On Hardness Assumptions Needed for "Extreme High-End" PRGs and Fast Derandomization

Ronen Shaltiel* Emanuele Viola†

November 20, 2023

Abstract

The hardness vs. randomness paradigm aims to explicitly construct pseudorandom generators $G : \{0,1\}^r \to \{0,1\}^m$ that fool circuits of size m, assuming the existence of explicit hard functions. A "high-end PRG" with seed length $r = O(\log m)$ (implying BPP=P) was achieved in a seminal work of Impagliazzo and Wigderson (STOC 1997), assuming THE HIGH-END HARDNESS ASSUMPTION: there exist constants $0 < \beta < 1 < B$, and functions computable in time $2^{B \cdot n}$ that cannot be computed by circuits of size $2^{\beta \cdot n}$.

Recently, motivated by fast derandomization of randomized algorithms, Doron et al. (FOCS 2020) and Chen and Tell (STOC 2021), construct "extreme high-end PRGs" with seed length $r = (1 + o(1)) \cdot \log m$, under qualitatively stronger assumptions.

We study whether extreme high-end PRGs can be constructed from the following scaled version of the assumption which we call THE EXTREME HIGH-END HARDNESS ASSUMPTION, and in which $\beta = 1 - o(1)$ and $B = 1 + o(1)$. We give a partial negative answer:

- Doron et al. compose a PEG (pseudo-entropy generator) with an extractor. The PEG is based on a hardness assumption for MA-type circuits. We show that black-box PEG constructions from THE EXTREME HIGH-END HARDNESS ASSUMPTION must have large seed length (and so cannot be used to obtain extreme high-end PRGs by applying an extractor). To prove this, we establish a new property of (general) black-box PRG constructions from hard functions: it is possible to fix many output bits of the construction while fixing few bits of the hard function. This property distinguishes PRG constructions from typical extractor constructions, and this may explain why it is difficult to design PRG constructions.
- Chen and Tell compose two PRGs: $G_1 : \{0,1\}^{(1+o(1)) \cdot \log m} \to \{0,1\}^{r_2 = m^{\Omega(1)}}$ and $G_2 : \{0,1\}^{r_2} \to \{0,1\}^m$. The first PRG is based on THE EXTREME HIGH-END HARDNESS ASSUMPTION, and the second PRG needs to run in time $m^{1+o(1)}$, and is based on one way functions. We show that in black-box proofs of hardness amplification to $\frac{1}{2} + 1/m$, reductions must make $\Omega(m)$ queries, even in the extreme high-end. Known PRG constructions from hard functions are black-box and use (or imply) hardness amplification, and so cannot be used to construct a PRG G_2 from THE EXTREME HIGH-END HARDNESS ASSUMPTION. The new feature of our hardness amplification result is that it applies even to the extreme high-end setting of parameters, whereas past work does not. Our techniques also improve recent lower bounds of Ron-Zewi, Shaltiel and Varma (ITCS 2021) on the number of queries of local list-decoding algorithms.

*Department of computer science, University of Haifa. E-mail: ronen@cs.haifa.ac.il. This research was supported by ISF grants 1628/17 and 1006/23.

†Khoury College of Computer Sciences, Northeastern University, E-mail: viola@ccs.neu.edu. Supported by NSF CCF award 1813930 and NSF CCF award 2114116.

Contents

1 Introduction

1.1 Background

The hardness vs. randomness paradigm (initiated in [Yao82, BM84, NW94] and followed up by a long sequence of work [BFNW93, Imp95, IW97, STV01, KvM02, MV05, ISW99, ISW06, SU05, Uma03, Uma09, SU06, AIKS16, DMOZ20, CT21b, CT21a]) aims to explicitly construct pseudorandom generators (PRGs) from explicit hard functions.

Definition 1.1 (PRGs)**.** *A function $G : \{0,1\}^r \to \{0,1\}^m$ is an ϵ-PRG for a function $D : \{0,1\}^m \to \{0,1\}$, if*

$$|\Pr[D(G(U_r)) = 1] - \Pr[D(U_m) = 1]| \leq \epsilon.$$

G is an ϵ-PRG for a class $\mathcal{D}$ of functions $D : \{0,1\}^m \to \{0,1\}$, if for every D in $\mathcal{D}$, G is an ϵ-PRG for D. If we omit ϵ or $\mathcal{D}$, the default choices are $\epsilon = 1/10$, and $\mathcal{D}$ is the class of circuits of size m.

Explicit pseudorandom generators have many applications in computer science. The signature application of PRGs is to derandomize randomized algorithms (by running the algorithm using all outputs of the PRG). This is quantitatively specified in the proposition below.

Proposition 1.2 (standard)**.** *If $G : \{0,1\}^r \to \{0,1\}^m$ is a PRG, then every randomized algorithm running in time m can be simulated by a deterministic algorithm in time $\mathrm{Time}_{\mathrm{all}}(G) + 2^r \cdot m$, where $\mathrm{Time}_{\mathrm{all}}(G)$ is the time it takes to compute the output of G on all 2^r inputs, and is obviously upper bounded by $2^r \cdot \mathrm{Time}(G)$, where $\mathrm{Time}(G)$ is the time it takes to compute G on a given input.*

High-end PRGs that imply BPP=P. A corollary of Proposition 1.2 is that a PRG $G : \{0,1\}^{r=O(\log m)} \to \{0,1\}^m$ with $\mathrm{Time}(G) = \mathrm{poly}(m)$ implies that BPP=P.[1]

Such PRGs are often referred to as *"high-end PRGs"*. Historically, this name aims to distinguish them from weaker *"low-end PRGs"* which have $r = m^{o(1)}$, and $\mathrm{Time}(G) = 2^{O(r)}$, which in turn imply the weaker conclusion that BPP is in subexponential time, see [ISW06, SU05] for a discussion.

Extreme high-end PRGs and fast derandomization. Recently, Doron et al. [DMOZ20] asked whether it is possible to obtain a faster derandomization. Here, the goal is to show that a randomized algorithm running in time m can be simulated by a deterministic algorithm running in time $O(m^c)$ for the smallest possible constant c.

The time of the deterministic simulation of Theorem 1.2 depends on both the seed length r, and $\mathrm{Time}_{\mathrm{all}}(G)$. Note that even if we take r to the extreme[2], and have a PRG with $r = 1 \cdot \log m$, then the time of the simulation is at least $2^r \cdot m = m^2$. This time can be achieved if furthermore, $\mathrm{Time}_{\mathrm{all}}(G) = O(2^r \cdot m) = O(m^2)$ (which follows if $\mathrm{Time}(G) = O(m)$). This means that we can hope to achieve $c = 2$ (that is, a quadratic time simulation) if we have such PRGs, which we will call *"extreme high-end"* PRGs.

Definition 1.3 (Extreme high-end PRGs)**.** *$G : \{0,1\}^r \to \{0,1\}^m$ is an extreme high-end PRG if:*

[1] Note that in this range of parameters there is no reason to distinguish between $\mathrm{Time}(G)$ and $\mathrm{Time}_{\mathrm{all}}(G)$, as $\mathrm{Time}_{\mathrm{all}}(G) = \mathrm{poly}(m)$ if and only if $\mathrm{Time}(G) = \mathrm{poly}(m)$, and this is why past work is stated in terms of $\mathrm{Time}(G)$ and not $\mathrm{Time}_{\mathrm{all}}(G)$.

[2] A PRG must have $r \geq \log m - O(\log \log m)$ as otherwise, a circuit of size m could be hardwired with prefixes of length $r + 1$ for all 2^r pseudorandom strings, and distinguish a uniform string from a pseudorandom string.

Pseudorandomness: G is a PRG with seed length $r = (1 + o(1)) \cdot \log m$.
Explicitness: $\mathrm{Time}_{\mathrm{all}}(G) = m^{2+o(1)}$ *(which follows if* $\mathrm{Time}(G) = m^{1+o(1)}$*).*

These parameters are chosen so that an extreme high-end PRG implies that randomized algorithms running in time m can be simulated deterministically in time $m^{2+o(1)}$.

There are reasons to think that this quadratic slowdown is the best possible if one seeks the smallest possible c such that every randomized algorithm running in time m can be simulated deterministic time m^c. More precisely, the problem of "univariate identity testing" is in $\mathrm{BPTIME}(\tilde{O}(n))$ but not in $\mathrm{DTIME}(n^{2-o(1)})$, under certain assumptions on "fine grained complexity" introduced by Carmosino et al. [CGI+16]. See [DMOZ20] for details and a discussion.

We remark that this lower bound still allows a deterministic simulation that runs in time $O(m \cdot n)$ (where n is the input length) and a simulation that approaches this time bound was obtained by Chen and Tell [CT21b] under certain hardness assumptions. See [CT21b] for details and a discussion.

Hardness implied by PRGs. PRGs immediately imply circuit lower bounds that are beyond our current ability. Consequently, constructing explicit PRGs, requires circuit lower bounds (namely the existence of explicit functions that cannot be computed by small circuits). In particular, high-end PRGs imply the existence of functions $f : \{0,1\}^\ell \to \{0,1\}$ which cannot be computed by circuits of size $2^{\Omega(\ell)}$, and $\mathrm{Time}(f) = 2^{O(\ell)}$.[3]

1.1.1 The Impagliazzo-Wigderson (high-end) hardness assumption

The goal of the hardness vs. randomness program is to construct PRGs based on lower bounds that are as strong (or almost as strong) as the ones implied by the PRG. A major milestone in this program was achieved by Impagliazzo and Wigderson [IW97].

Theorem 1.4 ([IW97])**.** *A high-end PRG follows from the following assumption:*

THE HIGH-END HARDNESS ASSUMPTION*: There exist constants $0 < \beta < 1 < B$, and a function $f : \{0,1\}^\ell \to \{0,1\}$ that satisfies:*

Hardness: f cannot be computed by circuits of size $2^{\beta \cdot \ell}$.
Explicitness: $\mathrm{Time}(f) \leq 2^{B \cdot \ell}$.

Theorem 1.4 converts hardness into pseudorandomness at close to the "correct rate" (as in the converse direction) if one does not care about the precise values of the constants β, B, and the constant hidden in the $O(\cdot)$ notation in the seed length of the high-end PRG.

1.1.2 Scaling the Impagliazzo-Wigderson assumption to the extreme high-end

In the case of extreme high-end PRGs, we insist on seed length $r \approx 1 \cdot \log m$ and the constants β, B from the THE HIGH-END HARDNESS ASSUMPTION do matter. Assuming that we don't expect to "improve" the hardness of the assumed explicit hard function, we must have $\beta \geq 1 - o(1)$ and

[3] More specifically, Impagliazzo, Shaltiel and Wigderson [ISW99] showed that if $G : \{0,1\}^r \to \{0,1\}^m$ is a PRG then for $\ell = r + 1$, there is a function $f : \{0,1\}^\ell \to \{0,1\}$ (defined by checking whether an input of length ℓ is a prefix of an output of G) such that f cannot be computed by circuits of size m, and $\mathrm{Time}_{\mathrm{all}}(f) \approx \mathrm{Time}_{\mathrm{all}}(G)$. Note that $\mathrm{Time}_{\mathrm{all}}(f) = 2^{O(\ell)}$ iff $\mathrm{Time}(f) = 2^{O(\ell)}$.

$B \leq 1 + o(1)$. Thus, imitating the approach of Impagliazzo and Wigderson [IW97] for the extreme high-end, leads to the following open problem.

Open Problem 1.5. *Show that an extreme high-end PRG follows from the following assumption:*

THE EXTREME HIGH-END HARDNESS ASSUMPTION*: There exists a function* $f : \{0,1\}^\ell \to \{0,1\}$ *that satisfies:*

Hardness: f *cannot be computed by circuits of size* $2^{(1-o(1))\cdot\ell}$.
Explicitness: $\text{Time}(f) \leq 2^{(1+o(1))\cdot\ell}$.[4]

The proof techniques of Impagliazzo and Wigderson [IW97] (as well as of later works [STV01, SU05, Uma03]) do not solve Open Problem 1.5. As we explain in Section 1.2.4, these proofs rely on the "hybrid argument" of [Yao82, GM84], and even assuming THE EXTREME HIGH-END HARDNESS ASSUMPTION, one can at best obtain a PRG $G : \{0,1\}^r \to \{0,1\}^m$ with seed length $r \geq A \cdot \log m$, where $A > 3$, and actual proofs do worse.

Recent work on extreme high-end PRGs. Recently, Doron et al. [DMOZ20], and Chen and Tell [CT21b] gave conditional constructions of extreme high-end PRGs, however, in both cases the assumption used is *stronger* than THE EXTREME HIGH-END HARDNESS ASSUMPTION. We will elaborate on these results later on.[5] An incomparable assumption was very recently used by Chen and Tell [CT21a] for constructing "targeted PRGs" (which are weaker than PRGs and yet suffice for derandomizing randomized algorithms).

Goals of this paper. In this paper, we investigate the problem of constructing explicit PRGs from explicit hard functions, focusing on open problem 1.5. More specifically, we investigate the power of "black-box proofs" that convert explicit hard functions into PRGs and related objects. We show limitations on certain recent approaches to solve Open Problem 1.5, and hope that this may help to point us in the direction of better constructions. A secondary goal of this paper is to survey recent work and point out the relationship between parameters, and potential barriers for improvement.

