Pseudorandom generators with optimal seed length for non-boolean poly-size circuits

[Extended Abstract]

Sergei Artemenko[*]
University of Haifa
Mount Carmel, Haifa
sartemen@gmail.com

Ronen Shaltiel[†]
University of Haifa
Mount Carmel, Haifa
ronen@cs.haifa.ac.il

ABSTRACT

A sampling procedure for a distribution P over $\{0,1\}^\ell$, is a function $C : \{0,1\}^n \to \{0,1\}^\ell$ such that the distribution $C(U_n)$ (obtained by applying C on the uniform distribution U_n) is the "desired distribution" P. Let $n > r \geq \ell = n^{\Omega(1)}$. An *nb-PRG* (defined by Dubrov and Ishai (STOC 2006)) is a function $G : \{0,1\}^r \to \{0,1\}^n$ such that for every $C : \{0,1\}^n \to \{0,1\}^\ell$ in some class of "interesting sampling procedures", $C'(U_r) = C(G(U_r))$ is close to $C(U_n)$ in *statistical distance.*

We construct poly-time computable nb-PRGs with $r = O(\ell)$ (which is best possible) for poly-size circuits. Previous nb-PRGs of Dubrov and Ishai have $r = \Omega(\ell^2)$. We rely on the assumption that: there exists $\beta > 0$, and a problem L in $\mathrm{E} = \mathrm{DTIME}(2^{O(n)})$ such that for every large enough n, nondeterministic circuits of size $2^{\beta n}$ that have NP-gates cannot solve L on inputs of length n. This assumption is a scaled nonuniform analogue of (the widely believed) $\mathrm{EXP} \neq \Sigma_2^{\mathrm{P}}$, and similar assumptions appear in various contexts in derandomization. The nb-PRGs of Dubrov and Ishai are based on very strong cryptographic assumptions, or alternatively, on non-standard assumptions regarding incompressibility of functions on random inputs.

When restricting to poly-size circuits $C : \{0,1\}^n \to \{0,1\}^\ell$ with Shannon entropy $H(C(U_n)) \leq k$, for $\ell > k = n^{\Omega(1)}$, our nb-PRGs have $r = O(k)$ which is best possible. The nb-PRGs of Dubrov and Ishai use seed length $r = \Omega(k^2)$ and require that the probability distribution of $C(U_n)$ is efficiently computable.

Our nb-PRGs follow from a notion of "conditional PRGs" which may be of independent interest. These are PRGs where $G(U_r)$ remains pseudorandom even when conditioned on a "large" event $\{A(G(U_r)) = 1\}$, for an arbitrary poly-size circuit A. A related notion was considered by Shaltiel and Umans (CCC 2005) in a different setup, and our proofs use ideas from that paper, as well as ideas of Dubrov and Ishai.

We also give an unconditional construction of a poly-time computable nb-PRGs for poly(n)-size, depth d circuits $C : \{0,1\}^n \to \{0,1\}^\ell$ with $r = O(\ell \cdot \log^{d+O(1)} n)$. This improves upon the previous work of Dubrov and Ishai that has $r \geq \ell^2$. Our nb-PRGs can be implemented by a uniform family of poly-size constant depth circuits (with slightly larger, but still almost linear seed length). The nb-PRG of Dubrov and Ishai computes large parities and cannot be computed in poly-size and constant depth.

This result follows by adapting a recent PRG construction of Trevisan and Xue (CCC 2013) to the case of nb-PRGs, and implementing it by constant-depth circuits.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*

General Terms

Theory

Keywords

ACM proceedings, LaTeX, text tagging

1. INTRODUCTION

A sampling procedure is a function $C : \{0,1\}^n \to \{0,1\}^\ell$ such that when C is applied on the uniform distribution U_n, the obtained distribution $C(U_n)$ is some "desired distribution" P over ℓ-bit strings. There are two natural complexity measures for sampling procedures: the *computational complexity* of the function C, and the *randomness complexity* which is the number of random bits used by the procedure (denoted by n). The reader is referred to [38], for a discussion on the complexity of sampling procedures. Dubrov and Ishai [7] considered the following natural problem: is it possible to reduce the randomness complexity of sampling procedures without substantially increasing their computational complexity? Specifically, given an efficient sampling procedure $C : \{0,1\}^n \to \{0,1\}^\ell$ with $n > \ell$, construct an

[*] Supported by ERC starting grant 279559.

[†] Supported by BSF grant 2010120, ISF grant 864/11, and ERC starting grant 279559.

STOC '14 May 31 - June 03 2014, New York, NY, USA
Copyright 2014 ACM 978-1-4503-2710-7/14/05 ...$15.00.
http://dx.doi.org/10.1145/2591796.2591846.

efficient sampling procedure $C' : \{0,1\}^r \to \{0,1\}^\ell$ which uses only $r \ll n$ random bits, and $C'(U_r)$ is close to the desired distribution $C(U_n)$ in *statistical distance.*[1] For this purpose, Dubrov and Ishai suggested the following notion of "pseudorandom generator against non-Boolean statistical tests".

Definition 1.1 (nb-PRG [7]). *A function $G : \{0,1\}^r \to \{0,1\}^n$ is an ϵ-nb-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 (and we say that G ϵ-fools C). G is an ϵ-nb-PRG for a class $\mathcal{C}$ of functions, if G is an ϵ-nb-PRG for every function in the class.*

Indeed, given an efficient nb-PRG G we can compute $C'(U_r) = C(G(U_r))$ and sample a distribution that is ϵ-close to $C(U_n)$ using only r random bits.[2] Note that if the class of sampling procedures that we consider contains the function $C : \{0,1\}^n \to \{0,1\}^\ell$ that outputs the first ℓ bits (and any reasonable complexity class does), then the seed length r has to be at least ℓ (assuming $\epsilon < 1/2$). nb-PRGs are a natural generalization of "standard PRGs" defined below.

Definition 1.2 (PRG). *A function $G : \{0,1\}^r \to \{0,1\}^n$ is an ϵ-PRG for a function $C : \{0,1\}^n \to \{0,1\}$ if $|\Pr[C(G(U_r)) = 1] - \Pr[C(U_m) = 1]| \le \epsilon$ (that is iff $C(G(U_r))$ and $C(U_n)$ are ϵ-close). G is an ϵ-PRG for a class $\mathcal{C}$ of functions, if G is an ϵ-PRG for every function in the class.*

Consequently, nb-PRGs are at least as hard to construct as (standard) PRGs. In this paper we will be interested nb-PRGs for two types of sampling procedures: polynomial-size circuits and circuits with polynomial-size and constant depth.

In addition to the application of reducing randomness of sampling procedure, nb-PRGs can also be used to reduce the communication in interactive protocols (by having a party send a seed to an nb-PRG and the other parties compute the next message function). Dubrov and Ishai [7] gave such applications in information theoretic cryptographic setups, and we believe that there may be many additional applications.

If $H(C(U_n))$ is small

If we are guaranteed that the Shannon entropy of $C(U_n)$ is small (say $H(C(U_n)) \le k$ for some parameter k) than we can hope for a shorter seed length $r \approx k$. There are efficiently samplable distributions P with entropy k, such that any distribution that is ϵ-close to P cannot be sampled using less than $O(k/\epsilon)$ bits.[3] Thus, an ϵ-nb-PRG for poly-size circuits that are guaranteed to produce distributions with entropy $\le k$, must have seed length $r = \Omega(k/\epsilon)$.

