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

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

Benny Applebaum Sergei Artemenko Ronen Shaltiel Guang Yang

November 27, 2014

Abstract

We say that 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 $x \leftarrow \{0,1\}^n$, 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 E is hard for almost exponential size nondeterministic circuits, we give polynomial time computable incompressible functions f for size n^c circuits. We can achieve a boolean f with $\ell = (1 - o(1)) \cdot n$ and $\epsilon = n^{-c}$. This result follows by starting with a simple probabilistic construction based on poly(n)-wise independent hash functions, and then derandomizing it using sufficiently strong PRGs for nondeterministic circuits. A similar approach allows to construct PRGs against nonboolean circuits improving and simplifying the previous construction of Shaltiel and Artemenko (STOC 14).

(2) Given the above results, it is natural to try and improve the error parameter ϵ to be a negligible function $n^{-\omega(1)}$, either by some form of hardness amplification or via a direct construction. We show that this goal cannot be achieved by "existing proof techniques". Namely, *nondeterministic reductions* cannot get $\epsilon = n^{-\omega(1)}$ for *boolean* incompressible functions. This holds even if the assumption is strengthened to allow circuits with oracle access to the polynomial time hierarchy. Our results also apply to constructions of standard Nissan-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).

(3) Our lower-bound raises the question of constructing nonboolean incompressible functions with negligible error $\epsilon = n^{-\omega(1)}$, an object which will be useful for our cryptographic applications. We show that that this is indeed possible by constructing (under strong, yet plausible assumption) a nonboolean function $f : \{0,1\}^n \to \{0,1\}^m$ which is incompressible by n^c circuits with $\ell = \Omega(n)$ and extremely small $\epsilon = n^{-c} \cdot 2^{-m}$. Our construction is based on a new notion of *relative error* deterministic extractors. Roughly speaking, the output distribution of such an extractor should assign to every event A the same probability that A gets under the uniform distribution up to $(1 \pm \epsilon)$ multiplicative error. This means that even if the error ϵ is noticeable, events that receive negligible probability under the uniform distribution cannot become noticeable under the output distribution of the extractor. This notion can be seen as a way to beat the aforementioned lower bounds, and constructions of such extractors give nonboolean incompressible functions with very small error, that can be used in cryptographic applications. We achieve such extractors by using ideas and techniques due to Trevisan and Vadhan (FOCS 00).

ISSN 1433-8092

1 Introduction

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 $f : \{0,1\}^n \to \{0,1\}^m$ 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 $C : \{0,1\}^n \to \{0,1\}^m$ in $\mathcal{C}$, $\Pr_{x \leftarrow U_n}[C(x) = f(x)] \leq \frac{1}{2^m} + \epsilon$. A function $G : \{0,1\}^r \to \{0,1\}^n$ is an ϵ-**PRG** for a class $\mathcal{C}$ of functions 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. We sum up some of the main achievements of this line of research in the definition and theorem below.

Definition 1.2. *We say that "E is hard for exponential size circuits" 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 size $2^{\beta n}$ fail to compute the characteristic function of L on inputs of length n.*

Theorem 1.3 ([Lip91, NW94, BFNW93, IW97, STV01])**.** *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]
- *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).*

We state Theorem 1.3 allowing the input length r of 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 $r = a \log n$. However, this statement emphasizes that if $r = n^{\Omega(1)}$ then we get incomputable functions/PRGs which run in time polynomial in their input length. Furthermore, even for $r = n^{\Omega(1)}$ we only get $\epsilon = n^{-c}$ which is polynomially small in the seed length. (This is in contrast to other settings in pseudorandomness where with seed length r, we can expect to get error $\epsilon = 2^{-\Omega(r)}$).

In this paper we consider generalizations of incomputable functions and PRGs that were introduced by Dubrov and Ishai [DI06].

1.1 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 introduced by Harnik and Naor [HN10] who studied the case where f is an NP-hard function.[2] Following Dubrov and Ishai [DI06], 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

[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.

[2] Similar notions were also studied by the Parameterized Complexity community, see [HN10] for references.

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. In the following, the reader should think of $m < \ell < n$.

Definition 1.4 (incompressible function [DI06]). *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 $D(C(x)) \neq x$. 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 [KRR14] that verifiable computation can be achieved with a 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 ([GGP10, CKV10, AIK10]) where the client is allowed to send long messages before seeing the input. Similar tight lower-bounds can be achieved for other related cryptographic tasks such as instance-hiding or garbled circuits (cf. [AIKW13, Section 6]).

Leakage-resilient storage [DDV10]. 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.[3] 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 Section B for a formal definition.)

Unfortunately, so far no explicit constructions of incompressible functions are known, even in the worst-case setting.

1.2 PRGs for nonboolean circuits

In the definition below the reader should think of $\ell \leq r < n$.

Definition 1.5 (PRG for boolean and nonboolean distinguishers). *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 $\mathcal{C}$.*

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

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

Note that a $(1, \epsilon)$-PRG is an ϵ-PRG. As explained in [DI06, AS14b], PRGs with large ℓ can be used to reduce the randomness of sampling procedures.

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 $C(U_n)$.*

We illustrate the usefulness of this notion with an example. 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^3. If we have a poly(n)-time computable (ℓ, ϵ)-PRG $G : \{0,1\}^r \to \{0,1\}^n$ against size n^4 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, and we can hope to obtain $r \ll n$. An obvious lower bound is $r \geq \ell$ (as $X = U_\ell$ is trivially samplable by size n^3 circuits.).

1.3 Extractors for samplable distributions

We also consider Extractors for samplable distributions (introduced by Trevisan and Vadhan [TV00]).

Definition 1.7 (Extractors for samplable distributions [TV00])**.** *A function $E : \{0,1\}^n \to \{0,1\}^m$ is a (k, ϵ)-**extractor for distributions samplable by** $\mathcal{C}$, where $\mathcal{C}$ is a class of functions that output n bits, if for every distribution X that is samplable by $\mathcal{C}$ and $H_\infty(X) \geq k$, $E(X)$ is ϵ-close to U_m.*[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). Furthermore, the model of samplable distributions (say by circuits of size n^3) is very general, and contains many subclasses of distributions studied in the literature on seedless extractors. Finally, Trevisan and Vadhan make the arguable 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^3), 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 $\geq \epsilon$. Note that for this application we want that the extractor to run in polynomial time.

1.4 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 is an A-circuit*

[5] In the definition above we don't set an a-priori bound on the input length of the functions in C. 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 the input length of circuits in $\mathcal{C}$.

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 derandomization [KvM02, MV05, TV00, GW02, SU05, SU06, BOV07]. Typically, the assumption used is identical to that of Definition 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.[7]

It is possible to extend Theorem 1.3 to every type of circuits considered in Defition 1.8.

Theorem 1.10 ([IW97, KvM02, SU05, SU06])**.** *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 $\epsilon = n^{-c}$.[8]

1.5 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 parameters in the theorem below are identical to that achieved in Theorem 1.3 for incomputable functions.

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).*

[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 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 $\mathrm{DTIME}^{\Sigma_i^P}(s^{\Theta(1)})/s^{\Theta(1)}$.

[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 AM $=$ NP, which indeed follows using sufficiently strong PRGs. [KvM02, MV05, SU05, SU06]. 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 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 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 B.)

1.6 A construction of PRGs for nonboolean circuits

Dubrov and Ishai [DI06] showed that incompressible functions imply PRGs for nonboolean distinguishers. More precisely, they used the analysis of the Nisan-Wigderson generator [NW94] 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 [AS14b]).

Artemenko and Shaltiel [AS14b], 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 seed length $r = O(\ell)$.

Theorem 1.12. *If E is hard for exponential size 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).*

1.7 Nondeterministic reductions

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 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. [SV10, AS14a]. (We give a more precise account of these results in Section D).

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 [FL97] on hardness of the permanent. More relevant to our setup are the following two results by Trevisan and Vadhan [TV00], and Drucker [Dru13], 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 [TV00])**.** *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 [Dru13])[9])**.** *For every*

[9] Drucker [Dru13] 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.

$c > 1$ 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 [AS14a] 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 that 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 to be a nondeterministic/Σ_i-circuit. Previous limitations on reductions [SV10, AS14a] do not hold in this case, and 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 that E is hard for exponential size Σ_i-circuits, for some $i > 0$.

1.8 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 Section D) 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.[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 Section E.

Informal Theorem 1.15. *For every $i \geq 0$, it is impossible to use "black-box reductions" to prove that the assumption that E is hard for Σ_i-circuits implies that for some sufficiently large n and $r \leq n$ there are poly(n)-time computable ϵ-incomputable functions, or ϵ-incompressible functions, or ϵ-PRGs (and in particular (ℓ, ϵ)-PRGs for any ℓ), or $(n-1, \epsilon)$-extractors for samplable distributions, with input length r and $\epsilon = n^{-\omega(1)}$. Furthermore, this holds 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 and the problem in E.*

[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 remark that our Theorem 1.11 holds also if we replace the assumption by the following: 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 $\ell = n^{\Omega(1)}$.

[11] It should be noted that there are proof techniques (see e.g. [GSTS07, GTS07]) that bypass analogous limitations in a related setup. See [GTS07] for a discussion.

[12] Another contrast between boolean and nonboolean hardness amplification was obtained by Shaltiel and Viola [SV10] for reductions that are non-adaptive constant depth circuits, and the reasons for the current contrast, are similar. Our proof follows the strategy of [SV10] as explained in detail in Section 2.

It is interesting to note that previous work on (deterministic) black-box reductions often cannot handle reductions that are both adaptive and nonuniform [GR08, SV10] (see [AS14a] for a discussion) and so the model of nondeterministic reductions that we consider is very strong.

1.9 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 $f : \{0,1\}^n \rightarrow \{0,1\}^m$ 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 we obtain that $f : \{0,1\}^n \rightarrow \{0,1\}^m$ is ϵ-incomputable by size n^c circuits, with $\epsilon = n^{-c} \cdot 2^{-m}$, meaning that circuits of size n^c, have probability at most $\frac{1+n^{-c}}{2^m}$ of computing $f(x)$. Note that in the aforementioned theorems the probability is larger than $2^{-(m/2)}$ which is large compared to 2^{-m}.

Finally, the function we get is not only ϵ-incomputable, but (ℓ, ϵ)-incompressible for large $\ell = \Omega(n)$, and this even holds 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.

1.10 Relative error extractors for samplable distributions with relative error

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 [TV00])**.** *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 $E : \{0,1\}^n \rightarrow \{0,1\}^m$ 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 1.3) 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. 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]

[13] In [TV00], 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 [TV00] 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 $\epsilon \geq 1$.

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.

In the definition below, we consider extractors for samplable distributions where the requirement that the output is ϵ-close to uniform is replaced by the requirement that the output is close to uniform with relative error.

Definition 1.19 (extractors for samplable distributions with relative error)**.** *A function $E : \{0,1\}^n \to \{0,1\}^m$ is a (k,ϵ)**-relative-error extractor for distributions samplable by** $\mathcal{C}$, if for every distribution X that is samplable by $\mathcal{C}$ and $H_\infty(X) \geq k$, $E(X)$ is ϵ-close to uniform with relative error.*

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

Theorem 1.20 (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).*

We believe that this makes extractors for samplable distributions more suitable for cryptographic applications.

2 Overview and Technique

2.1 Relative error extractors for recognizable distributions

In this section we present a high level overview of the techniques used to prove our results. Many of our results follow by considering the following notion of "relative error extractors for recognizable distributions". This notion is a variant (with relative error) of a notion of extractors for recognizable distributions introduce in [Sha11].

Definition 2.1 (Relative error extractors for recognizable distributions)**.** *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 $\{x : C(x) = 1\}$. A function $E : \{0,1\}^n \to \{0,1\}^m$ is a (k,ϵ)**-relative error extractor for distributions recognizable by** $\mathcal{C}$, if for every distribution X that is recognizable by $\mathcal{C}$ and $H_\infty(X) \geq k$, $E(X)$ is ϵ-close to U_m with relative error.*

We first observe that such extractors are incompressible functions with extremely small error.

Lemma 2.2. *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.

Application in the leakage resilient scenario. The same reasoning applies in the memory leakage scenario described in Section 1.1. 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.

2.2 Boolean incompressible functions

We now give 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. [TV00] 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 (with very small error).

By lemma 2.2, 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 AM $\subseteq$ NP/poly, it is sufficient to design an Arthur-Merlin protocols P, and furthermore by [BM88, GS86] 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).

[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 [IW98, TV07, GST03, SU09], but unfortunately, the relevant results only give hitting set generators, and using these we can only get incompressible function with $\epsilon = 1 - n^{-O(t)}$.

This approach of explicit construction by using PRGs to derandomize a probabilistic construction was suggested in full generality by Klivans and van Melkebeek [KvM02], and was used in many relevant works such as [SU06, AS14b]. 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 [AS14b].

2.3 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 $H = \left\{h_s : \{0,1\}^{2\ell} \to n\right\}$. 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 [AS14b]). 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 [GMW91]. 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 G 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 a 2ϵ-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 ϵ-PRG, then the distributions are ϵ-close and Merlin cannot distinguish with probability larger than $\frac{1}{2} + \epsilon/2$.

2.4 The power and limitations of nondeterministic reductions

The precise definitions of nondeterministic reductions and formal restatement of Theorem 1.15 appears in Section D. Below, we try to intuitively explain what makes nondterministic 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 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 2.3 (approximate counting [Sto83, Sip83, JVV86])**.** *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 ϵ'-approximation of $|\{x : C(x) = 1\}|$ (where we say that p is an ϵ-approximation of q if $(1 - \epsilon) \cdot p \leq q \leq (1 + \epsilon) \cdot p$).*

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 1.20 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 $f : \{0,1\}^n \to \{0,1\}^m$. 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 cannot succeed).

This explains why approximate counting is in some sense *necessary* for reductions that want to achieve negligible error. Using an argument similar to that of Furst, Saxe and Sipser [FSS84], we can 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 for a task that resembles majority, and cannot be computed by too small circuits with constant depth. This relationship uses ideas from [SV10].

2.5 Constructing relative error extractors for recognizable distributions

We now present a construction of extractors for recognizable distributions.

Theorem 2.4 (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 $m \leq \alpha n$ 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).*

In order to prove Theorem 2.4 we use tools and techniques from Trevisan and Vadhan [TV00], together with some key ideas that allow us to get relative error. The proof appears in Section E. 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). 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 [GL89] 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 very efficient list-decoding. In the case that $m > 1$, and we want to make the output "pseudorandom" then we need EC to be an extractor-code [TSZ01], that is $g(x,i) = T(f(x),i)$ 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".

It is known that seeded extractors cannot output m bits with error $< 2^{-(m/2)}$ [RTS00]. Instead, we suggest to use 2-source extractors, that can achieve error $\ll 2^{-m}$. In particular, if applied with error $\epsilon \ll 2^{-m}$, such extractors achieve "relative error", meaning that the probability of every output string is between $2^{-m} + \epsilon = (1 - \epsilon \cdot 2^m) \cdot 2^{-m}$ and $2^{-m} + \epsilon = (1 + \epsilon \cdot 2^m) \cdot 2^{-m}$.

The intuition outlined above doesn't seem sufficient to prove that g is incomputable. However, we observe that such extractors can be used as "inner codes" in the approach of [TV00] (which can be viewed as a more specialized concatenation of codes). In order to be used in the reduction of [TV00], these "codes" need to have efficient "(list)-decoding procedures", where in this setup "efficient" means polynomial size NP-circuits. For this, 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 2.4.

2.6 Relative error extractors for samplable distributions

We now explain how to construct relative error extractors for samplable distributions and prove Theorem 1.20. 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 2.4 just the same, if in the assumption we assume hardness for Σ_4-circuits, instead of Σ_3-circuits. This 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 1.20 follows from the (generalized version) of Theorem 2.4.

References

[AIK10] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. From secrecy to soundness: Efficient verification via secure computation. In *ICALP*, pages 152–163, 2010.

[AIKW13] 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.

[Ajt83] Miklós Ajtai. Σ_1^1-formulae on finite structures. *Ann. Pure Appl. Logic*, 24(1):1–48, 1983.

[AS14a] 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.

[AS14b] 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.

[BFNW93] 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.

[BGP00] M. Bellare, O. Goldreich, and E. Petrank. Uniform generation of np-witnesses using an np-oracle. *Inf. Comput.*, 163(2):510–526, 2000.

[BM88] 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.

[BOV07] B. Barak, S. J. Ong, and S. P. Vadhan. Derandomization in cryptography. *SIAM J. Comput.*, 37(2):380–400, 2007.

[BR94] Mihir Bellare and John Rompel. Randomness-efficient oblivious sampling. In *35th Annual Symposium on Foundations of Computer Science*, pages 276–287, 1994.

[CG88] B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, April 1988.

[CKV10] Kai-Min Chung, Yael Kalai, and Salil Vadhan. Improved delegation of computation using fully homomorphic encryption. In *CRYPTO*, pages 483–501, 2010.

[DDV10] 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.

[DEOR04] Y. Dodis, A. Elbaz, R. Oliviera, and R. Raz. Improved randomnness extractors from two independent sources. In *Proccedings of thw 8th International Workshop on Randomization and Computation*, 2004.

[DI06] 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.

[Dru13] 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.

[FL97] Uriel Feige and Carsten Lund. On the hardness of computing the permanent of random matrices. *Computational Complexity*, 6(2):101–132, 1997.

[FSS84] Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.

[GGP10] Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In *CRYPTO*, pages 465–482, 2010.

[GL89] 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.

[GMW91] 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.

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

[GS86] 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.

[GST03] 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.

[GSTS07] 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.

[GTS07] D. Gutfreund and A. Ta-Shma. Worst-case to average-case reductions revisited. In *APPROX-RANDOM*, pages 569–583, 2007.

[GW02] O. Goldreich and A. Wigderson. Derandomization that is rarely wrong from short advice that is typically good. In *APPROX-RANDOM*, pages 209–223, 2002.

[HN10] Danny Harnik and Moni Naor. On the compressibility of $\mathcal{NP}$ instances and cryptographic applications. *SIAM J. Comput.*, 39(5):1667–1713, 2010.

[IW97] R. Impagliazzo and A. Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *STOC*, pages 220–229, 1997.

[IW98] 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.

[JVV86] 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.

[KRR14] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. How to delegate computations: the power of no-signaling proofs. In *STOC*, 2014.

[KvM02] 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.

[Lip91] R. Lipton. New directions in testing. In *Proceedings of DIMACS Workshop on Distributed Computing and Cryptography*, volume 2, pages 191–202. ACM/AMS, 1991.

[MV05] P. Bro Miltersen and N. V. Vinodchandran. Derandomizing arthur-merlin games using hitting sets. *Computational Complexity*, 14(3):256–279, 2005.

[NW94] N. Nisan and A. Wigderson. Hardness vs. randomness. *JCSS: Journal of Computer and System Sciences*, 49, 1994.

[RTS00] J. Radhakrishnan and A. Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM Journal on Discrete Mathematics*, 13(1):2–24, 2000.

[Sha11] Ronen Shaltiel. Weak derandomization of weak algorithms: Explicit versions of yao's lemma. *Computational Complexity*, 20(1):87–143, 2011.

[Sip83] M. Sipser. A complexity theoretic approach to randomness. In *STOC*, pages 330–335, 1983.

[Sto83] L. J. Stockmeyer. The complexity of approximate counting. In *STOC*, pages 118–126, 1983.

[STV01] M. Sudan, L. Trevisan, and S. P. Vadhan. Pseudorandom generators without the xor lemma. *J. Comput. Syst. Sci.*, 62(2):236–266, 2001.

[SU05] R. Shaltiel and C. Umans. Simple extractors for all min-entropies and a new pseudorandom generator. *J. ACM*, 52(2):172–216, 2005.

[SU06] R. Shaltiel and C. Umans. Pseudorandomness for approximate counting and sampling. *Computational Complexity*, 15(4):298–341, 2006.

[SU09] R. Shaltiel and C. Umans. Low-end uniform hardness versus randomness tradeoffs for am. *SIAM J. Comput.*, 39(3):1006–1037, 2009.

[SV10] R. Shaltiel and E. Viola. Hardness amplification proofs require majority. *SIAM J. Comput.*, 39(7):3122–3154, 2010.

[TSZ01] A. Ta-Shma and D. Zuckerman. Extractor codes. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, 2001.

[TV00] L. Trevisan and S. P. Vadhan. Extracting randomness from samplable distributions. In *41st Annual Symposium on Foundations of Computer Science*, pages 32–42, 2000.

[TV07] L. Trevisan and S. Vadhan. Pseudorandomness and average-case complexity via uniform reductions. *Computational Complexity*, 16(4):331–364, 2007.

[Vaz87] U. Vazirani. Strong communication complexity or generating quasi-random sequences from two communicating semi-random sources. *Combinatorica*, 7:375–392, 1987.

[Vio06] Emanuele Viola. *The Complexity of Hardness Amplification and Derandomization*. PhD thesis, Harvard University, 2006. http://www.eccc.uni-trier.de/eccc.

A Approximate counting and uniform sampling of NP witnesses

We use the classical result on approximate counting and uniform sampling of NP-witnesses [Sto83, Sip83, JVV86, BGP00], which we now state in a way that is convenient for our application.

Definition A.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 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.

Theorem A.2 (approximate counting [Sto83, Sip83, JVV86]). *For every $i \geq 0$, every sufficiently large s and every $\epsilon > 0$, there is a size poly(s/ϵ) Σ_{i+1}-circuit that given a Σ_i-circuit C of size s outputs an ϵ-approximation of* $|\{x : C(x) = 1\}|$.

Theorem A.3 (uniform sampling [JVV86, BGP00]). *For every $i \geq 0$, every sufficiently large s and every $\delta > 0$, there is a randomized size poly$(s/log(1/\delta))$ Σ_{i+1}-circuit A that given a Σ_i-circuit $C : \{0,1\}^n \to \{0,1\}$ 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 $\{x : C(x) = 1\}$.*

B 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 against s-size circuits. This collection is efficiently computable but large (contains $2^{\text{poly}(s)}$ members). This step is based on a simple connection between incompressible functions to extractors against recognizable distributions, and on the fact that t-wise independent hash functions are good extractors. At then 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 non-deterministic 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 by 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.

Definition B.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 his oracle a total number of ℓ bits, and every oracle $D : \{0,1\}^* \to \{0,1\}^*$, we have that $\Pr_{x \leftarrow U_n}[D(C(x)) = f(x)] \le \frac{1}{2^m} + \epsilon$.*

Clearly, interactive incompressibility implies non-interactive incompressibility (with the same parameters).

Incompressible Functions from Extractors. We continue by relating incompressible functions (in the interactive setting) to extractors for recognizable distributions.

Lemma B.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 $(n - \ell - \Delta_1, \epsilon')$ 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 $(k, \epsilon' 2^m)$-relative-error extractor for s'-size recognizable distributions.*

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

Proof. 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 $\Pr_x[C^D(x) = f(x)] \ge 2^{-m} + \epsilon$. 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 $x \leftarrow \{0,1\}^n$ induce 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 $w(a) \ge \epsilon \cdot 2^{-\ell-1}$ and $\sigma(a) \ge 2^{-m} + \epsilon/2$. Indeed, we can write

$$2^{-m} + \epsilon \le \Pr_x[C^D(x) = f(x)] = \sum_{a : w(a) < \epsilon \cdot 2^{-\ell-1}} w(a) \cdot \sigma(a) + \sum_{a : w(a) \ge \epsilon \cdot 2^{-\ell-1}} w(a) \cdot \sigma(a),$$

since the first summand is upper-bounded by $\epsilon \cdot 2^{-\ell-1} \cdot 2^\ell \le \epsilon/2$, the second summand must be at least $1/2 + \epsilon/2$, 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 $w(a) \ge \epsilon \cdot 2^{-\ell-1}$, 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 $s' \le 2s$. (Indeed, compute $C(x)$, with the answers b hardwired, and accept if all the queries to D are consistent with t, i.e., if $C_1(x) = a_1$, and $C_i(x, b_{i-1}) = a_i$ for every $1 < i \le q$). Finally, we have, by assumption, that $\Pr[f(X) = y] \ge 2^{-m} + \epsilon/2$, 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. □

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

Definition B.3 (t-wise independent hashing)**.** *Let $\mathcal{H} = \{h_z : \{0,1\}^n \to \{0,1\}^m\}$ be a collection of functions indexed by $\tau = poly(n)$-bit identifiers $z \in \{0,1\}^\tau$. We say that $\mathcal{H}$ is 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 random choice of $z \leftarrow \{0,1\}^\tau$ are uniformly distributed in $(\{0,1\}^m)^t$. We assume that the collection is efficiently computable, namely, 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 $y = h_z(x)$.*

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 [TV00, Proposition A.1] show that t-wise independent hash functions $\mathcal{H}$ form a good collection of extractors against any fixed family $\mathcal{C}$ of N sources, provided that t is sufficiently large compared to N. Concretely, their analysis yields the following fact.