1.2 Black-box proofs

1.2.1 Black-box proofs for PRG constructions and hardness amplification

A black-box proof that converts hard functions into PRGs consists of two parts:

- A *construction map*. This is a map that given a candidate function $f : \{0,1\}^\ell \to \{0,1\}$ produces a candidate PRG $\mathsf{Con}(f)$. (To avoid clutter, we will denote the function $\mathsf{Con}(f)$ by $\mathsf{Con}_f : \{0,1\}^r \to \{0,1\}^m$).

[4] In the case of the extreme high-end, it does make sense to distinguish between $\text{Time}(f)$ and $\text{Time}_{\text{all}}(f)$, and one can consider starting from a weaker explicitness condition in which it is required that $\text{Time}_{\text{all}}(f) \leq 2^{(2+o(1))\cdot\ell}$.

[5] Both these papers aim for a slightly weaker goal. Rather than requiring a single PRG with seed length $r = (1 + o(1)) \cdot \log m$ and explicitness $m^{1+o(1)}$ as in Definition 1.3, their constructions show that for every $\gamma > 0$ there exists a PRG with seed length $r = (1 + \gamma) \cdot \log m$ and explicitness $m^{1+\gamma}$. We will not distinguish these two goals in the informal discussion in the introduction.

- A *reduction* establishing the correctness of the construction. This is an oracle procedure $\mathsf{Red}^{(\cdot)}$ which given oracle access to an adversary D that breaks the security of Con_f, implements an adversary C that breaks the security of f.

As we explain in Section 1.2.3, because of their combinatorial properties, reductions of this type must be *nonuniform* and receive a "nonuniform advice string" α (that may depend on the candidate function f and the adversary D). This is reflected in the formal definition below.

We will consider "black-box $\mathsf{A} \Rightarrow \mathsf{B}$ proofs" for several other choices of primitives A, B (and not just hard functions and PRGs). One such primitive is functions that are hard on average (meaning that small circuits cannot compute f correctly with high probability on a uniformly chosen input). Such functions can serve both as A and B in black-box $\mathsf{A} \Rightarrow \mathsf{B}$ proofs. In order to capture all various scenarios in one definition, we will use a terminology that will describe a primitive by what it means to "break the security" of the primitive.

Definition 1.6. *Let $\mathcal{F}_{n,m}$ denote the set of all functions from n bits to m bits.*

- *For $G \in \mathcal{F}_{r,m}$ and $D \in \mathcal{F}_{m,1}$, we say that D ϵ-PRG-breaks G, if G is not an ϵ-PRG for D.*
- *For $f, C \in \mathcal{F}_{\ell,1}$, we say that C ρ-hard-function-breaks f, if $\Pr_{x \leftarrow U_\ell}[C(x) = f(x)] \geq \rho$. We say that C hard-function-breaks f if C 1-hard-function-breaks f (meaning that $C = f$). We say that f is a ρ-hard-function for C if C does not ρ-hard-function-breaks f.*

A ρ-hard-function for circuits of a certain size, is an average-case hard function, and the case where $\rho = 1$ captures the previously considered notion of worst-case hard functions.

We now formally define black-box ρ-hard-function $\Rightarrow$ ϵ-PRG, and ρ-hard-function $\Rightarrow$ ρ'-hard-function proofs.

Definition 1.7 (Black-box proofs)**.** *Given parameters $\ell, r, m, a, \rho, \epsilon$ (resp. $\ell, \ell', a, \rho, \rho'$) a black-box ρ-hard-function $\Rightarrow$ ϵ-PRG proof (resp. a black-box ρ-hard-function $\Rightarrow$ ρ'-hard-function proof) is a pair $(\mathsf{Con}, \mathsf{Red})$ of:*

- *A construction map $\mathsf{Con} : \mathcal{F}_{\ell,1} \to \mathcal{F}_{r,m}$ (resp. $\mathsf{Con} : \mathcal{F}_{\ell,1} \to \mathcal{F}_{\ell',1}$). (We use Con_f to denote the function $\mathsf{Con}(f)$).*
- *An oracle procedure $\mathsf{Red}^{(\cdot)}(x, \alpha)$ such that:*

  For every $f \in \mathcal{F}_{\ell,1}$ and for every $D \in \mathcal{F}_{m,1}$ such that D ϵ-PRG-breaks Con_f
  (resp. for every $D \in \mathcal{F}_{\ell',1}$ such that D ρ'-hard-function-breaks Con_f),
  there exists $\alpha \in \{0,1\}^a$, such that the function $C \in \mathcal{F}_{\ell,1}$ defined by $C(x) = \mathsf{Red}^D(x, \alpha)$,
  ρ-hard-function-breaks f.

If we omit ρ, we mean $\rho = 1$. If we omit ϵ, we mean $\epsilon = 1/10$. We say that Red makes q queries, if for every $D \in \mathcal{F}_{m,1}$, $\alpha \in \{0,1\}^a$, and $x \in \{0,1\}^\ell$, $\mathsf{Red}^D(x, \alpha)$ makes at most q oracle queries.

1.2.2 Parameters of black-box proofs

To the best of our knowledge all hardness vs. randomness proofs of PRG constructions are black-box (or rely on components which are black-box). In a black-box proof, the advice length a and the number of queries q determine the "hardness loss" in the tradeoff. More specifically:

Proposition 1.8 (Number of queries determines hardness loss in black-box proofs)**.** *Let* $(\mathsf{Con}, \mathsf{Red})$ *be a black-box* hard-function $\Rightarrow$ PRG *(resp.* hard-function $\Rightarrow$ ρ'-hard-function*) proof in which* Red *makes* q *queries, and has advice length* a*. If we start from a function* $f : \{0,1\}^\ell \to \{0,1\}$ *that cannot be computed by circuits of size* s*, and apply the black-box proof, one can at best obtain that* Con_f *is a* PRG *(resp.* ρ'-hard-function*) for circuits of size* $m \leq \frac{s-a}{q} \leq \frac{s}{q}$.

Loosely speaking, Proposition 1.8 follows because, when measuring the size s of the circuit $C = \mathsf{Red}^D(\cdot, \alpha)$ that is implied by the reduction for a circuit D of size m, then this circuit is of size $q \cdot m + a$. This gives that $s \geq q \cdot m + a$, implying the proposition.

Every function $f : \{0,1\}^\ell \to \{0,1\}$ has circuits of size 2^ℓ. This means that in order for reductions to be useful in transforming hard-functions into PRGs (or average case hard functions) they must make $q \leq \frac{s-a}{m} \leq \frac{2^\ell - a}{m}$ queries, and are useless for this purpose, if $q \geq 2^\ell$. Moreover, in the extreme high-end, $m = 2^{(1-o(1))\cdot\ell}$ and so it is critical that $q \leq \frac{s}{m} \leq \frac{2^\ell}{m} = 2^{o(\ell)} = m^{o(1)} \ll m$.

Our notion of black-box does not guarantee explicitness. We place no limitation on the map Con, and so, the notion of black-box that we use, *does not* enforce that if f can be computed efficiently, then Con_f can be computed efficiently. This notion of black-box does not imply the *explicitness* of the constructed function Con_f. We make this choice, because we want to show impossibility results on black-box proofs, and this choice makes our results stronger.[6] We also remark that the terms "black-box" and "non-black-box" are used to mean many different things in the literature.[7]

Parameters for black-box proofs for the extreme high-end. As a consequence of Proposition 1.8, if we assume THE EXTREME HIGH-END HARDNESS ASSUMPTION, to obtain an extreme high-end PRG (as in Open Problem 1.5) using a black-box hard-function $\Rightarrow$ PRG proof, we first need to solve the following open problem:

Open Problem 1.9. *Does there exist a black-box* hard-function $\Rightarrow$ PRG *proof with:*

Seed length: $r = (1 + o(1)) \cdot \ell$*. (Any black-box proof must have* $r \geq \ell$*).*
Output length: $m = 2^{(1-o(1))\cdot\ell}$*. (Any black-box proof must have* $m \leq 2^\ell$*).*
Advice string length: $a = m^{1+o(1)} = 2^{(1-o(1))\cdot\ell}$*. (Any black-box proof must have* $a \geq m$*).*
Number of queries: $q = m^{o(1)} = 2^{o(\ell)}$.

We stress again that a positive answer to Open Problem 1.9 is a *necessary* condition for using a black-box proofs to construct an extreme high-end PRG from THE EXTREME HIGH-END HARDNESS ASSUMPTION, however, it is not a *sufficient* condition.

We do not know whether black-box hard-function $\Rightarrow$ PRG proofs as in Open Problem 1.9 exist.[8] In Section 1.3 we show obstacles on certain approaches to design a black-box hard-function $\Rightarrow$ PRG

[6] One way to preserve efficiency is to require that there is an oracle machine $A^{(\cdot)}$ (in some complexity class) such that for every $f \in \mathcal{F}_{\ell,1}$, A^f implements Con_f. See e.g., [Vio05].

[7] For example, sometimes the term "non-black-box" is used to denote a deterministic simulation of randomized algorithms that is tailored for the specific input supplied to the algorithm. This notion is used for example in the recent work of Chen and Tell [CT21a] which constructs "targeted-PRGs" that are targeted to the given input, and is unrelated to the notion of "black-box" used here.

[8] In fact, the only known lower bound on black-box ρ-hard-function $\Rightarrow$ ϵ-PRG proofs by Shaltiel, Grinberg and Viola [GSV18] shows that if $a \leq 2^{\nu\cdot\ell}$ for some constant $\nu > 0$, and $\rho < 1 - 2^{-\ell}$ is sufficiently larger than $\frac{1}{2} + \epsilon$ then $q \geq \Omega(\frac{\log(1/(1-\rho))}{\epsilon^2})$. In the case of interest where $\rho = 1$ and ϵ is constant, this gives a weak bound of $q \geq \Omega(\ell)$,

proof, meeting the parameters of Open Problem 1.9. More specifically, we show that certain approaches cannot yield reductions with few queries. Before we describe our results, we give more background on black-box proofs.

1.2.3 Black-box proofs as codes, extractors, and other applications

Black-boxness is often helpful in PRG constructions (as demonstrated in [NW94, KvM02, MV05, SU05]) as such proofs readily extend to other computational models (e.g, bounded depth circuits, or nondeterministic circuits). There are other motivations to study black-box proofs (in addition to PRG constructions and hardness amplification). In fact, in the connections and applications below, the "black-boxness" of the proofs is crucial and helpful.

List-decodable codes. Following Sudan, Trevisan and Vadhan [STV01], a Black-box $\mathsf{hard\text{-}function} \Rightarrow \rho'\text{-}\mathsf{hard\text{-}function}$ proof $(\mathsf{Con}, \mathsf{Red})$ yields a "list-decodable code" $E : \{0,1\}^{2^\ell} \to \{0,1\}^{2^{\ell'}}$, defined by $E(f)_y = \mathsf{Con}_f(y)$. A consequence of lower bounds on the rate of such codes is that $\ell' \geq \ell + 2 \cdot \log(1/\rho')$, and that $a \geq 2 \cdot \log(1/\rho')$. Viewing the reduction as a "list" of 2^a procedures (one for every advice string $\alpha \in \{0,1\}^a$) yields a variant of a "local list-decoding algorithm" for the defined code, with the same number of queries. Techniques developed for black-box proofs [SV10, GSV18, AASY16] have been useful in proving lower bounds on the number of queries of such codes [RSV21].

Randomness Extractors: Following Trevisan [Tre01], a black-box $\mathsf{hard\text{-}function} \Rightarrow \epsilon\text{-}\mathsf{PRG}$ proof $(\mathsf{Con}, \mathsf{Red})$ yields a "randomness extractor" $E : \{0,1\}^{2^\ell} \times \{0,1\}^r \to \{0,1\}^m$, defined by $E(f,y) = \mathsf{Con}_f(y)$, and extracting randomness from sources with min-entropy $k = a + \log(1/\epsilon) + O(1)$. A consequence of lower bounds on extractors [RTS00] is that $r \geq \ell + 2 \cdot \log(1/\epsilon)$, and that $a \geq m + \log(1/\epsilon) - O(1)$. Continuing the analogy of the previous item, the reduction can be viewed as a local list-decoding algorithm for an "extractor-code" [TZ04]. Local list-decoding algorithms for (standard) codes, and for extractor-codes are closely related to "hard-core bits" for cryptographic primitives (see e.g. [RSV21] for a discussion).

Other applications. In recent years, black-box $\mathsf{hard\text{-}function} \Rightarrow \mathsf{PRG}$ proofs have found numerous applications in areas that are not directly related to pseudorandomness, and rely on "black-boxness". Some examples are: Learning and compression algorithms (Carmosino et al. [CIKK16] and subsequent work), worst-case to average-case reductions within NP (Hirahara [Hir18] and subsequent work), and Kolmogorov Complexity (Allender et al. [ABK+06] and subsequent work).

1.2.4 Black-box proofs and the hybrid argument

The hybrid argument cannot be used in the extreme high-end. Most known black-box $\mathsf{hard\text{-}function} \Rightarrow \mathsf{PRG}$ proofs in the literature rely on *hardness amplification* in order to use the *hybrid argument* of [Yao82, GM84]. That is, to achieve an ϵ-PRG, the construction is a sequence of two black-box proofs: $\mathsf{hard\text{-}function} \Rightarrow (\frac{1}{2} + \frac{\epsilon}{m})\text{-}\mathsf{hard\text{-}function} \Rightarrow \epsilon\text{-}\mathsf{PRG}$. Thus, even for constant ϵ,

and even this does not apply in the extreme high-end where $a = 2^{(1-o(1))\cdot\ell}$. Moreover, if we start from average-case hardness, it is open to prove that $q > 1$ for a black-box $\rho\text{-}\mathsf{hard\text{-}function} \Rightarrow \epsilon\text{-}\mathsf{PRG}$ proofs with $\rho \leq \frac{1}{2} + \epsilon$, even for small a.

hardness amplification must be performed to $\rho' \leq (\frac{1}{2} + \frac{1}{m})$. By the bounds in Section 1.2.3, in the first step, a function f_1 with input length ℓ must be transformed into a function f' with input length $\ell' \geq \ell + 2 \cdot \log \frac{1}{\rho'} \geq \ell + 2 \log m$. The final PRG construction will have seed length $r \geq \ell' \geq \ell + 2 \log m \geq 3 \log m$. This is too large in the extreme high end, where we want $r = (1 + o(1)) \cdot \log m$.

The hybrid argument suffices for the high-end. We remark that taking $\ell' = O(\ell + \log m)$ and $r = O(\ell')$ does suffice for the (non-extreme) high-end, and this is how known constructions for the high-end [IW97, STV01, SU05, Uma03] are achieved.[9]

1.3 Our Results

We show that certain approaches cannot yield a black-box hard-function $\Rightarrow$ PRG proof with the parameters of Open Problem 1.9, and therefore cannot be used to solve Open Problem 1.5. Our results are summarized in Table 1 and Table 2.

Table 1: Black-box proofs for PRGs and PEGs

Result	Type	Range	Condition	Bound
[GSV18]	hard-function $\Rightarrow$ ϵ-PRG	$a \leq 2^{\nu \cdot \ell}$		$q \geq \Omega(\frac{\ell}{\epsilon^2})$
Thm 1.11	hard-function $\Rightarrow$ ϵ-PRG, constant ϵ	$a \leq \nu \cdot 2^\ell$	$\exists j = o(\frac{2^\ell}{\ell}) : \mathrm{Fix}_j(\mathsf{Con}) > a + j \cdot \ell)$	$q \geq 2^\ell$
Thm 1.14	hard-function $\Rightarrow$ ϵ-PEG, constant ϵ	$a \leq \nu \cdot 2^\ell$	$r < \ell - \log \ell - O(1)$	$q \geq 2^\ell$
[IW97]	hard-function $\Rightarrow$ ϵ-PRG, constant ϵ	$a \leq 2^{\nu \cdot \ell}$		$q \leq m^{\Theta(1)}$

Table 2: Black-box proofs for hardness amplification

Result	Type	Range	Bound
[GSV18]	hard-function $\Rightarrow$ $(\frac{1}{2} + \epsilon)$-hard-function	$a \leq 2^{\nu \cdot \ell}$	$q \geq \Omega(\frac{\ell}{\epsilon^2})$
Thm 1.15	hard-function $\Rightarrow$ $(\frac{1}{2} + \epsilon)$-hard-function	$a \leq \nu \cdot 2^\ell$	$q \geq \Omega(\frac{1}{\epsilon})$
Thm 1.15	hard-function $\Rightarrow$ $(\frac{1}{2} + \epsilon)$-hard-function, constant ϵ	$a \leq \nu \cdot 2^\ell$	$q \geq \Omega(\ell - \log(2a))$
[IW97, STV01]	hard-function $\Rightarrow$ $(\frac{1}{2} + \epsilon)$-hard-function	$a \leq \nu \cdot 2^\ell$	$q \leq \mathrm{poly}(\frac{\ell}{\epsilon})$

In both tables above the first three lines are lower bounds, while the last line is an upper bound, and $0 < \nu \leq \frac{1}{2}$ is some constant.

1.3.1 Limitations on constructions of black-box hard-function $\Rightarrow$ PRG proofs

We show that for any black-box hard-function $\Rightarrow$ PRG proof $(\mathsf{Con}, \mathsf{Red})$, if Red makes $q \leq 2^\ell$ queries, then Con must be structured in a way that allows "fixing many outputs, with small information cost". More precisely, we introduce a measure $\mathrm{Fix}_j(\mathsf{Con})$ defined to be the minimal number h, so

[9] More specifically, hardness amplification can be performed (by a black-box proof) using "local list-decodable codes" [STV01], and the second $(\frac{1}{2} + \frac{\epsilon}{m})$-hard-function $\Rightarrow$ PRG step is done using the Nisan-Wigderson generator [NW94], which is a black-box proof. Shaltiel and Umans [SU05] and Umans [Uma03] gave an alternative direct transformation from worst-case hard function into PRGs which achieves a better seed length in the "low-end". However, as it also relies on the hybrid argument (and implies hardness amplification), it cannot achieve the extreme high-end.

that when F is chosen at random from $\mathcal{F}_{\ell,1}$, it is possible to fix j outputs of Con_F, while reducing the information about F by only h bits of information.

Definition 1.10 (The cost of fixing j outputs)**.** *Given* $\mathsf{Con} : \mathcal{F}_{\ell,1} \to \mathcal{F}_{r,m}$ *we define* $\mathrm{Fix}_j(\mathsf{Con})$ *to be the minimal number* h *such that there exist* j *distinct outputs* $z_1, \ldots, z_j \in \{0,1\}^m$ *such that:*

$$\Pr_{F \leftarrow \mathcal{F}_{\ell,1}} [\forall i \in [j] : \exists y_i \in \{0,1\}^r \text{ s.t. } \mathsf{Con}_F(y_i) = z_i] \geq 2^{-h}.$$

We show that if Red makes a small number q of queries, then for every j that is not too large, $\mathrm{Fix}_j(\mathsf{Con}) \leq a + j \cdot (\log q + O(1))$. Loosely speaking, this means that after a "fixed cost" of a bits of information, a large number of outputs of Con_F can be fixed at the cost of roughly $\log q$ bits of information about F, per m-bit output. This is stated formally below:

Theorem 1.11. *There exists a constant* $\nu > 0$ *such that for every* ρ-$\mathsf{hard\text{-}function} \Rightarrow \epsilon$-$\mathsf{PRG}$ *proof* $(\mathsf{Con}, \mathsf{Red})$ *with parameters* $\ell, r, m, a \leq \nu \cdot 2^\ell, \epsilon \leq 1 - 2^{r-m}, \rho > 0.51$*, if* Red *makes* $q \leq 2^\ell$ *queries, then for every* $j \leq \nu \cdot \frac{2^\ell}{\ell}$*,*

$$\mathrm{Fix}_j(\mathsf{Con}) \leq a + j \cdot (\log q + O(1)) \leq a + j \cdot (\ell + O(1)).$$

Previous limitations on the number of queries for reductions in black-box proofs (of any type) do not apply when $a \geq 2^{\ell/2}$ and therefore are unapplicable in the extreme high-end

We stress that Theorem 1.11 is unrelated to the "hybrid argument" and applies even for constructions where the correctness of the reduction *does not* rely on the hybrid argument. Moreover, the result applies for the whole range of parameters, and regardless of the choices of seed length and output length. See Section 3.1 for a more general statement and a discussion.

In the next section we use Theorem 1.11 to show limitations on the "PEG + extractor" approach of [DMOZ20].

Distinction between black-box PRGs and typical extractors. Following Trevisan [Tre01] (see discussion in Section 1.2.3) we know that construction maps for black-box $\mathsf{hard\text{-}function} \Rightarrow \mathsf{PRG}$ proofs are extractors (regardless of the number of queries used by the reduction). In fact, extractors and black-box proofs are essentially equivalent if we do not restrict the number of queries made by the reduction.

It is standard that if we choose a construction map Con at random, it will be an extractor. Nevertheless, we show that it is unlikely that a random construction map Con will have $\mathrm{Fix}_j(\mathsf{Con}) \leq a + j \cdot \ell$ (for relevant values of j). This implies that:

Theorem 1.12 (informal)**.** *It is unlikely that a random construction map (which is an extractor w.h.p) will have a "useful" reduction with* $q < 2^\ell$*.*

More details, and a precise statement is given in Section 3.1.2. This demonstrates that requiring a construction to have $q < 2^\ell$ and be useful for PRGs (and not just for extractors) places limitations on the structure of the construction.

Our interpretation. Our interpretation of Theorem 1.11 is that in order to enable the reduction to make few queries, the construction must "create a backdoor" and introduce correlations between different outputs. These correlations are "slightly harmful" to the goal of being an extractor. More

precisely, having low $\mathrm{Fix}_j(\mathsf{Con})$ means that there is a source distribution with min-entropy that is very high (only lacking $\mathrm{Fix}_j(\mathsf{Con})$ bits of information) on which j outputs of the extractor are fixed. This will violate the extractor guarantee if j is close to 2^r (but is possible for $j \ll 2^r$ which is the case in Theorem 1.11).

Theorems 1.11 and 1.12 suggest that it is more difficult to design PRG constructions than extractors. In Section 3.1.3 we review the known hard-function $\Rightarrow$ PRG constructions in the literature (the Nisan-Wigderson PRG [NW94] and the Shaltiel-Umans PRG [SU05, Uma03]) observing how they achieve low $\mathrm{Fix}_j(\mathsf{Con})$ in the high-end, and why they do not achieve this in the extreme high-end. We hope that understanding conditions that hard-function $\Rightarrow$ PRG constructions must satisfy, may point us to new constructions that may be applicable in the extreme high-end.

Technique. We consider a "distinguisher" $D_f : \{0,1\}^m \to \{0,1\}$ that answers one iff its input is an output of Con_f. For every function $f \in \mathcal{F}_{\ell,1}$, as D_f PRG-breaks Con_f, by Definition 1.7, there must exist $\alpha \in \{0,1\}^a$ such that the function $C(x) = \mathsf{Red}^{D_f}(x,\alpha)$ satisfies $C = f$. However, D_f only answers one on 2^r out of the possible 2^m queries. If Red does not ask such "interesting queries", then it obtains no information on f, and cannot hope to reconstruct every $f \in \mathcal{F}_{\ell,1}$.

How does Red know to ask interesting queries? The advice string α (that depends on f) may give Red information about interesting queries. However, the information in the advice string is limited by its length a, and we show that if Red is able to find interesting queries for many choices of $f \in \mathcal{F}_{\ell,1}$ and $x \in \{0,1\}^\ell$, then after this "fixed cost" of a bits of information, it is still difficult for Red to find interesting queries, unless the construction Con is set up so that many interesting queries (that is, outputs of Con_f) have low information, giving that $\mathrm{Fix}_j(\mathsf{Con}) \leq a + j \cdot (\log q + O(1))$ for many values of j). The precise details are given in Section 3.

1.3.2 Limitations on the "PEG + Extractor" approach of [DMOZ20]

Doron et al. [DMOZ20] showed how to construct extreme high end PRGs from a strengthening of THE EXTREME HIGH-END HARDNESS ASSUMPTION of Open Problem 1.5. More specifically, rather than only assuming that f cannot be computed by circuits of size $2^{(1-o(1))\cdot\ell}$, it is assumed that this holds even for circuits that are allowed to use nondeterminism and randomness (and can be thought of as a nonuniform analog of the class MA). This assumption is significantly stronger then THE EXTREME HIGH-END HARDNESS ASSUMPTION (although, still plausible).

The PEG + extractor approach. The approach of [DMOZ20] is to construct a pseudo-entropy generator (PEGs) (for a specific notion of "computational entropy" suggested in [BSW03]). This type of PEG can be thought of as a weak notion of PRGs, that is only guaranteed to fool tests that accept a very small fraction of the 2^m inputs:

Definition 1.13 (PEGs). *A function $G : \{0,1\}^r \to \{0,1\}^m$ is a (k,ϵ)-PEG for a function $D : \{0,1\}^m \to \{0,1\}$, if $\Pr[D(U_m)=1] \leq 2^{k-m}$ then $\Pr[D(G(U_r))=1] - \Pr[D(U_m)=1] \leq \epsilon$. We say that D (k,ϵ)-PEG-breaks G, if G is not an ϵ-PEG for D.*[10]

[10] We remark that the requirement that $\Pr[D(G(U_r))=1] - \Pr[D(U_m)=1] \leq \epsilon$ is sometimes replaced by the stronger requirement that $\Pr[D(G(U_r))=1] \leq \epsilon$, or following [BSW03], by the requirement that $\Pr[D(G(U_r))=1] \leq \Pr[D(U_m)=1] \cdot 2^{m-k} + \epsilon$ which is stronger still, if we replace ϵ by $\epsilon' = \epsilon/2$ and k by $k' = k + \log(1/\epsilon')$. We are interested in proving limitations on PEG and so taking a weak definition only makes our results stronger (especially as we are interested in constant ϵ and $k \ll m$ and the distinction between k, ϵ and k', ϵ' is immaterial).

More specifically, when given a function $f : \{0,1\}^\ell \to \{0,1\}$ (which is hard against the stronger model of circuits equipped with nondeterminism and randomness) the construction of [DMOZ20] works in two steps:

1. PEG: Use the hard function to construct a PEG $\mathrm{PEG} : \{0,1\}^{r_{\mathrm{PEG}}=o(\ell)} \to \{0,1\}^{m_{\mathrm{PEG}}=2^{(1-o(1))\cdot\ell}}$ for $k_{\mathrm{PEG}} = 2^{(1-o(1))\cdot\ell}$.
2. Extractor: Use an explicit extractor $\mathrm{EXT} : \{0,1\}^{m_{\mathrm{PEG}}} \times \{0,1\}^{r_{\mathrm{EXT}}=(1+o(1))\cdot\ell} \to \{0,1\}^{m=2^{(1-o(1))\cdot\ell}}$ with entropy threshold k_{PEG} (such explicit constructions are known unconditionally).

The final PRG $G : \{0,1\}^{r=r_{\mathrm{PEG}}+r_{\mathrm{EXT}}} \to \{0,1\}^m$ is obtained by interpreting a string $y \in \{0,1\}^r$ as two strings $y_1 \in \{0,1\}^{r_{\mathrm{PEG}}}$ and $y_2 \in \{0,1\}^{r_{\mathrm{EXT}}}$, and setting $G(y) = \mathrm{EXT}(\mathrm{PEG}(y_1), y_2)$.

The seed length of a PEG. The final seed length of G is $r = r_{\mathrm{PEG}} + r_{\mathrm{EXT}}$. Using lower bounds on the seed length of extractors [RTS00], it follow that $r_{\mathrm{EXT}} \geq \log(m_{\mathrm{PEG}} - k_{\mathrm{PEG}}) \geq (1 - o(1)) \cdot \ell$. Therefore, in order to achieve $r = (1 + o(1)) \cdot \ell$ (as is the case in the extreme high-end) it is crucial to use a PEG with seed length $r_{\mathrm{PEG}} = o(\ell)$. (We note that unlike PRGs, PEGs can potentially achieve $r = o(\log m)$, whereas, as noted earlier, PRGs must have $r \geq \log m - O(\log\log m)$).

Summing up, the construction of Doron et al. [DMOZ20] relies on the observation that PEGs are weaker objects than PRGs (and are therefore easier to construct) and that PEGs may have seed length that is significantly shorter than PRGs, so that summing the two seed lengths can still yield an almost optimal seed length.

Impossibility for black-box hard-function $\Rightarrow$ PEG proofs. We show an obstacle on this approach when starting from THE EXTREME HIGH-END ASSUMPTION of Open Problem 1.5. More specifically, we show that black-box hard-function $\Rightarrow$ PEG proofs with $r < \ell - \log(\ell)$ that make $q \leq 2^\ell$ queries, do not exist.

This means that the seed length of each of the two steps must be roughly ℓ and so the total seed length of a PEG + extractor must be at least

$$r_{\mathrm{PRG}} = r_{\mathrm{PEG}} + r_{\mathrm{EXT}} \geq (\ell - o(1)) + (\ell - o(1)) = 2\ell - o(1) > (2 - o(1)) \cdot \log m,$$

showing an obstacle for achieving extreme high-end PRGs with this approach. This is stated formally in the next theorem.

Theorem 1.14 (Impossibility of black-box PEGs with $r < \log m$)**.** *There exists a constant $\nu > 0$ such that for every black-box ρ-hard-function $\Rightarrow$ (k,ϵ)-PEG proof (Con, Red) with parameters $\ell, r < k, m, a \leq \nu \cdot 2^\ell, \epsilon \leq 1 - 2^{r-m}, \rho \geq 0.51$ such that Red makes $q \leq 2^\ell$ queries, it follows that:*[11]

$$r \geq \ell - \log\ell - O(1).$$

Summing up, Theorem 1.14 shows that black-box proofs cannot be used to solve Open Problem 1.5 using the PEG + extractor approach of Doron et al. [DMOZ20].[12]

[11] We have not yet formally defined the notion of ρ-hard-function $\Rightarrow$ (k,ϵ)-PEG proof. However, this definition is obtained by simply replacing "PRG-break" with "PEG-break" in Definition 1.7. For completeness, we give the full definition in Section 3.2.

[12] In light of Theorem 1.14 one may ask how Doron et al. [DMOZ20] construct their PEG. Is their proof non-black-box? The answer is that their proof is black-box, but it allows the reduction Red to use nondeterminism and randomness (and it is this ability that enables the reduction to make few queries). The cost of using these resources is that the reduction only contradicts the hardness of f if it is assumed to be hard even for circuits equipped with these resources. See e.g., Applebaum et al. [AASY16] for a discussion on nondeterministic reductions.

Consequences of Theorem 1.14 for "quantified derandomization". The notion of PEGs in Definition 1.13 is closely related to "quantified derandomization" (introduced by Goldreich and Wigderson [GW14], see survey article by Tell [Tel21]). Quantified derandomization is concerned with derandomizing algorithms that err on very few (say less than 2^k) of the possible 2^m values of their m random bits. This means that PEGs are exactly the type of PRGs that are suitable for this derandomization (see [DMOZ20, Tel21] for a discussion).

Consequently, Theorem 1.14 can also be viewed as a limitation on black-box proofs that obtain PRGs with very short seed for quantified derandomization, starting from THE EXTREME HIGH-END HARDNESS ASSUMPTION.

Technique. Theorem 1.14 follows from Theorem 1.11 (noting that Theorem 1.11 also applies to PEGs). Loosely speaking, if r is small, then the number of outputs of Con is small, and it is impossible for $\mathrm{Fix}_j(\mathsf{Con})$ to be small for large values of j, ruling out black-box proofs in which r is small.

1.3.3 Lower bounds on black-box hardness amplification at the extreme high-end

Grinberg, Shaltiel and Viola [GSV18] (continuing a line of previous work [Vio06, SV10, AS14]) proved a lower bound of $q \geq \Omega(\frac{\ell}{\epsilon^2})$ on the number of queries in reductions for black-box $\mathsf{hard\text{-}function} \Rightarrow (\frac{1}{2}+\epsilon)\text{-}\mathsf{hard\text{-}function}$ proofs (a.k.a. hardness amplification). By Proposition 1.8, such bounds imply that using black-box proofs to convert a function f on ℓ bits, that cannot be computed by size s into one that is average case hard for circuits of size m, one must have $m \leq \frac{s-a}{q} \leq \frac{s}{q}$ which means that such transformation "lose a factor q in the hardness".

In this paper we prove a lower bound of $q \geq \Omega(\frac{1}{\epsilon})$, which is quantitatively weaker than that of [GSV18], but unlike [GSV18] it applies in the extreme high-end. That is, our result allows $a = 2^{(1-o(1))\cdot\ell}$ whereas [GSV18] (as well as all previous bounds) only works if $a \leq 2^{\nu\cdot\ell}$ for some constant $\nu > 0$. (It is open to match the bound of [GSV18] for large a).

Theorem 1.15 (Lower bounds on black-box hardness amplification at the extreme high-end)**.** *Let* $(\mathsf{Con}, \mathsf{Red})$ *be a* $\mathsf{hard\text{-}function} \Rightarrow (\frac{1}{2}+\epsilon)\text{-}\mathsf{hard\text{-}function}$ *proof with parameters* $\ell, \ell', a, \rho = 1, \rho' = \frac{1}{2}+\epsilon$*. If* $\epsilon \leq \frac{1}{10}$*,* $\ell' \geq \log(1/\epsilon) + \Omega(1)$ *and* $a \leq \frac{2^\ell}{10}$ *then* Red *must make at least* q *queries for*

$$q \geq \max\left(\Omega\left(\frac{1}{\epsilon}\right), \Omega\left(\ell - \log(2a)\right)\right).$$

To the best of our knowledge, Theorem 1.15 is the first bound on the number of queries in black-box hardness amplification that applies for $a \geq 2^{\ell/2}$ and to the extreme high-end.

Using Proposition 1.8, Theorem 1.15 implies that even if one starts from THE EXTREME HIGH-END HARDNESS ASSUMPTION, then to obtain a $(\frac{1}{2}+\frac{1}{m})\text{-}\mathsf{hard\text{-}function}$ for circuits of size m (and apply the hybrid argument as explained in Section 1.2.4) there must be a "hardness loss", and $m \leq \frac{2^\ell}{q} \leq \frac{2^\ell}{m}$, implying that $m \leq 2^{\ell/2}$.

Note that this limitation applies regardless of the length ℓ' of the input length of Con_f. This means that a black-box $\mathsf{hard\text{-}function} \Rightarrow \mathsf{PRG}$ proof that relies on hardness amplification and the hybrid argument (that is: $\mathsf{hard\text{-}function} \Rightarrow (\frac{1}{2}+\frac{1}{m})\text{-}\mathsf{hard\text{-}function} \Rightarrow \mathsf{PRG}$) must have $m \leq 2^{\ell/2}$, and this holds even if the seed length r of the PRG is large.

In the next section we show that a similar argument also gives limitations on using hardness amplification together with the "PRG composition" approach of Chen and Tell [CT21b].

Technique. The work of [GSV18] (as well as previous work in this area) relied on information theoretic techniques that break down for large a. Indeed, the proof of Theorem 1.15 uses a different argument. This argument builds on ideas of Applebaum et al. [AASY16] which connect the number of queries required by a reduction (or in the case of [RSV21] a local list-decoding algorithm) to the success that small size, constant depth circuits have in solving the "coin problem" (that is distinguishing a sequence of independent tosses of an unbiased coin from a sequence of independent tosses of a slightly biased coin). The proofs of [AASY16, RSV21] do not work for $a \geq 2^{\ell/2}$ and our results are obtained by proving tighter bounds on depth 3 circuits for specific versions of the coin problem that come up in the argument. The proof is given in Section 4. As a consequence, we also improve the bounds of [RSV21] on the number of queries of local list-decoding algorithms, see Section 5 for details.

1.3.4 The hybrid argument and the "PRG composition" approach of [CT21b]

Chen and Tell [CT21b] construct extreme high end PRGs if, in addition to THE EXTREME HIGH-END HARDNESS ASSUMPTION of Open Problem 1.5, one also assumes the existence of one-way functions (OWFs). The existence of OWFs is a standard and widely believed assumption in cryptography. Nevertheless, OWFs (or more generally cryptography) are not known to be implied by extreme high-end PRGs (or other PRGs in complexity theory). Assuming OWFs does not seem necessary.

The PRG composition approach. Chen and Tell [CT21b] start from a hard function $f : \{0,1\}^{\ell} \to \{0,1\}$ given by THE EXTREME HIGH-END HARDNESS ASSUMPTION. Their PRG $G : \{0,1\}^{(1+o(1))\cdot\ell} \to \{0,1\}^{2^{(1-o(1))\cdot\ell}}$ is obtained by PRG composition, namely $G(y) = G_2(G_1(y))$ where:

1. $G_1 : \{0,1\}^{(1+o(1))\cdot\ell} \to \{0,1\}^{m_1=2^{\Omega(\ell)}}$ is a PRG against circuits of size $2^{(1-o(1))\cdot\ell}$ that is constructed from THE EXTREME HIGH-END HARDNESS ASSUMPTION using hardness amplification, the Nisan-Wigderson PRG, and the hybrid argument.[13]
2. $G_2 : \{0,1\}^{m_1=2^{\Omega(\ell)}} \to \{0,1\}^{m=2^{(1-o(1))\cdot\ell}}$ is a PRG with modest stretch (polynomial rather than exponential) that suffices to push the output length from $m_1 = 2^{\Omega(\ell)}$ to $m = 2^{(1-o(1))\cdot\ell}$. Nevertheless, for the composition to be a PRG, it is crucial that G_2 can be computed in time $2^{(1-o(1))\cdot\ell}$ (that is in almost linear time in its output length m).[14] Such PRGs indeed follow from the existence of OWFs [HILL99].

A natural question is whether it is possible to construct the PRG G_2 from THE EXTREME HIGH-END HARDNESS ASSUMPTION. This will remove the need for OWFs.

PRGs with polynomial stretch follow from this assumption (and even from weaker versions like THE HIGH-END HARDNESS ASSUMPTION or "low-end" versions). This is good news, as it shows that hardness amplification and the hybrid argument *can* yield sufficient stretch in this case.

[13] More precisely, the cost of the hybrid argument (explained in Section 1.2.4) is measured in terms of the output length m (even if the PRG fools circuits of larger size, as is the case here). This means, that the goal of fooling circuits of size $2^{(1-o(1))\cdot\ell}$ can be achieved by known black-box proofs (in the same manner explained in Section 1.2.4) from the EXTREME HIGH-END HARDNESS ASSUMPTION for PRGs that output $m_1 = 2^{\Omega(\ell)}$ bits, rather than $m = 2^{(1-o(1))\cdot\ell}$ bits.

[14] This requirement is necessary as in the composition one needs to consider a distinguisher for G_1 that runs G_2 as a procedure, and G_1 cannot fool circuits of size larger than 2^{ℓ}. We also remark that in contrast to cryptography, in hardness vs. randomness, efficiency of components is rarely used in proving security of the final primitive, and this is one such rare instance.

The issue is that the PRGs constructed by these methods do not run in time that is nearly linear in their output length m (and instead run in time m^c where $c > 2$). This means that they are unsuitable for the PRG composition approach (and this is why [CT21b] relies on OWFs).

Our results. Theorem 1.15 implies that PRGs (even with modest stretch) cannot run in nearly linear time, if they are obtained using black-box hardness amplification and the hybrid argument, assuming THE EXTREME HIGH-END HARDNESS ASSUMPTION. This implies that if we only assume THE EXTREME HIGH-END HARDNESS ASSUMPTION (and do not assume the existence of OWFs) then current techniques cannot yield the PRG G_2 required by the PRG composition.

More precisely, we have already seen in Section 1.3.3 that in order to do a hybrid argument for output length m, one needs a hardness amplification result that amplifies to below $\frac{1}{2} + \frac{1}{m}$, and that Theorem 1.15 implies that $m \leq 2^{\frac{\ell}{2}}$. This holds in the extreme high-end, regardless of the relationship between r and m. In particular, it also holds when trying to construct PRGs with modest stretch like G_2, assuming THE EXTREME HIGH-END HARDNESS ASSUMPTION. On the other hand, assuming that computing the average-case hard function Con_f takes at least the time it takes to compute the worst-case hard function f, and recalling that f cannot be computed by circuits of size $2^{(1-o(1))\cdot\ell}$, we conclude that Con_f cannot be computed in time $2^{(1-o(1))\cdot\ell}$ (which is at least $m^{2-o(1)}$).

Summing up, after performing hardness amplification, there must be at least a quadratic gap between the time it takes to compute Con_f, and the circuit size for which it is hard on average. This gap is inherited by the final PRG G_2. Consequently, G_2 cannot run in time smaller than $m^{2-o(1)}$, and in particular, there is in obstacle for obtaining PRGs that run in time nearly linear in m, using these techniques (even if the stretch is modest).

This shows an obstacle for using current techniques (that rely on hardness amplification and the hybrid argument) to apply the PRG composition approach of [CT21b] assuming only THE EXTREME HIGH-END HARDNESS ASSUMPTION. This partially explains why [CT21b] need the additional assumption that OWFs exist in order to construct the PRG G_2.

1.4 Organization of this paper

In Section 2 we define some notation, and cite some previous work that we use. In Section 3 we prove our results on black-box proofs for PRGs and PEGs (Theorem 1.11 and Theorem 1.14). In Section 4 we prove our results on hardness amplification (Theorem 1.15). In Section 5 we use the methodology devised for Theorem 1.15 to improve the lower bounds of Ron-Zewi, Shaltiel and Varma [RSV21] on the number of queries of decoders for locally decodable codes. In Section 6 we mention some open problems.

2 Preliminaries

Distributions and Random Variables. We use $X \leftarrow D$ to denote the experiment in which X is chosen from distribution D. For a set A we use $X \leftarrow A$ to denote the experiment in which X is chosen uniformly from A. Two distributions X, Y over the same finite domain are ϵ-close if for every event A, $|\Pr[X \in A] - \Pr[Y \in A]| \leq \epsilon$. For a distribution X over $\{0,1\}^n$, we define $H_\infty(X) = \min_{x \in \{0,1\}^n} \log \frac{1}{\Pr[X=x]}$.

For $0 \leq p \leq 1$, we use U_n^p to denote the distribution of n i.i.d. random variables, where each one has probability p to be one. We use U_n to denote $U_n^{1/2}$ (the uniform distributions on n bit strings).

Hamming distance and weight. For two strings $x, y \in \{0,1\}^n$ we use $\mathsf{dist}(x,y)$ to denote the relative Hamming distance between x and y, namely, $\mathsf{dist}(x,y) = |\{i \in [n] : x_i \neq y_i\}|/n$. We use $\mathrm{weight}(x)$ to denote the absolute Hamming weight, namely $\mathrm{weight}(x) = |\{i \in [n] : x_i \neq y_i\}|$.

Restrictions. We use the standard notion of random restrictions. Namely, a restriction $\rho : [n] \to \{0,1,*\}$ restricts some of the variables of a function $f : \{0,1\}^n \to \{0,1\}$.

We use $\mathrm{Select}(\rho)$ to denote the subset of free variables, $\mathrm{Free}(\rho)$ to denote the number of free variables, and $\mathrm{Assign}(\rho)$ to be the value of the assigned variables. This is stated formally in the next definition.

Definition 2.1 (Restrictions)**.** *A restriction of n variables is a function $\rho : [n] \to \{0,1,*\}$. We define $\mathrm{Select}(\rho) = \{i : \rho(i) = *\}$, $\mathrm{Free}(\rho) = |\mathrm{Select}(\rho)|$, and $\mathrm{Assign}(\rho) \in \{0,1\}^{n-\mathrm{Free}(\rho)}$ by enumerating the fixed elements $(i_1 < \ldots < i_{n-\mathrm{Free}(\rho)}) = [n] \setminus \mathrm{Select}(\rho)$ and defining $\mathrm{Assign}(\rho)_j = \rho(i_j)$.*

Given a function $f : \{0,1\}^n \to \{0,1\}$ and a restriction $\rho : [n] \to \{0,1,*\}$, and $x \in \{0,1\}^{\mathrm{Free}(\rho)}$ we define $\mathrm{Fill}_\rho(x) \in \{0,1\}^n$ as follows: Let $i_1 < \ldots < i_{\mathrm{Free}(\rho)}$ be the indices on which ρ outputs '$*$',

$$\mathrm{Fill}_\rho(x)_i = \begin{cases} x_j, & \exists j \text{ s.t. } i = i_j \\ \rho(i), & \text{otherwise} \end{cases}$$

Given $f : \{0,1\}^n \to \{0,1\}$, we define $f_\rho : \{0,1\}^{\mathrm{Free}(\rho)} \to \{0,1\}$ by

$$f_\rho(x) = f(\mathrm{Fill}_\rho(x)).$$

We use R_p^n to denote the set of restrictions with $p \cdot n$ unrestricted variables. A formal definition is given below.

Definition 2.2 (The class R_p^n)**.** *For $0 \leq p \leq 1$, let R_p^n denote the set of restrictions $\rho : [n] \to \{0,1,*\}$ with $\mathrm{Free}(\rho) = p \cdot n$.*

We use the following switching lemma due to Hastad [Hås86]. The specific version used here, appears in Beame's primer [Bea94].

Theorem 2.3. *[Hås86, Bea94] Let C be a q-CNF over n variables. For every $0 \leq p \leq 1$, the probability over choosing $\rho \leftarrow \mathsf{R}_p^n$ that C_ρ does not have a decision tree of height h is at most $(7 \cdot p \cdot q)^h$.*

3 Limitations on black-box proofs for PRGs and PEGs

In this section we discuss our main results on black-box proofs for PRGs and PEGs (Theorem 1.11 and Theorem 1.14). In Section 3.1 we discuss Theorem 1.11, and in Section 3.2 we discuss Theorem 1.14. The proofs of both these Theorems is given in Section 3.3.

3.1 Limitations on constructions of black-box hard-function ⇒ PRG proofs

In this section we restate Theorem 1.11 and discuss its consequences and interpretation. In Section 3.1.1 we restate Theorem 1.11 and discuss its interpretation. In Section 3.1.2 we show that a corollary of Theorem 1.11 is that PRG constructions must differ from typical extractor constructions, giving a precise statement of Theorem 1.12. In Section 3.1.3 revisit the Nisan-Wigderson PRG [NW94] and the Shaltiel-Umans PRG [SU05, Uma03] and discuss them from the perspective of Theorem 1.11.

3.1.1 A general statement of Theorem 1.11

Review of the setup of Theorem 1.11. Theorem 1.11 shows that for any black-box hard-function ⇒ PRG proof $(\mathsf{Con}, \mathsf{Red})$, if Red is useful, and makes $q \leq 2^\ell$ queries, then Con must be structured in a way that allows "fixing many outputs, with small information cost". This is measured by $\mathrm{Fix}_j(\mathsf{Con})$ from Definition 1.10, which defines $\mathrm{Fix}_j(\mathsf{Con})$ to be the minimal number h, so that when F is chosen at random from $\mathcal{F}_{\ell,1}$, it is possible to fix j outputs of Con_F, while only reducing h bits of information about F.

Theorem 1.11 shows that in order for Red to be useful and have $q \leq 2^\ell$, it must be that for many sufficiently large choices of j, $\mathrm{Fix}_j(\mathsf{Con}) \leq a + j \cdot (\log q + O(1))$. This means that after a "fixed cost" of a bits of information, a large number of outputs of Con_F can be fixed at the cost of $\log q + O(1)$ bits of information about F, per output.

A more general statement. The following theorem is a generalized version of Theorem 1.11 which also allows the parameter ρ (measuring how hard on average is the function we start from) to be very close to $\frac{1}{2}$, and then the amortized cost $\log q + O(1)$, is replaced by $\log \frac{q}{\rho - \frac{1}{2}}$.

Theorem 3.1. *Let* $(\mathsf{Con}, \mathsf{Red})$ *be a black-box* ρ-hard-function ⇒ ϵ-PRG *proof for parameters* $\ell, r, m, a, \epsilon, \rho$ *such that* Red *makes at most* $q \leq 2^\ell$ *queries. If* $\rho = \frac{1}{2} + \eta$, $\eta \geq 2^{-\ell}$, $\epsilon \leq 1 - 2^{r-m}$, *and* $a \leq \nu \cdot \eta^2 \cdot 2^\ell$ *for some sufficiently small constant* $\nu > 0$, *then for* $j_{\max} = \nu \cdot \frac{\eta^2 \cdot 2^\ell}{\ell}$, *and every* $j \leq j_{\max}$,

$$\mathrm{Fix}_j(\mathsf{Con}) \leq a + j \cdot (\log q + \log \frac{4}{\eta}).$$

Theorem 1.11 immediately follows from Theorem 3.1 by setting $\rho = 0.51$. Theorem 3.1 is proven in Section 3.3.

A discussion of the parameters of Theorem 3.1. The parameters in Theorems 1.11 and Theorem 3.1 are applicable even for very large a, and therefore Theorem 3.1 applies in the extreme high-end (where $a = 2^{(1-o(1))\cdot\ell}$) as well as in less challenging ranges such as the high-end (where $a = 2^{\nu\cdot\ell}$ for a constant $\nu > 0$) and the low-end (where $a = \mathrm{poly}(\ell)$). We stress that to the best of our knowledge, previous limitations on the number of queries of a black-box reduction (of any kind) did not apply for $a \geq 2^{\ell/2}$ and in particular, for the extreme high-end. See Section 4 for a discussion on past lower bounds on the number of queries by reductions for black-box hardness amplification proofs.

Furthermore, Theorems 1.11 and Theorem 3.1 make no assumption on the stretch (namely, the relationship between r and m) of the constructed PRG, and apply even when the stretch is very

small, and m is only slightly larger than r. In addition, the theorem applies even when ϵ is very large, and approaches one (rather than zero).

Finally, Theorem 3.1 applies even when $\eta = \frac{1}{2} - \rho$ is very small (say $\eta = 2^{-\Omega(\ell)}$) and even for black-box hard-function $\Rightarrow$ PRG proofs like the Nisan-Wigderson PRG, which are only known to work when η is very small. See Section 3.1.3 for a discussion of the Nisan-Wigderson PRG.

3.1.2 black-box PRGs are different than typical extractors

In this section we prove Theorem 1.12 which loosely states that a random construction Con for a black-box hard-function $\Rightarrow$ PRG proof is likely to be an extractor, but unlikely to be useful when transforming hard functions into PRGs.

Formal connection between extractors and black-box hard-function $\Rightarrow$ PRG proofs. It is well known following Trevisan's breakthrough construction [Tre01] (and as explained in Section 1.2.3) that black-box hard-function $\Rightarrow$ PRG proofs, are closely related to randomness extractors. Let us formally specify this connection. We start with a formal definition of randomness extractors.

Definition 3.2 (extractors)**.** *A function $E : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}^m$ is a (k,ϵ)-extractor if for every distribution X over $\{0,1\}^n$, with $H_\infty(X) \geq k$, $E(X, U_r)$ is ϵ-close to U_m.*

In order to compare function $E : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}^m$ to functions $\mathsf{Con} : \mathcal{F}_{\ell,1} \to \mathcal{F}_{r,m}$ we use the following notation.

Definition 3.3 (Constructions and extractors)**.** *A string $f \in \{0,1\}^n$ can be viewed as a function $f : \{0,1\}^{\log n} \to \{0,1\}$ by $f(i) = f_i$. This also applies in the other direction, allowing a function $f : \{0,1\}^\ell \to \{0,1\}$ to be viewed as a string $f \in \{0,1\}^{2^\ell}$.*

Given a function $E : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}^m$, we define $\ell = \log n$, and $\mathsf{Con}^E : \mathcal{F}_{\ell,1} \to \mathcal{F}_{r,m}$ where $\mathsf{Con}^E_f : \{0,1\}^r \to \{0,1\}^m$ is defined by $\mathsf{Con}^E_f(y) = E(f,y)$. This also applies in the other direction, where a function $\mathsf{Con} : \mathcal{F}_{\ell,1} \to \mathcal{F}_{r,m}$ induces a function $E : \{0,1\}^{n=2^\ell} \times \{0,1\}^r \to \{0,1\}^m$ by $E(f,y) = \mathsf{Con}_f(y)$.

With this notation, extractors and black-box hard-function $\Rightarrow$ PRG proofs, are essentially equivalent. This is stated in the following standard proposition which shows that the two notions are roughly equivalent for $a \approx k$.

Proposition 3.4 (Extractors are essentially equivalent to black-box hard-function $\Rightarrow$ PRG proofs)**.** *Let $E : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}^m$, and let $\ell = \log n$.*

1. *If Con^E can be matched with an oracle procedure $\mathsf{Red}^{(\cdot)}$ such that the pair $(\mathsf{Con}^E, \mathsf{Red})$ is a black-box hard-function $\Rightarrow$ ϵ-PRG proof with advice length a, then E is a $(k, 2\epsilon)$-extractor, for $k = a + \log(1/\epsilon) + 1$.*

2. *If E is a (k,ϵ)-extractor then there exists an oracle procedure $\mathsf{Red}^{(\cdot)}$ such that the pair $(\mathsf{Con}^E, \mathsf{Red})$ is a black-box hard-function $\Rightarrow$ ϵ-PRG proof with advice length $a = k + 1$.*

Reductions must make few queries to be useful. Note however, that in order to be useful for PRG construction, a black-box hard-function $\Rightarrow$ PRG proof must have a reduction Red that make few queries. In particular, as can be seen in Proposition 1.8, the reduction is useless if it makes $q \geq 2^\ell$ queries (and even if it makes $q \geq \frac{2^\ell - a}{m}$ queries).

The equivalence of Proposition 3.4 does not mention the number of queries (which can be as large as 2^m). In other words, a black-box hard-function $\Rightarrow$ PRG proof is an extractor even when the number of queries is very large, but it must have $q \ll 2^\ell$ in order to be useful when converting hard functions into PRGs.

A typical extractor is not a useful PRG construction. We now show that for a wide range of parameters ℓ, r, m and a (and in particular for the parameters that correspond to the high-end and the extreme high-end) if we choose a function $E : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}^m$ uniformly at random then:

- It is unlikely that construction Con^E (associated with E) has small $\mathrm{Fix}_j(\mathsf{Con}^E)$, and therefore, by Theorem 1.11, it is unlikely that Con^E can be matched with a reduction Red that makes $q \leq 2^\ell$ queries.
- However, a standard argument shows that it is likely that E is an extractor, and therefore, by Proposition 3.4, it is likely that Con^E can be matched with a reduction Red (although Red may make $q = 2^m$ queries).

As explained in Section 1.3.1, this can be interpreted as saying that PRG constructions must be quite different than typical extractor constructions in order to allow the existence of a reduction that makes few queries.

A formal and quantitative version of this is stated in the next definition and theorem.

Definition 3.5 (Random construction)**.** *Let $E \leftarrow \mathsf{RndE}_{\ell,r,m}$ denote the experiment in which a function $E : \{0,1\}^{n=2^\ell} \times \{0,1\}^r \to \{0,1\}^m$ is chosen uniformly from the set of all such functions.*

Theorem 3.6. *Let ℓ, r, m, a, ϵ be parameters such that $m \geq 2r$ and the following holds:*

Parameters that allow E to be an extractor: $r \geq \ell + 2\log(1/\epsilon) + O(1)$, $2^\ell \geq a \geq m + 2\log(1/\epsilon) + O(1)$, *and* $\epsilon \leq \frac{1}{2}$.

Parameters that prevent fixing seeds: $a + j \cdot \ell \leq min(j \cdot (m - r) - O(1), \frac{2^\ell}{2})$.

In the experiment $E \leftarrow \mathsf{RndE}_{\ell,r,m}$ the following items hold:

- *For every $j > 4$, the probability that $\mathrm{Fix}_j(\mathsf{Con}^E) \leq a + j \cdot \ell$ is smaller than $2^{-2^{2^\ell/2}}$.*
  Consequently (by Theorem 1.11) the probability that Con^E can be matched with an oracle procedure $\mathsf{Red}^{(\cdot)}$ that makes $q < 2^\ell$ queries, is smaller than $2^{-2^{2^\ell/2}}$.
- *The probability that E is an $(a + O(1), \epsilon)$ extractor is larger than $1 - o(1)$. Consequently (by Proposition 3.4) the probability that Con^E can be matched with an oracle procedure $\mathsf{Red}^{(\cdot)}$ with advice length $a + O(1)$ is larger than $1 - o(1)$.*

The parameters in Theorem 3.6. We stress that Theorem 3.6 applies for a wide range of parameters including the extreme high-end, and the high-end. Furthermore, Theorem 3.6 does not require that m is much larger than r and applies even when the stretch is small. Furthermore, the probability in the first item is $2^{-2^{2^\ell}/2}$ which is quite small. We also note that the requirement that $a + j \cdot \ell$ is small are very mild in the following sense:

- For every j it is obvious that $\mathrm{Fix}_j(\mathsf{Con}) \leq j \cdot m$, as the amount of information in j outputs is at most $j \cdot m$. We are only requiring that $a + j \cdot \ell \leq j \cdot (m - r) - O(1)$ which is quite close to the obvious bound.
- The total amount of information in f is at most 2^ℓ. We are only requiring that $a + j \cdot \ell \leq \frac{2^\ell}{2}$ which is quite close to the obvious bound.

Proof. (of Theorem 3.6) The second item is a standard calculation which can be found for example in [RTS00]. Therefore, we will only prove the first item. Within this proof, unless otherwise specified, all probabilities are regarding the experiment $E \leftarrow \mathsf{RndE}_{\ell,r,m}$.

For a fixed set $S \subseteq \{0,1\}^n$ of size $K = 2^{n-(a+j\cdot\ell)}$ and a sequence $\bar{z} = (z_1, \ldots, z_j)$ of distinct elements in $\{0,1\}^m$, let $A_{S,\bar{z}}$ denote the event that for every $f \in S$, there exist $y_1, \ldots, y_j \in \{0,1\}^r$ such that for all $i \in [j]$, $E(f, y_i) = z_i$. This definition is made so that:

$$\Pr[\mathrm{Fix}_j(\mathsf{Con}^E) \leq a + j \cdot \ell] \leq \binom{2^n}{K} \cdot 2^{m \cdot j} \cdot \Pr[A_{S,\bar{z}}],$$

where the latter expression is obtained by a union bound over all choices of S and $\bar{z}$. For every such S and $\bar{z}$, and for every $f \in S$, let $A_{S,\bar{z},f}$ denote the event that there exist $y_1, \ldots, y_j \in \{0,1\}^r$ such that for all $i \in [j]$, $E(f, y_i) = z_i$. For every $f \in S$, by a union bound over all choices of $y_1, \ldots, y_j \in \{0,1\}^r$,

$$\Pr[A_{S,\bar{z},f}] \leq 2^{r \cdot j} \cdot 2^{-m \cdot j}.$$

These events are independent for the K different choices of $f \in S$, and therefore, the probability of their conjunction is the product of their individual probabilities.

$$\Pr[A_{S,\bar{z}}] \leq \left(2^{r \cdot j} \cdot 2^{-m \cdot j}\right)^K = 2^{-j \cdot K \cdot (m-r)}.$$

We conclude that:

$$\begin{aligned}\Pr[\mathrm{Fix}_j(\mathsf{Con}^E) \leq a + j \cdot \ell] &\leq \binom{2^n}{K} \cdot 2^{m \cdot j} \cdot 2^{-j \cdot K \cdot (m-r)} \\ &\leq \left(\frac{2^n \cdot e}{K}\right)^K \cdot 2^{m \cdot j} \cdot 2^{-j \cdot K \cdot (m-r)}\end{aligned}$$

Let $v = 2^{2^\ell/2}$. In order to show that this probability is smaller than 2^{-v} it is sufficient to show that:

1. $\left(\frac{2^n \cdot e}{K}\right)^K \cdot 2^{-j \cdot K \cdot (m-r)} \leq 2^{-2v}$, and
2. $2^{m \cdot j} \leq 2^v$.

These two conditions follow by the requirements of the theorem. More specifically, the requirement that $a + j \cdot \ell \leq \frac{2^\ell}{2}$ gives that $K \geq 2^{2^\ell/2} \geq v$. This gives that the first condition follows if $a + j \cdot \ell \leq j \cdot (m - r) - O(1)$ which is one of the requirements of the theorem.

The second condition follows because by our requirement $j \leq \frac{2^\ell}{2}$ and $m \leq 2^\ell$. Therefore for sufficiently large ℓ, $m \cdot j \leq 2^{2^\ell/2} = v$, □

3.1.3 Revisiting the Nisan-Wigderson PRG and the Shaltiel-Umans PRG

As surveyed in Section 1.2.4 there are only two known constructions of black-box $\mathsf{hard\text{-}function} \Rightarrow \mathsf{PRG}$ proofs in the literature, where Red is useful and makes q queries, for q that can be significantly less than 2^ℓ. These are the Nisan-Wigderson PRG [NW94] and an additional construction is due to Shaltiel and Umans [SU05] and Umans [Uma03].

The former is more versatile, and has many additional applications, mainly because its reduction requires less computational resources, and makes less queries. The latter has advantages over the Nisan-Wigderson PRG as it allows to achieve $r = O(\ell)$ even for small values of a (like $a = \mathrm{poly}(\ell)$, which corresponds to "low-end" PRGs).

Nevertheless, as explained in Section 1.2.4, both these approaches rely on hardness amplification and the hybrid argument, preventing them from achieving the extreme high-end.

Theorem 1.11 and 3.1 show that in *any* black-box proof with $q \leq 2^\ell$, *even one that does not rely on the hybrid argument*, $\mathrm{Fix}_j(\mathsf{Con})$ must be small. Let us review how the known PRGs achieve this.

The Nisan-Wigderson PRG. The Nisan-Wigderson PRG is a black-box $(\frac{1}{2} + \frac{\epsilon}{m})$-$\mathsf{hard\text{-}function} \Rightarrow \epsilon$-$\mathsf{PRG}$ proof $(\mathsf{Con}^{NW}, \mathsf{Red}^{NW})$ with parameters $\ell, r, m, a, \rho = \frac{1}{2} + \frac{\epsilon}{m}$ and ϵ, such that Red makes a *single* query. This means that the Nisan-Wigderson PRG is applicable in the extreme high-end, when starting from a $(\frac{1}{2} + \frac{\epsilon}{m})$-$\mathsf{hard\text{-}function}$. As explained in Section 1.3.3 current hardness amplification techniques are not applicable for $m \geq 2^{\ell/2}$. This means that they cannot be used in the extreme high-end, even if we are willing to allow large seed length r.

By Theorem 3.1, we have that:

$$\mathrm{Fix}_j(\mathsf{Con}^{NW}) \leq a + j \cdot (\log m + O(1)),$$

for constant $\epsilon > 0$, and many values of m and j. Let us review how Con^{NW} achieves this.

The construction Con^{NW} is defined using a "design", namely a collection $S_1, \ldots, S_m \subseteq [r]$ such that for every $i \in [m]$, $|S_i| = \ell$, and then for every $f \in \mathcal{F}_{\ell,1}$, $\mathsf{Con}^{NW}_f : \{0,1\}^r \to \{0,1\}^m$ is defined by

$$\mathsf{Con}^{NW}_f(x) = f(x|_{S_1}), \ldots, f(x|_{S_m}).$$

Another version that is often considered is a "seed extending" version $\overline{\mathsf{Con}}^{NW}_f(x) = x, \mathsf{Con}^{NW}_f(x)$ that outputs $m + r$ bits. The analysis of [NW94] shows that the parameter a is determined by the sizes of pairwise intersections of sets in the design. More specifically, using an improved analysis by Raz, Reingold and Vadhan [RRV02] (for a condition called "weak design") it follows that taking

$$a = \sum_{i \in [m]} \sum_{i' \neq i} 2^{|S_{i'} \cap S_i|} \leq m \cdot 2^{\max_{i \neq i'} |S_i \cap S_{i'}|},$$

suffices to obtain a reduction Red^{NW} which makes a *single* query.

The weak design property can be used to show that for every $1 \leq j \leq 2^\ell$,

$$\mathrm{Fix}_j(\overline{\mathsf{Con}}^{NW}) \leq a + j.$$

Loosely speaking, this is because the NW proof shows that for every i, after fixing a bits of information about f, we have that for every value of $x|_{[r] \setminus S_i}$, all $m - 1$ output bits of the form

$f(x|_{S_{i'}})$ for $i' \neq i$ are completely determined as a function of $x|_{S_i}$. This means that for every value of $x|_{S_i}$, the only output bit that is not yet fixed is $f(x|_{S_i})$, which can be fixed paying only *one bit of information.*

If we were to beat the hybrid argument, and construct a new black-box $\mathsf{hard\text{-}function} \Rightarrow \mathsf{PRG}$ proof that works directly from worst-case hard functions, we would have to come close to this behavior, and we hope that understanding this property may point us to new constructions, potentially utilizing the ability to ask a small number of queries (but more than just one query).

The Shaltiel-Umans PRG. The Shaltiel-Umans PRG is a black-box $\mathsf{hard\text{-}function} \Rightarrow \mathsf{PRG}$ proof $(\mathsf{Con}^{SU}, \mathsf{Red}^{SU})$ with parameters $\ell, r, m, a, \rho = 1$ and $\epsilon = \frac{1}{10}$, such that Red makes $q = m^{O(1)}$ queries.

Unlike the Nisan-Wigderson PRG, the Shaltiel-Umans PRG is stated for $\rho = 1$ (starting from worst-case hard functions). Nevertheless, the Shaltiel-Umans PRG builds on the hardness amplification techniques of Sudan, Trevisan and Vadhan [STV01] and makes $q \geq m$ queries. Furthermore, it implies hardness amplification to $\frac{1}{2} + \frac{1}{m}$. This means that it is not applicable at the extreme high-end.

By Theorem 3.1, we have that:

$$\mathrm{Fix}_j(\mathsf{Con}^{SU}) \leq a + j \cdot (O(\log m)),$$

for many values of m and j.

Let us review how Con^{SU} achieves this. The construction Con_f^{SU} relies on (a carefully chosen) black-box $\mathsf{hard\text{-}function} \Rightarrow (\frac{1}{2} + \frac{1}{m})\text{-}\mathsf{hard\text{-}function}$ proof (which has additional structure). This hardness amplification construction converts the worst-case hard function $f : \{0,1\}^\ell \to \{0,1\}$ into an average case hard function $f' : \{0,1\}^{\ell'} \to \{0,1\}$.

The PRG construction is then defined by setting $r = \ell'$ and defining:

$$\mathsf{Con}_f^{SU}(x) = f'(x), f'(g(x)), f'(g(g(x))), \ldots, f'(g^{(m-1)}(x),$$

where $g : \{0,1\}^{\ell'} \to \{0,1\}^{\ell'}$ is some specific function. Another version that is often considered is a "seed extending" version $\overline{\mathsf{Con}}_f^{SU}(x) = x, \mathsf{Con}_f^{SU}(x)$ that outputs $m + r$ bits.

This means that for every seed $x \in \{0,1\}^{\ell'}$ if we consider the ℓ "consecutive" seeds,

$$g(x), g(g(x)), \ldots, g^{(m-1)}(x),$$

all the outputs of Con_f^{SU} on these seeds, depend only on the values of f' on $g(x), g(g(x)), \ldots, g^{(2m)}(x)$. This means that m outputs of Con_f^{SU} can be fixed at the cost of fixing $2m$ bits of f'. In particular, for $j = a$ we get that $j = a$ outputs of Con_f can be fixed at the total cost of $2a = a + j \cdot 1$ bits of information about f. This means that after paying a fixed cost of a, a outputs can be fixed at amortized cost of one bit of information per output. This can be used to show that:

$$\mathrm{Fix}_a(\overline{\mathsf{Con}}^{SU}) \leq a + a \cdot 1.$$

Again, if we were to beat the hybrid argument, and construct a new black-box $\mathsf{hard\text{-}function} \Rightarrow \mathsf{PRG}$ proof that works directly from worst-case hard functions, we may hope to take this understanding, hoping to somehow avoid paying the price of hardness amplification to extremely hard on average functions.

3.2 Limitations on the "PEG + Extractor" approach of [DMOZ20]

In this section we restate Theorem 1.14 and discuss its consequences and interpretation.

3.2.1 Limitation on black-box hard-function ⇒ PEG constructions with short seed

Review of the setup of Theorem 1.14. Theorem 1.11 shows that for any black-box hard-function ⇒ PEG proof $(\mathsf{Con}, \mathsf{Red})$, if Red is useful, and makes $q \leq 2^\ell$ queries, then $r \geq (1 - o(1)) \cdot \ell \geq (1 - o(1)) \cdot \log m$. As explained in Section 1.3.2, this shows that the "PEG + extractor" approach of [DMOZ20] cannot be used to obtain an extreme high-end PRG from THE EXTREME HIGH-END HARDNESS ASSUMPTION if the PEG is constructed by a black-box hard-function ⇒ PEG proof.

Formal definition of black-box hard-function ⇒ PEG proofs. In the introduction, we did not give a formal definition of black-box hard-function ⇒ PEG proofs (and only explained how these are defined in relation to black-box hard-function ⇒ PRG proofs. We therefore start with a formal definition.

Definition 3.7 (Black-box hard-function ⇒ PEG proof)**.** *Given parameters $\ell, r, m, a, k, \epsilon, \rho$ a black-box ρ-hard-function ⇒ (k, ϵ)-PEG proof is a pair $(\mathsf{Con}, \mathsf{Red})$ of:*

- *A construction map $\mathsf{Con} : \mathcal{F}_{\ell,1} \to \mathcal{F}_{r,m}$.*
- *An oracle procedure $\mathsf{Red}^{(\cdot)}(x, \alpha)$ such that for every $f \in \mathcal{F}_{\ell,1}$ and for every $D \in \mathcal{F}_{m,1}$ such that D (k, ϵ)-PEG-breaks Con_f, there exists $\alpha \in \{0,1\}^a$ such that the function $C \in \mathcal{F}_{\ell,1}$ defined by $C(x) = \mathsf{Red}^D(x, \alpha)$, ρ-hard-function-breaks f.*

If we omit ρ, we mean $\rho = 1$. If we omit ϵ, we mean $\epsilon = 1/10$. We say that Red makes q queries, if for every $D \in \mathcal{F}_{m,1}$, $\alpha \in \{0,1\}^a$, and $x \in \{0,1\}^\ell$, $\mathsf{Red}^D(x, \alpha)$ makes at most q oracle queries.

A more general statement. The following Theorem is a generalized version of Theorem 1.14 which also allows the parameter ρ (measuring how hard on average is the function we start from) to be very close to $\frac{1}{2}$.

Theorem 3.8. *There exists constants $\nu > 0$ and $c > 1$ such that for every black-box ρ-hard-function ⇒ (k, ϵ)-PEG proof $(\mathsf{Con}, \mathsf{Red})$ with parameters $\ell, r, m, a, k, \epsilon, \rho$ such that*

$$r < \ell - \log \ell - 2 \cdot \log \frac{1}{\eta} - c.$$

If $\rho = \frac{1}{2} + \eta$, $\eta \geq 2^{-\ell}$, $k > r$, $\epsilon \leq 1 - 2^{r-m}$, and $a \leq \nu \cdot \eta^2 \cdot 2^\ell$, then Red must make at least $q > 2^\ell$ queries.

A discussion of the parameters of Theorem 3.8. The parameters in Theorems 1.14 and Theorem 3.8 are applicable even for very large a, and therefore Theorem 3.1 applies in the extreme high-end (where $a = 2^{(1-o(1))\cdot\ell}$) as well as in less challenging ranges such as the high-end (where $a = 2^{\nu\cdot\ell}$ for a constant $\nu > 0$) and the low-end (where $a = \mathrm{poly}(\ell)$). We stress once again that to the best of our knowledge, no known lower bound on the number of queries of a black-box reduction (of any kind) applies in the extreme high-end. See Section 4 for a discussion on past lower bounds on the number of queries by reductions for black-box hardness amplification proofs.

We also note that the requirements on the parameters in Theorem 3.8 are quite mild. We allow ϵ to approach one (rather than zero) and k to approach r (rather than m). For $\rho = 1$ (namely, starting from a worst-case hard functions) we have that $\eta = \frac{1}{2}$ and the theorem works even for $a = \Omega(2^\ell)$.

3.3 Proofs of Theorem 3.1 and Theorem 3.8

In this section we prove Theorem 3.1 and Theorem 3.8 (which in turn imply Theorem 1.11 and Theorem 1.14). Both Theorems will follow from the following theorem (which can be seen as a version of Theorem 3.1 for PEGs).

Theorem 3.9. *There exists a constant $\nu > 0$ such that for every black-box ρ-hard-function $\Rightarrow$ (k,ϵ)-PEG proof $(\mathsf{Con},\mathsf{Red})$ with parameters $\ell, r, m, a, k, \epsilon, \rho$ such that Red makes at most $q \leq 2^\ell$ queries. If $\rho = \frac{1}{2} + \eta$, $\eta \geq 2^{-\ell}$, $k > r$, $\epsilon \leq 1 - 2^{r-m}$, and $a \leq \nu \cdot \eta^2 \cdot 2^\ell$, then for $j_{\max} = \nu \cdot \frac{\eta^2 \cdot 2^\ell}{\ell}$, and every $j \leq j_{\max}$,*

$$\mathrm{Fix}_j(\mathsf{Con}) \leq a + j \cdot (\log q + \log \frac{4}{\eta}).$$

The proof of Theorem 3.9 is given in Section 3.3.1.

Theorem 3.1 follows from Theorem 3.9. Loosely speaking, this follows because any PRG is also a PEG, and therefore limitations on PEGs also hold for PRGs. More formally, any black-box ρ-hard-function $\Rightarrow$ ϵ-PRG proof $(\mathsf{Con},\mathsf{Red})$ in which $\epsilon = 1 - 2^{r-m}$ is also a black-box ρ-hard-function $\Rightarrow$ (k,ϵ)-PEG proof for $k = r$. Theorem 3.9 applies for the latter, and therefore also applies for the former, proving Theorem 3.1.

Theorem 3.8 follows from Theorem 3.9. The conditions in Theorem 3.8 satisfy the conditions of Theorem 3.9 except for the requirement made in Theorem 3.9 that $q \leq 2^\ell$. Therefore, under the conditions of Theorem 3.9 we can apply Theorem 3.9 and conclude that either $q > 2^\ell$ or the conclusion of Theorem 3.9 holds and for $j_{\max} = \nu \cdot \frac{\eta^2 \cdot 2^\ell}{\ell}$, and every $j \leq j_{\max}$, $\mathrm{Fix}_j(\mathsf{Con}) \leq a + j \cdot (\log q + \log \frac{4}{\eta})$. However, the latter option cannot happen. This is because by the restriction on r, we get that $2^r < j_{\max}$. As the number of outputs of Con_f is $2^r < j_{\max}$, it is impossible to fix $j_{\max}$ distinct outputs, and so $\mathrm{Fix}_j(\mathsf{Con})$ is undefined and can be viewed as infinity. Consequently, it can't be the case that $\mathrm{Fix}_j(\mathsf{Con})$ is smaller than some finite quantity. We conclude that $q > 2^\ell$.

3.3.1 Proof of Theorem 3.9

In this section we prove Theorem 3.9. We assume the assumption of Theorem 3.9, namely that $(\mathsf{Con},\mathsf{Red})$ is a black-box ρ-hard-function $\Rightarrow$ (k,ϵ)-PEG proof that satisfies the conditions of Theorem 3.9. We first define a simple distinguisher that answers one iff its input is a pseudorandom string.

Definition 3.10 (The simple distinguisher)**.** *For every $f \in \mathcal{F}_{\ell,1}$ we define $D_f : \{0,1\}^m \to \{0,1\}$ by:*

$$D_f(z) = \begin{cases} 1, & \exists y \in \{0,1\}^r \text{ s.t. } \mathsf{Con}_f(y) = z \\ 0, & \text{otherwise} \end{cases}$$

Claim 3.11. *There exists $\alpha' \in \{0,1\}^a$ and $A_0 \subseteq \mathcal{F}_{\ell,1}$ such that:*

1. $\Pr_{F \leftarrow \mathcal{F}_{\ell,1}}[F \in A_0] \geq 2^{-a}$.

2. *For every* $f \in A_0$*, the function* $C_f : \{0,1\}^\ell \to \{0,1\}$ *defined by* $C_f(x) = \mathsf{Red}^{D_f}(x, \alpha')$*, satisfies that* C_f ρ-hard-function-breaks *the function* f.

Proof. For every $f \in \mathcal{F}_{\ell,1}$, $\Pr[D_f(\mathsf{Con}_f(U_r)) = 1] = 1$, and $\Pr[D_f(U_m) = 1] \leq 2^{r-m}$. Therefore, D_f (k, ϵ) − PEG-breaks Con_f if $k > r$ and $\epsilon \geq 1 - 2^{r-m}$, which is assumed in the statement of Theorem 3.9. By the definition of black-box ρ-hard-function $\Rightarrow$ (k, ϵ)-PEG proof, we have that for every $f \in \mathcal{F}_{\ell,1}$, as D_f (k, ϵ)-PEG-breaks Con_f, there exists $\alpha \in \{0,1\}^a$ such that the function $C \in \mathcal{F}_{\ell,1}$ defined by $C(x) = \mathsf{Red}^D(x, \alpha)$, ρ-hard-function-breaks f. Therefore, by averaging, there exists $\alpha' \in \{0,1\}^a$ that works for a 2^{-a} fraction of the functions $f \in \mathcal{F}_{\ell,1}$, and let A_0 denote the subset of all functions $f \in \mathcal{F}_{\ell,1}$ for which α' works. □

We will now define a function $C : \{0,1\}^\ell \to \{0,1\}$ (that does not depend on f) with the hope of showing that if the conclusion of Theorem 3.9 does not hold, then $C(x)$ simulates $\mathsf{Red}^{D_f}(x, \alpha')$ quite well for "some" choices of $f \in \mathcal{F}_{\ell,1}$ and $x \in \{0,1\}^\ell$.

Definition 3.12. *For every* $x \in \{0,1\}^\ell$*, we define* $z_1^x, \ldots, z_q^x \in \{0,1\}^m$ *as follows: We consider an invocation of* $\mathsf{Red}^{(\cdot)}(x, \alpha')$ *in which all queries are answered by zero, and for* $1 \leq i \leq q$*, let* z_i^x *denote the* j*'th query made by* $\mathsf{Red}(x, \alpha')$ *in this invocation. Let* $C(x)$ *denote the output of* $\mathsf{Red}(x, \alpha')$ *in this invocation.*

Note that $z_1^x, \ldots, z_q^x \in \{0,1\}^m$ do not depend on the choice of the function D given to Red as oracle. We will be interested in the case that Red gets oracle access to the function D_f, for some $f \in \mathcal{F}_{\ell,1}$. Note that for every $f \in \mathcal{F}_{\ell,1}$, z_1^x is the first query made by $\mathsf{Red}^{D_f}(x, \alpha')$, and for $i > 1$, z_i^x may be different than the i'th query made by $\mathsf{Red}^{D_f}(x, \alpha')$), as the reduction is allowed to make adaptive queries. Nevertheless, the following obviously holds:

Claim 3.13. *For every* $f \in \mathcal{F}_{\ell,1}$ *and every* $x \in \{0,1\}^\ell$*, if* $D_f(z_1^x) = \ldots = D_f(z_q^x) = 0$ *then* $C(x) = \mathsf{Red}^{D_f}(x, \alpha')$.

By definition, for every $f \in \mathcal{F}_{\ell,1}$, we have that the number of $z \in \{0,1\}^m$ on which D_f answers one is small (at most 2^r). We will say that a $z \in \{0,1\}^m$ is *weak* with respect to some $A \subseteq \mathcal{F}_{\ell,1}$, if it is likely that $D_f(z) = 1$ when f is chosen uniformly in A.

Definition 3.14. *We say that* $z \in \{0,1\}^m$ *is* t*-weak with respect to a set* $A \subseteq \mathcal{F}_{\ell,1}$ *if*

$$\Pr_{F \leftarrow A}[D_F(z) = 1] \geq 2^{-t}.$$

We will now consider an iterative process in which we will iteratively fix outputs z of Con that are weak.

More precisely, we set $t = \log \frac{4 \cdot q}{\eta}$ and consider the following iterative process. We start with the set A_0 that we already obtained, and $W_0 = \emptyset$. We will maintain the following invariant:

Invariant: At step j we maintain that:

1. $\Pr_{F \leftarrow \mathcal{F}_{\ell,1}}[F \in A_j] \geq 2^{-(a+jt)}$.
2. $A_j \subseteq A_0$.
3. $|W_j| = j$.

4. For every $z \in W_j$ and every $f \in A_j$, $D_f(z) = 1$.

Note that this invariant indeed holds for $j = 0$. At step $0 < j < j_{\max}$ we do the following: If there does not exist a $z \notin W_{j-1}$ that is t-weak with respect to A_{j-1} then the process stops. Otherwise, if there exists $z \notin W_{j-1}$ that is t-weak with respect to A_{j-1}, we define:

- $W_j = W_{j-1} \cup \{z\}$.
- $A_j = \{f \in A_{j-1} : D_f(z) = 1\}$.

We observe that the invariant is indeed kept throughout this process.

Claim 3.15. *For every $j \leq j_{max}$ for which the process has not yet stopped, the invariant above holds.*

Proof. This is obvious for the second, third and fourth items. The first item follows because for every j if the process did not stop before step j, then A_j exists, and we have that:

$$\begin{aligned}\Pr_{F \leftarrow \mathcal{F}_{\ell,1}}[F \in A_j] &= \Pr_{F \leftarrow \mathcal{F}_{\ell,1}}[F \in A_j | F \in A_{j-1}] \cdot \Pr_{F \leftarrow \mathcal{F}_{\ell,1}}[F \in A_{j-1}] \\ &\geq \Pr_{F \leftarrow A_{j-1}}[F \in A_j] \cdot 2^{-(a+(j-1)\cdot t)} \\ &\geq 2^{-t} \cdot 2^{-(a+(j-1)\cdot t)} \\ &\geq 2^{-(a+j\cdot t)},\end{aligned}$$

where the third line is using t-weakness. □

Claim 3.16. *If the process stops at some $j^* < j_{max}$, then there exists $A \subseteq A_0$ such that:*

1. $\Pr_{F \leftarrow \mathcal{F}_{\ell,1}}[F \in A] \geq 2^{-(a+j^*\cdot t)+1}$.
2. $\Pr_{F \leftarrow A, X \in \{0,1\}^\ell}[C(X) = F(X)] \geq \frac{1}{2} + \frac{\eta}{2}$.

Proof. If the process stopped at some $j^* < j_{max}$, then there does not exist a $z \notin W_{j^*}$ that is t-weak with respect to A_{j^*}. This in particular means that for every $x \in \{0,1\}^\ell$, $z_1^x, \ldots, z_q^x$ are not t-weak with respect to A_{j^*} and therefore, by a union bound:

$$\Pr_{F \leftarrow A_{j^*}}[\exists i \in [q] : D_F(z_i^x) = 1] \leq q \cdot 2^{-t} \leq \frac{\eta}{4},$$

which means that:

$$\Pr_{F \leftarrow A_{j^*}}[\forall i \in [q] : D_F(z_i^x) = 0] \geq 1 - \frac{\eta}{4},$$

by Claim 3.13 we conclude that for every $x \in \{0,1\}^\ell$,

$$\Pr_{F \leftarrow A_{j^*}}[C(x) = \mathsf{Red}^{D_F}(x, \alpha')] \geq 1 - \frac{\eta}{4}.$$

For every $x \in \{0,1\}^\ell$, let V_x denote the random variable (over the probability space of $F \leftarrow A_{j^*}$) defined by:

$$V_x = \begin{cases} 1, & C(x) \neq \mathsf{Red}^{D_F}(x, \alpha') \\ 0, & \text{otherwise} \end{cases}$$

Let $V = \sum_{x \in \{0,1\}^\ell} V_x$. It follows that $\mathbb{E}_{F \leftarrow A_{j^*}}[V] \leq 2^\ell \cdot \frac{\eta}{4}$. By Markov's inequality we have that:

$$\Pr_{F \leftarrow A_{j^*}}[V > 2^\ell \cdot \frac{\eta}{2}] < \frac{1}{2}.$$

Let $A = \left\{ f \in A_{j^*} : V \leq 2^\ell \cdot \frac{\eta}{2} \right\}$. It follows that:

$$\Pr_{F \leftarrow \mathcal{F}_{\ell,1}}[F \in A] \geq \frac{1}{2} \cdot \Pr_{F \leftarrow \mathcal{F}_{\ell,1}}[F \in A_{j^*}] \geq 2^{-(a+j^* \cdot t)+1}.$$

For every $f \in A$, we have that the fraction of inputs $x \in \{0,1\}^\ell$ on which $V_x(f) = 1$ (meaning that $C(x) \neq \mathsf{Red}^{D_f}(x, \alpha'))$ is $\frac{V(f)}{2^\ell} \leq \frac{\eta}{2}$. This means that when we choose both $F \leftarrow A$ and $X \leftarrow \{0,1\}^\ell$ independently, we have that:

$$\Pr_{F \leftarrow A, X \leftarrow \{0,1\}^\ell}[C(X) \neq \mathsf{Red}^{D_F}(X, \alpha')] \leq \frac{\eta}{2}.$$

On the other hand, as $A \subseteq A_{j^*} \subseteq A_0$, by Claim 3.11 we have that for every $f \in A$, the function $C_f : \{0,1\}^\ell \to \{0,1\}$ defined by $C_f(x) = \mathsf{Red}^{D_f}(x, \alpha')$, satisfies that C_f ρ-hard-function-breaks the function f. This means that for every $f \in A$,

$$\Pr_{X \leftarrow \{0,1\}^\ell}[\mathsf{Red}^{D_f}(X, \alpha') = f(X)] \geq \rho.$$

This means that when we choose both $F \leftarrow A$ and $X \leftarrow \{0,1\}^\ell$ independently, we have that:

$$\Pr_{F \leftarrow A, X \leftarrow \{0,1\}^\ell}[\mathsf{Red}^{D_F}(X, \alpha') = F(X)] \geq \rho.$$

Putting things together, we have that:

$$\Pr_{F \leftarrow A, X \leftarrow \{0,1\}^\ell}[C(X) = F(X)] \geq \rho - \frac{\eta}{2} = \frac{1}{2} + \frac{\eta}{2}.$$

□

However, the next claim shows that if A is a large set, it is unlikely that a single function C is a good approximation to $F \leftarrow A$.

Claim 3.17. *For every $A \subseteq \mathcal{F}_{\ell,1}$, and every function $C : \{0,1\}^\ell \to \{0,1\}$, if $\Pr_{F \leftarrow \mathcal{F}_{\ell,1}}[F \in A] \geq 2^{-\Delta}$ then*

$$\Pr_{F \leftarrow A, X \leftarrow \{0,1\}^\ell}[C(X) = F(X)] = \frac{1}{2} + O\left(\sqrt{\frac{\Delta + \ell}{2^\ell}}\right).$$

Proof. For the purpose of contradiction we will set $\lambda = \Omega\left(\sqrt{\frac{\Delta+\ell}{2^\ell}}\right)$, and assume that

$$\Pr_{F \leftarrow A, X \leftarrow \{0,1\}^\ell}[C(X) = F(X)] \geq \frac{1}{2} + \lambda.$$

By an averaging argument, it follows that:

$$\Pr_{F \leftarrow A}\left[\Pr_{X \leftarrow \{0,1\}^\ell}[C(X) = F(X)] \geq \frac{1}{2} + \frac{\lambda}{2}\right] \geq \frac{\lambda}{2}.$$

Let $A' \subseteq A$ be the subset defined as follows:

$$A' = \left\{ f \in A : \Pr_{X \leftarrow \{0,1\}^\ell}[C(X) = f(X)] \geq \frac{1}{2} + \frac{\lambda}{2} \right\}.$$

It follows that $|A'| \geq \frac{\lambda}{2} \cdot |A|$. For every function $C \in \mathcal{F}_{\ell,1}$, the number of $f \in \mathcal{F}_{\ell,1}$ such that $\Pr_{X \leftarrow \{0,1\}^\ell}[C(X) = f(X)] \geq \frac{1}{2} + \frac{\lambda}{2}$ is bounded by the size of a Hamming ball in $\{0,1\}^{2^\ell}$ that is of radius $(\frac{1}{2} - \frac{\lambda}{2}) \cdot 2^\ell$. The latter quantity is bounded by $2^{H(\frac{1}{2} - \lambda/2) \cdot 2^\ell} \leq 2^{(1 - O(\lambda^2)) \cdot 2^\ell}$ where $H(\cdot)$ is Shannon's binary entropy function, and using the fact that $H(\frac{1}{2} - \lambda) = 1 - O(\lambda^2)$. We conclude that there exists a constant $c > 1$ such that:

$$|A| \leq \frac{2}{\lambda} \cdot |A'| \leq \frac{2}{\lambda} \cdot 2^{(1 - c \cdot \lambda^2) \cdot 2^\ell} = 2^{\log(\frac{1}{\lambda}) + 1 + (1 - c \cdot \lambda^2) \cdot 2^\ell} = 2^{2^\ell - (c \cdot \lambda^2 \cdot 2^\ell - \log(1/\lambda) - 1)}.$$

We also have that $|A| \geq 2^{2^\ell - \Delta}$, and so we conclude that $\Delta > c \cdot \lambda^2 \cdot 2^\ell - \log(1/\lambda) - 1$. This gives a contradiction if $\lambda = \Omega\left(\sqrt{\frac{\Delta + \ell}{2^\ell}}\right)$. □

Putting Claim 3.16 and Claim 3.17 together, we get that:

Claim 3.18. *The iterative process does not stop until $j = j_{\max}$.*

Proof. By Claim 3.16 and Claim 3.17, we conclude that if the process stops at some $j^* < j_{max}$ then there exists a constant $c > 1$ such that:

$$\frac{1}{2} + c \cdot \sqrt{\frac{a + j^* \cdot t + \ell}{2^\ell}} \geq \frac{1}{2} + \frac{\eta}{2},$$

Recalling that $t = \log \frac{4q}{\eta}$ this gives that:

$$\begin{aligned} q &\geq \frac{\eta}{4} \cdot 2^{\frac{\frac{2^\ell \cdot \eta^2}{4c} - a - \ell}{j^*}} \\ &\geq \frac{\eta}{4} \cdot 2^{\frac{\frac{2^\ell \cdot \eta^2}{5c}}{j^*}} \\ &> 2^\ell, \end{aligned}$$

where the first inequality follows because we have that $a \leq \nu \cdot \eta^2 \cdot 2^\ell$, and we can choose the constant $\nu > 0$ to be sufficiently small. The second inequality follows because $\eta \geq 2^{-\ell}$, and $j^* \leq j_{\max} \leq \nu \cdot \frac{\eta^2 \cdot 2^\ell}{\ell}$, and we can choose $\nu > 0$ to be sufficiently small.

Therefore, as we are assuming that $q \leq 2^\ell$, it is a contradiction if the process stops at some $j^* < j_{\max}$. □

As the process did not stop at any $j < j_{\max}$, then for every $j \leq j_{\max}$, the sets A_j, W_j are defined, and by Claim 3.15 they maintain the invariant.

By the invariant, for every $j \leq j_{\max}$ we conclude that for every $z \in W_j$ and every $f \in A_j$, $D_f(z) = 1$ (meaning that there exists $y \in \{0,1\}^r$ such that z is an output of Con_f). By the invariant, we know that $|W_j| = j$ therefore, if we denote the elements of W_j by $z_1, \ldots, z_j \in \{0,1\}^m$

we conclude that for every $f \in A_j$, and every $i \in [j]$ there exists $y_i \in \{0,1\}^r$ (that may depend on f) such that $\mathsf{Con}_f(y_i) = z_i$. From the invariant, we also have that:

$$\Pr_{F \leftarrow \mathcal{F}_{\ell,1}}[F \in A_j] \geq 2^{-(a+j\cdot t)} = 2^{-(a+j\cdot(\log q + \log \frac{4}{\eta}))}.$$

Putting everything together we get that:

$$\Pr_{F \leftarrow \mathcal{F}_{\ell,1}}[\forall i \in [j] : \exists y_i \in \{0,1\}^r \text{ s.t. } \mathsf{Con}_F(y_i) = z_i] \geq 2^{-(a+j\cdot(\log q + \log \frac{4}{\eta}))},$$

which gives that for every $j \leq j_{\max}$, $\mathrm{Fix}_j(\mathsf{Con}) \leq a + j \cdot (\log q + \log \frac{4}{\eta})$. This proves Theorem 3.9.

4 Limitations on black-box hardness amplification at the extreme high-end

In this section we prove Theorem 1.15 showing that reductions for black-box hard-function $\Rightarrow$ $(\frac{1}{2}+\epsilon)$-hard-function proofs must make many queries, even at the extreme high-end. Theorem 1.15 is a combination of two lower bounds, stated next:

Theorem 4.1 (Lower bound in terms of ