM 4

From Lemma 2, this reduces to the following coding-theoretical question: for which values of α and β do we have an $(\alpha, \beta, 1)$-list code?

We call $C : \{0,1\}^N \to \{0,1\}^M$ an $[N, M, \alpha]$ code if the (relative Hamming) 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 [21]): An $[N, M, \alpha]$ code with $\alpha \geq \frac{1}{2}$ must have $N \leq \log(2M)$.

Combining these two facts, we have the following.

Lemma 17: For some constant c and for any $\gamma \in (0,1)$, any $(\frac{1-\gamma}{4}, \frac{1}{4}, L)$-list code $C : \{0,1\}^N \rightarrow \{0,1\}^M$ with $c\gamma\sqrt{N} > \log(2M)$ must have $L \geq 2$.

Proof: From Fact 3, there exists a $[K, N, \frac{1-\gamma}{2}]$ 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 $C \circ C' : \{0,1\}^K \rightarrow \{0,1\}^M$ is a $[K, M, \frac{1}{2}]$ code with $K > \log(2M)$, which is impossible according to Fact 4. □

Then, from Lemma 2, we obtain Theorem 4.

ACKNOWLEDGMENT

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

REFERENCES

[1] L. Babai, L. Fortnow, N. Nisan, and A. Wigderson, "BPP has subexponential time simulations unless exptime has publishable proofs," *Comput. Complex.*, vol. 3, no. 4, pp. 307–318, 1993.

[2] M. Blum and S. Micali, "How to generate cryptographically strong sequences of pseudo random bits," in *Proc. 23rd Annu. IEEE Symp. Found. Comput. Sci.*, 1982, pp. 112–117.

[3] A. Bogdanov and L. Trevisan, "On worst-case to average-case reductions for NP problems," in *Proc. 44th Annu. Symp. Found. Comput. Sci.*, Cambridge, MA, 2003, pp. 11–14.

[4] M. L. Furst, J. B. Saxe, and M. Sipser, "Parity, circuits, and the polynomial-time hierarchy," *Math. Syst. Theory*, vol. 17, no. 1, pp. 13–27, 1984.

[5] O. Goldreich, N. Nisan, and A. Wigderson, "On Yao's XOR lemma," Electronic Colloquium on Computational Complexity, Tech. Rep. TR95–050, 1995.

[6] V. Guruswami, J. Håstad, M. Sudan, and D. Zuckerman, "Combinatorial bounds for list decoding," *IEEE Trans. Inf. Theory*, vol. 48, no. 5, pp. 1021–1034, May 2002.

[7] V. Guruswami and S. Vadhan, "A lower bound on list size for list decoding," in *Proc. 8th Int. Workshop Random. Comput.*, 2005, pp. 318–329.

[8] J. Håstad, "Computational limitations for small depth circuits," Ph.D. dissertation, Dept. Math., Massachusetts Inst. Technol., Cambridge, MA, 1986.

[9] A. Healy, S. P. Vadhan, and E. Viola, "Using nondeterminism to amplify hardness," in *Proc. 36th ACM Symp. Theory Comput.*, 2004, pp. 192–201.

[10] R. Impagliazzo, "Hard-core distributions for somewhat hard problems," in *Proc. 36th Annu. IEEE Symp. Found. Comput. Sci.*, 1995, pp. 538–545.

[11] R. Impagliazzo, R. Jaiswal, and V. Kabanets, "Approximately list-decoding direct product codes and uniform hardness amplification," in *Proc. 47th Annu. Symp. Found. Comput. Sci.*, 2006, pp. 187–196.

[12] R. Impagliazzo, R. Jaiswal, V. Kabanets, and A. Wigderson, "Uniform direct product theorems: Simplified, optimized, and derandomized," in *Proc. 40th ACM Symp. Theory Comput.*, 2008, pp. 579–588.

[13] R. Impagliazzo, R. Shaltiel, and A. Wigderson, "Extractors and pseudo-random generators with optimal seed length," in *Proc. 32nd ACM Symp. Theory Comput.*, 2000, pp. 1–10.

[14] R. Impagliazzo and A. Wigderson, "P = BPP if E requires exponential circuits: Derandomizing the XOR lemma," in *Proc. 29th ACM Symp. Theory Comput.*, 1997, pp. 220–229.

[15] R. Impagliazzo and A. Wigderson, "Randomness vs. time: De-randomization under a uniform assumption," in *Proc. 39th Annu. IEEE Symp. Found. Comput. Sci.*, 1998, pp. 734–743.

[16] A. Klivans and D. van Melkebeek, "Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses," *SIAM J. Comput.*, vol. 31, no. 5, pp. 1501–1526, 2002.

[17] N. Linial, Y. Mansour, and N. Nisan, "Constant depth circuits, Fourier transform, and learnability," *J. ACM*, vol. 40, no. 3, pp. 607–620, 1993.

[18] N. Nisan and A. Wigderson, "Hardness vs randomness," *J. Comput. Syst. Sci.*, vol. 49, no. 2, pp. 149–167, 1994.

[19] R. O'Donnell, "Hardness amplification within NP," *J. Comput. Syst. Sci.*, vol. 69, no. 1, pp. 68–94, 2004.

[20] C. Papadimitriou, *Computational Complexity*. Reading, MA: Addison-Wesley, 1994.

[21] M. Plotkin, "Binary codes with specified minimum distance," *IEEE Trans. Inf. Theory*, vol. 6, no. 4, pp. 445–450, Sep. 1960.

[22] R. Shaltiel and C. Umans, "Simple extractors for all min-entropies and a new pseudo-random generator," in *Proc. 42nd Annu. IEEE Symp. Found. Comput. Sci.*, 2001, pp. 648–657.

[23] M. Sudan, L. Trevisan, and S. Vadhan, "Pseudorandom generators without the XOR lemma," *J. Comput. Syst. Sci.*, vol. 62, no. 2, pp. 236–266, 2001.

[24] L. Trevisan, "List decoding using the XOR lemma," in *Proc. 23rd Annu. IEEE Symp. Found. Comput. Sci.*, 2003, pp. 126–135.

[25] L. Trevisan and S. P. Vadhan, "Pseudorandomness and average-case complexity via uniform reductions," *Comput. Complex.*, vol. 16, no. 4, pp. 331–364, 2007.

[26] C. Umans, "Pseudo-random generators for all hardnesses," *J. Comput. Syst. Sci.*, vol. 67, no. 2, pp. 419–440, 2003.

[27] E. Viola, "The complexity of constructing pseudorandom generators from hard functions," *Comput. Complex.*, vol. 13, no. 3-4, pp. 147–188, 2004.

[28] E. Viola, "On constructing parallel pseudorandom generators from one-way functions," in *Proc. 20th Comput. Complex. Conf.*, 2005, pp. 183–197.

[29] A. C.-C. Yao, "Theory and applications of trapdoor functions," in *Proc. 23rd Annu. IEEE Symp. Found. Comput. Sci.*, 1982, pp. 80–91.