Proposition B.4 (Collections of Extractors from Hashing [TV00])**.** *Fix a family $\mathcal{C}$ of N sources over n-bit string. Let k be some entropy bound, $\delta \in (0,1)$ be an error parameter, and let $\mathcal{H} = \{h_z\}$ be a family of t-wise independent hash functions that maps n bits to m bits where $t = 2\log(k + N + 1/\delta)$ and $m = k - 2\log(1/\epsilon) - \log t - 2$. Then $\mathcal{H}$ is a collection of (k,ϵ) extractors against $\mathcal{C}$ with failure probability δ. Moreover, the above holds even for relative-error extractors (with the same parameters).*

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 B.5. *Let $s = n^c, \epsilon = n^{-c}$ for some arbitrary constant $c > 1$, and let $k = k(n) \leq n$ be be some entropy bound which is lower-bounded by $\Delta_2 = 3c\log n - \log\log n + \Omega(1)$. 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)-extractor 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 $n^{O(c)}$.*

Partial Derandomization: Obtaining Small Collection From now on, we fix some constant $c > 1$, and set $s = 2n^c$, $\epsilon = 1/s$, $\Delta_1 = n - \ell - c\log n - 1$ as in Lemma B.2 and $\Delta_2 = 3c\log n - \log\log n + \Omega(1)$ as in Proposition B.5. We also choose some positive $\ell(n) \leq n - (\Delta_1 + \Delta_2) = n - 4c\log n - \log\log n + \Omega(1)$, and let $k = n - \ell - \Delta_1$ and $m = k - (\Delta_1 + \Delta_2)$. Let $H_{c,\ell} : \{0,1\}^\tau \times \{0,1\}^n \to \{0,1\}^m$ be the evaluation algorithm that satisfies Proposition B.5 with respect to the above parameters, and note that by Lemma B.2 all but $1 - 2^{-n}$ 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 B.5.1. *There exists a non-deterministic 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 (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 Goldwasser-Sipser [GS86] and Babai-Moran [BM88] transformations, we get that Π has a two-message public-coin AM protocol, which, in turn, means that Π has a non-deterministic circuit of poly(n)-size. □

Recall that, by Lemma B.2, all but a 2^{-n} fraction of the τ-bit strings z are NO-instances. As a result we obtain the following proposition.

Proposition B.6. *Let $G : \{0,1\}^r \to \{0,1\}^\tau$ be an n^{-b}-**PRG** for non-deterministic circuits of size n^b where $b > c$ is the constant from Claim B.5.1. Consider the ensemble $\mathcal{F}$ defined by the evaluation algorithm $f : \{0,1\}^r \times \{0,1\}^n \to \{0,1\}^m$ where*

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

Then, $\mathcal{F}$ is a collection of (ℓ, n^{-c})-incompressible functions for n^c-size circuits (in the non-interactive setting) with failure probability $2^{-n} + n^{-2c}$.

From Small Collection to a Single Function Finally, we need the following simple observation.

Claim B.6.1 (Combining incompressible functions). *If $\mathcal{F} = \{f_z : \{0,1\}^n \to \{0,1\}^m\}_{z\in\{0,1\}^\tau}$ is a collection of (ℓ,ϵ)-incompressible 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, there exists a string z for which: (1) The function f_z is (ℓ,ϵ)-incompressible; and (2) For randomly chosen $x \leftarrow \{0,1\}^n$ the compressor C succeeds on the input (z,x) with success probability $2^{-m} + \epsilon$. The claim follows by noting that f_z is (ℓ,ϵ)-compressed by $C(z,\cdot)$ (with respect to solver D), in contradiction to (1). □

We can now prove a stronger variant of Theorem 1.11.

Theorem B.7. *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 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. Furthermore, f is computable in time poly(n^c).*

Proof. Fix some constant $c > 1$, let $\ell(n) \le n - (\Delta_1 + \Delta_2) = n - 4c\log n - \log\log n + \Omega(1)$, 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 $b(c)$ be the corresponding constant from Claim B.5.1. By Thm. 1.10, our hypothesis implies the existence of n^{-b}-PRG $G : \{0,1\}^r \to \{0,1\}^{\tau(n)}$ against non-deterministic circuits of size n^b, where $r = a\log n$ for some constant a which depends on c. By Proposition B.6 and Claim B.6.1 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. Letting $N = r + n$ and $c' = c/2$ we have that f is $(\ell, N^{-c'})$-incompressible for $N^{c'}$-size circuits where $\ell \le N - (4c + a)\log n = N - \Omega_c(\log N)$ and $m = N - \ell - 5c\log n = N - \ell - \Omega_c(\log N)$, hence the theorem follows. □

B.1 The non-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 B.5.1 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.

Claim B.7.1. *Let $\mathcal{H} = \{h_z : \{0,1\}^n \to \{0,1\}^m\}$ be a collection of functions with an evaluation algorithm of complexity t, and let $\delta(n) < \frac{1}{2}$. Consider the promise problem Π whose YES instances are τ-bit strings for which $H(z,\cdot)$ is* not *$(k+1, 2\delta)$-relative error extractors for s-size recognizable sources, and the NO instances are τ-bit strings for which $H(z,\cdot)$ is (k,δ)-relative error extractors for s-size recognizable sources. Then there exists a nondeterministic NP-circuit A of size poly$(t, s, 1/\delta)$ that accepts the YES instance of Π and rejects its NO instances.*

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 assumptions of a poly(t, s)-size circuit. 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 larger then $1.25 \cdot 2^k$; (2) Ask for a $(1 \pm \delta^2)$ multiplicative approximation of the number of assignments x that satisfy $H(z, C(x)) = y$, and check that the result is either smaller than $2^{n-m} \cdot (1 - 1.5\delta)$ or larger than $2^{n-m} \cdot (1.5\delta)$. Such an approximation is implementable with an NP oracle [Sto83] with circuit complexity of poly$(t, s, 1/\delta)$. (Since we work in the non-uniform model we can assume that the approximation always succeeds.)

