Electronic Colloquium on Computational Complexity, Revision 2 of Report No. 51 (2015)

Incompressible Functions, Relative-Error Extractors, and the Power of Nondeterministic Reductions

Benny Applebaum[*] Sergei Artemenko[†] Ronen Shaltiel[‡] Guang Yang[§]

March 13, 2016

Abstract

A circuit C *compresses* a function $f : \{0,1\}^n \rightarrow \{0,1\}^m$ if given an input $x \in \{0,1\}^n$ the circuit C can shrink x to a shorter ℓ-bit string x' such that later, a computationally-unbounded solver D will be able to compute $f(x)$ based on x'. In this paper we study the existence of functions which are *incompressible* by circuits of some fixed polynomial size $s = n^c$. Motivated by cryptographic applications, we focus on average-case (ℓ, ϵ) incompressibility, which guarantees that on a random input $x \in \{0,1\}^n$, for every size s circuit $C : \{0,1\}^n \rightarrow \{0,1\}^\ell$ and any unbounded solver D, the success probability $\Pr_x[D(C(x)) = f(x)]$ is upper-bounded by $2^{-m} + \epsilon$. While this notion of incompressibility appeared in several works (e.g., Dubrov and Ishai, STOC 06), so far no explicit constructions of efficiently computable incompressible functions were known. In this work we present the following results:

(1) Assuming that E is hard for exponential size nondeterministic circuits, we construct a polynomial time computable *boolean* function $f : \{0,1\}^n \rightarrow \{0,1\}$ which is incompressible by size n^c circuits with communication $\ell = (1 - o(1)) \cdot n$ and error $\epsilon = n^{-c}$. Our technique generalizes to the case of PRGs against nonboolean circuits, improving and simplifying the previous construction of Shaltiel and Artemenko (STOC 14).

(2) We show that it is possible to achieve *negligible* error parameter $\epsilon = n^{-\omega(1)}$ for *nonboolean* functions. Specifically, assuming that E is hard for exponential size Σ_3-circuits, we construct a nonboolean function $f : \{0,1\}^n \rightarrow \{0,1\}^m$ which is incompressible by size n^c circuits with $\ell = \Omega(n)$ and extremely small $\epsilon = n^{-c} \cdot 2^{-m}$. Our construction combines the techniques of Trevisan and Vadhan (FOCS 00) with a new notion of *relative error* deterministic extractor which may be of independent interest.

(3) We show that the task of constructing an incompressible *boolean* function $f : \{0,1\}^n \rightarrow \{0,1\}$ with *negligible* error parameter ϵ cannot be achieved by "existing proof techniques". Namely, *nondeterministic reductions* (or even Σ_i reductions) cannot get $\epsilon = n^{-\omega(1)}$ for *boolean* incompressible functions. Our results also apply to constructions of standard Nisan-Wigderson type PRGs and (standard) boolean functions that are hard on average, explaining, in retrospect, the limitations of existing constructions. Our impossibility result builds on an approach of Shaltiel and Viola (STOC 08).

[*] Tel Aviv University, Tel Aviv, Israel; bennyap@post.tau.ac.il.

[†] University of Haifa, Haifa, Israel; sartemen@gmail.com.

[‡] University of Haifa, Haifa, Israel; ronen@cs.haifa.ac.il.

[§] Tsinghua University, Beijing, P. R. China; guang.research@gmail.com.

ISSN 1433-8092

1 Introduction

In this paper we study several non-standard pseudorandom objects including incompressible functions, non-boolean PRGs and relative-error extractors for samplable and recognizable distributions. We present new constructions of these objects, relate them to each other and to standard pseudorandom objects, and study their limitations. Following some background on "traditional" pseudorandom objects (Section 1.1), we define and motivate incompressible functions, non-boolean PRGs and extractors for samplable distributions (Section 1.2). We continue with additional background on Hardness assumptions (Section 1.3), and state our results in Sections 1.4 – 1.7.

1.1 Incomputable functions and pseudorandom generators

Functions that are hard to compute on a random input, and pseudorandom generators (PRGs) are both fundamental objects in Complexity Theory, Pseudorandomness and Cryptography.

Definition 1.1 (incomputable functions and pseudorandom generators)**.**

- A function $f : \{0,1\}^n \to \{0,1\}^m$ is **incomputable** by a function class $\mathcal{C}$ if f is not contained in $\mathcal{C}$. We say that f is **ϵ-incomputable** by $\mathcal{C}$ if for every function $C : \{0,1\}^n \to \{0,1\}^m$ in $\mathcal{C}$, $\Pr_{x \leftarrow U_n}[C(x) = f(x)] \leq \frac{1}{2^m} + \epsilon$.
- A length-increasing function $G : \{0,1\}^r \to \{0,1\}^n$ $(r < n)$ is an ϵ-**PRG** for a function class $\mathcal{C}$ if for every function $C : \{0,1\}^n \to \{0,1\}$ in $\mathcal{C}$, $|\Pr[C(G(U_r)) = 1] - \Pr[C(U_n) = 1]| \leq \epsilon$.

A long line of research is devoted to achieving constructions of *explicit* incomputable functions and PRGs. As we are unable to give unconditional constructions of such explicit objects, the focus of many previous works is on achieving conditional constructions, that rely on as weak as possible unproven assumption. A common assumption under which explicit incomputable functions and PRGs can be constructed is the assumption below:

Assumption 1.2 (E is hard for exponential size circuits)**.** *There exists a language L in $E = DTIME(2^{O(n)})$ and a constant $\beta > 0$, such that for every sufficiently large n, circuits of size $2^{\beta n}$ fail to compute the characteristic function of L on inputs of length n.*

A long line of research in complexity theory is concerned with "hardness amplification" (namely, conditional constructions of explicit ϵ-incomputable functions with small ϵ) and "hardness versus randomness tradeoffs" (namely, conditional constructions of explicit PRGs). We sum up some of the main achievements of this line of research in the theorem below.

Theorem 1.3 (Babai et al., 1993; Impagliazzo and Wigderson, 1997; Lipton, 1989; Nisan and Wigderson, 1994; Sudan et al., 2001)**.** *If E is hard for exponential size circuits, then for every constant $c > 1$ there exists a constant $a > 1$ such that for every sufficiently large n, and every r such that $a \log n \leq r \leq n$:*

- *There is a function $f : \{0,1\}^r \to \{0,1\}$ that is n^{-c}-incomputable for size n^c circuits. Furthermore, f is computable in time poly(n^c).*[1]

[1] A statement like this means that we consider a family $f = \{f_n\}$ for growing input lengths, and we think of $r = r(n)$ as a function. We use this convention throughout the paper.

- *There is a function $G : \{0,1\}^r \to \{0,1\}^n$ that is an n^{-c}-PRG for size n^c circuits. Furthermore, G is computable in time poly(n^c).*

In the statement of Theorem 1.3 we allow input length r (of the functions f and G) to vary between $a \log n$ and n. It should be noted that the case of $r > a \log n$ easily follows from the case of $r = a \log n$. We state the theorem this way, as we want to emphasize that by choosing $r = n^{\Omega(1)}$, we obtain incomputable functions/PRGs which run in time polynomial in their input length.

We also stress that in many settings in derandomization, increasing the input length r of a pseudorandom object allows achieving a very small error of $\epsilon = 2^{-\Omega(r)}$. In contrast, in Theorem 1.3 this dependance is not achieved. More precisely, if we set $r = n^{\Omega(1)}$, we only get $\epsilon = n^{-c} = r^{-\Omega(1)}$ which is polynomially small in the input length. We will elaborate on this limitation later on.

1.2 Additional pseudorandom objects

In this paper we consider generalizations of incomputable functions and PRGs that were introduced by Dubrov and Ishai (2006). We also consider the notion of extractors for samplable distributions introduced by Trevisan and Vadhan (2000).

1.2.1 Incompressible functions

Compression. Consider the following scenario. A computationally bounded machine C wishes to compute some complicated function f on an input x of length n. While C cannot compute $f(x)$ alone, it has a communication-limited access to a computationally-unbounded trusted "solver" D, who is willing to help. Hence, C would like to "compress" the n-bit input x to a shorter string x' of length ℓ (the communication bound) while preserving the information needed to compute $f(x)$.

This notion of compression was explicitly introduced by Dubrov and Ishai (2006) and was later extended by several works. Harnik and Naor (2010) studied the case where f is an NP-hard function and the compressor C runs in (arbitrary) polynomial-time. This form of compressibility is closely related to the notion of "kernelization" studied in the context of Parameterized Complexity (see Bodlaender et al., 2009 and references therein). In this setting, it is known that, under standard complexity-theoretic assumptions, some functions are unlikely to be compressible (cf. Bodlaender et al., 2009; Dell and van Melkebeek, 2014; Fortnow and Santhanam, 2011). One can also consider a different setting in which the compressor is coming from a low-complexity class (e.g., constant-depth circuits). In this case, it is possible to prove unconditional incompressibility results for explicit polynomial-time computable functions (cf. Chattopadhyay and Santhanam, 2012; Dubrov and Ishai, 2006; Oliveira and Santhanam, 2015).

Following Dubrov and Ishai (2006), we focus on an intermediate setting of the problem where the gap between the time-complexity of f to the time-complexity of the compressor C is some fixed polynomial (e.g., C runs in time n^2, while f is computable in time n^3). In this setting, the notion of *incompressibility* is a natural strengthening of incomputability (as defined in Definition 1.1). We proceed with a formal definition of incompressible functions. In the following, the reader should think of the input length n as being larger than the communication-bound ℓ, which is, in turn, larger than the output length m, i.e., $n > \ell > m$.

Definition 1.4 (incompressible function (Dubrov and Ishai, 2006))**.** A function $f : \{0,1\}^n \to \{0,1\}^m$ is **incompressible** by a function $C : \{0,1\}^n \to \{0,1\}^\ell$ if for every function $D : \{0,1\}^\ell \to \{0,1\}^n$, there exists $x \in \{0,1\}^m$ such that $D(C(x)) \neq f(x)$. We say that f is **ϵ-incompressible**

by C if for every function $D : \{0,1\}^\ell \to \{0,1\}^m$, $\Pr_{x \leftarrow U_n}[D(C(x)) = f(x)] \leq \frac{1}{2^m} + \epsilon$. We say that f is **ℓ-incompressible** (resp. **(ℓ, ϵ)-incompressible**) by a class $\mathcal{C}$ if for every $C : \{0,1\}^n \to \{0,1\}^\ell$ in $\mathcal{C}$, f is incompressible (resp. ϵ-incompressible) by C.

Incompressible functions are a generalization of incomputable functions in the sense that for every $\ell \geq 1$ an (ℓ, ϵ)-incompressible function is in particular ϵ-incomputable. However, incompressibility offers several additional advantages and yield some interesting positive and negative results.

Communication lower-bounds for verifiable computation. As an immediate example, consider the problem of *verifiable computation* where a computationally bounded client C who holds an input $x \in \{0,1\}^n$ wishes to delegate the computation of $f : \{0,1\}^n \to \{0,1\}$ (an n^3-time function) to a computationally strong (say n^{10}-time) untrusted server, while verifying that the answer is correct. This problem has attracted a considerable amount of research, and it was recently shown by Kalai et al. (2014) that verifiable computation can be achieved with one round of communication in which the client sends x to the server, and, in addition, the parties exchange at most a polylogarithmic number of bits. If $(1 - o(1)) \cdot n$-incompressible functions exist, then this is essentially optimal. Furthermore, this lower bound holds even in the preprocessing model (Applebaum et al., 2010; Chung et al., 2010; Gennaro et al., 2010) where the client is allowed to send long messages before seeing the input. Similar tight lower bounds can be shown for other related cryptographic tasks such as instance-hiding or garbled circuits (cf. Applebaum et al., 2015, Section 6).

Leakage-resilient storage (Davì et al., 2010). On the positive side, consider the problem of storing a cryptographic key K on a computer that may leak information. Specifically, assume that our device was hacked by a computationally-bounded virus C who reads the memory and sends at most ℓ bits to a (computationally unbounded) server D.[2] Is it possible to securely store a cryptographic key in such a scenario? Given an (ℓ, ϵ)-incompressible function $f : \{0,1\}^n \to \{0,1\}^m$ we can solve the problem (with information-theoretic security) by storing a random $x \leftarrow \{0,1\}^n$ and, whenever a cryptographic key K is needed, compute $K = f(x)$ on-the-fly without storing it in the memory. For this application, we need average-case incompressibility (ideally with negligible ϵ), and a large output length m. Furthermore, it is useful to generalize incompressibility to the interactive setting in which the compressor C is allowed to have a multi-round interaction with the server D.

Unfortunately, so far it is unknown how to construct (based on "standard assumptions") explicit functions which are incompressible by bounded-size circuits, even in the worst-case setting.[3]

1.2.2 PRGs for nonboolean circuits

Dubrov and Ishai (2006) considered a generalization of pseudorandom generators, which should be secure even against distinguishers that output many bits. In the definition below, the reader should think of $\ell \leq r < n$.

[2] One may argue that if the outgoing communication is too large, the virus may be detected.

[3] As already mentioned, functions which are incompressible by depth-limited circuits (e.g., AC0) can be constructed unconditionally (cf. Chattopadhyay and Santhanam, 2012; Dubrov and Ishai, 2006; Oliveira and Santhanam, 2015). Such functions were also used in the context of leakage resilient cryptography by Faust et al. (2014).

Definition 1.5 (PRG for boolean and nonboolean distinguishers (Dubrov and Ishai, 2006))**.** A function $G : \{0,1\}^r \to \{0,1\}^n$ is an ϵ-**PRG** for a function $C : \{0,1\}^n \to \{0,1\}^\ell$ if the distributions $C(G(U_r))$ and $C(U_n)$ are ϵ-close.[4] G is an (ℓ,ϵ)-**PRG** for a class $\mathcal{C}$ of functions, if G is an ϵ-PRG for every function $C : \{0,1\}^n \to \{0,1\}^\ell$ in $\mathcal{C}$.

Indeed, note that a $(1,\epsilon)$-PRG is simply an ϵ-PRG. Dubrov and Ishai noted that PRGs with large ℓ can be used to reduce the randomness of sampling procedures. We now explain this application. In the definition below, the reader should think of $\ell \le n$.

Definition 1.6 (Samplable distribution)**.** We say that a distribution X on ℓ bits is samplable by a class $\mathcal{C}$ of functions $C : \{0,1\}^n \to \{0,1\}^\ell$ if there exists a function C in the class such that X is distributed as $C(U_n)$.

Imagine that we can sample from some interesting distribution X on $\ell = n^{1/10}$ bits using n random bits, by a procedure C that runs in time n^2. If we have a poly(n)-time computable (ℓ,ϵ)-PRG $G : \{0,1\}^r \to \{0,1\}^n$ against size n^2 circuits, then the procedure $P(s) = C(G(s))$ is a polynomial time procedure that samples a distribution that is ϵ-close to X (meaning that even an unbounded adversary cannot distinguish between the two distributions). Furthermore, this procedure uses only r random bits (rather than n random bits) and we can hope to obtain $r \ll n$.

1.2.3 Extractors for samplable distributions

Deterministic (seedless) extractors are functions that extract randomness from "weak sources of randomness". The reader is referred to Shaltiel (2002, 2011b) for survey articles on randomness extractors.

Definition 1.7 (deterministic extractor)**.** Let $\mathcal{C}$ be a class of distributions over $\{0,1\}^n$. A function $E : \{0,1\}^n \to \{0,1\}^m$ is a (k,ϵ)-extractor for $\mathcal{C}$ if for every distribution X in the class $\mathcal{C}$ such that $H_\infty(X) \ge k$, $E(X)$ is ϵ-close to uniform.[5]

Trevisan and Vadhan (2000) considered extractors for the class of distributions samplable by small circuits (e.g., distributions samplable by circuits of size n^2).[6] The motivation presented by Trevisan and Vadhan is to extract randomness from "weak sources of randomness" in order to generate keys for cryptographic protocols. Indeed, extractors for samplable distributions are *seedless* and require no additional randomness (in contrast to seeded extractors). Note that for this application we would like extractors that run in polynomial time. The model of samplable distributions (e.g. circuits of size n^2) is very general, and contains many subclasses of distributions studied in the literature on seedless extractors. Finally, Trevisan and Vadhan make the philosophical assumption that distributions obtained by nature must be efficiently samplable.

Summing up, if we are convinced that the physical device that is used by an honest party as a "weak source of randomness" has low complexity, (say size n^2), then even an unbounded adversary that gets to *choose* or *affect* the source, cannot distinguish between the output of the extractor and a uniformly random string with advantage $\ge \epsilon$.

[4] We use U_n to denote the uniform distribution on n bits. Two distributions X, Y over the same domain are ϵ-close if for any event A, $|\Pr[X \in A] - \Pr[Y \in A]| \le \epsilon$.

[5] For a distribution X over $\{0,1\}^n$, $H_\infty(X) := \min_x \log \frac{1}{\Pr[X=x]}$, where the minimum is taken over all strings x in the support of X.

[6] In this paper we won't explicitly set a bound on the input length of the sampling circuit as such a bound is implied by the bound on its size.

1.3 Hardness assumptions against nondeterministic and Σ_i-circuits

In contrast to incomputable functions and (standard) PRGs, poly(n)-time constructions of the three objects above (incompressible functions, PRGs for nonboolean distinguishers and extractors for samplable distributions) are not known to follow from the assumption that E is hard for exponential size circuits. We now discuss stronger variants of this assumption under which such constructions can be achieved.

Definition 1.8 (nondeterministic circuits, oracle circuits and Σ_i-circuits)**.** A *nondeterministic* circuit C has additional "nondeterministic input wires". We say that the circuit C evaluates to 1 on x iff there exist an assignment to the nondeterministic input wires that makes C output 1 on x. An oracle circuit $C^{(\cdot)}$ is a circuit which in addition to the standard gates uses an additional gate (which may have large fan in). When instantiated with a specific boolean function A, C^A is the circuit in which the additional gate is A. Given a boolean function $A(x)$, an A-circuit is a circuit that is allowed to use A gates (in addition to the standard gates). An NP-circuit is a SAT-circuit (where SAT is the satisfiability function) a Σ_i-circuit is an A-circuit where A is the canonical Σ_i^P-complete language. The size of all circuits is the total number of wires and gates.[7]

Note, for example, that an NP-circuit is different than a nondeterministic circuit. The former is a nonuniform analogue of P^{NP} (which contains coNP) while the latter is an analogue of NP. Hardness assumptions against nondeterministic/NP/Σ_i circuits appear in the literature in various contexts of complexity theory and derandomization (Barak et al., 2007; Drucker, 2013; Feige and Lund, 1997; Goldreich and Wigderson, 2002; Gutfreund et al., 2003; Klivans and van Melkebeek, 2002; Miltersen and Vinodchandran, 2005; Shaltiel and Umans, 2005, 2006, 2009; Trevisan and Vadhan, 2000). Typically, the assumption used is identical to that of Assumption 1.2 except that "standard circuits" are replaced by one of the circuit types defined above. For completeness we restate this assumption precisely.

Definition 1.9. We say that "E is hard for exponential size circuits of type X" if there exists a problem L in E $=$ DTIME($2^{O(n)}$) and a constant $\beta > 0$, such that for every sufficiently large n, circuits of type X with size $2^{\beta n}$ fail to compute the characteristic function of L on inputs of length n.

Such assumptions can be seen as the nonuniform and scaled-up versions of assumptions of the form EXP $\neq$ NP or EXP $\neq \Sigma_2^{\mathrm{P}}$ (which are widely believed in complexity theory). As such, these assumptions are very strong, and yet plausible - the failure of one of these assumptions will force us to change our current view of the interplay between time, nonuniformity and nondeterminism.[8]

Hardness assumptions against nondeterministic or Σ_i-circuits appear in the literature in several contexts (most notably as assumptions under which AM $=$ NP). It is known that Theorem 1.3 extends to every type of circuits considered in Definition 1.8.

[7] An alternative approach is to define our model using the Karp-Lipton notation for Turing machines with advice. For $s \geq n$, a size $s^{\Theta(1)}$ deterministic circuit is equivalent to DTIME($s^{\Theta(1)}$)/$s^{\Theta(1)}$, a size $s^{\Theta(1)}$ nondeterministic circuit is equivalent to NTIME($s^{\Theta(1)}$)/$s^{\Theta(1)}$, a size $s^{\Theta(1)}$ NP-circuit is equivalent to $\mathrm{DTIME}^{\mathrm{NP}}(s^{\Theta(1)})/s^{\Theta(1)}$, a size $s^{\Theta(1)}$ nondeterministic NP-circuit is equivalent to $\mathrm{NTIME}^{\mathrm{NP}}(s^{\Theta(1)})/s^{\Theta(1)}$, and a size $s^{\Theta(1)}$ Σ_i-circuit is equivalent to $\mathrm{DTIME}^{\Sigma_i^P}(s^{\Theta(1)})/s^{\Theta(1)}$.

[8] Another advantage of constructions based on this type of assumptions is that any E-complete problem (and such problems are known) can be used to implement the constructions, and the correctness of the constructions (with that specific choice) follows from the assumption. We do not have to consider and evaluate various different candidate functions for the hardness assumption.

Theorem 1.10 (Impagliazzo and Wigderson (1997); Klivans and van Melkebeek (2002); Shaltiel and Umans (2005, 2006))**.** *For every $i \geq 0$, the statement of Theorem 1.3 also holds if we replace every occurrence of the word "circuits" by "Σ_i-circuits" or alternatively by "nondeterministic Σ_i-circuits".*

Thus, loosely speaking, if E is hard for exponential size circuits of type X, then for every $c > 1$ we have PRGs and incomputable functions for size n^c circuits of type X, and these objects are poly(n^c)-time computable, and have error $\epsilon = n^{-c}$.[9]

1.4 New constructions based on hardness for nondeterministic circuits

Our first results are explicit constructions of incompressible functions and PRGs for non-boolean distinguishers from the assumption that E is hard for exponential size nondeterministic circuits.

1.4.1 A construction of incompressible functions

Our first result is a construction of polynomial time computable incompressible functions, based on the assumption that E is hard for exponential size nondeterministic circuits. This is the first construction of incompressible functions from "standard assumptions". The theorem below is stated so that the input length of the function is n. However, The input length can be shortened to any $\Omega(\log n) \leq r \leq n$ as in the case of incomputable functions stated in Theorem 1.3.

Theorem 1.11. *If E is hard for exponential size nondeterministic circuits, then for every constant $c > 1$ there exists a constant $d > 1$ such that for every sufficiently large n, there is a function $f : \{0,1\}^n \rightarrow \{0,1\}$ that is (ℓ, n^{-c})-incompressible for size n^c circuits, where $\ell = n - d \cdot \log n$. Furthermore, f is computable in time poly(n^c).*

The theorem smoothly generalizes to the case of non-boolean functions $f \colon \{0,1\}^n \rightarrow \{0,1\}^{n-\ell-d\log n}$, and can also be extended to the interactive setting at the expense of strengthening the assumption to "E is hard for exponential size nondeterministic NP-circuits". (See Section 4.)

1.4.2 A construction of PRGs for nonboolean circuits

Dubrov and Ishai (2006) showed that incompressible functions imply PRGs for nonboolean distinguishers. More precisely, they used the analysis of the Nisan-Wigderson generator by Nisan and Wigderson (1994) to argue that an incompressible function with the parameters obtained by Theorem 1.11 implies that for every constant $c > 1$, and every sufficiently large n and $n^{\Omega(1)} \leq \ell < n$, there is a poly($n^c$)-time computable (ℓ, n^{-c})-PRG $G : \{0,1\}^{r=O(\ell^2)} \rightarrow \{0,1\}^n$ for circuits of size n^c. Using this relationship, one can obtain such PRGs under the assumption that E is hard for exponential size nondeterministic circuits. Note that a drawback of this result is that the seed length r is *quadratic* in ℓ, whereas an optimal PRG can have seed length $r = O(\ell)$. This difference

[9] Historically, the interest in PRGs for nondeterministic/NP circuits was motivated by the goal of proving that AM = NP, which indeed follows using sufficiently strong PRGs (Klivans and van Melkebeek, 2002; Miltersen and Vinodchandran, 2005; Shaltiel and Umans, 2005, 2006). It is important to note, that in contrast to PRGs against deterministic circuits, PRGs for nondeterministic circuits are trivially impossible to achieve, if the circuit can simulate the PRG. Indeed, this is why we consider PRGs against circuits of size n^c that are computable in larger time of poly(n^c).

is significant in the application of reducing the randomness of sampling procedures (as explained in detail by Artemenko and Shaltiel (2014b)).

Artemenko and Shaltiel (2014b) constructed PRG for nonboolean circuits with the parameters above, while also achieving seed length $r = O(\ell)$. However, they used the stronger assumption that E is hard for nondeterministic NP-circuits. In the theorem below we obtain the "best of both worlds": We start from the assumption that E is hard for nondeterministic circuits and obtain PRGs with the optimal seed length of $r = O(\ell)$.

Theorem 1.12. *If E is hard for exponential size nondeterministic circuits, then there exists a constant $b > 1$ such that for every constant $c > 1$ there exists a constant $a > 1$ such that for every sufficiently large n, and every ℓ such that $a \log n \leq \ell \leq n$, there is a function $G : \{0,1\}^{b \cdot \ell} \to \{0,1\}^n$ that is an (ℓ, n^{-c})-PRG for size n^c circuits. Furthermore, G is computable in time poly(n^c).*

It should be noted that if $\ell \leq c \log n$ then standard PRGs against size $2 \cdot n^c$ circuits are also non-boolean PRGs. This is because any statistical test on $\ell = c \log n$ bits can be implemented by a circuit of size n^c.

1.5 The power and limitations of nondeterministic reductions

1.5.1 Negligible error in pseudorandom objects?

A common theme in Theorems 1.3, 1.10, 1.11 and 1.12 is that we can get $\epsilon = n^{-c}$, but we never get $\epsilon = n^{-\omega(1)}$ which would be desired, for example, for the virus application. This holds even if we are allowed to increase the input/seed length r, and let r approach n (say $r = n^{\Omega(1)}$). More generally, in all these results (and in fact, in all the literature on achieving incomputable functions/PRGs from the assumption that E is hard for exponential size *deterministic* circuits) $1/\epsilon$ is always smaller than the running time of the constructed object. Consequently, polynomial time computable constructs do not obtain negligible error of $\epsilon = n^{-\omega(1)}$. This phenomenon is well understood, in the sense that there are general results showing that "current proof techniques" cannot beat this barrier (Artemenko and Shaltiel, 2014a; Shaltiel and Viola, 2010). We give a more precise account of these results in Section 6.

However, there are examples in the literature where assuming hardness against *nondeterministic* (or more generally Σ_i) circuits, it is possible to beat this barrier. The first example is the seminal work of Feige and Lund (1997) on hardness of the permanent. More relevant to our setup are the following two results by Trevisan and Vadhan (2000), and Drucker (2013), stated precisely below. Note that in both cases, the target function is a polynomial time computable function that is ϵ-incomputable for negligible $\epsilon = n^{-\omega(1)}$.

Theorem 1.13 (Nonboolean incomputable function with negligible error (Trevisan and Vadhan, 2000)[10]). *If E is hard for exponential size NP-circuits, then there exists some constant $\alpha > 0$ such that for every constant $c > 1$ and for every sufficiently large n, there is a function $f : \{0,1\}^n \to \{0,1\}^m$ that is ϵ-incomputable by size n^c circuits for $m = \alpha n$ and $\epsilon = 2^{-(m/3)} = 2^{-\Omega(n)}$. Furthermore, f is computable in time poly(n^c).*

[10] Theorem 1.13 is not stated in this form in Trevisan and Vadhan (2000). Nevertheless, it directly follows from Trevisan and Vadhan (2000). See Artemenko et al. (2016) (Section 7) for an explanation.

Theorem 1.14 (Nonboolean incomputable function with negligible error (corollary of Drucker, 2013)[11]). *For every $c > 1$ there is a constant $c' > c$ such that if there is a problem in P that for every sufficiently large n is $(\frac{1}{2} - \frac{1}{n})$-incomputable by nondeterministic circuits of size $n^{c'}$, then for every sufficiently large n, there is a function $f : \{0,1\}^n \to \{0,1\}^{\sqrt{n}}$ that is ϵ-incomputable by circuits of size n^c, for $\epsilon = 2^{-n^{\Omega(1)}}$. Furthermore, f is computable in time poly(n^c).*[12]

It is important to note that in both cases above the target function that is constructed is *nonboolean.* We stress that the aforementioned lower bounds of Artemenko and Shaltiel (2014a) apply also to the case of nonboolean target functions, and the proofs above bypass these limitations by using *nondeterministic reductions.*

More precisely, assuming that the target function can be computed too well, the proofs need to contradict the assumption that E is hard for nondeterministic/Σ_i-circuits. They do this by designing a reduction. This reduction uses a deterministic circuit that computes the target function too well, in order to construct a nondeterministic/Σ_i-circuit that contradicts the assumption. This setting allows the reduction itself to be a nondeterministic/Σ_i-circuit. A precise definition of nondeterministic reductions appears in Section 6.

Nondeterministic reductions are very powerful and previous limitations on reductions (Artemenko and Shaltiel, 2014a; Shaltiel and Viola, 2010) do not hold for nondeterministic reductions. (Indeed, Theorems 1.13 and 1.14 beat the barrier and achieve polynomial time computable functions that are $n^{-\omega(1)}$-incomputable).

Our Theorems 1.11 and 1.12 are also proven using nondeterministic reductions. This raises the question whether nondeterministic reductions can achieve error $\epsilon = n^{-\omega(1)}$ in these cases. More generally, given the success of Trevisan and Vadhan, and Drucker, it is natural to hope that we can get $\epsilon = n^{-\omega(1)}$ in the classical results stated in Theorem 1.3, if we are willing to assume the stronger assumption that E is hard for exponential size Σ_i-circuits, for some $i > 0$. Assuming this stronger assumption will allow the proof to use nondeterministic reductions (and the aforementioned lower bounds do not hold).

1.5.2 Limitations on nondeterministic reductions

In this paper we show that nondeterministic reductions (or, more generally, Σ_i-reductions) cannot be used to obtain a polynomial time $n^{-\omega(1)}$-incomputable *boolean* function, starting from the assumption that E is hard for exponential size Σ_i-circuits (no matter how large i is). To the best of our knowledge, our model of nondeterministic reductions (that is explained in Section 6) is sufficiently general to capture all known proofs in the literature on hardness amplification and PRGs.[13] This is a startling contrast between boolean and non-boolean hardness amplification - the latter

[11] Drucker (2013) considers a more general setting, on which we will not elaborate, and proves a direct product result. The result we state is a corollary that is easy to compare to the aforementioned results.

[12] The assumption of Theorem 1.14 is known to follow from the assumptions E is hard for exponential size nondeterministic circuits by Theorem 1.10. Consequently, the assumption used in Theorem 1.14 follows from the assumption in Theorem 1.13. The converse does not hold. We also remark that our Theorem 1.11 holds also if we replace the assumption by the following assumption that is similar in structure to Drucker's assumption: For every $c > 1$ there is a constant $c' > c$ such that there is a problem in P that for every sufficiently large n is $(\frac{1}{2} - \frac{1}{n})$-incomputable by NP-circuits of size $n^{c'}$. The same holds for our Theorem 1.12 if we make the additional requirement that $\ell = n^{\Omega(1)}$.

[13] It should be noted that there are proof techniques (see e.g. Gutfreund and Ta-Shma, 2007; Gutfreund et al., 2007) that bypass analogous limitations in a related setup. See Gutfreund and Ta-Shma (2007) for a discussion.

can achieve negligible error, while the former cannot.[14] Our results provide a formal explanation for the phenomenon described above, and in particular, explains why Trevisan and Vadhan, and Drucker, did not construct boolean functions.

We show that the same limitations hold, also for incompressible functions, PRGs against both boolean and nonboolean distinguishers, and extractors for samplable distributions. Our results are summarized informally below, and the precise statement of our limitations appears in Section 6.

Informal Theorem 1.15. *For every $i \geq 0$ and $c > 0$, it is impossible to use "black-box reductions" to prove that the assumption that E is hard for exponential size Σ_i-circuits implies that for $\epsilon = n^{-\omega(1)}$, there is a poly$(n)$-time computable:*

- *functions $f : \{0,1\}^n \to \{0,1\}$ which are ϵ-incomputable by size n^c circuits, or*
- *ϵ-PRG $G : \{0,1\}^r \to \{0,1\}^n$ for size n^c circuits (the limitation holds for every $r \leq n-1$), or*
- *(ℓ, ϵ)-PRG $G : \{0,1\}^r \to \{0,1\}^n$ for size n^c circuits (the limitation holds for every $r \leq n-1$), or*
- *(k, ϵ)-extractor $E : \{0,1\}^n \to \{0,1\}^m$ for size n^c circuits (the limitation holds for every $m \geq 1$ and $k \leq n - \log(1/\epsilon)$).*

Furthermore, these limitations hold even if we allow reductions to perform Σ_i-computations, make adaptive queries to the "adversary breaking the security guarantee" and receive arbitrary polynomial-size nonuniform advice about the adversary.

It is interesting to note that previous work on (deterministic) black-box reductions often cannot handle reductions that are both adaptive and nonuniform (Gutfreund and Rothblum, 2008; Shaltiel and Viola, 2010) (see Artemenko and Shaltiel, 2014a for a discussion) and so the model of nondeterministic reductions that we consider is very strong.

Related work on limitations on black-box hardness amplification A "black-box proof of hardness amplification" consists of two components: A *construction* (showing how to to compute the target function given access to the hardness assumption) and a *reduction* (showing that an adversary that is able to compute the target function too well, can be used to break the initial hardness assumption). We stress that in this paper we prove limitations on *reductions.* Our limitation holds without placing limitations on the complexity of the construction (and this only makes our results stronger).

There is an orthogonal line of work which is interested in proving limitations on low complexity constructions. Some of these results (Lu et al., 2007, 2008; Viola, 2005) show lower bounds on constructions implementable in the polynomial time hierarchy. However, this line of work is incomparable to ours, and is not relevant to the setting that we consider.

More specifically, we want to capture cases in which the hardness assumption is for a problem in exponential time. All previous limitations on the complexity of the *construction* do not hold in this setting. We elaborate on our model and the meaning of our results in Section 6.

[14] Another contrast between boolean and nonboolean hardness amplification was obtained by Shaltiel and Viola (2010) for reductions that are non-adaptive constant depth circuits, and the reasons for the current contrast, are similar. Our proof follows the strategy of Shaltiel and Viola (2010) as explained in detail in Section 2.

1.6 Nonboolean incompressible functions with negligible error

In light of the previous discussion, if we want to achieve poly-time computable ϵ-incompressible functions with $\epsilon = n^{-\omega(1)}$ we must resort to nonboolean functions. In the next theorem we give such a construction.

Theorem 1.16 (Nonboolean incompressible function with negligible error)**.** *If E is hard for exponential size Σ_3-circuits then there exists a constant $\alpha > 0$ such that for every constant $c > 1$ and every sufficiently large n, and $m \leq \alpha \cdot n$ there is a function $f : \{0,1\}^n \to \{0,1\}^m$ that is $(\ell, n^{-c} \cdot 2^{-m})$-incompressible for size n^c circuits, where $\ell = \alpha \cdot n$. Furthermore, f is computable in time poly(n^c).*

We remark that the proof of Theorem 1.16 uses different techniques from the proof of Theorem 1.11. More specifically, the incompressible function of Theorem 1.16 is a consequence of a construction of an object that we call "relative-error extractor for recognizable distributions". We elaborate on this object and explain the connection to incompressible functions in Section 1.7.

We also note that the conclusion of Theorem 1.16 is stronger than that of Theorems 1.13 and 1.14, even if we restrict our attention to $\ell = 1$. Specifically for $m = \Omega(n)$, we obtain that $f : \{0,1\}^n \to \{0,1\}^{\Omega(n)}$ is ϵ-incomputable by size n^c circuits, with $\epsilon = n^{-c} \cdot 2^{-\Omega(n)}$, meaning that circuits of size n^c, have probability at most $\frac{1+n^{-c}}{2^m}$ of computing $f(x)$. This should be compared to the probability of random guessing which is $\frac{1}{2^m}$. Note that in the aforementioned theorems of (Drucker, 2013; Trevisan and Vadhan, 2000) the probability is larger than $2^{-(m/2)}$ which is large compared to 2^{-m}.

Moreover, the function we get is not only ϵ-incomputable, but (ℓ, ϵ)-incompressible for large $\ell = \Omega(n)$, and we will show that this holds even in the interactive setting. Getting back to the memory leakage scenario, we will later see that (variants of) the theorem allows us to achieve a constant rate scheme (an m bit key is encoded by $n = O(m)$ bits) which resists an n^c-time virus that (interactively) leaks a constant fraction of the stored bits.

1.7 Deterministic extractors with relative error

1.7.1 Previous work on extractors for samplable distributions

Trevisan and Vadhan constructed extractors for distributions samplable by size n^c circuits. The precise statement appears below.

Theorem 1.17 (Extractors for samplable distributions Trevisan and Vadhan (2000))**.** *If E is hard for exponential size Σ_4-circuits then there exists a constant $\alpha > 0$ such that for every constant $c > 1$ and sufficiently large n, and every $m \leq \alpha n$ there is a $((1-\alpha) \cdot n, \frac{1}{n^c})$-extractor $E : \{0,1\}^n \to \{0,1\}^m$ for distributions samplable by size n^c circuits. Furthermore, E is computable in time poly(n^c).*[15]

As explained earlier, our limitations explain why Trevisan and Vadhan did not achieve $\epsilon = n^{-\omega(1)}$. This may be a significant drawback in applications. In particular, if we use the extractor to generate keys for cryptographic protocols (as explained in Section 1.2.3) then it might be that an

[15] In Trevisan and Vadhan (2000), this is stated with $m = 0.5 \cdot c \cdot \log n$, but a more careful argument can give the stronger result that we state here. Another result that appears in Trevisan and Vadhan (2000) allows m to be $(1-\delta) \cdot n$ for an arbitrary constant $\delta > 0$, and then Σ_4 is replaced by Σ_5, $\epsilon = 1/n$ and the running time is $n^{b_{c,\delta}}$ for a constant $b_{c,\delta}$ that depends only on c and δ.

adversary that has a negligible probability of attacking the protocol under the uniform distribution, has a noticeable probability of attacking under the distribution output by the extractor.[16]

1.7.2 Extractors with relative error

In order to circumvent this problem we suggest the following revised notion of statistical distance, and extractors.

Definition 1.18 (statistical distance with relative error)**.** We say that a distribution Z on $\{0,1\}^m$ is **ϵ-close to uniform with relative error** if for every event $A \subseteq \{0,1\}^m$, $|\Pr[Z \in A] - \mu(A)| \leq \epsilon \cdot \mu(A)$ where $\mu(A) = |A|/2^m$.[17]

Note that if Z is ϵ-close to uniform with relative error, then it is also ϵ-close to uniform. However, we now also get that for every event A, $\Pr[Z \in A] \leq (1+\epsilon) \cdot \mu(A)$ and this implies that events that are negligible under the uniform distributions cannot become noticeable under Z.

We now introduce a revised definition of deterministic extractors by replacing the requirement that the output is ϵ-close to uniform by the requirement that the output is close to uniform with relative error.

Definition 1.19 (deterministic extractor with relative error)**.** Let $\mathcal{C}$ be a class of distributions over $\{0,1\}^n$. A function $E : \{0,1\}^n \to \{0,1\}^m$ is a (k,ϵ)-relative-error extractor for $\mathcal{C}$ if for every distribution X in the class $\mathcal{C}$ such that $H_\infty(X) \geq k$, $E(X)$ is ϵ-close to uniform with relative error.

To the best of our knowledge, this concept of "relative-error extractor" was not previously considered in the literature. We first observe that a standard probabilistic argument shows existence of such extractors for any small class of distributions. This follows by proving that random functions satisfy this property with high probability (using the same calculation as in the case of standard extractors). Moreover, this probabilistic argument works with random t-wise independent functions. Specifically, the following theorem was implicitly proven by Trevisan and Vadhan (2000) (Proof of Proposition A.1):

Theorem 1.20 (Existence of relative-error extractors)**.** *Let $\mathcal{C}$ be a class of at most N distributions on $\{0,1\}^n$. Then there exists a (k,ϵ)-relative-error extractor $E : \{0,1\}^n \to \{0,1\}^m$ for $\mathcal{C}$ with $m = k - 2\log(1/\epsilon) - O(\log\log N)$. Furthermore, with probability at least $1-2^{-n}$ a random $O(n+\log N)$-wise independent function $h : \{0,1\}^n \to \{0,1\}^m$ is a (k,ϵ)-relative-error extractor $E : \{0,1\}^n \to \{0,1\}^m$ for $\mathcal{C}$.*

1.7.3 New constructions of relative-error extractors for samplable distributions

We are able to extend Theorem 1.17 to hold with this new definition. Specifically:

Theorem 1.21 (Extractors for samplable distributions with relative error)**.** *If E is hard for exponential size Σ_4-circuits then there exists a constant $\alpha > 0$ such that for every constant $c > 1$ and sufficiently large n, and every $m \leq \alpha n$ there is a $((1-\alpha) \cdot n, \frac{1}{n^c})$-relative-error extractor $E : \{0,1\}^n \to \{0,1\}^m$ for distributions samplable by size n^c circuits. Furthermore, E is computable in time poly(n^c).*

[16] A function $\alpha(n) : \mathbb{N} \to [0,1]$ is noticeable if it is lower-bounded by some inverse polynomial, i.e., $\alpha(n) > 1/n^{\Omega(1)}$.

[17] While we'll use this definition mostly with $\epsilon < 1$, note that it makes sense also for $\epsilon \geq 1$.

As previously explained this means that events that receive negligible probability under the uniform distribution also receive negligible probability under the output distribution of the extractor. We believe that this makes extractors for samplable distributions more suitable for cryptographic applications.

1.7.4 Relative-error extractors for recognizable distributions

We say that a circuit $C : \{0,1\}^n \to \{0,1\}$ "recognizes" a distribution X over $\{0,1\}^n$ if X is uniform over the set of satisfying assignments of C. Correspondingly, Shaltiel (2011a) introduced the notion of "recognizable distributions" which can be viewed as dual to the notion of efficiently samplable distributions.

Definition 1.22 (Recognizable distribution Shaltiel (2011a))**.** We say that a distribution X on n bits is **recognizable** by a class $\mathcal{C}$ of functions $C : \{0,1\}^n \to \{0,1\}$ if there exists a function C in the class such that X is uniform over $\{x : C(x) = 1\}$.

As we will later see extractors for distributions recognizable by small circuits translate into incompressible functions. Furthermore, relative-error extractors with large error translate into non-boolean incompressible functions with very small error.

Lemma 1.23.

- *An $(n - (\ell + \log(1/\epsilon) + 1), \epsilon/2)$-extractor for distributions recognizable by size $2n^c$ circuits, is an (ℓ, ϵ)-incompressible function for size n^c circuits.*
- *An $(n - (\ell + \log(1/\epsilon) + m + 1), \epsilon/2)$ relative-error extractor $f : \{0,1\}^n \to \{0,1\}^m$ for distributions recognizable by size $2n^c$ circuits, is an $(\ell, \epsilon \cdot 2^{-m})$-incompressible function for size n^c circuits.*

This argument demonstrates (once again) the power of extractors with relative error. More precisely, note that even if ϵ is noticeable (i.e., $1/n^{\Omega(1)}$), we get guarantees on probabilities that are negligible! This lemma shows that in order to construct nonboolean incompressible functions with very low error, it is sufficient to construct extractors for recognizable distributions with relative error that is noticeable.

This lemma follows because if we choose $X \leftarrow U_n$ and consider the distribution of $(X|C(X) = a)$ for some compressed value $a \in \{0,1\}^\ell$ that was computed by the compressor C, then this distribution is recognizable, and for most a, it has sufficiently large min-entropy for the extractor f. It follows that $f(X)$ is close to uniform with relative error even after seeing $C(X)$. However, in a distribution that is ϵ-close to uniform with relative error, no string has probability larger than $(1 + \epsilon) \cdot 2^{-m}$, and so even an unbounded adversary that sees $C(X)$ cannot predict $f(X)$ with advantage better than $\epsilon \cdot 2^{-m}$ over random guessing. We give a full proof in a more general setup in the formal section (Section 4).

Our next result is a construction of a relative-error extractor for recognizable distributions.

Theorem 1.24 (Extractors with relative error for recognizable distributions)**.** *If E is hard for exponential size Σ_3-circuits then there exists a constant $\alpha > 0$ such that for every constant $c > 1$ and sufficiently large n, and every $m \leq \alpha n$ there is a $((1-\alpha) \cdot n, \frac{1}{n^c})$-relative-error extractor $E : \{0,1\}^n \to \{0,1\}^m$ for distributions recognizable by size n^c circuits. Furthermore, E is computable in time poly(n^c).*

Application in the leakage resilient scenario. The same reasoning applies in the memory leakage scenario described in Section 1.2.1. Using a relative-error extractor for recognizable distributions f, we can achieve a constant rate scheme (an m bit key is encoded by $n = O(m)$ bits) which resists an n^c-time virus who (interactively) leaks a constant fraction of the stored bits in the following strong sense: Say that the key $K = f(x)$ is used as the key of some cryptographic scheme F_K, and that the scheme F_K is secure in the sense that the probability that an adversary breaks the scheme is negligible (under a uniform key); then the scheme remains secure even in the presence of the additional information that was released by the virus.

1.8 Organization of the paper

In Section 2 we give a high-level overview of the ideas and techniques used in our results. In Section 3 we give some preliminaries and state some previous classical results on approximate counting and sampling of NP witnesses. In Section 4 we construct incompressible functions and discuss the "fully interactive setting" in which compression is done by a bounded communication two-way interactive protocol. In Section 5 we construct PRGs for non-boolean circuits. In Section 6 we give a formal model of nondeterministic reductions and prove limitations on achieving negligible ϵ for pseudorandom objects by nondeterministic reductions. In section 7 we construct relative-error extractors for distributions that are recognizable by small circuits; this, in turn, gives incompressible functions with negligible ϵ and relative-error extractors for samplable distributions.

2 Overview and Technique

In this section we present a high-level overview of the techniques used to prove our results.

2.1 Boolean incompressible functions with error n^{-c}

We start with an overview of the proof of Theorem 1.11. Our goal is to construct a boolean incompressible function for size n^c circuits. Consider a family of poly(n^c)-wise independent hash functions $H = \{h_s : \{0,1\}^n \to \{0,1\}\}$. We can sample from such a family using $t = n^{O(c)}$ random bits. An easy counting argument (see e.g. Trevisan and Vadhan, 2000) shows that for every not-too-large class of distributions with min-entropy k (such as the class of distributions recognizable by size n^c circuits) a random $h_s \leftarrow H$, is with high probability an extractor for distributions in the class.

By Lemma 1.23, a random $h \leftarrow H$ is, with high probability, an (ℓ, ϵ)-incompressible function for $\ell = (1 - o(1)) \cdot n$ and negligible ϵ. We are assuming that E is hard for exponential size nondeterministic circuits, and by Theorem 1.10, there is a poly(n^t)-time computable PRG $G : \{0,1\}^n \to \{0,1\}^t$ for size $n^{O(t)}$ nondeterministic circuits (where $t = n^{O(c)}$ is the number of bits needed to specify a hash function from H). We construct an incompressible function $f : \{0,1\}^{2n} \to \{0,1\}$ as follows:

$$f(x, y) = h_{G(y)}(x).$$

Note that f is computable in polynomial time. In order to show that f is (ℓ, n^{-c})-incompressible, it is sufficient to show that for a $(1 - n^{-c}/2)$-fraction of seeds $y \in \{0,1\}^n$, $f(\cdot, y) = h_{G(y)}(\cdot)$ is $(\ell, n^{-c}/2)$-incompressible.

We will show that for $\epsilon = 1/\text{poly}(n)$, there exists a polynomial-size nondeterministic circuit P, that when given $s \in \{0,1\}^t$, accepts if h_s is not $(\ell, 2\epsilon)$-incompressible, and rejects if h_s is (ℓ, ϵ)-incompressible. A key observation is that as $\text{AM} \subseteq \text{NP/poly}$, it is sufficient to design an Arthur-Merlin protocol P, and furthermore by using the results of Babai and Moran (1988) and Goldwasser and Sipser (1986) we can allow this protocol to be a private-coin, constant-round protocol, with small (but noticeable) gap between completeness and soundness.

We now present the protocol P: Merlin (who is claiming that h_s is not $(\ell, 2\epsilon)$-incompressible) sends a circuit $C : \{0,1\}^n \to \{0,1\}^\ell$ of size n^c (which is supposed to compress the function well). Arthur, chooses private coins $x \leftarrow U_n$, and sends $C(x)$ to Merlin. Merlin responds by guessing $h_s(x)$, and Arthur accepts if Merlin guessed correctly. It is immediate that this protocol has completeness $\frac{1}{2} + 2\epsilon$ and soundness $\frac{1}{2} + \epsilon$ and the gap is large enough to perform amplification.

It follows that for a uniform y, w.h.p. $h_{G(y)}$ is 2ϵ-incompressible, as otherwise the nondeterministic circuit P distinguishes the output of G from uniform.[18]

We remark that this approach can be extended to yield nonboolean incompressible functions. However, using this approach we cannot get $\epsilon = n^{-\omega(1)}$. This is because the error of the final function f is at least the error of the PRG G, which cannot be negligible. We later present our construction of nonboolean incompressible functions with very low error (as promised in Theorem 1.16), which works by giving a construction of relative-error extractors for recognizable distributions (using quite different techniques).

The proof of Theorem 1.11 can be viewed under the general paradigm of using a PRG to derandomize a probabilistic construction. This paradigm was abstracted by Klivans and van Melkebeek (2002), and was also used in many relevant works such as Shaltiel and Umans (2006) and Artemenko and Shaltiel (2014b). However, in contrast to previous works, we strongly rely on AM protocols with *private coins*. This allows us to come up with very simple proofs that improve upon previous work. An example is our next result that improves a recent construction of Artemenko and Shaltiel (2014b).

2.2 PRGs for nonboolean distinguishers

We now give an overview of the proof of Theorem 1.12 and show how to construct PRGs against nonboolean distinguishers. The argument is similar to that of the previous section. This time we take a $\text{poly}(n^c)$-wise independent family of hash functions $H = \{h_s : \{0,1\}^{2\ell} \to \{0,1\}^n\}$. First, we show that w.h.p. a random $h_s \leftarrow H$ is an (ℓ, ϵ)-PRG with very small ϵ. Indeed, by a standard calculation, h_s is, w.h.p, a $(\epsilon \cdot 2^{-\ell})$-PRG for size n^c, and this easily implies that it is also an (ℓ, ϵ)-PRG (Artemenko and Shaltiel, 2014b). Next, we derandomize the collection using a PRG G against $\text{poly}(n^c)$-size nondeterministic circuits. Namely, our final PRG is $G'(x, y) = h_{G(y)}(x)$.

Following our earlier strategy, it is sufficient to design a private-coin, constant-round AM protocol P with noticeable gap ϵ between completeness and soundness, such that given $s \in \{0,1\}^t$, P distinguishes the case that h_s is not an $(\ell, 2\epsilon)$-PRG from the case that h_s is an (ℓ, ϵ)-PRG.

[18] Note that for this argument it is sufficient to have a PRG $G : \{0,1\}^n \to \{0,1\}^{t=n^{O(c)}}$ that has polynomial stretch. Therefore, any assumption that implies such a PRG suffices for our application, and we chose the assumption that E is hard for exponential size nondeterministic circuits, for the ease of stating it. Furthermore, it is sufficient for us that G fools *uniform* AM protocols, and we don't need to fool *nonuniform* nondeterministic circuits. There is a line of work on constructing PRGs against uniform classes under uniform assumption (Gutfreund et al., 2003; Impagliazzo and Wigderson, 2001; Shaltiel and Umans, 2009; Trevisan and Vadhan, 2007), but unfortunately, the relevant results only give hitting set generators, and using these we can only get incompressible function with $\epsilon = 1 - n^{-O(t)}$.

We now present such a protocol, that is similar in spirit to the graph non-isomorphism protocol by Goldreich et al. (1991). Merlin (who is claiming that h_s is not a good PRG) sends a circuit $C : \{0,1\}^n \to \{0,1\}^\ell$ (that is supposed to distinguish the output of h_s from random). Arthur tosses a private fair coin, and either sends $C(y)$ for $y \leftarrow U_n$, or $C(h_s(x))$ for $x \leftarrow U_{2\ell}$, depending on the value of the coin. Merlin is supposed to guess Arthur's coin. Note that if h_s is not an $(\ell, 2\epsilon)$-PRG, then the two distributions $C(U_n)$ and $C(h_s(U_{2\ell}))$ are not 2ϵ-close and Merlin can indeed guess Arthur's coin with probability $\frac{1}{2} + \epsilon$. If h_s is an (ℓ, ϵ)-PRG, then the distributions are ϵ-close and Merlin cannot distinguish with probability larger than $\frac{1}{2} + \epsilon/2$.
The precise argument appears in Sections 4 and 5. Figure 1 includes a roadmap for the approach described above.

2.3 The power and limitations of nondeterministic reductions

The precise definitions of nondeterministic reductions and formal restatement of Theorem 1.15 appear in Section 6. Below, we try to intuitively explain what makes nondeterministic reductions more powerful than deterministic reductions, and why this additional power is more helpful when constructing nonboolean functions, and less helpful when constructing boolean functions.

Recall that we observed that nondeterministic reductions can be used to achieve negligible error $\epsilon = n^{-\omega(1)}$ when constructing incomputable functions $f : \{0,1\}^n \to \{0,1\}^m$ for large m, and we want to show that they cannot achieve this for $m = 1$. A powerful tool used by several nondeterministic reductions is *approximate counting.*

Theorem 2.1 (approximate counting (Jerrum et al., 1986; Sipser, 1983; Stockmeyer, 1983))**.** *For every sufficiently large n, and every $\epsilon' > 0$ there is a size poly(n/ϵ') randomized NP-circuit that, given oracle access to a function $C : \{0,1\}^n \to \{0,1\}$, outputs with probability $1 - 2^{-n}$ an integer p which ϵ'-approximates the value $q = |\{x : C(x) = 1\}|$ in the sense that $(1 - \epsilon') \cdot p \leq q \leq (1 + \epsilon') \cdot p$.*

We want the oracle circuit above to have size poly(n), and so we can only afford $\epsilon' = n^{-c}$. Suppose that we are using approximate counting with this ϵ' on some function $C : \{0,1\}^n \to \{0,1\}$, to try and distinguish the case that $q = |\{x : C(x) = 1\}|/2^{-n}$ satisfies $q \leq 2^{-m}$ from the case that $q \geq 2^{-m} + \epsilon$, for negligible $\epsilon = n^{-\omega(1)}$. Note that an n^{-c}-approximation can indeed perform this distinguishing task if $m \geq \log(1/\epsilon)$, but it cannot do so if $m = 1$.

The reductions that we describe in the proofs of Theorems 1.16 and 1.21 construct functions with m bit outputs, and critically rely on this property. We now observe that in order to be useful for constructing functions with output length m, reductions must be able to distinguish the two cases above.

Let us focus on the task of constructing incomputable functions $f : \{0,1\}^n \to \{0,1\}^m$. Such reductions receive oracle access to a circuit $C : \{0,1\}^n \to \{0,1\}^m$, and if C computes f too well on average, the reduction needs to contradict the hardness assumption. Loosely speaking, we observe that the reduction must be able to distinguish the case that it is given a *useful* circuit C, namely one such that $\Pr_{x \leftarrow U_n}[C(x) = f(x)] \geq 2^{-m} + \epsilon$ (on which the reduction must succeed) from the case that it is given a *useless* circuit C', which ignores its input and outputs a random value, so that $\Pr_{x \leftarrow U_n}[C'(x) = f(x)] = 2^{-m}$ (and as this circuit is useless, the reduction receives no information on f, and cannot succeed).

This explains why approximate counting is in some sense *necessary* for reductions that want to achieve negligible error. In the formal proof, we use an argument similar to that of Furst et al. (1984), to show that even reductions that are Σ_i-circuits, cannot approximately count with the

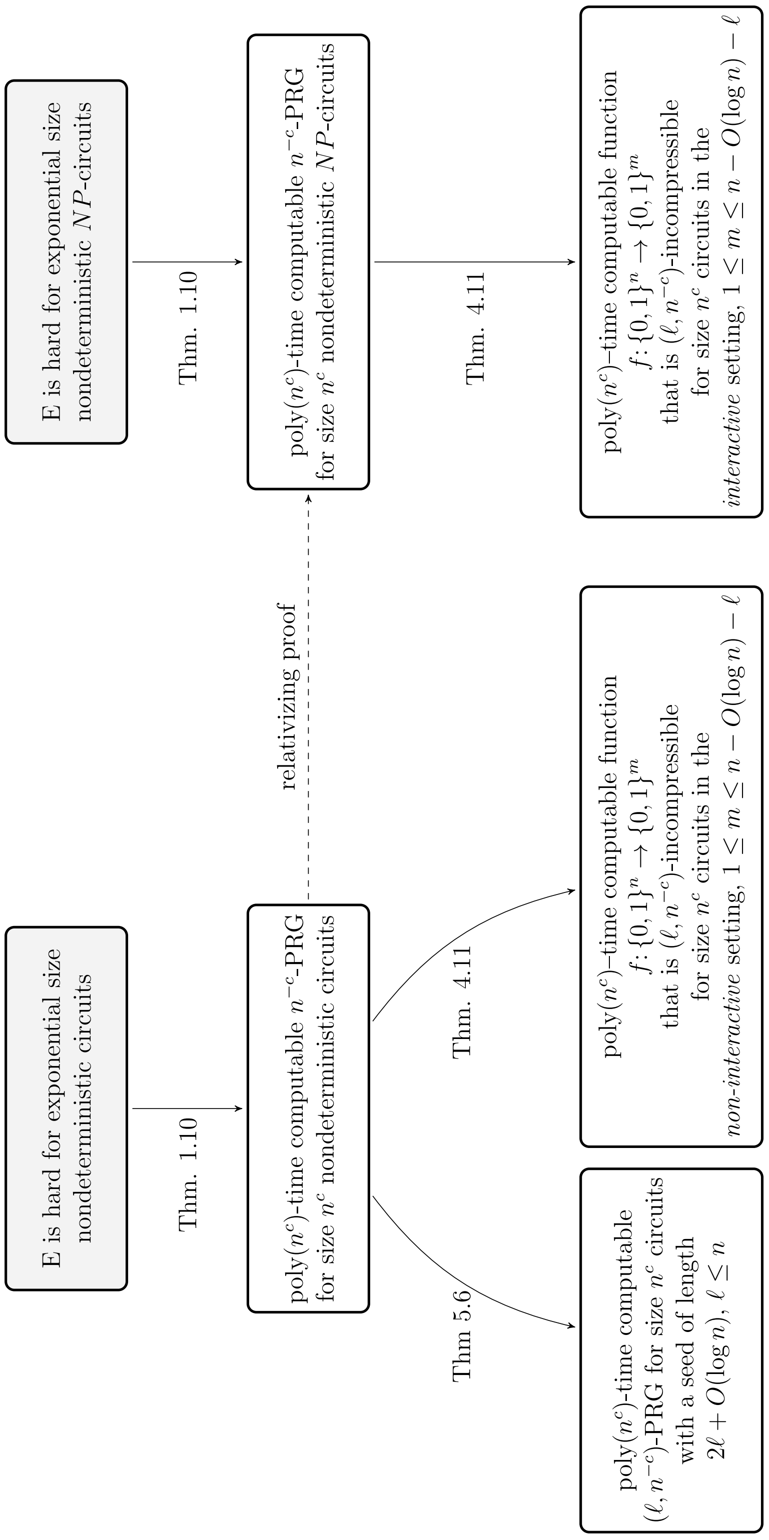
E is hard for exponential size nondeterministic circuits
Thm. 1.10
poly(n^c)-time computable n^{-c}-PRG for size n^c nondeterministic circuits
Thm 5.6
poly(n^c)-time computable (ℓ, n^{-c})-PRG for size n^c circuits with a seed of length $2\ell + O(\log n)$, $\ell \le n$
Thm. 4.11
poly(n^c)–time computable function $f\colon \{0,1\}^n \to \{0,1\}^m$ that is (ℓ, n^{-c})-incompressible for size n^c circuits in the *non-interactive* setting, $1 \le m \le n - O(\log n) - \ell$
relativizing proof
E is hard for exponential size nondeterministic NP-circuits
Thm. 1.10
poly(n^c)-time computable n^{-c}-PRG for size n^c nondeterministic NP-circuits
Thm. 4.11
poly(n^c)–time computable function $f\colon \{0,1\}^n \to \{0,1\}^m$ that is (ℓ, n^{-c})-incompressible for size n^c circuits in the *interactive* setting, $1 \le m \le n - O(\log n) - \ell$

Figure 1: Roadmap: from hardness assumptions to incompressible functions and nb-PRGs.

precision needed for distinguishing the cases above if $m = 1$. This is shown by relating the quality of such reductions to the quality of AC^0-circuits that need to perform some task (for which there are known lower bounds). This relationship uses ideas from the previous lower bounds of Shaltiel and Viola (2010).

2.4 Constructing relative-error extractors for recognizable distributions

By lemma 1.23 it is sufficient to construct relative-error extractors for recognizable distributions in order to obtain non-boolean incompressible functions with negligible error. We now explain how to construct such extractors and prove Theorem 1.24. We use tools and techniques from Trevisan and Vadhan (2000), together with some key ideas that allow us to get relative error. The full proof appears in Section 7.

It is complicated to explain the precise setting, and instead we attempt to explain what enables us to obtain relative-error. For this purpose, let us restrict our attention to the problem of constructing an ϵ-incomputable function $g : \{0,1\}^n \to \{0,1\}^m$ for $\epsilon = n^{-c} \cdot 2^{-m}$, which means that the function cannot be computed with probability larger than $(1 + n^{-c}) \cdot 2^{-m}$ on a random input.

We will start from a function that is already very hard on average, say $f : \{0,1\}^n \to \{0,1\}^{n'}$ that is ϵ-incomputable for $\epsilon = 2^{-n'/3}$ (and we indeed have such a function by Theorem 1.13 for $n' = \Omega(n)$). We want to reduce the output length of f from n' to $m \approx \log(1/\epsilon)$ while preserving ϵ. This will make ϵ small compared to 2^{-m}.

A standard way to reduce the output length while preserving security is via the "hardcore theorem" of Goldreich and Levin (1989), or, more generally, by concatenating with a "good" inner code. More precisely, it is standard to define $g(x, i) = EC(f(x))_i$ for some error-correcting code $EC : \{0,1\}^{n'} \to (\{0,1\}^m)^t$ that has sufficiently efficient list-decoding. Typically, the inner code that we use is binary (that is, $m = 1$). However, we want to choose codes with large alphabet that have extremely strong list decodability. One way to get such behavior is to use "extractor codes" (defined by Ta-Shma and Zuckerman, 2004). More precisely, one sets $g(x, i) = T(f(x), i)$ where $T : \{0,1\}^{n'} \times [t] \to \{0,1\}^m$ is a "seeded extractor". This guarantees that for every event $A \subseteq \{0,1\}^m$, there aren't "too many" x's for which $T(x, \cdot)$ lands in A with "too large probability" (this is the kind of "combinatorial list-decoding" guarantee that we are interested in). It turns out that for our application we need to replace "seeded extractors" with "2-source extractors". A useful property of 2-source extractors is that they can achieve error $\ll 2^{-m}$. In particular, if applied with error $\epsilon \ll 2^{-m}$, such extractors can be thought of as achieving "relative error" - the probability of every output string is between $2^{-m} - \epsilon = (1 - \epsilon \cdot 2^m) \cdot 2^{-m}$ and $2^{-m} + \epsilon = (1 + \epsilon \cdot 2^m) \cdot 2^{-m}$. This can be seen as a relative approximation with error $\epsilon' = \epsilon \cdot 2^m$.

We observe that such extractors can be used as "inner codes" in the approach of Trevisan and Vadhan (2000) (which can be viewed as a more specialized concatenation of codes). Precise details appear in the formal proof.

As in the case of Goldreich-Levin, these "codes" need to have efficient "list-decoding procedures". In this setup "efficient" means: a list-decoding procedure implementable by a polynomial-size NP-circuit. In order to obtain such a list-decoding procedure (for very small ϵ) we critically use that approximate counting can indeed distinguish 2^{-m} from $2^{-m} + \epsilon$ for negligible ϵ using a noticeable approximation precision $\epsilon' = n^{-c}$, as explained in Section 2.3.

2.5 Relative-error extractors for samplable distributions

We now explain how to construct relative-error extractors for samplable distributions and prove Theorem 1.21. In this high-level overview, let us restrict our attention to samplable distributions that are flat, that is uniform over some subset $S \subseteq \{0,1\}^n$. Let X be such a distribution, and let $C : \{0,1\}^t \to \{0,1\}^n$ be a circuit that samples X (that is $X = C(U_t)$). It immediately follows that X is recognizable by the NP-circuit that given x accepts iff there exists $y \in \{0,1\}^t$ such that $C(y) = x$. This means that it suffices to construct a relative-error extractor for distributions recognizable by NP-circuits. This follows from Theorem 1.24 in an analogous manner, if in the assumption we assume hardness for Σ_4-circuits, instead of Σ_3-circuits. This follows by observing that the proof of Theorem 1.24 relativizes. The argument sketched above gives an extractor for flat samplable distributions. In order to extend this to distributions that are not flat, we generalize the notion of recognizable distributions to non-flat distributions and then Theorem 1.21 follows from the (generalized version) of Theorem 1.24.

Our constructions of relative error extractors appear in Section 7. Figure 2 includes a roadmap for the approach used to construct relative error extractors and consequences.

3 Preliminaries

We use the classical result on approximate counting and uniform sampling of NP-witnesses (Bellare et al., 2000; Jerrum et al., 1986; Sipser, 1983; Stockmeyer, 1983), which we now state in a way that is convenient for our application.

Definition 3.1 (relative approximation)**.** We say that a number p is an ϵ-relative approximation to q if $(1-\epsilon) \cdot p \leq q \leq (1+\epsilon) \cdot p$.

It is useful to note that if p is an ϵ-approximation to q, then q is a 2ϵ-approximation to p. If p is an ϵ-approximation to q and q is an ϵ-approximation to w, then p is 2ϵ-approximation to w. If p' is an ϵ-approximation to p and q' is an ϵ-approximation to q, then a p'/q' is a 2ϵ-approximation to p/q. (The last property does not hold if we replace relative approximations with additive approximations). In particular, this means that even if only want an additive approximation of some quantity $a = p/q$, then it is sufficient to have relative approximations to p and q, whereas an additive approximation does not suffice.

Theorem 3.2 (approximate counting [Jerrum et al. (1986); Sipser (1983); Stockmeyer (1983)])**.** *For every $i \geq 0$, every sufficiently large s and every $\epsilon > 0$, there is a Σ_{i+1}-circuit of size poly(s/ϵ) that given a Σ_i-circuit C of size s outputs an ϵ-approximation of $|\{x : C(x) = 1\}|$.*

Theorem 3.3 (uniform sampling [Bellare et al. (2000); Jerrum et al. (1986)])**.** *For every $i \geq 0$, every sufficiently large s and every $\delta > 0$, there is a randomized Σ_{i+1}-circuit A of size poly$(s/\log(1/\delta))$ that given a Σ_i-circuit $C\colon \{0,1\}^n \to \{0,1\}$ of size s outputs a value in $\{0,1\}^n \cup \perp$ such that for every size s Σ_i-circuit, $\Pr[A(C) = \perp] \leq \delta$ and the distribution $(A(C) | A(C) \neq \perp)$ is uniform over $\{x : C(x) = 1\}$.*

4 Incompressible Functions from Hard Functions

Outline. Our construction is based on a simple three-step approach. First, we construct a collection of efficiently computable functions $\mathcal{H}$ that most of its members are incompressible functions

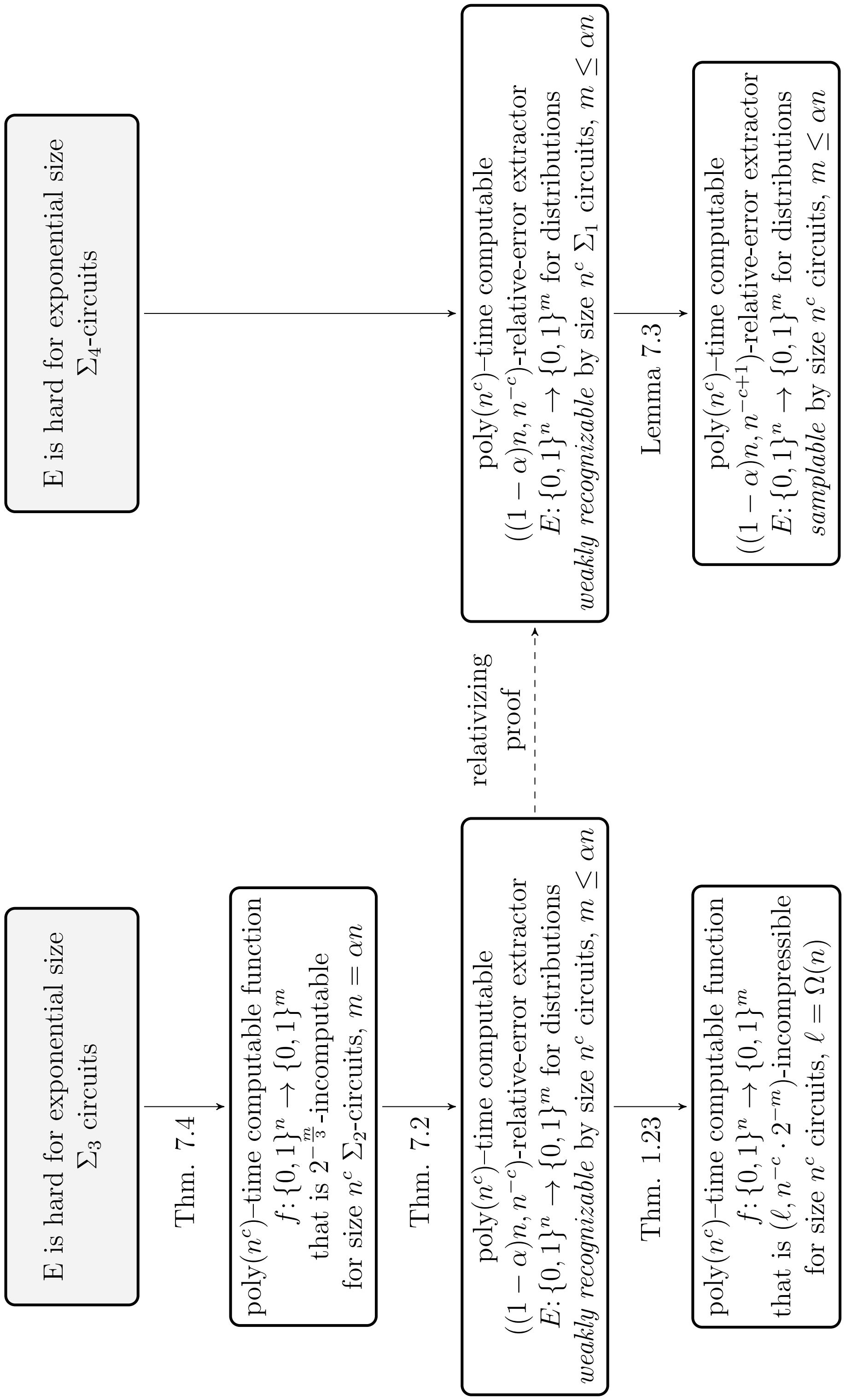
E is hard for exponential size
Σ_3 circuits
Thm. 7.4
poly(n^c)–time computable function
$f\colon\{0,1\}^n \to \{0,1\}^m$
that is $2^{-\frac{m}{3}}$-incomputable
for size n^c Σ_2-circuits, $m = \alpha n$
Thm. 7.2
poly(n^c)–time computable
$((1-\alpha)n, n^{-c})$-relative-error extractor
$E\colon\{0,1\}^n \to \{0,1\}^m$ for distributions
weakly recognizable by size n^c circuits, $m \le \alpha n$
Thm. 1.23
poly(n^c)–time computable function
$f\colon\{0,1\}^n \to \{0,1\}^m$
that is $(\ell, n^{-c} \cdot 2^{-m})$-incompressible
for size n^c circuits, $\ell = \Omega(n)$
relativizing
proof
E is hard for exponential size
Σ_4-circuits
poly(n^c)–time computable
$((1-\alpha)n, n^{-c})$-relative-error extractor
$E\colon\{0,1\}^n \to \{0,1\}^m$ for distributions
weakly recognizable by size n^c Σ_1 circuits, $m \le \alpha n$
Lemma 7.3
poly(n^c)–time computable
$((1-\alpha)n, n^{-c+1})$-relative-error extractor
$E\colon\{0,1\}^n \to \{0,1\}^m$ for distributions
samplable by size n^c circuits, $m \le \alpha n$

Figure 2: Roadmap: from hardness assumptions to relative error extractors and incompressible (nonboolean) functions with negligible error.

against s-size circuits. This collection is efficiently computable but large (contains $2^{\mathrm{poly}(s)}$ members). This step is based on a simple connection between incompressible functions and extractors against recognizable distributions, and on the fact that t-wise independent hash functions are good extractors. At the second step, we reduce the size of the collection to poly(s). This partial derandomization is based on the observation that good functions in the large collection $\mathcal{H}$ can be identified by poly(s)-size nondeterministic circuits, and so one can sub-sample functions from the large collection via an appropriate (standard) NW-PRG. Finally, we note that collections of incompressible functions $\mathcal{F}$ can be easily combined into a single incompressible function while increasing the input length by $\log |\mathcal{F}|$. For small collections of size poly(s), this leads to a minor logarithmic loss in the parameters.

Interactive compressibility. We begin by extending the notion of incompressibility to the interactive setting in which the compressor C is allowed to interact with an unbounded solver D in (arbitrarily) many rounds. As in the non-interactive setting we assume that C is a circuit of size s and restrict the total communication from C to D to be at most ℓ bits. We do not restrict the communication from D to C, though it is implicitly restricted by the circuit size of C. It will be convenient to think of such an interactive compression protocol (C, D) as an s-size circuit C with oracle gates to a *stateful* oracle $D : \{0,1\}^* \to \{0,1\}^*$, with the restriction that the total bit-length of all calls to D is upper-bounded by ℓ. Formally, a stateful oracle is defined by an initial state M_0 and a mapping $D : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^* \times \{0,1\}^*$ which takes a (current) state M and a query q and generates a new state M' and an output z. Since the oracle is computationally unbounded we may assume, without loss of generality, that the state M consists of the full history of all queries that were asked so far.

Definition 4.1 (Incompressible function in the interactive setting)**.** We say that $f : \{0,1\}^n \to \{0,1\}^m$ is (ℓ, ϵ)-incompressible by s-size circuits in the **interactive setting** if for every s-size oracle-aided circuit C which sends to its oracle a total number of ℓ bits, and every *stateful* oracle D, we have that $\Pr_{x \leftarrow U_n}[C^D(x) = f(x)] \leq \frac{1}{2^m} + \epsilon$.

Obviously, interactive incompressibility implies non-interactive incompressibility (with the same parameters). Interactive compressibility was first presented by Dell and van Melkebeek (2014) (for NP-hard problems) and was further studied for depth-limited compressors by Chattopadhyay and Santhanam (2012); Oliveira and Santhanam (2015).

4.1 Incompressible functions from extractors

We proceed by relating incompressible functions to extractors for recognizable distributions. The following lemma extends Lemma 1.23 to the interactive setting.

Lemma 4.2 (Incompressible functions from Extractors)**.** *For every integers $m < \ell < n$ and s, and every real $\epsilon \in [0,1]$ the following holds. If $f : \{0,1\}^n \to \{0,1\}^m$ is an $(n - \ell - \Delta_1, \epsilon')$ extractor for s'-size recognizable distributions, then f is (ℓ, ϵ)-incompressible by s-size circuits in the interactive setting where $\Delta_1 = 1 + \log(1/\epsilon)$, $\epsilon' = \epsilon/2$ and $s' = 2s$. Moreover, this holds even if f is an $(n - \ell - \Delta_1, \epsilon' 2^m)$-relative-error extractor for s'-size recognizable distributions.*

Concretely for any constant $c > 0$, we get that an $(n - \ell - c \log n - 1, n^{-c}/2)$ extractor for $2n^c$-size recognizable distributions is (ℓ, n^{-c})-incompressible by n^c-size circuits.

Proof. Let $f : \{0,1\}^n \to \{0,1\}^m$ be an $(n-\ell-\Delta_1, \epsilon')$ extractor for s'-size recognizable distributions or $(n-\ell-\Delta_1, \epsilon' 2^m)$-relative-error extractor for s'-size recognizable distributions. Assume, towards a contradiction, that there exists an s-size interactive compressor C that makes q calls to an unbounded solver D with total communication of ℓ, such that $\Pr_x[C^D(x) = f(x)] \geq 2^{-m} + \epsilon$. A *transcript* $a = (a_1, \ldots, a_q) \in \{0,1\}^\ell$ is a concatenation of all the messages sent by C to the solver oracle D. Such a transcript uniquely defines the answers $(b_1, \ldots, b_q)$ of the solver as well as the final output of the protocol y. The compressor C (together with D) defines a mapping ρ from an input $x \in \{0,1\}^n$ to a transcript $a \in \{0,1\}^\ell$, and so a random choice of $x \leftarrow \{0,1\}^n$ induces a probability distribution over the transcripts. Let $w(a) = \Pr_x[\rho(x) = a]$ denote the weight of a transcript a, and $\sigma(a)$ denote the conditional success probability $\Pr_x[C^D(x) = f(x) | \rho(x) = a]$.

We first claim there exists a *good* transcript a for which both $w(a) \geq \epsilon \cdot 2^{-\ell-1}$ and $\sigma(a) \geq 2^{-m} + \epsilon/2$. Indeed, we can write

$$2^{-m} + \epsilon \;\leq\; \Pr_x[C^D(x) \;=\; f(x)] \;=\; \sum_{a: w(a) < \epsilon \cdot 2^{-\ell-1}} w(a) \cdot \sigma(a) + \sum_{a: w(a) \geq \epsilon \cdot 2^{-\ell-1}} w(a) \cdot \sigma(a). \quad (4.3)$$

As the first summand is upper-bounded by $\epsilon \cdot 2^{-\ell-1} \cdot 2^\ell \leq \epsilon/2$, the second summand must be at least $2^{-m} + \epsilon/2$, and so the claim follows by an averaging argument.

Let us fix a good transcript $a = (a_1, \ldots, a_q) \in \{0,1\}^\ell$, let $b = (b_1, \ldots, b_q)$ be the corresponding answers of the decoding oracle, and let y be the final outcome of C^D. Consider the uniform distribution X over n-bit strings which are mapped (by ρ) to the transcript a. Since $w(a) \geq \epsilon \cdot 2^{-\ell-1}$, the min-entropy of X is at least $k = n - \ell - 1 - \log(1/\epsilon)$. Furthermore, X is recognizable by a circuit C' of size $s' \leq 2s$. (Indeed, compute $C(x)$, with the answers b hardwired, and accept if all the queries to D are consistent with the transcript a, i.e., if $C_1(x) = a_1$, and $C_i(x, b_1, \ldots, b_{i-1}) = a_i$ for every $1 < i \leq q$ where C_i denotes th sub-circuit that computes the i-th query based on the input x and the answers of the previous queries). Finally, we have, by assumption, that $\Pr[f(X) = y] \geq 2^{-m} + \epsilon/2$, and so f is neither (k, ϵ') extractor nor $(k, \epsilon' 2^m)$-relative-error extractor for s'-recognizable sources, in contradiction to our hypothesis. □

4.2 Ensembles of extractors

Our next goal is to construct a polynomial-time computable ensemble of functions almost all of whose members are extractors for recognizable distributions. For this purpose we will make use of t-wise independent hashing.

