Incompressible Functions, Relative-Error Extractors, and the Power of Nondeterminsitic Reductions (Extended Abstract)*

Benny Applebaum[1], Sergei Artemenko[2], Ronen Shaltiel[2], and Guang Yang[3]

1 **Tel Aviv University, Tel Aviv, Israel,** bennyap@post.tau.ac.il
2 **Haifa University, Haifa, Israel,** sartemen@gmail.com, ronen@cs.haifa.ac.il
3 **Tsinghua University, Beijing, China,** guang.research@gmail.com

Abstract

A circuit C *compresses* a function $f : \{0,1\}^n \to \{0,1\}^m$ 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 $C : \{0,1\}^n \to \{0,1\}^\ell$ 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 $\mathbf{E}$ is hard for exponential size nondeterministic circuits, we construct a polynomial time computable *boolean* function $f : \{0,1\}^n \to \{0,1\}$ 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 $\mathbf{E}$ is hard for exponential size Σ_3-circuits, we construct a nonboolean function $f : \{0,1\}^n \to \{0,1\}^m$ 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 $f : \{0,1\}^n \to \{0,1\}$ 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 retrospective, the limitations of existing constructions. Our impossibility result builds on an approach of Shaltiel and Viola (STOC 08).

1998 ACM Subject Classification F.1.2 Modes of Computation

Keywords and phrases compression, pseudorandomness, extractors, nondeterministic reductions

Digital Object Identifier 10.4230/LIPIcs.CCC.2015.p

* Full version appears in ECCC, TR15-051.

30th Conference on Computational Complexity (CCC'15).
Editor: David Zuckerman; pp. 1–19
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

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. The reader is referred to [1] for a full version of the paper.

1.1 Incomputable functions and Pseudorandom generators

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

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

- A function $f : \{0,1\}^n \rightarrow \{0,1\}^m$ is **incomputable** by a class $\mathcal{C}$ of functions if f is not contained in $\mathcal{C}$. We say that f is **ϵ-incomputable** by $\mathcal{C}$ if for every function $C : \{0,1\}^n \rightarrow \{0,1\}^m$ in $\mathcal{C}$, $\Pr_{x \leftarrow U_n}[C(x) = f(x)] \leq \frac{1}{2^m} + \epsilon$.
- A function $G : \{0,1\}^r \rightarrow \{0,1\}^n$ is an **ϵ-PRG** for a class $\mathcal{C}$ of functions if for every function $C : \{0,1\}^n \rightarrow \{0,1\}$ 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 problem L in $\mathrm{E} = \mathrm{DTIME}(2^{O(n)})$ 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** ([30, 34, 6, 25, 44]). *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 $f : \{0,1\}^r \rightarrow \{0,1\}$ that is n^{-c}-incomputable for size n^c circuits. Furthermore, f is computable in time poly(n^c).*[1]
- *There is a function $G : \{0,1\}^r \rightarrow \{0,1\}^n$ 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

[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.

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 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 [12]. We also consider the notion of extractors for samplable distributions introduced by Trevisan and Vadhan [47].

1.2.1 Incompressible functions

1.2.1.1 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 introduced by Harnik and Naor [24] who studied the case where f is an NP-hard function. (Similar notions were also studied by the Parameterized Complexity community, see [24] for references.) Following Dubrov and Ishai [12], we focus on a scaled-down version of the problem where the gap between the complexity of f to the 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. It is more useful to define the notion of "incompressibility" rather than "compressibility". In the following, the reader should think of $m < \ell < n$.

▶ **Definition 1.4** (incompressible function [12]). 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 $D : \{0,1\}^\ell \to \{0,1\}^m$, 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.

1.2.1.2 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 [28] 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 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 ([15, 9, 2]) 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. [3, Section 6]).

1.2.1.3 Leakage-resilient storage [10].

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 an 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. (See the full version [1] for a formal definition.)

Unfortunately, so far no explicit constructions of incompressible functions (based on “standard assumptions”) are known, even in the worst-case setting.

1.2.2 PRGs for nonboolean circuits

Dubrov and Ishai [12] 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 $\ell \le r < n$.

▶ **Definition 1.5** (PRG for boolean and nonboolean distinguishers [12]). 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.[3] 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 $C(U_n)$.

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

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

[3] 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, $|\Pr[X \in A] - \Pr[Y \in A]| \le \epsilon$.

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 [35, 36] 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 $H_\infty(X) \geq k$, $E(X)$ is ϵ-close to uniform.[4]

Trevisan and Vadhan [47] considered extractors for the class of distributions samplable by small circuits (e.g., distributions samplable by circuits of size n^2).[5] 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 (say by 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 the random string with advantage $\geq \epsilon$.

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 *non-deterministic* 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

[4] For a distribution X over $\{0,1\}^n$, $H_\infty(X) := \min_{x \in \{0,1\}^n} \log \frac{1}{\Pr[X=x]}$.

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

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.[6]

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 [13, 29, 33, 47, 37, 16, 23, 38, 8, 39, 11]. 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 $\mathrm{E} = \mathrm{DTIME}(2^{O(n)})$ 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 $\mathrm{EXP} \neq \mathrm{NP}$ or $\mathrm{EXP} \neq \Sigma_2^{\mathrm{P}}$ (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.[7]

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

▶ **Theorem 1.10** ([25, 29, 37, 38]). *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 $\mathrm{poly}(n^c)$-time computable, and have error $\epsilon = n^{-c}$.[8]

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.

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

[7] 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.

[8] Historically, the interest in PRGs for nondeterministic/NP circuits was motivated by the goal of proving that $\mathrm{AM} = \mathrm{NP}$, which indeed follows using sufficiently strong PRGs [29, 33, 37, 38]. 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 $\mathrm{poly}(n^c)$.

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 function 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 $f : \{0,1\}^n \to \{0,1\}$ 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 $f : \{0,1\}^n \to \{0,1\}^{n-\ell-d\log n}$, 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 the full version [1].)

1.4.2 A construction of PRGs for nonboolean circuits

Dubrov and Ishai [12] showed that incompressible functions imply PRGs for nonboolean distinguishers. More precisely, they used the analysis of the Nisan-Wigderson generator [34] 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 $G : \{0,1\}^{r=O(\ell^2)} \to \{0,1\}^n$ 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 is significant in the application of reducing the randomness of sampling procedures (as explained in detail by Artemenko and Shaltiel [5]).

Artemenko and Shaltiel [5] constructed PRGs 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 non-deterministic 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 nb-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. [40, 4]. (We give a more precise account of these results in the full version [1]).

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 [13] on hardness of the permanent. More relevant to our setup are the following two results by Trevisan and Vadhan [47], and Drucker [11], 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 [47]). *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).*

▶ **Theorem 1.14** (Nonboolean incomputable function with negligible error (corollary of [11])[9]). *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).*[10]

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 [4] 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 the full version [1].

Nondeterministic reductions are very powerful and previous limitations on reductions [40, 4] 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).

[9] Drucker [11] 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.

[10] 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 $\ell = n^{\Omega(1)}$.

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 even 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 reduction (that is explained in the full version [1]) is sufficiently general to capture all known proofs in the literature on hardness amplification and PRGs.[11] This is a startling contrast between boolean and non-boolean hardness amplification - the latter can achieve negligible error, while the former cannot.[12] 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 the full version [1].

▶ 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 Σ_i-circuits implies that for $\epsilon = n^{-\omega(1)}$, there is a poly($n$)-time computable:

- ϵ-incomputable functions $f : \{0,1\}^n \to \{0,1\}$ 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 $k \leq n-1$).

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 [20, 40] (see [4] for a discussion) and so the model of nondeterministic reductions that we consider is very strong.

[11] It should be noted that there are proof techniques (see e.g. [21, 22]) that bypass analogous limitations in a related setup. See [22] for a discussion.

[12] Another contrast between boolean and nonboolean hardness amplification was obtained by Shaltiel and Viola [40] for reductions that are non-adaptive constant depth circuits, and the reasons for the current contrast, are similar. Our proof follows the strategy of [40] as explained in detail in Section 2.

1.5.2.1 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. There is a superficial similarity to our work in that some of these results [48, 31, 32] 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. Specifically, we want to capture cases in which the hardness assumption is for a function in exponential time. Typical polynomial time constructions use the hardness assumption on inputs of length $O(\log n)$ where n is the input length of the target function (so that the initial function is computable in time polynomial in n) and this allows the construction to inspect the entire truth table of the function in the hardness assumption. All previous limitations on the complexity of the *construction* trivially do not hold in this setting. We elaborate on our model and the meaning of our results in the full version [1].

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. 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 [47, 11] 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 [47]). *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).*[13]

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 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.

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 $\mu(A) = |A|/2^m$.[14]

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

[13] In [47], 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 [47] 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.

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

t-wise independent functions. Specifically, the following theorem was implicitly proven by Trevisan and Vadhan [47] (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).*

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

Shaltiel [41] introduced a notion of "recognizable distributions".

▶ **Definition 1.22** (Recognizable distributions [41]). 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 $\{x : C(x) = 1\}$.

It is easy to see that 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 n^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 n^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, 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.
Our next result is a construction of a relative-error extractor for recognizable distributions.

▶ **Theorem 1.24** (Extractors for 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 recognizable by size n^c circuits. Furthermore, E is computable in time poly(n^c).*

1.7.4.1 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.

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. [47]) 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 w.h.p. 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. 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 $(1-n^{-c}/2)$-fraction of seeds $y \in \{0,1\}^n$, $f(y,\cdot) = h_{G(y)}(\cdot)$ 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 $\mathrm{AM} \subseteq \mathrm{NP/poly}$, it is sufficient to design an Arthur-Merlin protocols P, and furthermore by [7, 19] 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.[15]

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 function 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).

This approach of explicit construction by using PRGs to derandomize a probabilistic construction was suggested in full generality by Klivans and van Melkebeek [29], and was used in many relevant works such as [38, 5]. However, the use of AM protocols with *private coins* enables 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 [5].

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 poly(n^c)-wise independent family of hash functions $H = \left\{h_s : \{0,1\}^{2\ell} \to \{0,1\}^n\right\}$. We show that w.h.p. a random $h_s \leftarrow H$ is an (ℓ, ϵ)-PRG with very small ϵ. (This follows because by a standard calculation, w.h.p, h_s is a $(\epsilon \cdot 2^{-\ell})$-PRG for size n^c, and this easily implies that it is an (ℓ, ϵ)-PRG [5]). Our final PRG is again $G'(x, y) = h_{G(y)}(x)$ for the same PRG G as in the previous section.

Following our earlier strategy, it is sufficient to design a constant round, private coin 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.

We now present such a protocol, that is similar in spirit to the graph non-isomorphism protocol [18]. 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 $x \leftarrow U_{2\ell}$,

[15] 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 classed under uniform assumption [26, 46, 23, 39], but unfortunately, the relevant results only give hitting set generators, and using these we can only get incompressible function with $\epsilon = 1 - n^{-O(t)}$.

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 $\frac{1}{2} + \epsilon/2$.

2.3 The power and limitations of nondeterministic reductions

The precise definitions of nondeterministic reductions and formal restatement of Theorem 1.15 appears in the full version [1]. 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 [43, 42, 27])**.** *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 task distinguish if $m \geq \log(1/\epsilon)$, but it cannot distinguish 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, Saxe and Sipser [14], to show that even reductions that are Σ_i-circuits, cannot approximately count with the 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 [40].

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 [47], together with some key ideas that allow us to get relative error. The full proof appears in the full version [1].

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 the Goldreich-Levin theorem [17] or more generally, 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 deocdability. One way to get such behavior is to use "extractor codes" (defined by Ta-Shma and Zuckerman [45]). More precisely, to set $g(x,i) = T(f(x),i)$ 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 [47] (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 samplable by NP-circuits. This follows from Theorem 1.24 just the same, 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.

Acknowledgement

We thank Yuval Ishai for helpful discussions. 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.

References

1 Benny Applebaum, Sergei Artemenko, Ronen Shaltiel, and Guang Yang. Incompressible functions, relative-error extractors, and the power of nondeterminsitic reductions. *Electronic Colloquium on Computational Complexity (ECCC)*, 15(51), 2015.

2 Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. From secrecy to soundness: Efficient verification via secure computation. In *ICALP*, pages 152–163, 2010.

3 Benny Applebaum, Yuval Ishai, Eyal Kushilevitz, and Brent Waters. Encoding functions with constant online rate or how to compress garbled circuits keys. In *CRYPTO*, pages 166–184, 2013.

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

5 Sergei Artemenko and Ronen Shaltiel. Pseudorandom generators with optimal seed length for non-boolean poly-size circuits. In *Symposium on Theory of Computing, STOC*, pages 99–108, 2014.

6 L. Babai, L. Fortnow, N. Nisan, and A. Wigderson. Bpp has subexponential time simulations unless exptime has publishable proofs. *Computational Complexity*, 3:307–318, 1993.

7 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.

8 B. Barak, S. J. Ong, and S. P. Vadhan. Derandomization in cryptography. *SIAM J. Comput.*, 37(2):380–400, 2007.

9 Kai-Min Chung, Yael Kalai, and Salil Vadhan. Improved delegation of computation using fully homomorphic encryption. In *CRYPTO*, pages 483–501, 2010.

10 Francesco Davì, Stefan Dziembowski, and Daniele Venturi. Leakage-resilient storage. In *Security and Cryptography for Networks, 7th International Conference, SCN 2010*, pages 121–137, 2010.

11 Andrew Drucker. Nondeterministic direct product reductions and the success probability of SAT solvers. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 736–745, 2013.

12 B. Dubrov and Y. Ishai. On the randomness complexity of efficient sampling. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 711–720, 2006.

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

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

15 Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In *CRYPTO*, pages 465–482, 2010.

16 O. Goldreich and A. Wigderson. Derandomization that is rarely wrong from short advice that is typically good. In *APPROX-RANDOM*, pages 209–223, 2002.

17 Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 25–32, 1989.

18 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.

19 Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, pages 59–68, 1986.

20 D. Gutfreund and G. Rothblum. The complexity of local list decoding. In *12th Intl. Workshop on Randomization and Computation (RANDOM)*, 2008.

21 D. Gutfreund, R. Shaltiel, and A. 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.

22 D. Gutfreund and A. Ta-Shma. Worst-case to average-case reductions revisited. In *APPROX-RANDOM*, pages 569–583, 2007.

23 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.

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

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

26 R. Impagliazzo and A. Wigderson. Randomness vs. time: De-randomization under a uniform assumption. In *39th Annual Symposium on Foundations of Computer Science*. IEEE, 1998.

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

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

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

30 R. Lipton. New directions in testing. In *Proceedings of DIMACS Workshop on Distributed Computing and Cryptography*, volume 2, pages 191–202. ACM/AMS, 1991.

31 C.-J. Lu, S.-C. Tsai, and H.-L. Wu. On the complexity of hardness amplification. *IEEE Transactions on Information Theory*, 54(10):4575–4586, 2008.

32 Chi-Jen Lu, Shi-Chun Tsai, and Hsin-Lung Wu. Impossibility results on weakly black-box hardness amplification. In *FCT*, pages 400–411, 2007.

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

34 N. Nisan and A. Wigderson. Hardness vs. randomness. *JCSS: Journal of Computer and System Sciences*, 49, 1994.

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

36 R. Shaltiel. An introduction to randomness extractors. In *Automata, Languages and Programming - 38th International Colloquium*, pages 21–41, 2011.

37 R. Shaltiel and C. Umans. Simple extractors for all min-entropies and a new pseudorandom generator. *J. ACM*, 52(2):172–216, 2005.

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

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

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

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

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

43 L. J. Stockmeyer. The complexity of approximate counting. In *STOC*, pages 118–126, 1983.

44 M. Sudan, L. Trevisan, and S. P. Vadhan. Pseudorandom generators without the xor lemma. *J. Comput. Syst. Sci.*, 62(2):236–266, 2001.

45 A. Ta-Shma and D. Zuckerman. Extractor codes. In *STOC*, 2001.

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

47 L. Trevisan and S. P. Vadhan. Extracting randomness from samplable distributions. In *41st Annual Symposium on Foundations of Computer Science*, pages 32–42, 2000.

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