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) $p = \Pr_x[H(z, C(x)) = y] \notin (1 \pm 2\delta)2^{-m}$. Part (a) means that the set $C^{-1}(1)$ is larger than 2^{k+1} and so the approximated value must larger than $1.25 \cdot 2^k$, 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 $2^{n-m}(1 - 2\delta)$ or larger than $2^{n-m}(1 + 2\delta)$. In the first case, the approximated value will be at most $2^{n-m}(1-2\delta)(1+\delta^2)$ which is smaller than $2^{n-m} \cdot (1.5\delta)$ for $\delta < \frac{1}{2}$. In the second case, the approximated value will be at least $2^{n-m}(1+2\delta)(1-\delta^2)$ which is larger than $2^{n-m} \cdot (1.5\delta)$. 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 $p = \Pr_x[H(z, C(x)) = y] \in [2^{-m}(1 \pm \delta)]$ and so the approximation computed in the second part of the test must be in the interval $2^{n-m}(1 \pm \delta)(1 \pm \delta^2) \subset 2^{n-m}(1 \pm 1.5\delta)$ which means that the second check fails. This completes the proof of the claim. □

The rest of the argument is similar to the one used in the previous section. Specifically, we derive the following corollary. (A proof will be given in the full version).

Theorem B.8. *If E is hard for exponential size 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 interactive setting. Furthermore, f is computable in time poly(n^c).*

C Constructing nb-PRGs

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

Proposition C.1. *If G is an $(1, 2^{-\ell}\epsilon)$-PRG for size s circuits then its is also (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 C.1.1 (Collections of PRGs from Hashing)**.** *For every s and $\epsilon, \delta \in [0, 1]$ let $t = 4s \log s + 2 \log(1/\delta)$ and let $r > 2 \log(1/\epsilon) + \log t$. Then 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.*

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 [BR94, Lemma 2.2], we have

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

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 C.2. *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* $(\ell, 2n^{-c})$-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 C.2.1. *There exists a non-deterministic 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-Sipser [GS86] and Babai-Moran [BM88] transformations, we get that Π has a two-message public-coin AM protocol, which, in turn, means that Π has a non-deterministic circuit of poly(n)-size. □

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

Theorem C.3. *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 $\ell \leq n$, 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 a function which satisfies $a \log n \leq \ell \leq n$. 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 C.2.1, is recognizable by a non-deterministic circuit of size $n^{c'}$ where $c' > c$ 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 non-deterministic circuits of size $n^{c'}$, where $r' = a' \log n$ for some constant a' which depends on c'_c. Consider the function $G : \{0,1\}^{r'} \times \{0,1\}^r \to \{0,1\}^n$ defined by

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

Recall that, by Corollary C.2, 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, overall G is a $(\ell, 3n^{-c})$-PRG. Overall, G has an input 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 now follows. □

D Limitations on nondeterministic reductions

D.1 black-box hardness amplification and nondeterministic reductions

The task of converting a worst-case hard function f into an average-case hard function g if 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. This notion is formally defined below. In fact, below, we define a more general setting in which the function f that we start from in the hardness amplification result is assumed to be not only hard on the worst case, but mildly average-case hard. This makes it easier to prove hardness amplification results, and so lower bounds in this setting are more general.

Definition D.1 (black-box hardness amplification [SV10][16]). *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 $Con_f : \{0,1\}^n \to \{0,1\}$.*
- *A reduction Red is an oracle procedure $Red^{(\cdot)}(x, \alpha)$ that accepts two inputs $x \in \{0,1\}^k$ and $\alpha \in \{0,1\}^a$ which is called "nonuniform advice string". Red also receives oracle access to a function $C : \{0,1\}^n \to \{0,1\}$.*

Furthermore, for every functions $f : \{0,1\}^k \to \{0,1\}$ and $C : \{0,1\}^n \to \{0,1\}$ such that

$$\Pr_{y \leftarrow U_n}[C(y) = Con_f(y)] \geq \frac{1}{2} + \epsilon,$$

[16] We use a different notation from [SV10]. More precisely, in [SV10] 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.

there exists $\alpha \in \{0,1\}^a$ such that

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

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

Sudan, Trevisan and Vadhan [STV01] observed that if $(\mathrm{Con}, \mathrm{Red})$ is a worst-case $\rightarrow (\frac{1}{2} - \epsilon)$-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 D.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 $a = 0$).

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 $\mathcal{C}$.

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

Definition D.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 $C : \{0,1\}^n \rightarrow \{0,1\}$:*

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

where Q stands for "$\exists$" if i is odd, and for "$\forall$" if i is even.[17]

Note that the more general definition captures deterministic reductions (for $i = 0$) and nondeterministic reductions for $i = 1$. Let us discuss some aspects of Defintion D.2. This definition captures nondeterministic reductions that are used in the literature [FL97, KvM02, TV00, Dru13]. Indeed, if there is a size s, i-nondeterministic reduction Red and some construction Con such that $(\mathrm{Con}, \mathit{Red})$ are a worst-case $\rightarrow (\frac{1}{2} - \epsilon)$-black box hardness amplification, then it follows that for every s' if f is incomputable by size $s \cdot s' + a$ Σ_i-circuits, then $g = \mathrm{Con}_f$ is ϵ-incomputable by (deterministic) circuits of size s'.

Remark D.3 (Comparison of this model to previous limitations on reductions)**.** *Previous lower bounds on black-box hardness amplification [SV10, AS14a] cannot handle nondeterministic reductions. This is because previous lower bounds assume that when 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 $\mathit{Red}^C(x,\alpha)$ did not make. This indeed holds for small size deterministic reductions, as the number of queries that $\mathit{Red}^{(C}(x,\alpha)$ makes is bounded by the total size of the reductions. Note, that in contrast, nondeterministic reductions are "armed with quantifiers", and the computation of $\mathit{Red}^C(x,\alpha)$ 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 [FL97, TV00, Dru13] to bypass the limitations on deterministic reductions.*

We are interested in obtaining functions $g = \mathrm{Con}_f$ that are computable in time $\mathrm{poly}(n)$ and yet are $n^{-\omega(1)}$-incomputable by size n^c circuits. Assuming that the construction of $g = \mathit{Con}_f$ invokes f at least

[17] 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 $z_1, \ldots, z_i$.

once, it follows that f is 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), then 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.[18] This discussion motivates the choice of parameters in the next theorem, which is the formal restatement of Theorem 1.15.

Theorem D.4 (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 $\to (1/2 - \epsilon)$ black-box hardness amplification where Red is a size n^d, i-nondeterministic reduction with $\epsilon = n^{-\omega(1)}$.*

Indeed, Theorem D.4 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.

Note that in Theorem D.4 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 D.4 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.

Theorem D.5 (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 $\frac{1}{2}$.

D.2 Proof of the lower bound

In this section we prove Theorem D.5. It will be helpful to identify boolean functions $C : \{0,1\}^n \to \{0,1\}$ with their truth table $C \in \{0,1\}^{2^n}$. We need the following definition.

Definition D.6 (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 function $N_p : \{0,1\}^n \to \{0,1\}$.*

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 $C = \text{Con}_f \oplus N_{\frac{1}{2}-2\epsilon}$ (where for two functions $A, B : \{0,1\}^n \to \{0,1\}$, $A \oplus B : \{0,1\}^n \to \{0,1\}$ is defined by $(A \oplus B)(y) = A(y) \oplus B(y)$. Note that xoring with random noise with $p = \frac{1}{2} - 2\epsilon$ is extremely likely to give a function C such that $\Pr_{y \leftarrow U_n}[C(y) = \text{Con}_f(y)] \geq \frac{1}{2} + \epsilon$, on which the reduction Red must perform. This is stated precisely below.

Lemma D.7. *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 $\Pr_{x \leftarrow U_k}[\mathit{Red}^{\mathit{Con}_f \oplus N}(x, \alpha) = f(x)] \geq 1 - \delta$.*

[18] 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 $2^{O(k)} = \text{poly}(n)$.

The second oracle we consider is $C = \text{Con}_f \oplus N_{\frac{1}{2}}$. 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 D.8. *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 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, Saxe and Sipser [FSS84] 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 D.9. *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 D.2). For every fixed x, α and $z_1, \ldots, z_i$. $A^C(x, \alpha, z_1, \ldots, z_i)$ 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 input long 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 D.10. *There is a circuit B of size $2^{O(n^d)}$ and depth $i+O(1)$ 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 $C(y) = N(y) \oplus \text{Con}_f(y)$. For every $x \in \{0,1\}^k$ and $\alpha \in \{0,1\}^a$, A_f computes $B_{x,\alpha}(C)$. For every $\alpha \in \{0,1\}^k$, 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 [Ajt83]). 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 D.7, 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 D.8 and a union bound over all $\alpha \in \{0,1\}^a$ $\Pr[A_F(N_{\frac{1}{2}}) = 1] \leq 2^a \cdot 2^{-(1-H(\delta+\frac{1}{n}))\cdot 2^k} = o(1)$

by our choice of parameters. By an averaging argument, there exists an f such that A_f distinguishes the two distributions with probability $1 - o(1) \geq 0.99$. $\square$

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 [Vio06, SV10].

Theorem D.11. *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 D.5 follows.

D.3 Extending the limitations to PRGs and extractors for samplable distributions

It is not difficult to extend Definition D.1 and Theorem D.5 to the case where ϵ-incomputable functions are replaced with PRGs or extractors for samplable distributions. The details are deferred to the full version.

E Extractors for recognizable distributions and incompressible functions with low error

Our constructions of nonboolean incompressible function with low error (stated in Theorem 1.16) and of extractors for samplable distributions with relative error (stated in Theorem 1.20), both follow from a construction of *extractors for recognizable distributions with relative error.*

E.1 Relative error extractors for recognizable distributions

We generalize Definition 2.1, introducing the notion of "weakly recognizable" distributions.

Definition E.1 (recognizable distributions)**.** *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 $\{x : C(x) = 1\}$. X 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 $x \in \{0,1\}^n$, $\Pr[X = x] = \frac{C(x)}{\sum_{x \in \{0,1\}^n} C(x)}$.*[19]

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 in [Sha11]. We give a construction of extractors for weakly recognizable distributions that have relative error. This notion is formalized in the next definition (which also discusses extractors for weakly recognizable distributions).

Definition E.2 (Extractors for recognizable distributions with relative error)**.** *A function $E : \{0,1\}^n \to \{0,1\}^m$ is a (k, ϵ)-**relative-error extractor for distributions recognizable (resp. weakly recognizable) by $\mathcal{C}$**, if for every distribution X that is recognizable (resp. weakly recognizable) by $\mathcal{C}$ and $H_\infty(X) \geq k$, $E(X)$ is ϵ-close to uniform with relative error.*

We show that we can construct such extractors if E is hard for exponential size Σ_3-circuits.

Theorem E.3 (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 $m \leq \alpha n$ there is a $((1-\alpha) \cdot n, \frac{1}{n^c})$-relative error extractor $E : \{0,1\}^n \to \{0,1\}^m$ for distributions weakly recognizable by size n^c circuits. Furthermore, E is computable in time poly(n^c).*

[19] 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.

E.2 Obtaining nonboolean incompressible functions with very low error

We now observe that extractors for recognizable distributions are incompressible functions. This means that Theorem E.3 also gives a construction of incompressible functions, proving Theorem 1.16.

Lemma E.4 (extractors are incompressible functions)**.** *Let $\Delta = \ell + m + \log(1/\epsilon) + 1$. If $E : \{0,1\}^n \to \{0,1\}^m$ is an $(n - \Delta, \epsilon/2)$-relative error extractor for distributions recognizable by size $n^c + O(\ell)$ circuits, then E is an $(\ell, \epsilon \cdot 2^{-m})$-incompressible function for circuits of size n^c.*

Proof. Let $C : \{0,1\}^n \to \{0,1\}^\ell$ be some size n^c circuit. We say that $z \in \{0,1\}^\ell$ is *light* if $\Pr_{x \leftarrow U_n}[C(x) = z] \leq 2^{-\Delta}$ and *heavy* otherwise. We denote the set of light z by L. It follows that $\Pr_{x \leftarrow U_n}[C(x) \in L] \leq 2^{\ell - \Delta} \leq \epsilon \cdot 2^{-(m+1)}$. For every $z \in \{0,1\}^\ell$ we denote the uniform distribution over $\{x : C(x) = z\}$ by X_z, and note that this distribution is recognizable by a circuit C'_z of size $n^c + O(\ell)$ (which when given $x \in \{0,1\}^n$, checks whether $C(x) = z$). For every heavy $z \in \{0,1\}^\ell$, $H_\infty(X) \geq n - \Delta$ and therefore, for every heavy z, $E(X_z)$ is $\epsilon/2$-close to uniform with relative error. It follows that for any function $D : \{0,1\}^\ell \to \{0,1\}$, $\Pr_{x \leftarrow U_n}[D(z) = E(x) | C(x) = z] \leq (1 + \frac{\epsilon}{2}) \cdot 2^{-m}$. Overall, we have that the probability that

$$\Pr_{X \leftarrow U_n}[D(C(x)) = E(x)] \leq \epsilon \cdot 2^{-(m+1)} + (1 + \frac{\epsilon}{2}) \cdot 2^{-m} \leq (1 + \epsilon) \cdot 2^{-m}.$$

□

Theorem 1.16 now follows directly from Theorem E.3.

E.3 Obtaining relative-error extractors for samplable distributions

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

Lemma E.5 (connection between samplable and recognizable distributions)**.** *If a distribution X on $\{0,1\}^n$ is samplable by a size $s \geq n$ 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, $|\Pr[X \in A] - \Pr[X' \in A]| \leq \epsilon \cdot min(\Pr[X \in A], \Pr[X' \in A])$.*

Proof. Lemma E.5 follows from Theorem A.3 as given x, an NP-circuit can approximate the integer $\Pr[X = x] \cdot 2^s$ with small relative error, and this recognizes a distribution X' as required. □

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 E.3 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 samplable by size n^c NP-circuits. This gives a construction of extractors for samplable distributions, and proves Theorem 1.20.

E.4 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 [TV00]. We imitate the overall argument structure of [TV00]. 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 E.6 ([TV00]). *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 $\epsilon = 2^{-(n'/3)} = 2^{-\Omega(n)}$. 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 E.

Our construction will use 2-source extractors, defined below.

Definition E.7. *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 E.8 ([CG88, Vaz87, DEOR04]). *There exists a constant $\alpha > 0$ such that for every sufficiently large n' and every $m \leq \alpha n'$ there is a $(0.2n, 0.9n, 2^{-3m})$-2-source extractor $T : \{0,1\}^{n'} \times \{0,1\}^{n'} \to \{0,1\}^m$.*

We will construct a function $E : \{0,1\}^{\bar{n}} \to \{0,1\}^m$ where $\bar{n} = n + n'$. We will set the parameters so that $n' \leq n$ and so $\bar{n} \leq 2n$. 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 E.9. *Let T be the 2-source extractor of Theorem E.8. For every sufficiently large n', and every m that satisfies the requirements of Theorem E.8, 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. Otherwise, set R_1 to be the uniform distribution over such y, and R_2 to be the uniform distribution over S and note that $\Pr[T(R_1, R_2) = a] \geq (1-\epsilon) \cdot 2^{-m} \geq 2^{-m} + 2^{-2m}$ which shows the failure of T. □

Theorem E.10. *There exists a constant $\alpha > 0$, such that for every constant $c > 1$, and for every sufficiently large n, if $f : \{0,1\}^n \to \{0,1\}^{n'}$ is $2^{-(2\Delta + m + 0.2 \cdot n' + c \cdot \log n)}$-incomputable for size n^c, Σ_3-circuits, and T is taken so that ϵ from Lemma E.9 is $n^{-\alpha c}/9$, and $n' \geq 10 \cdot (\Delta + m + c \log n)$ then E is an $(n - \Delta, n^{-\alpha c})$-relative error extractor for distributions weakly-recognizable by size $n^{\alpha c}$ circuits.*

Theorem E.3 now follows by choosing $n' = \nu \cdot n$ for a sufficiently small constant $\nu > 0$, and choosing $m = \alpha n$, and $\Delta = \alpha n$ for a sufficiently small constant $\alpha > 0$ that is smaller than ν so that the requirements hold.

E.5 Proof of Theorem E.10

In this section we Prove Theorem E.10. The proof is by contradiction. Assume that the conclusion on Theorem E.10 does not hold. That is, that for some sufficiently large n, E is not an $(n - \Delta, 9\epsilon)$-relative-error extractor for distributions weakly-recognizable by size $n^{\alpha \cdot c}$ circuits. Where $\alpha > 0$ is a constant that we choose later. By an averaging argument we get that:

Lemma E.11. *There exists a distribution $Z' = (X', I')$ over $\{0,1\}^{\bar{n}}$ 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 $\Pr[E(Z') = a] \leq (1-2\epsilon)\cdot 2^{-m}$ follows the same way using the fact that in Lemma E.9 we have control over both cases). We need the following definition.

Definition E.12 (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 E.13. $\Pr[X'$ *is useful*$] \geq 2^{-\Delta}\cdot\epsilon\cdot 2^{-m}$.

The same averaging argument appears in [TV00]. In the final version, we will provide a proof of Lemma E.13 for completeness. We would like to get an estimate on the probability of useful inputs, according to the uniform distribution.

Lemma E.14. $\Pr_{x\leftarrow U_n}[x$ *is useful*$] \geq 2^{-2\Delta}\cdot\epsilon\cdot 2^{-m}$.

Proof. Let G denote the set of useful x. We have that $H_\infty(Z') \geq n - \Delta$ and therefore $H_\infty(X') \geq n - \Delta$. Thus, for every $x \in \{0,1\}^n$, $\Pr[X' = x] \leq 2^{-(n-\Delta)}$. By Lemma E.13, $\Pr[X' \in G] \geq \rho$ for $\rho = 2^{-\Delta}\cdot\epsilon\cdot 2^{-m}$. This means that $|G| \geq \rho/2^{-(n-\Delta)} = \rho\cdot 2^n/2^\Delta$ which means that $\Pr_{x\leftarrow U_n}[x \in G] \geq \rho/2^\Delta$. □

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_3-circuit A, but as we are in a distributional setup, the random coins of A can be later hardwired, to produce the same success probability. 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 a 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 $p' \geq (1+1.5\cdot\epsilon)\cdot 2^{-m}$. 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 $p \geq (1+2\epsilon)\cdot 2^m$ and A_x answers zero if $p \leq (1+\epsilon)\cdot 2^{-m}$.

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

Using Theorems A.2 and A.3 it follows that:

Lemma E.15. *For a sufficiently small constant $\alpha > 0$, A can be implemented by a (probabilistic) size poly*$(n^{\alpha c}/\epsilon') = n^c$*,* Σ_2*-circuit.*

The algorithm A indeed suceeds with not too small probability on "useful inputs".

Lemma E.16. *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, We have that the distribution $R_2 = (I'|X' = x)$ has $H_\infty(R_2) \geq 0.9n'$. Therefore, by Lemma E.9 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 $\Pr[A(x) = f(x)] \geq 2^{-2\Delta} \cdot \epsilon \cdot 2^{-m} \cdot 2^{-0.2n'}$ as required.

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 specigically, 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 E.17. *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

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

We can compute an approximation of the denominator by considering the circuit $C_2'(z,y)$ which outputs $C'(z,y)$ is $C_2(z) = 1$ 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 enumerator. □

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.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de