Definition 4.4 (t-wise independent hashing)**.** A collection of functions $\mathcal{H} = \{h_z : \{0,1\}^n \to \{0,1\}^m\}$ which is indexed by τ-bit identifiers $z \in \{0,1\}^\tau$ is called **t-wise independent hash function** if for every t distinct strings $x_1, \ldots, x_t$ the random variables $h_z(x_1), \ldots, h_z(x_t)$ induced by a uniformly random choice of $z \leftarrow \{0,1\}^\tau$ are uniformly distributed in $(\{0,1\}^m)^t$.

We will make use of *efficiently computable* families of t-wise independent hash functions. That is, we assume that there exists a poly(n)-time evaluation algorithm $H : \{0,1\}^\tau \times \{0,1\}^n \to \{0,1\}^m$ which takes an identifier $z \in \{0,1\}^\tau$ and an input $x \in \{0,1\}^n$ and outputs $y = h_z(x)$. It is well known that for every polynomial $t(n) \in \text{poly}(n)$ efficient families of t-wise independent hash functions exists (e.g., based on Reed-Solomon codes).

We say that $\mathcal{H} = \{h_z : \{0,1\}^n \to \{0,1\}^m\}$ is a collection of (k, ϵ) extractors for a family of sources $\mathcal{C}$ with a failure probability δ if

$$\Pr_z[h_z \text{ is a } (k,\epsilon) - \text{extractor for } \mathcal{C}] > 1 - \delta.$$

Collections of (k, ϵ) incompressible functions (or PRGs) are defined analogously. Trevisan and Vadhan (2000) show that t-wise independent hash functions $\mathcal{H}$ form a good collection of relative-error extractors against any fixed family $\mathcal{C}$ of N sources, provided that $t = O(n + \log N)$. (See Theorem 1.20.) Since the number of s-size circuits is $N \leq 2^{2s \log s + 5s}$, and since there are families of t-wise independent functions computable by (uniform) circuits of size $T = \text{poly}(t, n)$ (say quadratic in nt) we derive the following proposition.

Proposition 4.5. *Let $s = s(n) = \omega(n)$ be some polynomially-bounded function, $\epsilon = 1/s$, and let $k = k(n) \leq n$ be some entropy bound which is lower-bounded by $\Delta_2 = 3 \log s - \log \log s + \Omega(1)$. Then, there exists a collection $\mathcal{H} = \{h_z : \{0,1\}^n \to \{0,1\}^m\}$ with $m = k - \Delta_2$ of (k, ϵ)-(relative-error)-extractors for s-size recognizable sources with failure probability $\delta = 2^{-n}$. Furthermore, $\mathcal{H}$ is computable by an evaluation algorithm $H : \{0,1\}^\tau \times \{0,1\}^n \to \{0,1\}^m$ of complexity poly(s).*

4.3 Partial derandomization: obtaining a small collection

From now on, we fix some constant $c > 1$, and some arbitrary function $\ell(n) \leq n - \Omega_c(\log n)$ and strive for an (ℓ, n^{-c})-incompressible for n^c-size compressors. Concretely, we let $\Delta_1 = 1 + c \log n$ as in Lemma 4.2 and $\Delta_2 = 3c \log n - \log \log n + \Omega(1)$ as in Proposition 4.5 instantiated with $s = 2n^c$, and require that $\ell(n) \leq n - (\Delta_1 + \Delta_2) = n - \Omega_c(\log n)$. We also let $k = n - \ell - \Delta_1$ be an entropy bound, and $m = k - \Delta_2 = n - \ell - (\Delta_1 + \Delta_2) = n - \ell - \Omega_c(\log n)$ be the desired output length.

Let $H_{c,\ell} : \{0,1\}^\tau \times \{0,1\}^n \to \{0,1\}^m$ be the evaluation algorithm that satisfies Proposition 4.5 with respect to the parameters s and k (defined above), and note that by Lemma 4.2, all but 2^{-n} fraction of the functions in $\mathcal{H}$ are (ℓ, n^{-c})-incompressible for n^c-size compressors. Consider the promise problem $\Pi_{c,\ell}$ whose YES instances are the $\tau(n)$-bit strings z for which $H(z, \cdot)$ is $(\ell, 2n^{-c})$-*compressible* by n^c-size circuits in the non-interactive setting, and the NO instances $\tau(n)$-bit strings z for which $H(z, \cdot)$ is (ℓ, n^{-c})-*incompressible* by n^c-size circuits in the non-interactive setting. (The interactive setting will be discussed later.)

Claim 4.6. *There exists a nondeterministic circuit of a size n^b which accepts the YES instances of $\Pi_{c,\ell}$ and rejects the NO instances of $\Pi_{c,\ell}$, where b is a constant which depends on c.*

Proof. We prove a stronger statement, namely that $\Pi_{c,\ell}$ is in AM. Consider the following interactive proof system. On joint input z, Merlin sends a description of an n^c-size (non-interactive) compressing circuit $C : \{0,1\}^n \to \{0,1\}^\ell$, Arthur samples a random string $x \leftarrow \{0,1\}^n$ and sends to Merlin $C(x)$, who responds with $y \in \{0,1\}^m$. Arthur accepts if $y = H_{c,\ell}(z, x)$. Clearly, if z is a YES instance then Merlin can make Arthur accept with probability $1/2 + 2n^{-c}$, whereas on NO instance the acceptance probability is upper-bounded by $\frac{1}{2} + n^{-c}$. Using standard amplification techniques together with the transformations of Goldwasser and Sipser (1986) and Babai and Moran (1988), we get that $\Pi_{c,\ell}$ has a two-message public-coin AM protocol, which, in turn, means that $\Pi_{c,\ell}$ has a nondeterministic circuit of poly(n)-size. □

Recall that, by Proposition 4.5 and Lemma 4.2, all but 2^{-n} fraction of the functions in $\mathcal{H}$ are (ℓ, n^{-c})-incompressible for n^c-size compressors. Hence, all but a 2^{-n} fraction of the τ-bit strings z are NO-instances. As a result we obtain the following proposition.

Proposition 4.7. *Let $G : \{0,1\}^r \to \{0,1\}^\tau$ be an n^{-b}-PRG for nondeterministic circuits of size n^b where $b > c$ is the constant from Claim 4.6. Consider the algorithm $f : \{0,1\}^r \times \{0,1\}^n \to \{0,1\}^m$ where*

$$f(w,x) = H_{c,\ell}(G(w),x).$$

Then, the ensemble $\mathcal{F} = \{f(w,\cdot) : \{0,1\}^n \to \{0,1\}^m\}_{w\in\{0,1\}^r}$ is a collection of (ℓ, n^{-c})-incompressible functions for n^c-size circuits (in the non-interactive setting) with failure probability $2^{-n} + n^{-2c}$.

The interactive setting. We would like to prove an analogous statement for incompressible functions in the interactive setting. However, we do not know how to recognize compressible functions with few alternations. (Note that a straightforward generalization of Claim 4.6 yields an interactive proof with polynomially many rounds of interaction.) An easier task is to identify extractors for recognizable distributions. In fact, we will make use of the fact that it suffices to recognize extractors with relative error.

Concretely, let $\Pi'_{c,\ell}$ denote the promise problem whose YES instances are the $\tau(n)$-bit strings z for which $H(z,\cdot)$ is *not* $(k+1, 2\epsilon)$-relative-error extractors for s-size recognizable sources, and the NO instances are τ-bit strings for which $H(z,\cdot)$ is (k,ϵ)-relative-error extractors for s-size recognizable sources. We prove the following claim.

Claim 4.8. *There exists a nondeterministic NP-circuit A of size n^b that accepts the YES instance of $\Pi'_{c,\ell}$ and rejects its NO instances, where b is a constant that depends on c.*

Proof. The circuit A takes a string z as an input, and a witness (C,y) where $C : \{0,1\}^n \to \{0,1\}$ is an s-size circuit and $y \in \{0,1\}^m$ is a string. The circuit A will test whether C recognizes a high-entropy distribution, and that the quantity $p = \Pr_x[H(z,C(x)) = y]$ is far enough from 2^{-m}. Both checks will be implemented by using the NP-oracle to approximate the number of satisfiable assignments of a poly(T,s)-size circuit, where T is the time complexity of the evaluation algorithm of $\mathcal{H}$. (Recall that both T and s are polynomials in n^c.)

Formally, A will perform the following checks: (1) Ask for a $(1 \pm 1/4)$ multiplicative approximation of the number of assignments that satisfy C, and check that the result is at least $1.5 \cdot 2^k$; (2) Ask for a $(1 \pm \epsilon^2)$ multiplicative approximation of the number of assignments x that satisfy $H(z,C(x)) = y$, and check that the result is *not* in the interval $2^{n-m} \cdot (1 \pm 1.5\epsilon)$. Such an approximation is implementable with an NP oracle (Stockmeyer, 1983) with circuit complexity of poly$(T,s,1/\epsilon) = \text{poly}(n^c)$. (See Theorem 3.2.)

The analysis is straightforward. First, assume that z is a yes instance. Then there exists a witness (C,y) such that: (a) C recognizes a distribution with min-entropy $k+1$; and (b) $p = \Pr_x[H(z,C(x)) = y] \notin (1 \pm 2\epsilon)2^{-m}$. Part (a) means that the set $C^{-1}(1)$ is larger than 2^{k+1} and so the approximated value must be larger or equal to $1.5 \cdot 2^k$, and the first check will go through. Similarly, (b) means that the number of x's which satisfy $H(z,C(x)) = y$ is either less than $2^{n-m}(1-2\epsilon)$ or larger than $2^{n-m}(1+2\epsilon)$. In the first case, the approximated value will be at most $2^{n-m}(1-2\epsilon)(1+\epsilon^2) < 2^{n-m} \cdot (1-1.5\epsilon)$ for $\epsilon < 1$. In the second case, the approximated value will be at least $2^{n-m}(1+2\epsilon)(1-\epsilon^2)$ which is larger than $2^{n-m} \cdot (1+1.5\epsilon)$ for $\epsilon < 0.3$. (Recall that $\epsilon = o(1)$ in our setting.) Hence, in both cases the second check passes, and A accepts.

On the other hand, it is not hard to show that a NO instance z is always rejected. Indeed, fix a potential witness (C,y) if C passes the first check then it must sample a source with at least k bits of min-entropy. For such a source we know that $p = \Pr_x[H(z,C(x)) = y] \in [2^{-m}(1-\epsilon), 2^{-m}(1+\epsilon)]$

and so the approximation computed in the second part of the test must be in the interval

$$[2^{n-m}(1-\epsilon)(1-\epsilon^2), 2^{n-m}(1+\epsilon)(1+\epsilon^2)] \subset [2^{n-m}(1-1.5\epsilon), 2^{n-m}(1+1.5\epsilon)],$$

which means that the second check fails. This completes the proof of the claim. □

By construction, all but a 2^{-n} fraction of the τ-bit strings z are NO-instances of $\Pi'_{c,\ell}$. Furthermore, by Lemma 4.2, for each of these strings, the function $H(z,\cdot)$ is (ℓ, n^{-c})-*incompressible* by n^c-size circuits in the interactive setting. As a result we obtain the following proposition (which is analogous to Proposition 4.7).

Proposition 4.9. *Let $G : \{0,1\}^r \to \{0,1\}^\tau$ be an n^{-b}-PRG for nondeterministic NP-circuits of size n^b, where $b > c$ is the constant from Claim 4.8. Consider the algorithm $f : \{0,1\}^r \times \{0,1\}^n \to \{0,1\}^m$ where*

$$f(w,x) = H_{c,\ell}(G(w), x).$$

Then, the ensemble $\mathcal{F} = \{f(w,\cdot) : \{0,1\}^n \to \{0,1\}^m\}_{w\in\{0,1\}^r}$ is a collection of (ℓ, n^{-c})-incompressible functions for n^c-size circuits in the interactive setting with failure probability $2^{-n} + n^{-2c}$.

4.4 From a small collection to a single function

Finally, we need the following simple observation.

Claim 4.10 (Combining incompressible functions)**.** *Let*

$$\mathcal{F} = \{f_z : \{0,1\}^n \to \{0,1\}^m\}_{z\in\{0,1\}^\tau}$$

be a collection of (ℓ,ϵ)-incompressible functions for s-size circuits with failure probability δ, and let $f(z,x)$ be the evaluator of $\mathcal{F}$. Then the function $f(z,x)$ viewed as a single-input function over $(\tau+n)$-bit strings is $(\ell, \epsilon+\delta)$-incompressible for s-size circuits. Furthermore, this holds both in the interactive and non-interactive setting.

Proof. Assume, towards a contradiction, that f is compressible by an s-size circuit C with communication of ℓ-bits and solver D with success probability larger than $2^{-m} + \epsilon + \delta$. Then, for more than δ fraction of the strings $z \in \{0,1\}^\tau$ it holds that

$$\Pr_{x\leftarrow\{0,1\}^n}[C^D(z,x) = f(z,x)] \geq 2^{-m} + \epsilon.$$

Observe that for each of these z's, the function f_z is (ℓ,ϵ)-compressed by $C(z,\cdot)$ (with respect to solver D). It follows that more than a δ fraction of the f_z are (ℓ,ϵ)-compressible, in contradiction to our assumption. □

We can now prove a stronger variant of Theorem 1.11.

Theorem 4.11. *If E is hard for exponential size nondeterministic circuits (resp., nondeterministic NP-circuits), then for every constant $c > 1$ there exists a constant $d > 1$ such that for every sufficiently large n and every $\ell < n - d\log n$, there is a function $f : \{0,1\}^n \to \{0,1\}^{n-\ell-d\log n}$ that is (ℓ, n^{-c})-incompressible for size n^c circuits in the non-interactive setting (resp., in the interactive setting). Furthermore, f is computable in time poly(n^c).*

Proof. We will prove that for every constant $c > 1$ and every integer-valued function $\ell(n) \leq n - \Omega_c(\log n)$ there exists an efficiently computable function $f\colon \{0,1\}^{n+\Omega_c(\log n)} \to \{0,1\}^{n-\ell-\Omega_c(\log n)}$ which is $(\ell, 2n^{-c})$-incompressible for n^c-size circuits in the non-interactive setting (resp., in the interactive setting). The theorem then follows by renaming the input length and by starting with a larger constant, e.g., $c' = c + 1$.

To prove the above statement, fix some constant $c > 1$, and use the parameters in the previous subsection. That is, let $\ell(n) \leq n - (\Delta_1 + \Delta_2) = n - 4c \log n - \log\log n + \Omega(1)$, and let $m = n - \ell - 4c \log n - \log\log n + \Omega(1)$. Let $H_{c,\ell} : \{0,1\}^\tau \times \{0,1\}^n \to \{0,1\}^m$ be the collection defined in the previous section and let $b(c) > c$ be the corresponding constant from Claim 4.6 (resp., Claim 4.8). By Theorem 1.10, our hypothesis implies the existence of n^{-b}-PRG $G : \{0,1\}^r \to \{0,1\}^{\tau(n)}$ against nondeterministic circuits (resp., nondeterministic NP-circuits) of size n^b, where $r = a \log n$ for some constant a which depends on c. By Proposition 4.7 (resp., Proposition 4.9) and Claim 4.10 the function $f : \{0,1\}^{r+n} \to \{0,1\}^m$ defined via the mapping $(w, x) \mapsto H(G(w), x)$ is $(\ell, n^{-c} + n^{-b})$-incompressible for n^c-size circuits in the non-interactive setting (resp., in the interactive setting). □

5 Constructing non-boolean PRGs

In this section we prove Theorem 1.12 by following the outline sketched in Section 2. We begin with a simple observation.

Proposition 5.1. *If G is an $(1, 2^{-\ell}\epsilon)$-PRG for size s circuits then it is also an (ℓ, ϵ)-PRG for size s circuits.*

A standard probabilistic argument shows that a function h which is randomly chosen from a t-wise collection of hash functions $\mathcal{H}$ is a good ϵ-PRGs against any s-size circuits with probability $1 - \delta$, where $t = \Omega(s \log s + \log(1/\delta))$.

Claim 5.2 (Collections of PRGs from Hashing)**.** *For every s and $\epsilon, \delta \in [0,1]$ a family of t-wise independent hash functions $\mathcal{H} = \{h_z : \{0,1\}^r \to \{0,1\}^n\}$ is a collection of ϵ-PRG against s-size circuits with failure probability δ, where $t = 4s \log s + 2\log(1/\delta)$ and $r = 2\log(1/\epsilon) + \log t$.*

Proof. Fix an s-size distinguisher C and let $\mu = \Pr[C(U_n) = 1]$. For every fixed string $x \in \{0,1\}^r$ let v_x be the random variable which takes the value 1 if $h(x) \in C^{-1}(1)$ where the probability is taken over a choice of a random $h \leftarrow \mathcal{H}$. Note that the expectation of v_x is μ and that the random variables $\{v_x\}_{x\in\{0,1\}^r}$ are t-wise independent. Applying a tail inequality for sums of t-wise independent variables from (Bellare and Rompel, 1994, Lemma 2.2), we have

$$\Pr_h\left[|\Pr[C(h(U_r)) = 1] - \Pr[C(U_n) = 1]| > \epsilon\right] = \Pr_h[|(2^{-r} \cdot \sum_x v_x) - \mu| > \epsilon] \leq \left(\frac{t}{\epsilon^2 2^r}\right)^{t/2} \leq \delta/N, \tag{5.3}$$

where $N = 2^{2s \log s + 5s}$ upper-bounds the number of all s-size circuits. The claim now follows by taking a union-bound over all distinguishers of size s. □

We derive the following corollary.

Corollary 5.4. *For every constant $c > 1$ and function $\ell(n)$ the following holds. With probability $1 - n^{-c}$, a random $t = n^{c+1}$-wise independent hash function $h : \{0,1\}^r \to \{0,1\}^n$ with input length of $r = 2\ell + (2c+1)\log n$ forms an (ℓ, n^{-c})-PRG against n^c-size circuits.*

For some $c > 1$ and ℓ, let $\mathcal{H}_{c,\ell}$ denote an efficiently computable hash function which satisfies the corollary, and let H be its evaluation algorithm whose complexity is n^b for some constant $b = b(c)$. Define a promise problem $\Pi_{c,\ell}$ whose YES instances are the strings z for which $H(z,\cdot)$ is *not* an $(\ell, 2n^{-c})$-PRG against n^c-size circuits, and the NO instances are the strings z for which $H(z,\cdot)$ is (ℓ, n^{-c})-PRG against n^c-size circuits.

Claim 5.5. *There exists a nondeterministic circuit of size $n^{c'}$ which accepts the YES instances of $\Pi_{c,\ell}$ and rejects the NO instances of $\Pi_{c,\ell}$, where c' is a constant which depends on c.*

Proof. We prove a stronger statement, namely that $\Pi_{c,\ell}$ is in AM. Consider the following interactive proof system. On joint input z, Merlin sends a description of an n^c-size nonboolean distinguisher $C : \{0,1\}^n \to \{0,1\}^\ell$, Arthur samples a pair of strings $y_0 = C(U_n)$ and $y_1 = C(H(z, U_r))$, tosses a coin $\sigma \leftarrow \{0,1\}$ and sends to Merlin the sample y_σ. Merlin guesses a bit σ' and Arthur accepts if $\sigma = \sigma'$. It is not hard to verify that YES instances are accepted with probability $\frac{1}{2} + n^{-c}$ and NO instances are accept with probability at most $\frac{1}{2} + n^{-c}/2$. Using standard amplification techniques together with the Goldwasser and Sipser (1986) andBabai and Moran (1988) transformations, we get that Π has a two-message public-coin AM protocol, which, in turn, means that Π has a nondeterministic circuit of poly(n)-size. □

We can now prove Theorem 1.12 (restated here in a stronger form).

Theorem 5.6. *If E is hard for exponential size nondeterministic circuits, then for every constant $c > 1$ there exists a constant $a > 1$ such that for every sufficiently large n, and every $\ell \leq n$, there is a function $G : \{0,1\}^{2\cdot\ell + a\log n} \to \{0,1\}^n$ that is an (ℓ, n^{-c})-PRG for size n^c circuits. Furthermore, G is computable in time poly(n^c).*

Proof. Fix some constant $c > 1$, let a_c be a constant whose value will be determined later and let ℓ be an arbitrary function which satisfies $\ell \leq n$. Let $H_{c,\ell} : \{0,1\}^\tau \times \{0,1\}^r \to \{0,1\}^n$ be the collection defined above, where $r = 2\ell + (2c+1)\log n$. Let $\Pi_{c,\ell}$ be the corresponding promise problem which, by Claim 5.5, is recognizable by a nondeterministic circuit of size $n^{c'}$ where c' depends on c. By Thm. 1.10, our hypothesis implies the existence of (standard) $n^{-c'}$-PRG $G' : \{0,1\}^{r'} \to \{0,1\}^{\tau(n)}$ against nondeterministic circuits of size $n^{c'}$, where $r' = a' \log n$ for some constant a' which depends on c' (and therefore depends on c). Consider the function $G : \{0,1\}^{r'} \times \{0,1\}^r \to \{0,1\}^n$ defined by

$$G(w, x) = H_{c,\ell}(G'(w), x).$$

Recall that, by Corollary 5.4, a random w is a NO instance of Π with probability $1 - n^{-c}$. It follows that, for $1 - 2n^{-c}$-fraction of the w's, we have that $G(w,\cdot)$ is (ℓ, n^{-c})-PRG against n^c-size circuits. Therefore, G is a $(\ell, 3n^{-c})$-PRG. Overall, G has an input length of $r + r' = 2\ell + (2c+1)\log n + a' \log n$ which, for some constant a_c, simplifies to $2\ell + a_c \log n$. The theorem follows. □

6 Limitations on nondeterministic reductions

In this section we formally state and prove the limitation stated loosely in Theorem 1.15. Theorem 1.15 discusses several different objects: incomputable functions, incompressible functions, pseudorandom generators (for boolean or nonboolean distinguishers) and extractors for samplable distributions. Since incompressible functions are incomputable, and PRGs for nonboolean distinguishers are also standard PRGs, we do not need to discuss them separately.

We start by proving limitations on the derivation of incomputable functions from more weakly incomputable functions (as this task, known as "hardness amplification", is extensively studied with many previous results showing limitations). In Sections 6.2 we extend our approach to limitations on pseudorandom generators. In Section 6.3 we explain how to extend our approach to extractors for samplable distributions.

6.1 Black-box hardness amplification and nondeterministic reductions

6.1.1 Formal definition of black-box hardness amplification

We start by considering limitations on obtaining incomputable functions from the assumption that E is hard for exponential size Σ_i-circuits. The task of converting a worst-case hard function f into an average-case hard function g is called "hardness amplification".

To the best of our knowledge, all proofs of such results in the literature work by presenting two components: A *construction* which specifies how to transform a given function $f : \{0,1\}^k \to \{0,1\}$ into a boolean function g where $g : \{0,1\}^n \to \{0,1\}$. The second component of the proof is a *reduction* showing that if there exists a "too small" circuit C such that $\Pr_{y \leftarrow U_n}[C(y) = g(y)] \geq \frac{1}{2} + \epsilon$ then there exists a "too small" circuit A that computes f.

In typical hardness amplification results such as those stated in Theorem 1.3 one sets $k = O(\log n)$ and lets f be the characteristic function of an E-complete problem. However, in our formal definition we do not require this dependence between k and n, which only makes our limitations stronger. In the setting of hardness amplification it is also common to consider the stronger assumption that the initial function f is hard on average (making the hardness amplification task easier). Our limitations also hold in this stronger setup (which make the results stronger) and the formal definition (given below) already discusses this more general case.

Definition 6.1 (black-box hardness amplification Shaltiel and Viola (2010)[19]). A $\delta \to (\frac{1}{2} - \epsilon)$ **black-box hardness amplification** with input lengths k and n, and list size 2^a is a pair (Con, Red) such that:

- A construction Con is a map from functions $f : \{0,1\}^k \to \{0,1\}$ to functions $\mathrm{Con}_f : \{0,1\}^n \to \{0,1\}$.
- A reduction Red is an oracle procedure $\mathrm{Red}^{(\cdot)}(x, \alpha)$ that accepts two inputs $x \in \{0,1\}^k$ and $\alpha \in \{0,1\}^a$ which is called a "nonuniform advice string". Red also receives oracle access to a function $C : \{0,1\}^n \to \{0,1\}$.

We say that a function $C : \{0,1\}^n \to \{0,1\}$ ϵ-breaks a function $f : \{0,1\}^k \to \{0,1\}$ if

$$\Pr_{y \leftarrow U_n}[C(y) = \mathrm{Con}_f(y)] \geq \frac{1}{2} + \epsilon.$$

[19] We use a different notation from Shaltiel and Viola (2010). More precisely, in Shaltiel and Viola (2010) instead of a single reduction Red that receives an a bit long advice string α, they define a list/class of reductions of size 2^a. This notation is clearer for the results that we present.

We require that for all functions $f : \{0,1\}^k \to \{0,1\}$ and $C : \{0,1\}^n \to \{0,1\}$ such that C ϵ-breaks f, there exists $\alpha \in \{0,1\}^a$ such that

$$\Pr_{x \leftarrow U_k}[\mathrm{Red}^C(x,\alpha) = f(x)] \geq 1 - \delta.$$

We omit δ and say that (Con, Red) is a **worst-case → $(\frac{1}{2} - \epsilon)$ black-box amplification** if $\delta < 2^{-k}$ (which means that the requirement above translates to $\mathrm{Red}^C(x,\alpha)$ computes $f(x)$ correctly on all inputs x).

We now elaborate on the formal choices made in the definition.

The role of the nonuniform advice string. Sudan et al. (2001) observed that if (Con, Red) is a worst-case → $(\frac{1}{2} - \epsilon)$-black-box hardness amplification then Con is a $(\frac{1}{2} - \epsilon, 2^a)$-list decodable code. It is known that for $\epsilon < 1/4$ such codes do not allow unique decoding, and so the notions of "list size" and "nonuniform advice string" that show up in Definition 6.1 are necessary for studying hardness amplification with small ϵ. (In other words, it is not interesting to rule out black-box hardness amplification with $a = 0$). The reader is referred to Shaltiel and Viola (2010) for additional discussion on this issue.

Usefulness of definition. Let us observe that black-box hardness amplification indeed proves hardness amplification results. Indeed, if f is $(1-\delta)$-incomputable by some class $\mathcal{D}$ of circuits, and for every circuit C in $\mathcal{C}$ and every $\alpha \in \{0,1\}^a$, the function $D(x) = \mathrm{Red}^C(x,\alpha)$ is in $\mathcal{D}$, then we can indeed conclude that $g = \mathrm{Con}_f$ is ϵ-incomputable by $\mathcal{C}$.

Hardness assumptions against nondeterministic circuits We will be interested in the case where $\mathcal{D}$ consists of poly-size Σ_i-circuits, and $\mathcal{C}$ consists of poly-size deterministic circuits. This allows the reduction Red to use nondeterminism and motivates the following definition.

Definition 6.2 (nondeterministic reductions)**.** A reduction Red is **size s deterministic**, if Red is a size s deterministic oracle circuit. Red is **size s nondeterministic**, if Red is a size s nondeterministic oracle circuit. More generally, Red is **size s, i-nondeterministic** if there exists a deterministic size s oracle circuit $A^{(\cdot)}$ such that for every $x \in \{0,1\}^k$, $\alpha \in \{0,1\}^a$ and $C : \{0,1\}^n \to \{0,1\}$:

$$\mathrm{Red}^C(x,\alpha) = 1 \;\Leftrightarrow\; \exists z_1 \forall z_2 \exists z_3 \forall z_4 \ldots \mathrm{Q} z_i : \; A^C(x,\alpha,z_1,\ldots,z_i) = 1$$

where Q stands for "∃" if i is odd, and for "∀" if i is even.[20]

Note that the more general definition captures deterministic reductions (for $i = 0$) and nondeterministic reductions for $i = 1$. Let us discuss some aspects of Definition 6.2. This definition captures nondeterministic reductions that are used in the literature (Drucker, 2013; Feige and Lund, 1997; Klivans and van Melkebeek, 2002; Trevisan and Vadhan, 2000). Indeed, if there is a size s, i-nondeterministic reduction Red and some construction Con such that (Con, *Red*) is a worst-case → $(\frac{1}{2} - \epsilon)$-black box hardness amplification, then it follows that if f is incomputable by Σ_i-circuits of size $s \cdot s' + a$, then $g = \mathrm{Con}_f$ is ϵ-incomputable by (deterministic) circuits of size s'.

[20] We make no explicit bound on the length of $z_1, \ldots, z_i$. However, the fact that A is a size s circuit, places a bound of s on the total length of $z_1, \ldots, z_i$.

Remark 6.3 (Comparison of this model to previous limitations on reductions)**.** *Previous lower bounds on black-box hardness amplification (Artemenko and Shaltiel, 2014a; Shaltiel and Viola, 2010) cannot handle nondeterministic reductions. This is because previous lower bounds rely on the following assumption: given an $x \in \{0,1\}^k$ and $\alpha \in \{0,1\}^a$, there must be many queries y to the oracle C, that the reduction $Red^C(x,\alpha)$ did not make. This indeed holds for small size deterministic reductions, as the number of queries that $Red^C(x,\alpha)$ makes is bounded by the total size of the reduction. Note, that in contrast, nondeterministic reductions are "armed with quantifiers", and the computation of $Red^C(x,\alpha)$ may (when varying over all choices of z_1) query the oracle on all $y \in \{0,1\}^n$. Indeed, this is the property that is used in the aforementioned positive results of (Drucker, 2013; Feige and Lund, 1997; Trevisan and Vadhan, 2000) to bypass the limitations on deterministic reductions.*

Remark 6.4 (Comparison of this model to previous limitations on constructions)**.** *An orthogonal line of work, initiated by Viola (2005) (see also Lu et al., 2007, 2008) considers the setting in which we place no limitations on the reduction, and instead require that the construction is defined by $Con_f(x) = A^f(x)$ where A is a "low complexity" oracle procedure. Here low complexity, typically means: the polynomial time hierarchy. These works prove impossibility results for black-box hardness amplification in this setting. In our model, the construction is unbounded, and we place computational limitations on the reduction.*

We stress that previous work on restricting the construction does not seem relevant to the setting that we consider. This line of work is concerned with hardness amplification results where (the characteristic function of) f is computable in the polynomial time hierarchy (whereas we want to rule out the case that f is in E).

More specifically, hardness amplification results that start from a hardness assumption on E, typically set $k = O(\log n)$. This means that in time poly(n), A can read the entire truth table of f. The aforementioned limitations do not hold in such a case (and in fact, the intuition behind these limitations is based on the fact that A cannot perform poly-time computations on the entire truth table of f). Summing up, our results are the first limitations that are applicable when assuming a hardness assumption of the form E is hard for exponential size Σ_i-circuits.

6.1.2 Formal statement of our limitations

We now state our limitations formally. We start by stating limitations for reductions which are based on worst case assumptions.

Theorem 6.5 (Limitations on nondeterministic reductions)**.** *For every constants $i \geq 0$ and $d > 1$, and every sufficiently large n and k such that $2d \log n \leq k \leq n$, $a \leq n^d$, there does not exist a worst-case $\rightarrow (1/2 - \epsilon)$ black-box hardness amplification with input lengths k and n where Red is a size n^d, i-nondeterministic reduction with $\epsilon = n^{-\omega(1)}$.*

We now elaborate on the meaning of Theorem 6.5. In our hardness amplification setting we want to obtain $g = Con_f$ that is computable in time poly(n) and is $n^{-\omega(1)}$-incomputable by size n^c circuits. In our formal model we make no assumption about the complexity of the construction. However, when interpreting the result let us make the assumption that the implementation of $g = Con_f$ generates a function g with time complexity that is at least the time complexity of f. (This follows immediately if $g = Con_f$ is obtained by some oracle "construction procedure" that invokes f at least once, as is the case in all known constructions). We are interested in the case where g is computable in time poly(n) and by the former discussion, this implies that f must be computable in time poly(n).

We are assuming that f is incomputable by circuits of some size, and since f is computable in time poly(n), this size must be smaller than n^d for some constant d. Thus, in order to contradict the assumption that f is incomputable by size n^d circuits, the reduction cannot have size larger than n^d.[21] This discussion motivates the choice of parameters in the theorem.

Indeed, Theorem 6.5 shows that if we start with some function $f : \{0,1\}^k \to \{0,1\}$ that is incomputable by size poly(n) circuits (e.g. if we set $k = O(\log n)$ and let f be the characteristic function of an E-complete problem) then we cannot use nondeterministic reductions to obtain an ϵ-incomputable function for $\epsilon = n^{-\omega(1)}$, even if we are willing to assume that E is hard for exponential size Σ_i-circuits, for a large i.

Finally, let us make the technical remark that in Theorem 6.5 we indeed must require that $k > d \log n$ as otherwise, the reduction Red could ask for an n^d long nonuniform advice string that is the truth table of f, and the theorem will not hold. Moreover, the case that $k \leq d \log n$ is uninteresting, as in this case, circuits of size $2^k = n^d$ can compute f and we will not be able to assume that f is incomputable by circuits of size larger than n^d.

Theorem 6.5 is a special case of the following more general result, in which we rule out black-box hardness amplification even starting from average case hardness, and we also give a more precise estimate on what is the smallest ϵ that can be achieved. In the following, let $H : [0,1] \to [0,1]$ denote the binary entropy function defined by $H(p) = p \log_2(1/p) + (1-p)\log_2(1/(1-p))$.

Theorem 6.6 (Limitations on nondeterministic reductions (general case))**.** *There exists a constant $c > 1$ such that for every constants $i \geq 0$, $d > 1$ and for every sufficiently large n, and every k such that $2d \log n \leq k \leq n$, $a \leq n^d$, and $\delta < \frac{1}{2} - \frac{1}{n}$ such that $H(\delta + \frac{1}{n}) \leq 1 - \frac{1}{n^{2d}}$, and $\epsilon > 0$, there does not exist a $\delta \to (1/2 - \epsilon)$ black-box hardness amplification where Red is a size n^d, i-nondeterministic reduction with $\epsilon = n^{-(i+c)\cdot d}$.*

Note that we can allow δ to be any constant $\delta < \frac{1}{2}$ and even slowly approach $\frac{1}{2}$.

6.1.3 Proof of the lower bound

In this section we prove Theorem 6.6. Our approach is based on a previous lower bound of Shaltiel and Viola (2010). It will be helpful to identify boolean functions $C : \{0,1\}^n \to \{0,1\}$ with their truth tables $C \in \{0,1\}^{2^n}$. We also need the following definition.

Definition 6.7 (Noise vectors and oracles)**.** For $0 \leq p \leq 1$, and an integer t, we use N_p^t to denote the distribution of t i.i.d. bits where each of them has probability p to evaluate to one. We omit t if it is 2^n, and note that by our conventions, we can think of N_p as a probability distribution over functions $N_p : \{0,1\}^n \to \{0,1\}$.

Let Red be a size n^d, i-nondeterministic reduction as in the statement of the theorem. We will show that ϵ cannot be small. For a function $f : \{0,1\}^k \to \{0,1\}$, we will consider two distributions over oracles. The first is

$$C_1 = \mathrm{Con}_f \oplus N_{\frac{1}{2}-2\epsilon},$$

[21] We remark that the reductions used to prove Theorem 1.3 use $k = O(d \cdot \log n)$ and so it indeed follows that if f is the characteristic function of a problem in E then it is computable in time $2^{O(k)} = \mathrm{poly}(n)$.

where for two functions $A, B : \{0,1\}^n \to \{0,1\}$, the function $A \oplus B : \{0,1\}^n \to \{0,1\}$ is defined by $(A \oplus B)(y) = A(y) \oplus B(y)$.

By a multiplicative Chernoff bound, C_1 is likely to agree with Con_f on a fraction of $\frac{1}{2} + \epsilon$ of the inputs (i.e., C_1 is likely to ϵ-break f), and so, by assumption, the reduction is likely to succeed. This is stated precisely below.

Lemma 6.8. *For every $f : \{0,1\}^k \to \{0,1\}$, with probability $1 - 2^{-\Omega(2^n)}$ over the choice of a "noise function" $N : \{0,1\}^n \to \{0,1\}$ from the distribution $N_{\frac{1}{2}-2\epsilon}$, we get that there exists $\alpha \in \{0,1\}^k$ such that $\Pr_{x \leftarrow U_k}[Red^{Con_f \oplus N}(x, \alpha) = f(x)] \geq 1 - \delta$.*

Proof. By the multiplicative Chernoff bound the probability that the $N_{\frac{1}{2}-2\epsilon}$ evaluates to one on a fraction of inputs less than $\frac{1}{2} - \epsilon$ is at least $1 - 2^{-\Omega(2^n)}$. If this event occurs, then $\mathrm{Con}_f \oplus N$ agrees with Con_f on a $\frac{1}{2} + \epsilon$ fraction of the inputs. By the guarantee of the reduction, this implies that there exists $\alpha \in \{0,1\}^k$ such that $\Pr_{x \leftarrow U_k}[\mathrm{Red}^{\mathrm{Con}_f \oplus N}(x, \alpha) = f(x)] \geq 1 - \delta$. □

The second oracle we consider is

$$C_2 = \mathrm{Con}_f \oplus N_{\frac{1}{2}}.$$

This oracle is distributed like $N_{\frac{1}{2}}$ and carries no information on the function f. Therefore, given such an oracle, the reduction Red will not be able to approximate a function f that is chosen at random. This is stated in the next lemma.

Lemma 6.9. *For every $\alpha \in \{0,1\}^k$, with probability*

$$1 - 2^{-(1 - H(\delta + \frac{1}{n})) \cdot 2^k}$$

over the choice of a "noise function" $N : \{0,1\}^n \to \{0,1\}$ from the distribution $N_{\frac{1}{2}}$, and a uniform function $f : \{0,1\}^k \to \{0,1\}$, we get that $\Pr_{x \leftarrow U_k}[Red^{Con_f \oplus N}(x, \alpha) = f(x)] < 1 - (\delta + 1/n)$.

Proof. The oracle given to the reduction is independent of f. Thus, for every $\alpha \in \{0,1\}^k$, the reduction computes a function that is independent of f. The reduction succeeds if these two functions (viewed as strings) have relative Hamming distance $\leq \delta + \frac{1}{n}$, and the number of strings of length t that have relative Hamming distance distance $\leq \alpha$ from some fixed string is upper-bounded by $2^{H(\alpha) \cdot t}$. □

Thus, loosely speaking, Red can be used to distinguish $N_{\frac{1}{2}-2\epsilon}$ from $N_{\frac{1}{2}}$. Following Furst et al. (1984) we can convert a size n^d, i-noneterministic reduction into a (deterministic) circuit of size $2^{O(n^d)}$ and depth $i+2$ that receives C as a 2^n bit-long input.

Lemma 6.10. *There exists a constant $e > 1$ such that for every $x \in \{0,1\}^k$, $\alpha \in \{0,1\}^a$ there exists (deterministic) circuit $B_{x,\alpha} : \{0,1\}^{2^n} \to \{0,1\}$ of size $2^{e \cdot n^d}$ such that for every $x \in \{0,1\}^k$, $\alpha \in \{0,1\}^a$ and $C : \{0,1\}^n \to \{0,1\}$, $B_{x,\alpha}(C) = Red^C(x, \alpha)$ (where on the l.h.s. we think of C as a string $C \in \{0,1\}^{2^n}$ and on the r.h.s. we think of $C : \{0,1\}^n \to \{0,1\}$ as a function).*

Proof. Let A be the size n^d deterministic circuit that is used by the reduction (as define in Definition 6.2). For every fixed x, α and $z_1, \ldots, z_i$, the computation $A^C(x, \alpha, z_1, \ldots, z_i)$ can be viewed as a depth n^d decision tree that makes queries to C. It can thus be implemented by a depth 2 circuit

$B_{x,\alpha,z_1,\ldots,z_i}(C)$ of size $2^{O(n^d)}$ that receives (the 2^n bit long) input C. We now consider the function $B_{x,\alpha}(C)$ defined to be one iff $\exists z_1 \forall z_2 \ldots \mathrm{Q} z_i : B_{x,\alpha,z_1,\ldots,z_i}(C) = 1$. Note that this function can be implemented by a circuit of depth $i+2$ and size $2^{O(n^d)}$ times the size of a circuit $B_{x,\alpha,z_1,\ldots,z_i}$. Overall, we get a depth $i+2$, size $2^{O(n^d)}$ circuit. □

We now show that Red can be used to construct a constant depth circuit of size $2^{O(n^d)}$ that distinguishes between $N_{\frac{1}{2}-2\epsilon}$ and $N_{\frac{1}{2}}$.

Lemma 6.11. *There is a circuit B of size $2^{O(n^d)}$ and depth $i+O(1)$ such that $|\Pr[B(N_{\frac{1}{2}-2\epsilon}) = 1] - \Pr[B(N_{\frac{1}{2}}) = 1]| \geq 0.99$.*

Proof. For every function $f : \{0,1\}^k \to \{0,1\}$ let us consider the circuit $A_f : \{0,1\}^n \to \{0,1\}$ defined as follows: The circuit A_f is hardwired with f and Con_f. Upon receiving an input $N \in \{0,1\}^{2^n}$ it computes $C \in \{0,1\}^{2^n}$ defined by $C(y) = N(y) \oplus \mathrm{Con}_f(y)$. For every $x \in \{0,1\}^k$ and $\alpha \in \{0,1\}^a$, A_f computes $B_{x,\alpha}(C)$. For every $\alpha \in \{0,1\}^a$, the circuit A_f approximately counts the fraction of $x \in \{0,1\}^k$ such that $f(x) = B_{x,\alpha}(C)$. If there exists an $\alpha \in \{0,1\}^a$ such that this fraction is at least $1-\delta$ then the circuit accepts. If for all $\alpha \in \{0,1\}^a$ the fraction is smaller than $1-(\delta+\frac{1}{n})$ the circuit rejects. The difference of $1/n$ was chosen so that this approximate counting task can be done by a circuit of size $2^{O(n)}$ and constant depth (as proven by Ajtai, 1983). Overall, the circuit A_f described above can be implmented by a depth $i+O(1)$ circuit of size $2^{O(n^d)}$.

We now use the probabilistic method to show that there exists $f : \{0,1\}^k \to \{0,1\}^n$, for which $B = A_f$ is the circuit that we need to construct. We choose $F : \{0,1\}^k \to \{0,1\}$ uniformly at random. By Lemma 6.8, For every function f, $\Pr[A_f(N_{\frac{1}{2}-2\epsilon}) = 1] \geq 1 - 2^{-\Omega(2^n)}$ and therefore, $\Pr[A_F(N_{\frac{1}{2}-2\epsilon}) = 1] \geq 1 - 2^{-\Omega(2^n)}$. By Lemma 6.9 and a union bound over all $\alpha \in \{0,1\}^a$, we have $\Pr[A_F(N_{\frac{1}{2}}) = 1] \leq 2^a \cdot 2^{-(1-H(\delta+\frac{1}{n}))\cdot 2^k} = o(1)$ by our choice of parameters. By an averaging argument, there exists an f such that A_f distinguishes the two distributions with probability $1-o(1) \geq 0.99$. □

However, it is known that such circuits do not exist for small ϵ. This follows by reduction to lower bounds on constant depth circuits that compute the majority function, and appears e.g. in (Shaltiel and Viola, 2010; Viola, 2006).

Theorem 6.12. *There exists a constant $a > 1$, such that for every sufficiently small $\epsilon > 0$, circuits of depth k and size $s = exp((\frac{1}{\epsilon})^{\frac{1}{k+a}})$ cannot distinguish $N^t_{\frac{1}{2}}$ and $N^t_{\frac{1}{2}-\epsilon}$ with advantage 0.99 for any $t \leq s$.*

Theorem 6.6 follows.

6.2 Extending the limitations to pseudorandom generators

We now explain how to modify the definitions and argument of Section 6.1.1 to rule out constructions of pseudorandom generator. For this purpose we will modify Definition 6.1 in two ways, so that it captures constructions and reductions for pseudorandom generators:

- the function $g = \mathrm{Con}(f)$ will now be a function $g : \{0,1\}^r \to \{0,1\}^n$ where $r \leq n-1$ (as we are now considering constructions of a pseudorandom generator g).

- We will say that $C : \{0,1\}^n \to \{0,1\}$ ϵ-breaks f if C distinguishes the output of the construction from uniform, that is, if $|\Pr[C(\mathrm{Con}_f(U_r)) = 1] - \Pr[C(U_n) = 1] > \epsilon$.

With these modifications, Definition 6.1 now captures pairs of construction/reduction for PRGs (rather than incomputable functions). We claim that Theorems 6.5 and Theorem 6.6 hold exactly as stated for the modified definition. We sketch this argument below.

The proof repeats the argument of Section 6.1.3 with the following modifications:

Let $Dist_f : \{0,1\}^n \to \{0,1\}$ denote the boolean function that accepts an input $x \in \{0,1\}^n$ iff there exists $s \in \{0,1\}^r$ such that $\mathrm{Con}_f(r) = x$. Loosely speaking, this is the optimal distinguisher against the candidate pseudorandom generator Con_f. In Section 6.1.3 we use two oracles C_1, C_2 which we now modify as follows: given a function $f : \{0,1\}^k \to \{0,1\}$, we set $C_1 = Dist_f \oplus N_{\frac{1}{2}-2\epsilon}$ and $C_2 = Dist_f \oplus N_{\frac{1}{2}}$.

Note that this is similar to the previous proof except that Con_f is replaced by $Dist_f$. The reason for this modification is that in order to ϵ-break f, a circuit C needs to distinguish Con_f from uniform (rather than compute Con_f as was the case previously). The proof can now proceed as before (with these modifications). Indeed, the analogs of Lemmas 6.8 and 6.9 follow with these modifications. Loosely speaking, this is because for every f, with high probability C_1 ϵ-breaks f, and on the other hand, for every f, C_2 is uniformly distributed (and thus gives no information on f). These are the two properties that come up in the proof.

6.3 Extending the limitations to extractors

In this section we explain how to extend our limitations so that they hold for extractors for distributions samplable/recognizable by small circuits. Loosely speaking, this follows because an $(n - \log(1/\epsilon), \epsilon)$-extractor is in particular a 2ϵ-incomputable function. We now explain this argument.

We first consider a $(n - \log(1/\epsilon), \epsilon)$-extractor $E : \{0,1\}^n \to \{0,1\}$ for distributions recognizable by circuits of size n^c. We claim that E is 2ϵ-incomputable for circuits of size n^c. Indeed, let $C : \{0,1\}^n \to \{0,1\}$ be a size n^c circuit, and let $X \leftarrow U_n$. We have that:

$$\Pr[C(X) = E(X)] = \sum_{b \in \{0,1\}} \Pr[C(X) = E(X) | C(X) = b] \cdot \Pr[C(X) = b] = \sum_{b \in \{0,1\}} \Pr[E(X) = b | C(X) = b] \cdot \Pr[C(X) = b]. \quad (6.13)$$

Note that the conditional distributions $(X|C(X) = b)$ above are recognizable by size n^c circuits. If $\Pr[C(X) = b] \geq \epsilon$ then the conditional distribution has min-entropy at least $n - \log(1/\epsilon)$ and so, by the property of the extractor $\Pr[E(X) = b|C(X) = b] \leq \frac{1}{2} + \epsilon$. If this holds for both values of b, then $\Pr[C(X) = E(X)] \leq \frac{1}{2} + \epsilon$. If it doesn't hold, then there is a single value b', for which $\Pr[C(X) = b'] < \epsilon$. In that case, the sum above can be bounded by $\frac{1}{2} + 2\epsilon$. In both cases, the claim follows.

We now consider extractors for samplable distributions, and sketch the argument for the impossibility result. The key observation is that by Jerrum et al. (1986) and Bellare et al. (2000) the distribution $(X|C(X) = b)$ is samplable by NP-circuits. Thus, an $(n - \log(1/\epsilon), \epsilon)$-extractor for distributions samplable by size n^c NP-circuits is 2ϵ-incomputable by deterministic circuits of slightly smaller size. This means that an i-nondeterministic reduction showing that E is an $(n - \log(1/\epsilon), \epsilon)$-extractor for samplable distributions, implies an $(i+1)$-nondeterministic reduction showing that E is a 2ϵ-incomputable, and we have already ruled out such reductions for negligible ϵ in Section 6.1.1.

7 Extractors for recognizable distributions and incompressible functions with low error

Our constructions of nonboolean incompressible functions with low error (stated in Theorem 1.16) and of extractors for samplable distributions with relative error (stated in Theorem 1.21), both follow from a construction of *extractors for recognizable distributions with relative error.*

7.1 Relative-error extractors for weakly recognizable distributions

We generalize Definition 1.22, introducing the notion of "weakly recognizable" distributions.

Definition 7.1 (weakly recognizable distributions)**.** We say that a distribution X on n bits is **weakly recognizable** by a class $\mathcal{C}$ of functions $C : \{0,1\}^n \to \mathbb{N}$ if there exists a function C in $\mathcal{C}$ such that for every $x \in \{0,1\}^n$, $\Pr[X = x] = \frac{C(x)}{\sum_{x' \in \{0,1\}^n} C(x')}$.[22]

Note that a recognizable distribution is in particular weakly recognizable. However, the notion of weakly recognizable distributions allows distributions that are not flat. The notion of extractors for recognizable distributions (with standard error) was introduced by Shaltiel (2011a). We give a construction of extractors for weakly recognizable distributions that have relative error under the assumption that E is hard for exponential size Σ_3-circuits.

Theorem 7.2 (Extractors for weakly recognizable distributions with relative error)**.** *If E is hard for exponential size Σ_3-circuits then there exists a constant $\alpha > 0$ such that for every constant $c > 1$ and sufficiently large n, and every $m \le \alpha n$ there is a $((1-\alpha) \cdot n, \frac{1}{n^c})$-relative-error extractor $E : \{0,1\}^n \to \{0,1\}^m$ for distributions weakly recognizable by size n^c circuits. Furthermore, E is computable in time poly(n^c).*

7.2 Obtaining relative-error extractors for samplable distributions

We now show that extractors for distributions that are weakly recognizable by NP-circuits are extractors for distributions samplable by (deterministic) circuits.

Lemma 7.3 (connection between samplable and recognizable distributions)**.** *If a distribution X on $\{0,1\}^n$ is samplable by a size $s \ge n$ circuit then for every $\epsilon > 0$ there exists a distribution X' on $\{0,1\}^n$ that is weakly recognizable by size poly(s/ϵ) NP-circuits, and for every event A, $|\Pr[X \in A] - \Pr[X' \in A]| \le \epsilon \cdot min(\Pr[X \in A], \Pr[X' \in A])$.*

Proof. Let $C : \{0,1\}^{n'} \to \{0,1\}^n$ be a size s circuit that samples the distribution X, and let $\epsilon > 0$. By Theorem 3.3 there exists a size poly(s/ϵ) NP-circuit $C'(x)$ which given x, computes an $\epsilon/10$-relative approximation of the integer $\Pr[C(U_{n'}) = x] \cdot 2^{n'}$. Let X' be the distribution on $\{0,1\}^n$

[22] In the definition above we don't set an a-priori bound on length of integers used. In this paper we will always have that $\mathcal{C}$ will be size s circuits, and the size bound implies an upper bound of s on length of integers output by C.

that is recognized by C'. That is, for every $x \in \{0,1\}^n$,

$$\Pr[X' = x] = \frac{C'(x)}{\sum_{x' \in \{0,1\}^n} C'(x')}$$

Note that for every $x \in \{0,1\}^n$, this quantity is an ϵ-relative approximation of

$$\frac{\Pr[C(U_{n'}) = x]}{\sum_{x' \in \{0,1\}^n} \Pr[C(U_{n'}) = x']} = \Pr[X = x],$$

and this indeed gives that for every event A,

$$|\Pr[X \in A] - \Pr[X' \in A]| \leq \epsilon \cdot min(\Pr[X \in A], \Pr[X' \in A]). \qquad \square$$

It follows that for every constant $c > 1$, if we have a $(k, n^{-(c+1)})$-relative-error extractor for distributions weakly recognizable by size $n^{O(c)}$ NP-circuits, then this extractor is also a (k, n^{-c})-relative-error extractor for distributions samplable by size n^c circuits. Theorem 7.2 can be pushed "one level up the hierarchy". That is, assume that E is hard for exponential size Σ_4-circuits, and conclude that the extractor works for distributions weakly recognizable by size n^c NP-circuits. This gives a construction of extractors for samplable distributions, and proves Theorem 1.21.

7.3 Constructing relative-error extractors for recognizable distributions

We now give our construction of a relative-error extractor for recognizable distributions. The construction and its analysis relies on components and ideas of Trevisan and Vadhan (2000). We imitate the overall argument structure of Trevisan and Vadhan (2000). The key point is that we are able to obtain extractors with relative error.

We start by using our assumption to obtain a function that is ϵ-incomputable by Σ_2-circuits. This is done by observing that the proof of Theorem 1.13 is relativizing, and so we can "push it" up the hierarchy and get:

Theorem 7.4 (Trevisan and Vadhan, 2000). *If E is hard for exponential size Σ_3-circuits, then there exists some constant $\alpha > 0$ such that for every constant $c > 1$ and for every sufficiently large n, there is a function $f : \{0,1\}^n \to \{0,1\}^{n'}$ that is ϵ-incomputable by size n^c Σ_2-circuits for $n' = \alpha n$ and $\epsilon = 2^{-(n'/3)}$. Furthermore, f is computable in time poly(n^c).*

The statement above is identical to Theorem 1.13 except that we assume hardness for Σ_3-circuits and conclude incomputability by Σ_2-circuits. An additional modification is that we now denote the output length by n' (and not m). This is because we reserve m for the output length of the extractor (to be defined next).

Our construction will use 2-source extractors, defined below (see e.g., the survey article Shaltiel, 2011b).

Definition 7.5. A (k_1, k_2, ϵ)-**2-source extractor** is a function $T : \{0,1\}^{n'} \times \{0,1\}^{n'} \to \{0,1\}^m$ such that for any two independent random variables R_1, R_2 with $H_\infty(R_1) \geq k$ and $H_\infty(R_2) \geq k$, $T(R_1, R_2)$ is ϵ-close to U_m.

Theorem 7.6 (Chor and Goldreich, 1988; Dodis et al., 2004; Vazirani, 1987). *There exists a constant $\alpha > 0$ such that for every sufficiently large n' and every $m \leq \alpha n'$ there is a poly-time computable $(0.2n', 0.9n', 2^{-3m})$-2-source extractor $T : \{0,1\}^{n'} \times \{0,1\}^{n'} \to \{0,1\}^m$.*

The construction: Let $0 < \nu \leq 1$ be some constant. Let $n' = \nu \cdot n$ and let $\bar{n} = n + n'$. These three parameters are going to serve as input lengths for various functions in our construction, and the reader should keep in mind that the three parameters are polynomially related.

Given functions $f : \{0,1\}^n \to \{0,1\}^{n'}$ and $T : \{0,1\}^{n'} \times \{0,1\}^{n'} \to \{0,1\}^m$ we construct a function $E : \{0,1\}^{\bar{n}} \to \{0,1\}^m$ as follows: Given an input $z \in \{0,1\}^{\bar{n}}$ we split it into two parts: $x \in \{0,1\}^n$ and $i \in \{0,1\}^{n'}$ and set

$$E(z) = T(f(x), i).$$

This construction should be compared to the more standard idea of code-concatenation, or the Goldreich-Levin theorem. Indeed, the standard way to take a nonboolean function that is ϵ-incomputable and convert it to a boolean function that is ϵ'-incomputable for some ϵ' related to ϵ is to take $E(z) = EC(f(x))_i$ where EC is a boolean error-correcting code, with sufficiently efficient decoding. Here, the input to E is not uniform, but rather a high min-entropy distribution, and code concatenation does not work. Nevertheless, it turns out that a 2-source extractor gives us the following "list-decoding" guarantee that will be used in the proof.

Lemma 7.7. *Let $\alpha > 0$ and T be the constant and 2-source extractor of Theorem 7.6. For every sufficiently large n', and every $m \leq \alpha n'$, every $\epsilon \geq 2^{-m}$, every $a \in \{0,1\}^m$, and every distribution R_2 with $H_\infty(R_2) \geq 0.9n'$, there are at most $2^{0.2 \cdot n'}$ strings $y \in \{0,1\}^{n'}$ such that $\Pr[E(y, R_2) = a] \geq (1+\epsilon)2^{-m}$, and similarly there are at most $2^{0.2 \cdot n'}$ strings $y \in \{0,1\}^{n'}$ such that $\Pr[E(y, R_2) = a] \leq (1-\epsilon)2^{-m}$.*

Proof. We prove the first conclusion, and the second follows in the same manner. Fix some R_2 with $H_\infty(R_2) \geq 0.9n'$. If there exists an $a \in \{0,1\}^m$ with more than $2^{0.2 \cdot n'}$ strings $y \in \{0,1\}^{n'}$ such that $\Pr[E(y, R_2) = a] \geq (1+\epsilon)2^{-m}$, then let R_1 be the uniform distribution over these strings, and note that $H_\infty(R_1) \geq 0.2 \cdot n'$. It follows that

$$\Pr[T(R_1, R_2) = a] \geq (1+\epsilon) \cdot 2^{-m} \geq 2^{-m} + 2^{-2m} = \Pr_{Z \leftarrow U_n}[Z = a] + 2^{-2m}.$$

This contradicts the guarantee that T is a $(0.2n', 0.9n', 2^{-3m})$-2-source extractor □

The next theorem shows the correctness of our construction.

Theorem 7.8. *There exist constants $\nu, \beta > 0$, such that for every constant $c > 1$, there exists a constant $\alpha > 0$, such that for every sufficiently large n, $n' = \nu \cdot n$, $\bar{n} = n + n'$ the following holds.*

- *Let $f : \{0,1\}^n \to \{0,1\}^{n'}$ be a $2^{-n'/3}$-incomputable for size n^c, Σ_3-circuits.*
- *Let T be the $(0.2n', 0.9n', 2^{-3m})$-2-source extractor $T : \{0,1\}^{n'} \times \{0,1\}^{n'} \to \{0,1\}^{m=\alpha \cdot n}$ of Theorem 7.6.*

It follows that $E : \{0,1\}^{\bar{n}} \to \{0,1\}^m$ is an $(\bar{n} - \Delta, n^{-\beta c})$-relative-error extractor for distributions weakly recognizable by size $n^{\beta c}$ circuits, for $\Delta = \alpha \cdot n \geq \alpha \cdot \bar{n}/2$.

Theorem 7.2 now follows, as with these choices the function f of Theorem 7.4 satisfies the requirements Theorem 7.8, and noting that $n, n', \bar{n}$ are linearly related and so polynomials in one can be replaced by polynomials in another.

7.4 Proof of Theorem 7.8

In this section we prove Theorem 7.8. The proof is by contradiction. Let $\beta > 0$ be a constant that we choose later, and let $\epsilon = n^{-\beta c}/9$. Assume that the conclusion on Theorem 7.8 does not hold. That is, that for some sufficiently large n, E is not an $(\bar{n} - \Delta, 9\epsilon)$-relative-error extractor for distributions weakly recognizable by size $n^{\beta \cdot c}$ circuits. By an averaging argument over all $a \in \{0,1\}^m$ we get that:

Lemma 7.9. *There exists a distribution $Z' = (X', I')$ over $\{0,1\}^{\bar{n}}$ with $H_\infty(Z) \geq \bar{n} - \Delta$ that is weakly recognizable by a circuit C of size n^c, and there exists $a \in \{0,1\}^m$ such that $\Pr[E(Z') = a]$ is either at least $(1 + 9\epsilon) \cdot 2^{-m}$ or at most $(1 - 9\epsilon) \cdot 2^{-m}$.*

From now on we assume that $\Pr[E(Z') = a] \geq (1 + 9\epsilon) \cdot 2^{-m}$. (The case that $\Pr[E(Z') = a] \leq (1 - 9\epsilon) \cdot 2^{-m}$ follows the same way using the fact that in Lemma 7.7 we have control over both cases). We need the following definition.

Definition 7.10 (useful inputs)**.** We say that $x \in \{0,1\}^n$ is *useful* if

- $\Pr[E(x, I') = a | X' = x] = \Pr[T(f(x), I') = a | X' = x] \geq (1 + 2\epsilon) \cdot 2^{-m}$, and
- $H_\infty(I' | X' = x) \geq 0.9 \cdot n'$.

The parameters were chosen so that by another averaging argument we get that:

Lemma 7.11. $\Pr[X'$ *is useful*$] \geq \epsilon \cdot 2^{-m}$.

Proof. Let $\Delta' = m + \log(1/\epsilon)$. We say that $x \in \{0,1\}^n$ is *sufficiently heavy* if $\Pr[X' = x] \geq 2^{-(n+\Delta')}$ and let H be the set of x''s that are sufficiently heavy. Note that for every $x \in H$ and every $i \in \{0,1\}^{n'}$ we have that:

$$\Pr[I' = i | X' = x] = \frac{\Pr[I' = i \text{ and } X' = x]}{\Pr[X' = x]} \leq \frac{2^{-(n+n'-\Delta)}}{2^{-(n+\Delta')}} = 2^{-(n'-(\Delta+\Delta'))}. \quad (7.12)$$

which means that for every $x \in H$,

$$H_\infty(I' | X' = x) \geq n' - (\Delta + \Delta') \geq 0.9 \cdot n'$$

where the last inequality follows for appropriately chosen constant $\alpha, \nu > 0$. More specifically,

$$\Delta + \Delta' = \alpha \cdot n + m + \log(1/\epsilon) = 2\alpha \cdot n + c \log n.$$

We have that $n' = \nu \cdot n$ and and so by choosing $\alpha > 0$ to be a sufficiently small constant smaller than $\nu > 0$ we can make $\Delta + \Delta' \leq 0.1 \cdot n' = 0.1 \cdot \nu n$, for sufficiently large n.

This means that every sufficiently heavy x satisfies the second item of Def. 7.10. Note that $\Pr[X' \notin H] \leq 2^n \cdot 2^{-(n+\Delta')} = 2^{-\Delta'} \leq \frac{\epsilon}{2^m}$. It follows that:

$$\begin{aligned} \Pr[E(X', I') = a \text{ and } X' \in H] &\geq \Pr[E(X', I') = a] - \Pr[X' \notin H] \\ &\geq (1 + 9\epsilon) \cdot 2^{-m} - \epsilon \cdot 2^{-m} = (1 + 8\epsilon) \cdot 2^{-m}. \quad (7.13) \end{aligned}$$

We also have that:

$$\Pr[E(X',I') = a \text{ and } X' \in H] = \sum_{x\in H} \Pr[X' = x] \cdot \Pr[E(x,I') = a | X' = x] \quad (7.14)$$

Let G be the set of all $x \in H$ that satisfy the first item and let B be the set of all $x \in H$ that don't satisfy the first item. We can divide the sum according to these sets yielding an upper-bound of:

$$7.14 \le \sum_{x\in B} \Pr[X' = x] \cdot \Pr[E(x,I') = a|X' = x] + \Pr[X' \in G] < (1 + 2\epsilon) \cdot 2^{-m} + \Pr[X' \in G]. \quad (7.15)$$

Every $x \in G$ satisfies the two items of Def. 7.10, and overall we have that:

$$\Pr[X' \in G] \ge (1 + 8\epsilon) \cdot 2^{-m} - (1 + 2\epsilon) \cdot 2^{-m} \ge \epsilon \cdot 2^{-m}.$$

The lemma follows. □

We would like to get an estimate on the probability of useful inputs, according to the uniform distribution.

Lemma 7.16. $\Pr_{x\leftarrow U_n}[x \textit{ is useful}] \ge 2^{-\Delta} \cdot \epsilon \cdot 2^{-m}$.

Proof. Let G denote the set of useful x. We have that $H_\infty(Z') \ge \bar{n} - \Delta$ and therefore $H_\infty(X') \ge n - \Delta$. Thus, for every $x \in \{0,1\}^n$, $\Pr[X' = x] \le 2^{-(n-\Delta)}$. By Lemma 7.11, $\Pr[X' \in G] \ge \rho$ for $\rho = \epsilon \cdot 2^{-m}$. This means that $|G| \ge \rho/2^{-(n-\Delta)} = \rho \cdot 2^n/2^\Delta$ which means that $\Pr_{x\leftarrow U_n}[x \in G] \ge \rho/2^\Delta$. □

We now present a Σ_2-circuit A such that $\Pr_{x\leftarrow U_n}[A(x) = f(x)]$ is not small, and this will give a contradiction. We will present a probabilistic Σ_2-circuit A. By averaging over the random coins of the circuits, the random coins of A can be later hardwired, to produce the same success probability with a circuit that is not probabilistic.

A good intuition to keep in mind is that we want A to succeed with not too small probability on every useful input. It is simpler and instructive to first consider the case where Z' is recognized by C. We will later explain how to modify the proof if Z' is only weakly recognized by C.
When given $x \in \{0,1\}^n$, A acts as follows:

1. Let $\epsilon' = \epsilon/10$. Let $A_x(y)$ be an NP-circuit such that given $y \in \{0,1\}^{n'}$, A_x uses it's NP-oracle to compute:

   - an ϵ'-approximation $W'_{x,y}$ to the integer
     $W_{x,y} = |\{i : C(x,i) = 1 \wedge T(y,i) = a\}|$, and
   - an ϵ'-approximation V'_x to the integer
     $V_x = |\{i : C(x,i) = 1\}|$.

   The circuit A_x then computes $p' = W'_{x,y}/V'_x$ and it answers one iff $p' \ge (1 + 1.5 \cdot \epsilon) \cdot 2^{-m}$. Note that p' is a $2\epsilon'$-approximation to $p = W_{x,y}/V_x = \Pr[T(y,I') = a|X' = x]$. In particular, A_x answers one if $p \ge (1 + 2\epsilon) \cdot 2^m$ and A_x answers zero if $p \le (1 + \epsilon) \cdot 2^{-m}$.

2. A samples a uniform y from the set $\{y : A_x(y) = 1\}$, and outputs y.

Using Theorems 3.2 and 3.3 it follows that:

Lemma 7.17. *For a sufficiently small constant $\beta > 0$, A can be implemented by a (probabilistic) Σ_2-circuit of size poly$(n^{\beta c}/\epsilon') = n^c$.*

We observe that algorithm A indeed succeeds with not too small probability on "useful inputs".

Lemma 7.18. *If $x \in \{0,1\}^n$ is useful, then* $\Pr[A(x) = f(x)] \geq 2^{-0.2n'}$ *where the probability is over the random choices of A.*

Proof. Let $x \in \{0,1\}^n$ be useful. By the second item of Def. 7.10, we have that the distribution $R_2 = (I'|X' = x)$ has $H_\infty(R_2) \geq 0.9n'$. Therefore, by Lemma 7.7 the set S of strings $y \in \{0,1\}^{n'}$ such that $\Pr[T(y, R_2) = a] \geq (1+\epsilon)2^{-m}$ satisfies $|S| \leq 2^{0.2n'}$. The circuit A_x samples a uniform y from some subset of $S' \subseteq S$, and note that S' contains $f(x)$ because $\Pr[T(f(x), I') = a|X' = x] \geq (1+2\epsilon) \cdot 2^{-m}$. It follows that the probability that A hits $f(x)$ is at least $2^{-0.2 \cdot n'}$. □

This implies that

$$\Pr_{x \leftarrow U_n}[A(x) = f(x)] \geq \Pr_{x \leftarrow U_n}[x \text{ is useful}] \cdot 2^{-0.2n'} \geq 2^{-\Delta} \cdot \epsilon \cdot 2^{-m} \cdot 2^{-0.2n'} = 2^{-2\alpha n - 0.2n'} \cdot O(n^{-c})$$

The latter quantity can be made at least $2^{-n'/3}$ by choosing $\alpha > 0$ to be a sufficiently small constant much smaller than μ so that $2\alpha n \leq 0.1n' = 0.1\nu n$. This gives the required contradiction in the case the distribution Z is recognized by C.

We now consider the case that Z' is weakly recognizable by C. The only place where we used the fact that Z is recognizable, rather than weakly recognizable, is in step 1 of the algorithm A. More specifically, we need to show that in this case we can also get an NP-circuit $A_x(y)$ that can compute an ϵ'-approximation p' to $p = \Pr[T(y, I') = a|X' = x]$. For this purpose we observe that:

Lemma 7.19. *If a distribution Z on $\{0,1\}^n$ is weakly recognizable by circuits of size s, and $C_1, C_2 : \{0,1\}^n \to \{0,1\}$ are some size s circuits, then there exists an NP-circuit of size poly(s/ϵ) that computes an ϵ-approximation of* $\Pr[C_1(Z) = 1|C_2(Z) = 1]$.

Proof. Let $C : \{0,1\}^n \to \mathbb{N}$ be the circuit that weakly recognizes Z, and note that the integer that it outputs are between 0 and $2^s - 1$. Consider, the circuit $C' : \{0,1\}^n \times \{0,1\}^s \to \{0,1\}$, defined by $C'(z, y) = 1$ iff $C(z) \leq y$ where here we interpret y as a number between 0 and $2^s - 1$. Note that for any $z \in \{0,1\}^n$, $C(z) = |\{y : C'(z, y) = 1\}|$. This means that

$$Pr[C_1(Z) = 1|C_2(Z) = 1] = \frac{\Pr[C_1(Z) = 1 \wedge C_2(Z) = 1]}{\Pr[C_2(Z) = 1]} = \frac{\sum_{z:C_1(z)=1 \wedge C_2(z)=1} C(z)}{\sum_{z:C_2(z)=1} C(z)} \quad (7.20)$$

We can compute an approximation of the denominator by considering the circuit $C_2'(z, y)$ which outputs $C'(z, y)$ if $C_2(z) = 1$ and 0 otherwise. Note that approximately counting the accepting inputs of C_2' gives an approximation for the denominator. The same reasoning can be applied to the numerator. □

This means that we can indeed get an NP-circuit A_x that computes an ϵ'-approximation of $p = Pr[T(y, I') = a|X' = x]$, and this suffices for the proof.

Acknowledgements

We thank to Yuval Ishai for helpful discussions and to anonymous referees for helpful comments and corrections.

The first author was supported by ERC starting grant 639813 ERC-CLC, ISF grant 1155/11, Israel Ministry of Science and Technology (grant 3-9094), GIF grant 1152/2011, and the Check Point Institute for Information Security. The second author was supported by ERC starting grant 279559. The third author was supported by BSF grant 2010120, ISF grant 864/11, and ERC starting grant 279559. The work of the forth author was done while visiting the first author in Tel-Aviv University.

A preliminary version of this paper appears in Proceedings of the 30th Computational Complexity Conference (CCC'15).

References

Miklós Ajtai. Σ_1^1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24(1):1 – 48, 1983.

Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. From secrecy to soundness: Efficient verification via secure computation. In *ICALP (1)*, volume 6198 of *Lecture Notes in Computer Science*, pages 152–163. Springer, 2010.

Benny Applebaum, Yuval Ishai, Eyal Kushilevitz, and Brent Waters. Encoding functions with constant online rate, or how to compress garbled circuit keys. *SIAM J. Comput.*, 44(2):433–466, 2015.

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

Sergei Artemenko and Ronen Shaltiel. Pseudorandom generators with optimal seed length for non-boolean poly-size circuits. In *STOC*, pages 99–108. ACM, 2014b.

Sergei Artemenko, Russell Impagliazzo, Valentine Kabanets, and Ronen Shaltiel. Pseudorandomness when the odds are against you. In *Conference on Computational Complexity*, 2016.

László Babai and Shlomo Moran. Arthur-merlin games: A randomized proof system, and a hierarchy of complexity classes. *J. Comput. Syst. Sci.*, 36(2):254–276, 1988.

László Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3:307–318, 1993.

Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in cryptography. *SIAM J. Comput.*, 37(2):380–400, 2007.

Mihir Bellare and John Rompel. Randomness-efficient oblivious sampling. In *FOCS*, pages 276–287. IEEE Computer Society, 1994.

Mihir Bellare, Oded Goldreich, and Erez Petrank. Uniform generation of np-witnesses using an np-oracle. *Inf. Comput.*, 163(2):510–526, 2000.

Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.

Arkadev Chattopadhyay and Rahul Santhanam. Lower bounds on interactive compressibility by constant-depth circuits. In *FOCS*, pages 619–628. IEEE Computer Society, 2012.

Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM J. Comput.*, 17(2):230–261, 1988.

Kai-Min Chung, Yael Tauman Kalai, and Salil P. Vadhan. Improved delegation of computation using fully homomorphic encryption. In *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 483–501. Springer, 2010.

Francesco Davì, Stefan Dziembowski, and Daniele Venturi. Leakage-resilient storage. In *SCN*, volume 6280 of *Lecture Notes in Computer Science*, pages 121–137. Springer, 2010.

Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *J. ACM*, 61(4):23:1–23:27, 2014.

Yevgeniy Dodis, Ariel Elbaz, Roberto Oliveira, and Ran Raz. Improved randomness extraction from two independent sources. In *APPROX-RANDOM*, volume 3122 of *Lecture Notes in Computer Science*, pages 334–344. Springer, 2004.

Andrew Drucker. Nondeterministic direct product reductions and the success probability of SAT solvers. In *FOCS*, pages 736–745. IEEE Computer Society, 2013.

Bella Dubrov and Yuval Ishai. On the randomness complexity of efficient sampling. In *STOC*, pages 711–720. ACM, 2006.

Sebastian Faust, Tal Rabin, Leonid Reyzin, Eran Tromer, and Vinod Vaikuntanathan. Protecting circuits from computationally bounded and noisy leakage. *SIAM J. Comput.*, 43(5):1564–1614, 2014.

Uriel Feige and Carsten Lund. On the hardness of computing the permanent of random matrices. *Computational Complexity*, 6(2):101–132, 1997.

Lance Fortnow and Rahul Santhanam. Infeasibility of instance compression and succinct pcps for NP. *J. Comput. Syst. Sci.*, 77(1):91–106, 2011.

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

Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In *CRYPTO*, volume 6223 of *Lecture Notes in Computer Science*, pages 465–482. Springer, 2010.

Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *STOC*, pages 25–32. ACM, 1989.

Oded Goldreich and Avi Wigderson. Derandomization that is rarely wrong from short advice that is typically good. In *RANDOM*, volume 2483 of *Lecture Notes in Computer Science*, pages 209–223. Springer, 2002.

Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity for all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(3):691–729, 1991.

Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *STOC*, pages 59–68. ACM, 1986.

Dan Gutfreund and Guy N. Rothblum. The complexity of local list decoding. In *APPROX-RANDOM*, volume 5171 of *Lecture Notes in Computer Science*, pages 455–468. Springer, 2008.

Dan Gutfreund and Amnon Ta-Shma. Worst-case to average-case reductions revisited. In *APPROX-RANDOM*, volume 4627 of *Lecture Notes in Computer Science*, pages 569–583. Springer, 2007.

Dan Gutfreund, Ronen Shaltiel, and Amnon Ta-Shma. Uniform hardness versus randomness tradeoffs for arthur-merlin games. *Computational Complexity*, 12(3-4):85–130, 2003.

Dan Gutfreund, Ronen Shaltiel, and Amnon Ta-Shma. If NP languages are hard on the worst-case, then it is easy to find their hard instances. *Computational Complexity*, 16(4):412–441, 2007.

Danny Harnik and Moni Naor. On the compressibility of *NP* instances and cryptographic applications. *SIAM J. Comput.*, 39(5):1667–1713, 2010.

Russell Impagliazzo and Avi Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *STOC*, pages 220–229. ACM, 1997.

Russell Impagliazzo and Avi Wigderson. Randomness vs time: Derandomization under a uniform assumption. *J. Comput. Syst. Sci.*, 63(4):672–688, 2001.

Mark Jerrum, Leslie G. Valiant, and Vijay V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theor. Comput. Sci.*, 43:169–188, 1986.

Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. How to delegate computations: the power of no-signaling proofs. In *STOC*, pages 485–494. ACM, 2014.

Adam Klivans and Dieter van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM J. Comput.*, 31(5):1501–1526, 2002.

Richard J. Lipton. New directions in testing. In *Distributed Computing And Cryptography*, volume 2 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 191–202. DIMACS/AMS, 1989.

Chi-Jen Lu, Shi-Chun Tsai, and Hsin-Lung Wu. Impossibility results on weakly black-box hardness amplification. In *FCT*, volume 4639 of *Lecture Notes in Computer Science*, pages 400–411. Springer, 2007.

Chi-Jen Lu, Shi-Chun Tsai, and Hsin-Lung Wu. On the complexity of hardness amplification. *IEEE Transactions on Information Theory*, 54(10):4575–4586, 2008.

Peter Bro Miltersen and N. V. Vinodchandran. Derandomizing arthur-merlin games using hitting sets. *Computational Complexity*, 14(3):256–279, 2005.

Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994.

Igor Carboni Oliveira and Rahul Santhanam. Majority is incompressible by acˆ0[p] circuits. In *Conference on Computational Complexity*, volume 33 of *LIPIcs*, pages 124–157. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

Ronen Shaltiel. Recent developments in explicit constructions of extractors. *Bulletin of the EATCS*, 77:67–95, 2002.

Ronen Shaltiel. Weak derandomization of weak algorithms: Explicit versions of yao's lemma. *Computational Complexity*, 20(1):87–143, 2011a.

Ronen Shaltiel. An introduction to randomness extractors. In *ICALP (2)*, volume 6756 of *Lecture Notes in Computer Science*, pages 21–41. Springer, 2011b.

Ronen Shaltiel and Christopher Umans. Simple extractors for all min-entropies and a new pseudo-random generator. *J. ACM*, 52(2):172–216, 2005.

Ronen Shaltiel and Christopher Umans. Pseudorandomness for approximate counting and sampling. *Computational Complexity*, 15(4):298–341, 2006.

Ronen Shaltiel and Christopher Umans. Low-end uniform hardness versus randomness tradeoffs for AM. *SIAM J. Comput.*, 39(3):1006–1037, 2009.

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

Michael Sipser. A complexity theoretic approach to randomness. In *STOC*, pages 330–335. ACM, 1983.

Larry J. Stockmeyer. The complexity of approximate counting (preliminary version). In *STOC*, pages 118–126. ACM, 1983.

Madhu Sudan, Luca Trevisan, and Salil P. Vadhan. Pseudorandom generators without the XOR lemma. *J. Comput. Syst. Sci.*, 62(2):236–266, 2001.

Amnon Ta-Shma and David Zuckerman. Extractor codes. *IEEE Transactions on Information Theory*, 50(12):3015–3025, 2004.

Luca Trevisan and Salil P. Vadhan. Extracting randomness from samplable distributions. In *FOCS*, pages 32–42. IEEE Computer Society, 2000.

Luca Trevisan and Salil P. Vadhan. Pseudorandomness and average-case complexity via uniform reductions. *Computational Complexity*, 16(4):331–364, 2007.

Umesh V. Vazirani. Strong communication complexity or generating quasirandom sequences form two communicating semi-random sources. *Combinatorica*, 7(4):375–392, 1987.

Emanuele Viola. The complexity of constructing pseudorandom generators from hard functions. *Computational Complexity*, 13(3-4):147–188, 2005.

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