1.1 nb-PRGs for polynomial-size circuits

The setup

The most natural setup of parameters for sampling procedures is the case where $C : \{0,1\}^n \to \{0,1\}^\ell$, where $\ell = n^e$ for some constant $0 < e < 1$. We fix this choice of parameters for this discussion and consider the case where C is a size $s = s(n) = \mathrm{poly}(n)$. For the application of reducing randomness for sampling procedures, the size s is *known* to the PRG, and the PRG may be allowed to run in time $p(n)$ for a polynomial p that is larger than s. In the terminology of PRGs, this setup is often referred to as the "Nisan-Wigderson setting" [25]. However, note that as $\ell = n^e$ and the seed length must be at least ℓ, we are interested in PRGs $G : \{0,1\}^r \to \{0,1\}^{r^{O(1)}}$ (often referred to as polynomial stretch). The application also dictates that G run in time polynomial in r. This is in contrast to the "Nisan-Wigderson setting" in which PRGs are often allowed to run in time exponential in their seed length (because intended applications plan to enumerate all seed anyway).[4] Many (standard) PRG constructions in the Nisan-Wigderson setting [3, 32, 17, 28, 35, 36] critically use the ability to run in time exponential in the seed length (usually for encoding strings of length 2^r by error-correcting codes). In fact, the sole exception, is the original construction of Nisan and Wigderson [25] which (when used by itself without a pre-processing step of hardness amplification/error correction) can run in time polynomial in the output length (at least under some very specific hardness assumptions).

Using cryptographic PRGs

A very natural approach to construct nb-PRGs is to reduce to constructing standard PRGs. It is immediate that a standard PRG for circuits of size $s + 2^\ell$ is an nb-PRG for circuits $C : \{0,1\}^n \to \{0,1\}^\ell$ of size s. (This is because any statistical test on ℓ bits can be implemented by a circuit of size 2^ℓ). This means that a (standard) PRG $G : \{0,1\}^r \to \{0,1\}^n$ that fools circuits of size $s + 2^\ell = \Theta(2^{n^{1/e}})$ is an nb-PRG with the desired parameters. These parameters are obviously impossible in the Nisan-Wigderson setting (where a PRG that runs in polynomial time cannot fool a circuit of size superpolynomial). However, one can hope to achieve such parameters using "cryptographic PRGs" such as the Blum-Micali-Yao [6, 39] or HILL [14, 12, 15, 13]. Such PRGs imply (and therefore require) cryptographic assumptions such as the existence of one-way functions. Indeed, Dubrov and Ishai observe that if there exist one-way permutations $f : \{0,1\}^r \to \{0,1\}^r$ that cannot be inverted with noticeable probability by circuits of size $2^{O(\ell)}$, then the PRG construction of Blum, Miali and Yao [6, 39] gives an nb-PRG with seed length r. A weakness of this approach is that in order to achieve seed length $r = O(\ell^c)$ we need one-way permutations that cannot be inverted by circuits of size $2^{\Omega(r^{1/c})}$. This means that we can achieve seed length $r = O(\ell)$ only if we have permutations that cannot be inverted with noticeable probability by size $2^{\Omega(r)}$ circuits. This is a very strong assumption that is known not to hold for some of the candidate one-way permutations.[5] This assumption be-

[1] Two distributions over the same domain are ϵ-close if the probability that they assign to any event differs by at most ϵ.

[2] It is important to observe that $C'(U_r)$ is required to be *statistically indistinguishable* from $C(U_n)$. Standard PRGs suffice if we relax the requirement to *computational indistinguishability.*

[3] Let $2^{-n} \le \epsilon \le 1/10$. Fix some $x \in \{0,1\}^n$ and consider the distribution P over $\{0,1\}^n$ which gives weight $1 - 4 \cdot \epsilon$ to x and $4 \cdot \epsilon/(2^n - 1)$ to every other string. Note that $H(P) = O(\epsilon n)$, and yet, for every distribution Q that is samplable using less than $n/2$ random bits, Q is not ϵ-close to P.

[4] We remark that a similar setup (in the boolean setting) arises in "typically-correct derandomization" [27, 21, 26].

[5] The reason for using one-way permutations rather than one-way functions is that the best known PRG constructions

comes plausible for constants $c \gg 1$ and gives nb-PRGs with seed length $r = O(\ell^c)$. Dubrov and Ishai show that this approach also yields nb-PRGs with seed length $r = O((k/\epsilon)^c)$ for polynomial-size circuits C which are guaranteed to sample distributions with Shannon entropy $\leq k$.[6]

Function compression

Dubrov and Ishai show an interesting connection between nb-PRGs and "function compression". A function $f : \{0,1\}^n \rightarrow \{0,1\}$ is *compressed* by a circuit $C : \{0,1\}^n \rightarrow \{0,1\}^\ell$ if an unbounded procedure can compute $f(x)$ given $C(x)$ (without receiving x). We say that f is $(1/2+\epsilon, \ell)$-compressible by size s circuits, if there exits a size s circuit $C : \{0,1\}^n \rightarrow \{0,1\}^\ell$ such that $f(x)$ can be recovered correctly from $C(x)$ on at least $(1/2+\epsilon)$-fraction of the inputs. Dubrov and Ishai suggested to base nb-PRG constructions on the assumption that there exist explicit incompressible boolean functions. The high level idea is that some PRG constructions in the literature, are proven by a reduction showing that a small distinguisher circuit for the PRG can be converted into a small circuit computing the supposedly hard function. Some of these reductions can also convert a non-boolean distinguisher into a non-boolean circuit that compresses the function. This approach allows using one-way permutations $f : \{0,1\}^r \rightarrow \{0,1\}^r$ against poly-size circuits (rather than exponential size circuits), if the permutations have hard-core bits that are not only secure, but are also not $(1/2 + r^{-\omega(1)}, \Omega(r))$-compressible by polynomial size circuits. Assuming the existence of such one-way permutations, Dubrov and Ishai show that the Blum-Micali-Yao PRG yields an nb-PRG and has seed length $r = O(\ell)$. We are not aware of research that attempts to evaluate the validity of this assumption. We also point out, that this nb-PRG does not extend to have seed length proportional to the entropy, when it is guaranteed that the entropy of the sampled distribution $C(U_n)$ is small.

Nisan-Wigderson PRG with incompressible functions

Dubrov and Ishai show that a polynomial time computable nb-PRG that fools circuits of size n^c is obtained under the following assumption: There is a function $f : \{0,1\}^{O(\ell)} \rightarrow \{0,1\}$ computable in polynomial time that is not $(1/2 + \Omega(\epsilon/\ell), \Omega(l))$-compressible by circuits $C : \{0,1\}^n \rightarrow \{0,1\}^\ell$ of size $n^{O(c)}$. This result follows by using the function f in the Nisan-Wigderson generator [25], and follows by a clever argument showing that the security proof of [25] applies in this setting. However, a well known inefficiency of the Nisan-Wigderson generator dictates that even under this assumption the obtained seed length cannot be linear in ℓ and must be at least quadratic, that is $r = \Omega(\ell^2)$.[7]

1.1.1 Hardness assumptions for exponential size circuits

We give new constructions of nb-PRGs in the "Nisan-Wigderson setting". Our constructions achieve seed length $r = O(\ell)$ under strong but plausible assumptions. In order to discuss our assumptions we need a quick review of nondeterministic circuits and oracle circuits.

DEFINITION 1.3 (NONDETERMINISTIC CIRCUITS). *A nondeterministic circuit C has additional "nondeterministic input wires". We say that the circuit C evaluates to 1 on x iff there exist an assignment to the nondeterministic input wires that makes C output 1 on x. Given a boolean function $A(x)$, an A-circuit is a circuit that is allowed to use A gates (in addition to the standard gates). An NP-circuit is a SAT-circuit (where SAT is the satisfiability function) a Σ_i-circuit is an A-circuit where A is the canonical Σ_i^P-complete language. The size of all circuits is the total number of wires and gates.*[8]

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. Similarly, a nondeterministic NP-circuit is the nonuniform analogue of $\Sigma_2^{\mathrm{P}} = \mathrm{NP}^{\mathrm{NP}}$ and is thus weaker than a Σ_2-circuit (which is analogous to $\mathrm{P}^{\Sigma_2^P}$). Hardness assumptions against nondeterministic/NP/Σ_i circuits appear in the literature in various contexts of derandomization [22, 23, 33, 9, 28, 29, 4]. Typically, the assumption is of the following form: E is hard for exponential size circuits (where the type of circuits is one of the types discussed above). More specifically:

DEFINITION 1.4. *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.[9]

from one-way functions [14, 12, 15, 13] have a polynomial blow-up in the seed length.

[6] This is achieved by showing that there exists a circuit $D : \{0,1\}^\ell \rightarrow \{0,1\}^{O(k/\epsilon)}$ of size roughly 2^k such that if $C(U_n)$ and $C(G(U_r))$ are not ϵ-close then $D(C(U_n))$ and $D(C(G(U_r))$ are not $\Omega(\epsilon)$-close, meaning that an nb-PRG that fools $D \circ C$ also fools C, and in this setup an nb-PRG can handle very large circuits anyway. Note that this reduction is specific to this setup.

[7] This inefficiency was the focus of several works that construct improved PRGs (in the boolean setting) [17, 28, 35, 36], but all these approaches give PRGs with running time exponential in the seed length and do not make sense in this setup.

[8] An alternative approach is to define using the Karp-Lipton notation for Turing machines with advice. For $s \geq n$, a size $s^{\Theta(1)}$ deterministic circuit is equivalent to $\mathrm{DTIME}(s^{\Theta(1)})/s^{\Theta(1)}$, a size $s^{\Theta(1)}$ nondeterministic circuit is equivalent to $\mathrm{NTIME}(s^{\Theta(1)})/s^{\Theta(1)}$, a size $s^{\Theta(1)}$ NP-circuit is equivalent to $\mathrm{DTIME}^{\mathrm{NP}}(s^{\Theta(1)})/s^{\Theta(1)}$, a size $s^{\Theta(1)}$ nondeterministic NP-circuit is equivalent to $\mathrm{NTIME}^{\mathrm{NP}}(s^{\Theta(1)})/s^{\Theta(1)}$, and a size $s^{\Theta(1)}$ Σ_i-circuit is equivalent to $\mathrm{DTIME}^{\Sigma_i^P}(s^{\Theta(1)})/s^{\Theta(1)}$. With this view, we can also differentiate between circuits that make adaptive calls to their oracle, and circuits that make nonadaptive calls to their oracle, and the latter are called *parallel circuits*.

[9] Another advantage of constructions based on this type of assumptions is that any E-complete problem (and such prob-

1.1.2 New constructions of nb-PRGs

We give a construction of nb-PRGs with seed length $r = O(\ell)$ under the assumption that E is hard for exponential size nondeterministic NP-circuits.

THEOREM 1.5 (NB-PRGS WITH SHORT SEED). *There is a constant $b > 1$ such that if E is hard for exponential size nondeterministic NP-circuits then for every constants $e > 0$ and $c > 1$ there is a poly(n)-time computable ϵ-nb-PRG $G : \{0,1\}^{b \cdot \ell} \to \{0,1\}^n$ for size n^c circuits $C : \{0,1\}^n \to \{0,1\}^\ell$, as long as $\ell \geq n^e$, $\epsilon \geq n^{-c}$.*[10]

Note that G runs in time polynomial in n and this polynomial depends on c, e.

If $H(C(U_n))$ is small

We also consider the subclass of poly-size circuits C such that $H(C(U_n)) \leq k$. Recall that here, the best we can shoot for is seed length $r = O(k/\epsilon)$. We achieve this under the same hardness assumption.

THEOREM 1.6 (NB-PRGS FOR LOW ENTROPY). *There is a constant $b > 1$ such that if E is hard for exponential size nondeterministic NP-circuits then for every constants $e > 0$ and $c > 1$ there is a poly(n)-time computable ϵ-nb-PRG $G : \{0,1\}^{b \cdot k/\epsilon} \to \{0,1\}^n$ for size n^c circuits $C : \{0,1\}^n \to \{0,1\}^\ell$ which satisfy $H(C(U_n)) \leq k$, as long as $k \geq n^e$, $\ell \leq n^c$.*

Following the discussion in previous sections, we point out that in this setup, nb-PRGs with this seed length were only known under the assumption that there are one-way permutations with hardness $2^{\Omega(n)}$, and this is known not to hold for some candidate one-way permutations. Dubrov and Ishai [7] were able to achieve nb-PRGs in this setup under some of the other assumptions discussed in Section 1.1. However, these nb-PRGs achieve seed length $\geq (k/\epsilon)^2$, and require an additional assumption: that it is feasible to compute the quantity $p(z) = \Pr[C(U_n) = z]$ given $z \in \{0,1\}^\ell$.

Alternative hardness assumptions for our theorems

Our main technical construction will rely on the following assumption:

ASSUMPTION 1.7. *For every constant $c > 1$ there exists an (n^{-c})-PRG $G' : \{0,1\}^n \to \{0,1\}^{n^c}$ for nondeterministic NP-circuits of size n^c, and G' is computable in time $p(n)$ where p is a polynomial that depends on c.*

This assumption is known to follow from the assumption stated in Theorems 1.5 and 1.6 by the following argument: By the "downward collapse theorem" of Shaltiel and Umans [29] the assumption that E is hard for exponential size nondeterministic NP circuits implies that E is hard for exponential size Σ_2-circuits that make non-adaptive calls to their oracle. By [22, 18] the latter assumption implies a PRG G' with the required properties.

In fact, the PRG G' obtained in [18, 22] has better parameters than we asked for. It has "exponential stretch" and stretches $O(c \cdot logn)$ bits into n^c bits. This suggests that the assumption stated in Theorems 1.5 and 1.6 is stronger than what is actually needed (as we only need polynomial stretch). We can therefore use a hardness assumption against *polynomial size* Σ_2-circuits. However, as we are shooting for a PRG which is computable in polynomial time (rather than exponential time), we cannot afford "worst-case to average-case hardness amplification" (which takes exponential time and is known not to be possible in polynomial time by black-box techniques [37]). Instead, we can use Yao's XOR-Lemma (see [8] for a survey) which does not blow up the running time. The price of this modification is that instead of a "worst-case hardness assumption" we require a "mildly average-case hardness assumption". Summing up, we get that Assumption 1.7 (and therefore the conclusion of the two main theorems) follow from the following assumption:

ASSUMPTION 1.8. *For every constant $c > 1$ there exists a problem L in P such that for every sufficiently large n, every size n^c Σ_2-circuit fails to compute the characteristic function of L on at least a $1/n$-fraction of the inputs of length n.*

This assumption gives assumption 1.7 by using Yao's XOR-Lemma on (the characteristic function of) L, and then plugging the amplified function to the Nisan-Wigderson generator. We remark that the same assumption is also suggested (and relied on) in a construction of Goldreich and Wigderson [9] in a different context.

1.2 nb-PRGs for constant-depth circuits

In this section we discuss *unconditional constructions* of nb-PRGs against poly-size circuits that have constant-depth. There are many surprising instances where interesting distributions can be sampled by procedures with very low computational complexity see e.g., [1] and following work. The reader is referred to [38] for examples of low complexity sampling procedures. Dubrov and Ishai [7] considered the following setup: Let c, d, e be positive constants and consider a sampling procedure $C : \{0,1\}^n \to \{0,1\}^\ell$ that is a circuit of size n^c and depth d which outputs $\ell = n^e$ bits. Note that this is the setup considered in the previous section, with the additional restriction that circuits have constant depth. Dubrov and Ishai gave the following construction of nb-PRG.

THEOREM 1.9. *[7] Let c, d, e be positive constants. For every constant $\delta > 0$ there is an ϵ-nb-PRG $G : \{0,1\}^r \to \{0,1\}^n$ for circuits of size n^c, depth d and output length $\ell = n^e$. Furthermore, $r = \ell^{2+\delta}$, $\epsilon = n^{-\omega(1)}$ and G is computable in time poly(n).*

The construction of Dubrov and Ishai uses the Nisan-Wigderson generator [24, 25] with the parity function, and is based on showing that the parity function cannot be compressed by small constant-depth circuits. However, the aforementioned bottleneck in the Nisan-Wigderson generator causes the seed length r to be larger than ℓ^2 whereas the obvious lower bound is (once again) ℓ. Our first result is an nb-PRG which achieves seed length $\tilde{O}(\ell)$.

lems 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.

[10] Our approach can potentially work for smaller ℓ, ϵ, but will require stronger hardness assumptions that are not as nice to state. More details are deferred to the full version.

Theorem 1.10 (nb-PRGs with short seed). *Let c, d, e be positive constants. There is an ϵ-nb-PRG $G : \{0,1\}^r \rightarrow \{0,1\}^n$ for circuits of size n^c, depth d and output length $\ell = n^e$. Furthermore, $r = O(\ell \cdot \log^{a_d} n)$ (where $a_d = d + O(1)$ is a constant that depends only on d), $\epsilon = n^{-\omega(1)}$ and G is computable in time poly(n).*

Our construction gives a general result for arbitrary size, depth, output length and error, and Theorem 1.10 above is a special case of a more general theorem appears in Section 5. Our proof is based on adapting a recent boolean PRG construction of Trevisan and Xue [34] (which avoids the Nisan-Wigderson generator) to the case on nb-PRGs.

A drawback of both Theorem 1.9 and Theorem 1.10 is that the pseudorandom generator G is guaranteed to run in polynomial time, but is not necessarily implementable by a poly-size circuit with constant depth. This means that if we use G to sample the output distribution of some sampling procedure $C : \{0,1\}^n \rightarrow \{0,1\}^\ell$ that is a poly-size constant depth circuit, then the resulting sampling procedure $C'(\cdot) = C(G(\cdot))$ is implementable in poly-time but not necessarily in constant depth. Our next result gives an nb-PRG which is implementable by a uniform family of poly-size constant depth circuits. This PRG achieves seed length roughly $\ell^{1+\alpha}$ (where $\alpha > 0$ is an arbitrary small constant). This is worse than the seed length of Theorem 1.10, but still better than that achieved by Dubrov and Ishai in Theorem 1.9.

Theorem 1.11 (nb-PRGs in constant depth). *Let c, d, e be integer constants. For every $\alpha > 0$ there is an ϵ-nb-PRG $G : \{0,1\}^r \rightarrow \{0,1\}^n$ for circuits of size n^c, depth d and output length $\ell = n^e$. Furthermore, $r = O(\ell^{1+\alpha} \cdot \log^{a_d} n)$ (where $a_d = O(1/\alpha + d)$ is a constant that depends only on d, α), $\epsilon = n^{-\omega(1)}$ and G is computable by a family of uniform circuits of size poly$(n, c \log n)$ and depth $O(1/\alpha)$ (where the constant hidden in the $O(\cdot)$ is universal, and the depth does not depend on c, d).*[11]

We obtain this result, by giving an implementation of a variant of the nb-PRG of Theorem 1.10 by constant depth circuits. For this, we use an approach of Viola [38] to show that k-wise independent distributions can be sampled with competitive seed length by constant depth circuits. The proof is deferred to the full version.

We stress that the nb-PRG of Dubrov and Ishai from Theorem 1.9 is *not* computable by small constant depth circuits. This is because it computes the parity function on inputs of length $\geq \ell$.

2. TECHNIQUE

We aim to reduce the task of constructing nb-PRGs to that of constructing standard PRGs. Our first attempt is the following trivial observation: An $(\epsilon/2^\ell)$-PRG for size $s + O(\ell)$ circuits is also an ϵ-nb-PRG for size s circuits. This follows because if $C(U_n)$ and $C(G(U_r))$ are not ϵ-close, then there exists $z \in \{0,1\}^\ell$ such that the probability assigned to z by the two distributions differ by $\epsilon/2^\ell$. This means that a boolean circuit $C'(x)$ which outputs 1 iff $C(x) = z$ is not $(\epsilon/2^\ell)$-fooled by G.

Using the Nisan-Wigderson generator, we can construct such PRGs given a poly-time computable function $f : \{0,1\}^{O(\ell)} \rightarrow \{0,1\}$ on which every circuit of size $s^{O(1)}$ errs on at least a $(1/2 - 1/2^{O(\ell)})$-fraction of inputs. (Because of the aforementioned inefficiency of the Nisan-Wigderson PRG, this approach cannot give seed smaller than $\Omega(\ell^2)$). However, "existing techniques" cannot produce such a function f from the assumption that E is hard for exponential size circuits (or even from the weaker assumption: E is mildly average-case hard for exponential size circuits) [30, 2]. Trevisan and Vadhan [33] suggested that these limitations can be bypassed if we assume that E is hard for exponential size nondeterministic circuits (or more generally Σ_i-circuits for some $i \geq 1$). They were able to start from such assumptions and obtain their goal (which is extractors for samplable distributions). They were not, however, able to construct average-case hard functions (or PRGs) with very low error.[12]

Inspired by the success of Trevisan and Vadhan, we aim to construct nb-PRGs starting from a worst-case hardness assumption for Σ_i-circuits (for some small i). In order to achieve this, we would like a reduction, showing that a circuit $C : \{0,1\}^n \rightarrow \{0,1\}^\ell$ that is not ϵ-fooled by some candidate PRG can be transformed into a boolean test that is not ϵ'-fooled by the PRG. Our boolean test may be complex (and allowed to use nondeterminism) but we require that ϵ' is not much smaller than ϵ. This intuition is captured in the following lemma (which can be seen as a more careful version of our first attempt).

Lemma 2.1. *There exists a constant $B > 0$ such that for every constant $C > 0$ the following holds: Let R and V be distributions over $\{0,1\}^\ell$ that are not ℓ^{-C}-close (the reader should think of $R = C(U_n)$ and $V = C(G(U_r))$). There exist a $z \in \{0,1\}^\ell$ and $i \in [\ell]$ such that*

$$|\Pr[R_i = z_i | R_{1,\ldots,i-1} = z_{1,\ldots,i-1}] - \Pr[V_i = z_i | V_{1,\ldots,i-1} = z_{1,\ldots,i-1}]| > \ell^{-(C+5)},$$

and $\Pr[R_{1,\ldots,i-1} = z_{1,\ldots,i-1}] \geq 2^{-B \cdot \ell}$.

The proof of Lemma 2.1 is deferred to the full version. We explain the high level idea below. We use the following lemma (which follows by a case analysis).

[11] Note that this nb-PRG is "cryptographic" in the sense that the PRG is implementable by a uniform family of circuits of size $n^{c'}$ and depth d', for some constants c', d' and is able to fool circuits of depth d and size n^c for larger d, c for every sufficiently large n.

[12] Hardness amplification from worst-case to average case typically rely on binary error-correcting codes with certain "sub-linear time" list-decoding procedures [32]. These techniques cannot achieve low error with low complexity procedures [30, 2]. The approach of Trevisan and Vadhan is to allow these procedures to be nondeterministic, or more generally to allow them access to a Σ_i-oracle. Indeed, using this approach, Trevisan and Vadhan were able to construct non-binary codes with the required properties. They were not able to extend these results to binary codes (which is typically easy by code concatenation). We note that the impossibility results of Shaltiel and Viola [30] provide an explanation for this failure, showing that there is an inherent difference between the binary and non-binary cases. These impossibility results rule out the possibility of obtaining binary codes with the desired properties, even if the decoding procedures are given oracle access to languages in the polynomial time hierarchy. This is because [30] show that such decoding procedures imply a small constant depth circuit for the majority function, and a PH computation can be seen as a constant depth circuit over the queries to the oracle.

Lemma 2.2. *Let R, V be two distributions over some finite set S, such that R and V are not α-close. Let $\rho, \nu \geq 0$ and let $f : S \to \{0,1\}$ be a function such that $p = \Pr[f(R) = 0] \leq \frac{1}{2}$ then at least one of the following holds:*

- $|\Pr[f(R) = 1] - \Pr[f(V) = 1]| > \rho$.
- *$(R|f(R) = 1)$ and $(V|f(V) = 1)$ are not $((\alpha - \rho) \cdot (1 - \nu))$-close.*
- *$(R|f(R) = 0)$ and $(V|f(V) = 0)$ are not $((\alpha - \rho) \cdot (1 + \nu/2p))$-close.*

For the proof of Lemma 2.1 we apply Lemma 2.2 iteratively ℓ times as follows. At each step we concentrate on the first bit of R and V (by setting $f(z) = z_1$ or $f(z) = 1 - z_1$ depending on whether the first bit of R is more likely to be one or zero). By Lemma 2.2 either the first bit distinguishes the two distributions, or we can condition the two distributions R and V on the event that the first bit is fixed and obtain two distributions on less bits which are not close. By iterating this argument, we eventually obtain a boolean distinguisher that distinguishes R and V conditioned on the event that a prefix of the bits are fixed.

The second conclusion in Lemma 2.1 is that the event that we condition on has probability $2^{-O(\ell)}$. This will be important later on, as the exponent in this probability will be the main factor in the final seed length of the nb-PRG. Consequently, we need to be careful that the event that we condition on is not of too low probability. In each step we are happy if we condition on an event with probability $\geq \frac{1}{2}$ (as ℓ such steps do not reduce the probability below $2^{-\ell}$). We are concerned in steps that reduce the probability to less than half. However, by the third condition of Lemma 2.2, each such step increases the statistical distance (which cannot exceed one). This gives us control over the total contribution of "risky steps" and we can show that this total contribution does not reduce the probability below $2^{-O(\ell)}$.

The next definition captures the kind of tests that distinguishes $C(U_n)$ from $C(G(U_r))$ according to the lemma above.

Definition 2.3 (Conditional test [29]). *A conditional test is a pair (A, D) where $A : \{0,1\}^n \to \{0,1\}$ is called* condition, *and $D : \{0,1\}^n \to \{0,1\}$ is called* distinguisher. *We say that a conditional test (A, D) is ϵ-fooled by a distribution P over $\{0,1\}^n$ if*

$$|\Pr_{X \leftarrow P}[D(X) = 1|A(X) = 1] - \Pr_{X \leftarrow U_n}[D(X) = 1|A(X) = 1]| \leq \epsilon$$

The density *of a condition A is $\Pr_{X \leftarrow U_n}[A(X) = 1]$. The density of a conditional test (A, D) is the density of A, and we say that the test has size s if both A, D are circuits of size at most s.*

With this terminology, Lemma 2.1 can be rephrased as follows:

Corollary 2.4. *There exists a constant $B > 0$ such that for every constant $C > 0$ the following holds: Let $G : \{0,1\}^r \to \{0,1\}^n$, and let $E : \{0,1\}^n \to \{0,1\}^\ell$ be a size s circuit. If G is not ℓ^{-C}-fooled by E then there exist a conditional test (A, D) of size $s + O(\ell)$ and density $\geq 2^{-O(\ell)}$ which is not $\ell^{-(C+5)}$-fooled by G.*

2.1 PRGs against conditional tests

In light of Corollary 2.4, we would like to construct a PRG that fools poly-size conditional tests. Shaltiel and Umans [29] constructed PRGs against conditional tests. However, the setup considered there is quite different. It is not required that the PRG is computable by a deterministic procedure, and the PRG is allowed to have access to an NP-oracle. Moreover, the PRG receives the condition A before producing its output. This allows the PRG to find "interesting" elements $x \in \{x : A(x) = 1\}$ using its NP oracle, and this ability is critically used by the PRG. There are also additional advantages to having an NP oracle, which we can't use in our setup. On the other hand, we have an advantage over the setup of [29], we are guaranteed that the density of A is roughly $2^{-\ell}$ and are shooting for seed length $O(\ell)$ (whereas in [29] one wants seed $O(\log n)$) regardless of the density. We are not able to construct polynomial time PRGs against conditional tests in our setup. However, we are able to achieve the following weaker objects:

Definition 2.5 (cd-PRG). *A function $G : \{0,1\}^{r_1+r_2} \to \{0,1\}^n$ is an ϵ-cd-PRG for a class $\mathcal{C}$ of conditional tests, if with probability $1 - \epsilon/2$ over choosing $s_1 \leftarrow U_{r_1}$, every conditional test (A, D) in $\mathcal{C}$ is $\epsilon/2$-fooled by $G_{s_1}(U_{r_2})$, where $G_{s_1} : \{0,1\}^{r_2} \to \{0,1\}^n$ is defined by $G_{s_1}(s_2) = G(s_1 \circ s_2)$. G is an ϵ-wcd-PRG if for every conditional test (A, D) in $\mathcal{C}$, with probability $1 - \epsilon/2$ over choosing $s_1 \leftarrow U_{r_1}$, (A, D) is $\epsilon/2$-fooled by $G_{s_1}(U_{r_2})$.*

Note that a cd-PRG is in particular a wcd-PRG. Loosely speaking, in both notions, this definition allows the PRG to choose a uniform $s_1 \in \{0,1\}^{r_1}$ which will not be affected by the condition A. Only the second part of the seed (namely, s_2) is affected by conditioning.[13]

We are able to use the machinery developed by Shaltiel and Umans [29] (together with additional ideas) to construct cd-PRGs under the assumption that E is hard for exponential size nondeterministic NP-circuits, and wcd-PRGs under the weaker assumption that E is hard for exponential size nondeterministic circuits. We elaborate on this in Section 3. We are not able to show that wcd-PRGs give nb-PRGs, but are able to show that the stronger notion of cd-PRGs give nb-PRGs. This follows by the next lemma.

Lemma 2.6 (cd-PRGs are nb-PRGs). *Let C be a constant and let $G : \{0,1\}^{r_1+r_2} \to \{0,1\}^n$ be an $(\ell^{-(C+5)}/2)$-cd-PRG for conditional tests of size $s + O(\ell)$ and density at least $2^{-O(\ell)}$ then G is a ℓ^{-C}-nb-PRG for size s circuits $E : \{0,1\}^n \to \{0,1\}^\ell$.*

Proof. Let $E : \{0,1\}^n \to \{0,1\}^\ell$ be a circuit of size s and assume that E is not ℓ^{-C}-fooled by G, meaning that $C(E(U_r))$ and $E(U_n))$ are not ℓ^{-C}-close. By an averaging argument, for an $\ell^{-C}/2$-fraction of $s_1 \in \{0,1\}^{r_1}$, E is not $(\ell^{-C}/2)$-fooled by G_{s_1}, and call such s_1 *useful.* by Corollary 2.1 for every useful s_1, there exists a conditional test

[13] For perspective, let us consider an analogous definition to the standard setup of PRGs (or even nb-PRGs): Let $\mathcal{C}$ be a class of functions $C : \{0,1\}^n \to \{0,1\}^\ell$. If for every test in the class $\mathcal{C}$, with probability $1 - \epsilon/2$, over $s_1 \leftarrow U_{r_1}$, $G_{s_1}(U_{r_2})$ $\epsilon/2$-fools the test, then G is an ϵ-nb-PRG. Thus, for the more standard notion of ϵ-PRGs (or even ϵ-nb-PRGs) the modifications made in Definition 2.5 are immaterial and this is why this notion is not usually defined in these setups.

(A_{s_1}, D_{s_1}) of size $s + O(\ell)$ and density $2^{-O(\ell)}$ which is not $(\ell^{-(C+5)}/2)$-fooled by G_{s_1}. □

We remark that the fact that G_{s_1} fools *all* circuits $s+O(\ell)$ is an overkill for the argument. However, we do need G_{s_1} to fool many tests simultaneously, and this is why the argument does not work with wcd-PRGs.

3. A CONSTRUCTION OF CD-PRGS

We now show that Assumption 1.7 implies cd-PRGs. The construction is specified in Figure 1. The intuition for the construction builds on ideas of Shaltiel and Umans [29] (together with additional ideas) and we give a high level intuition in the next paragraph.

Intuition for the construction

Recall that we are aiming to construct a cd-PRG for conditional tests with density at least 2^{-d}. The first r_1 bits of the seed will be used to select a poly(n)-wise independent permutation $h : \{0,1\}^n \to \{0,1\}^n$. This costs $r_1 > n$ random bits (that we cannot afford) and we will derandomize this choice using a PRG G_1 for nondeterministic NP-circuits.

Recall that the definition of cd-PRGs discusses choosing a random seed of length r_1 and fixing its value. Therefore, it will be sufficient to argue that with high probability we obtain a fixed permutation $h : \{0,1\}^n \to \{0,1\}^n$ such that for every relevant condition circuit $A : \{0,1\}^n \to \{0,1\}$, truncating the output of h to $n-d$ bits, we achieve a hash function that hashes $\{x : A(x) = 1\}$ without many collisions. Formally, we will truncate to length $n - v$ bits, for $v = d+O(\log n)$ bits, and will argue that the obtained hash function has the property that no output $z \in \{0,1\}^{n-v}$ has more than a polynomial number of preimages in $\{x : A(x) = 1\}$. However, for the sake of this informal explanation, let us oversimplify, and pretend that by choosing $v = d$, we can truncate h to obtain a hash function that is one-to-one on $\{x : A(x) = 1\}$. We now continue describing the second part of the seed. We use a short seed s_2 and apply a PRG G_2 against NP-circuits to generate a pseudorandom string $G_2(s_2)$ of length $n - v$. For each such pseudorandom string $z = G_2(s_2)$, its preimage in $\{x : A(x) = 1\}$ under the truncated version of h is unique, and can be obtained by $h^{-1}(G_2(s_2) \circ w)$ for some unique $w \in \{0,1\}^v$. In other words, the distribution $R = h^{-1}(G_2(s_2) \circ w)$ where s_2 is a uniform seed of G_2 and w is chosen uniformly from $\{0,1\}^v$ has the property that $(R|A(R) = 1)$ is a bijection of the pseudorandom strings generated by G_2. Note that an NP-circuit can compute this bijection, and as G_2 fools such circuit, we obtain a cd-PRG. (The actual argument is more technical as we are not guaranteed that the hash function is one to one, but follows using the same intuition).

We now prove that the construction yields cd-PRGs.

THEOREM 3.1. *There exist a constant c such that for every constant b the following holds: Let $v = d + O(b \cdot \log n)$ and $\epsilon \geq n^{-b}$. If G_1, G_2 are $(\epsilon/100)$-PRGs for nondeterministic NP circuit of size $n^{b^2 \cdot c}$ then G is an ϵ-cd-PRG for conditional tests of size n^b and density $\geq 2^{-d}$.*

Note that for every constant $\delta > 0$, Assumption 1.7 guarantees the existence of G_1, G_2 with $d_1 = d_2 = n^\delta$ that can be computed in time poly(n) and have error n^{-2b}, giving that:

COROLLARY 3.2. *If Assumption 1.7 holds then for every constants $b \geq 1, \delta > 0$ and parameter d, there exists a poly(n)-time computable (n^{-b})-cd-PRG $G : \{0,1\}^{d+n^\delta+O(b\log n)} \to \{0,1\}^n$ for conditional tests of size n^b and density $\geq 2^{-d}$.*

Theorem 1.5 follows from Lemma 2.6, Corollary 3.2 and the discussion in Section 1.1.1 showing that Assumption 1.7 follows from the assumption in Theorem 1.5.

3.1 Analysis of the construction

We now prove Theorem 3.1. The seed s_1 is used to pick a permutation $h_{G(s_1)} : \{0,1\}^n \to \{0,1\}^n$. We want this permutation to be good in the following respect:

DEFINITION 3.3 (SPLITTING FUNCTION). *Given a function $h : \{0,1\}^n \to \{0,1\}^n$, let $h' : \{0,1\}^n \to \{0,1\}^{n-v}$ be the function obtained by truncating the last v output bits of h. Let $\delta > 0$. A function $h : \{0,1\}^n \to \{0,1\}^n$ is δ-*splitting *for $A : \{0,1\}^n \to \{0,1\}$ if for every $y \in \{0,1\}^{n-v}$, the quantities $a_y := |\{x : A(x) = 1 \wedge h'(x) = y\}|$ and $a := |\{x : A(x) = 1\}|$, satisfy $a_y \leq (1+\delta) \cdot a \cdot 2^{-(n-v)}$.*

We set $\delta = n^{-2b}$ and will show that a poly(n)-wise independent permutation is δ-splitting for a condition A with high probability. The full proof (which relies on a Chernoff bound for t-wise independent permutations) is deferred to the full version.

LEMMA 3.4. *Let $A : \{0,1\}^n \to \{0,1\}$ be a condition with density $\geq 2^{-v+10\log(n/\delta)+2}$. The probability over $s \leftarrow U_q$ that h_s is not $(\delta/4)$-splitting for A is at most $2^{-n^{5b}}$.*

By a union bound over all circuits of size n^b we get that:

COROLLARY 3.5. *The probability over $s \leftarrow U_q$ that h_s is not $(\delta/4)$-splitting for all circuits A of size n^b with density $\geq 2^{-v+10\log(n/\delta)+2}$ is at most 2^{-n}.*

We can achieve a similar result in the experiment $s \leftarrow G_1(U_{d_1})$ rather than $s \leftarrow U_q$.

LEMMA 3.6. *The probability over $s_1 \leftarrow U_{d_1}$ that $h_{G_1(s_1)}$ is not δ-splitting for all circuits A of size n^b with density $\geq 2^{-v+10\log(n/\delta)}$ is at most $2^{-n} + \epsilon/100 \leq \epsilon/50$.*

Lemma 3.6 follows from noticing that given an $s \in \{0,1\}^q$, The test $T(s)$ which checks whether there exists a condition A with density $\geq 2^{-v+10\log(n/\delta)}$ such that h_s is not δ-splitting for A, can be implemented by a size $n^{O(b^2)}$ nondeterministic NP-circuit. We have that G_1 fools such tests, which means that the probabilities in the experiments $s \leftarrow G_1(U_{d_1})$ and $s \leftarrow U_q$ are close.

To implement the test $T(s)$ note that by "approximate counting of NP witnesses" [31, 19, 5], given a condition A, an NP-circuit can check whether the density is at least $2^{-v+10\log(n/\delta)}$, and compute very good approximations to the quantities a and a_y. This means that the test $T(s)$ can be expressed as: "does there exists an A and y such that A has sufficiently large density and $a_y/a > (1+\delta) \cdot 2^{-(n-v)}$".

Figure 1: A cd-PRG

Parameters: n and v, b, d_1, d_2.

Ingredients:

- A family H of 2^q permutations, for $q = n^{O(b)}$, indexed by strings $s \in \{0,1\}^q$. Given s, the permutations h_s, h_s^{-1} are required to be computable in time $n^{O(b)}$. H should be $(2^{-n^{11b}})$-close to (n^{10b})-wise independent, namely, for every distinct $x_1, \ldots, x_{n^{10b}} \in \{0,1\}^n$, $(h_s(x_1), \ldots, h_s(x_{n^{10b}}))_{s \leftarrow U_q}$ is $2^{-n^{11b}}$-close to the uniform distribution on n^{10b} distinct elements from $\{0,1\}^n$. There are such explicit constructions in the literature [11, 16, 20].
- A function $G_1 : \{0,1\}^{d_1} \rightarrow \{0,1\}^q$ that is computable in time polynomial in n.
- A function $G_2 : \{0,1\}^{d_2} \rightarrow \{0,1\}^{n-v}$ that is computable in time polynomial in n.

Both G_1, G_2 will be assumed to be pseudorandom generators for nondeterministic NP-circuits of size $n^{O(b)}$ which exist by assumption 1.7.

The cd-PRG: $G : \{0,1\}^{r := d_1 + d_2 + v} \rightarrow \{0,1\}^n$ is defined as follows: Given $s_1 \in \{0,1\}^{d_1}$, $s_2 \in \{0,1\}^{d_2}$ and $w \in \{0,1\}^v$, we define $G(s_1, s_2, w) = h^{-1}_{G_1(s_1)}(G_2(s_2) \circ w)$. We set $r_1 = d_1$ and $r_2 = d_2 + v$ so that $r = r_1 + r_2$.

This test can be implemented by a nondeterministic NP-circuit. A full proof is deferred to the full version.[14]

Finally, we show that if a good s_1 (namely, one for which $h_{G(s_1)}$ is δ-splitting) is chosen. Then the distribution $G(s_1, U_{r_2}) = h^{-1}_{G_1(s_1)}(G_2(U_{d_2}) \circ U_v)$ $\epsilon/2$-fools every relevant conditional test (A, D).

LEMMA 3.7. *Let $h : \{0,1\}^n \rightarrow \{0,1\}^n$ be a δ-splitting permutation, and (A, D) be a conditional test of size n^b and density $\geq 2^{-v+10\log(n/\delta)}$. Then $h^{-1}(G_2(U_{d_2}) \circ U_v)$ $(O(\delta) + \epsilon/100)$-fools (A, D).*

This concludes the proof of Theorem 3.1. The proof of Lemma 3.7 is deferred to the full version, and is based on a similar argument of Shaltiel and Umans [29]. For Lemma 3.7, it is sufficient that G_2 fools NP-circuits that make nonadaptive calls to their oracle, and by [29], such PRGs can be obtained from the assumption: E is hard for exponential size nondeterministic circuits. The idea behind the proof is that U_n can be presented as $R_1 = (h^{-1}(Y_1 \circ W))_{Y_1 \leftarrow U_{n-v}, W \leftarrow U_v}$ and the distribution considered in the lemma is $R_2 = h^{-1}(Y_2 \circ W))_{Y_2 \leftarrow G_2(U_{d_2}), W \leftarrow U_v}$. For fixed $y \in \{0,1\}^{n-v}$, these coincide. NP-circuits can approximate the quantities $\ell_y = |\{x : A(x) = 1 \wedge D(x) = 1 \wedge h'(x) = y\}|$, and $a = |\{x : A(x) = 1\}|$, and therefore an NP-circuit can output 1 on distribution Y_i with probability proportional to $\mathbb{E}_{y \leftarrow Y_i}[\frac{\ell_y \cdot 2^v}{a}]$ which is equal to $\Pr[D(R_i) = 1 | A(R_i) = 1]$ if h is 0-splitting. Consequently, if (D, A) distinguishes R_1 from R_2 then an NP-circuit distinguishes Y_1 from Y_2. A more careful argument shows that this holds also for h that is δ-splitting and not 0-splitting.

We now make an observation that will be helpful for the proof of Theorem 1.6: The proof of Theorem 3.1 follows just the same if we allow conditions A to be *nondeterministic circuits* rather than deterministic circuits. This is because the only properties of A used, is that an NP-circuit can approximate the size of sets of the form $\{x : A(x) = 1 \wedge B(x) = 1\}$, where B is some deterministic circuit. This holds also for nondeterministic circuits A.

Another observation that will be useful for proving Theorem 1.6 is that G of Corollary 3.2 fools all nondeterministic circuits C of size n^b. This follows as for fixed s_1, G is a poly(n)-size permutation of the distribution $G_2(U_{d_2}) \circ U_v$ which fools nondeterministic circuits of size $n^{O(b)}$.

4. NB-PRGS FOR SAMPLING PROCEDURES WITH LOW ENTROPY

In this section we prove Theorem 1.6. Our proof uses some ideas by Dubrov and Ishai [7]. We are shooting to construct an ϵ-nb-PRG for size n^c circuits $C : \{0,1\}^n \rightarrow \{0,1\}^n$ with $H(C(U_n)) \leq k$. Let $\epsilon' = \Omega(\epsilon^2/k)$ and let $G : \{0,1\}^r \rightarrow \{0,1\}^n$ be an ϵ'-cd-PRG for conditional tests of size $n^{O(c)}$ and density $d = O(k/\epsilon)$. Such a poly(n)-time G follows from Assumption 1.7 by Corollary 3.2. Using the assumption that $k \geq n^e$, we have that $r = O(d) = O(k/\epsilon)$ as required. As noted earlier, G also fools conditional tests where the condition A is a size $n^{O(c)}$ nondeterministic circuit, and it fools nondeterministic circuits of size $n^{O(c)}$.

Assume that some size n^c circuit C with $H(C(U_n)) \leq k$ is not ϵ-fooled by G. We will try to show that there exists a circuit C' of size $n^{O(c)}$ and output length $O(k/\epsilon)$ that is not fooled by G. We can then use the previous argument with C'.

LEMMA 4.1. *There exists a nondeterministic circuit $A : \{0,1\}^n \rightarrow \{0,1\}$ of size $n^{O(c)}$ and density $\geq 1 - \epsilon$, such that $|\{x : A(x) = 1\}| \leq 2^{O(k/\epsilon)}$ and the distributions $R = (C(X) | A(X) = 1)_{X \leftarrow U_n}$, and $V = (C(G(Y)) | A(C(G(Y))) = 1)_{Y \leftarrow U_r}$ are not $\epsilon/10$-close.*

The proof of Lemma 4.1 is deferred to the full version. We provide a sketch here. Let A be a nondeterministic circuit that accepts $S = \left\{z : \Pr[C(U_n) = z] \geq 2^{-10k/\epsilon}\right\}$. Such a circuit follows by the aforementioned AM-protocol of Gold-

[14] This is the place where nondeterministic NP-circuits come up. We remark that by the AM protocol of Goldwasser and Sipser [10], a nondeterministic circuit can check whether the number of inputs accepted by a given circuit A is larger than some constant quantity. If we are shooting to construct wcd-PRGs (rather than cd-PRGs), then this observation (together with some small modifications in the proofs above) leads to an implementation of $T(s)$ by a nondeterministic circuit. This enables us to relax Assumption 1.7 and replace nondeterministic NP-circuits with nondeterministic circuits.

wasser and Sipser [10]. Since $H(C(U_n)) \leq k$, the distribution $C(U_n)$ cannot place a lot of weight on S^c, implying that $\Pr[A(C(U_n)) = 1] \geq 1 - \epsilon/10$. We know that $A(C(\cdot))$ is ϵ'-fooled by G and $\epsilon' \leq \epsilon/10$. Therefore, "some of the statistical distance" between $C(U_n)$ and $C(G(U_r))$ must be "preserved" under the condition A, and so R and V are not $\epsilon/10$-close.

The set S is of size $\leq 2^{ck/\epsilon}$ for some constant c. It is standard that with positive probability, picking a random function from a pairwise independent family of hash functions $h : \{0,1\}^n \rightarrow \{0,1\}^{2ck/\epsilon}$ is one to one on S, and such a function can be implemented by a poly-size circuit. It follows that $R = (h(C(X))|A(X) = 1)_{X \leftarrow U_n}$, and $V = (h(C(G(Y)))|A(C(G(Y))) = 1)_{Y \leftarrow U_r}$ are not $\epsilon/10$-close. We can now apply Lemma 2.1 on R and V (which are on $O(k/\epsilon)$ output bits) to obtain a conditional test (A', D') of size $n^{O(c)}$ that distinguishes between them with advantage $\frac{\epsilon/10}{O(k/\epsilon)} = \Omega(\epsilon^2/k) \geq \epsilon'$. Furthermore, A' has density $2^{-O(k/\epsilon)}$. This means that the conditional test (A'', D) where $A''(x) = A(x) \wedge A'(x)$ is a nondeterministic circuit of size $n^{O(c)}$ with density $\geq 2^{-O(k \cdot \log\log(1/\epsilon)/\epsilon)-1}$ such that (A'', D) is not ϵ'-fooled by G. This is a contradiction, as G is an ϵ'-cd-PRG against size $n^{O(c)}$ conditional tests with this density.

5. NB-PRGS FOR POLY-SIZE CONSTANT DEPTH CIRCUITS

The following theorem generalizes Theorem 1.10 and Theorem 1.11.

Theorem 5.1. *Let $\ell \leq n < M$ be positive integers, and let $\epsilon \geq 2^{-n}$ be a parameter. There is a procedure $G : \{0,1\}^r \rightarrow \{0,1\}^n$ such that for every circuit $C : \{0,1\}^n \rightarrow \{0,1\}^\ell$ of size M and depth d, the distribution $C(G(U_r))$ is ϵ-close to $C(U_n)$, and it is possible to take:*

- *$r = \ell \cdot O(\log M)^{d+7} \cdot \log^7(1/\epsilon)$ and then G can be computed in time poly$(n, \log^d M)$.*
- *$r = \ell^{1+\alpha} \cdot (\log M)^d \cdot (\log(M/\epsilon))^{O(1/\alpha)}$ for an arbitrary constant $\alpha > 0$, and then G can be computed by a uniform family of circuits of size poly$(n, \log^d M)$ and depth $O(1/\alpha)$.*

We remark that the procedure G needs to know the parameters ℓ, n, d, M and ϵ. However, the running time/size of G is a fixed polynomial in $(n, \log^d M)$, and the depth depends of G depends only on α.

5.1 Adapting the pseudorandom generator of [34]

There does not seem to be a general method to transform Boolean PRGs for constant depth circuits into nb-PRGs for constant depth circuits. Indeed, the nb-PRG of Dubrov and Ishai relies (amongst other things) on specific properties of the proof of correctness of the Nisan-Wigderson generator. In order to prove Theorem 1.10 we will exploit specific properties of the recent Boolean PRG construction of Trevisan and Xue [34].

The technique of [34] is based on pseudorandom restrictions - namely, a randomness efficient way to sample a small (but noticeable) subset S of the n input variables of a circuit $C : \{0,1\}^n \rightarrow \{0,1\}$ of poly-size and constant depth, such that if the remaining bits are chosen at random, then the restricted circuit (which now gets $|S|$ bits as input) simplifies into a poly-logarithmic depth decision tree. It is easy to verify that their approach works also on non-boolean circuits $C : \{0,1\}^n \rightarrow \{0,1\}^\ell$ and these simplify into a poly-logarithmic depth decision forests.

A PRG (with very low stretch) against boolean circuits can be constructed by sampling a set S pseudorandomly, and then sampling the values of the bits outside of S uniformly, and the value of bits inside S by poly-logarithmic-wise independence. The rationale is that following the restriction the simplified circuit cannot distinguish a uniform $|S|$ bit string from a poly-logarithmic-wise independent one. This gives a PRG with almost linear seed, and the seed can be reduced by recursion in which the step of sampling the value of bits outside of S uses the PRG recursively, rather than a uniform string. This argument can be extended to the non-boolean setting at the cost of multiplying the independence by ℓ. A full proof is deferred to the full version.

6. DISCUSSION AND OPEN PROBLEMS

The notions of cd-PRGs and wcd-PRGs are quite strong, and we are expecting that they will find applications in various setups.

We are using hardness against nondeterministic NP-circuits to construct nb-PRGs. Is it possible to use hardness for a weaker class? (Say nondeterministic circuits or NP circuits). We remark that we can construct wcd-PRGs under hardness for nondeterministic circuits (which seems like the best that can be done). However, we were not able reduce nb-PRGs to wcd-PRGs. This suggests that we may be able to achieve the weaker hardness assumption by improving the reduction so that we can use a weaker notion than cd-PRGs and showing how to construct such PRGs using hardness assumptions for nondeterministic circuits.

Recently, it is shown by [1] (and following work) that many cryptographic primitives can be computed by low circuit classes (and in particular, by poly-size constant depth circuits). We expect that nb-PRGs for poly-size constant depth circuits can be useful in this setup (as they fool the circuits that implement the primitives). Moreover, our PRGs are also implementable in poly-size and constant depth, and this may be helpful (even in the boolean case), as low complexity security reductions can run them.

Is it possible to give unconditional constructions of nb-PRGs against size $s = n^c$ and depth d circuits $C : \{0,1\}^n \rightarrow \{0,1\}^\ell$ with seed length $O(\ell) + O(\log^{a_d} s)$ for a constant a_d that depends only on d? Note that we achieve the multiplication of the two terms, but it may be possible to achieve the sum (even without new progress on circuit lower bounds for constant depth circuits).

7. REFERENCES

[1] B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography in nc^0. *SIAM J. Comput.*, 36(4):845–888, 2006.

[2] 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.

[3] 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.

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

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

[6] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal on Computing*, 13(4):850–864, Nov. 1984.

[7] 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.

[8] O. Goldreich, N. Nisan, and A. Wigderson. On Yao's XOR lemma. Technical Report TR95–050, *Electronic Colloquium on Computational Complexity*, March 1995. www.eccc.uni-trier.de/.

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

[10] S. Goldwasser and M. 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.

[11] W. T. Gowers. An almost m-wise independent random permutation of the cube. *Combinatorics, Probability and Computing*, 5:119–130, 6 1996.

[12] I. Haitner, D. Harnik, and O. Reingold. On the power of the randomized iterate. *SIAM J. Comput.*, 40(6):1486–1528, 2011.

[13] I. Haitner, O. Reingold, and S. P. Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, pages 437–446, 2010.

[14] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[15] T. Holenstein. Pseudorandom generators from one-way functions: A simple construction for any hardness. In *TCC*, pages 443–461, 2006.

[16] S. Hoory, A. Magen, S. Myers, and C. Rackoff. Simple permutations mix well. *Theor. Comput. Sci.*, 348(2-3):251–261, 2005.

[17] R. Impagliazzo, R. Shaltiel, and A. Wigderson. Reducing the seed length in the nisan-wigderson generator. *Combinatorica*, 26(6):647–681, 2006.

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

[19] 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.

[20] E. Kaplan, M. Naor, and O. Reingold. Derandomized constructions of k-wise (almost) independent permutations. *Algorithmica*, 55(1):113–133, 2009.

[21] J. Kinne, D. van Melkebeek, and R. Shaltiel. Pseudorandom generators, typically-correct derandomization, and circuit lower bounds. *Computational Complexity*, 21(1):3–61, 2012.

[22] 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.

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

[24] N. Nisan. Pseudorandom bits for constant depth circuits. *Combinatorica*, 11(1):63–70, 1991.

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

[26] R. Shaltiel. Typically-correct derandomization. *SIGACT News*, 41(2):57–72, 2010.

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

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

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

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

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

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

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

[34] L. Trevisan and T. K. Xue. A derandomized switching lemma and an improved derandomization of ac0. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:116, 2012.

[35] C. Umans. Pseudo-random generators for all hardnesses. *Journal of Computer and System Sciences*, 67:419–440, 2003.

[36] C. Umans. Reconstructive dispersers and hitting set generators. *Algorithmica*, 55(1):134–156, 2009.

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

[38] E. Viola. The complexity of distributions. *SIAM J. Comput.*, 41(1):191–218, 2012.

[39] A. C. Yao. Theory and applications of trapdoor functions (extended abstract). In *FOCS*, pages 80–91, 1982.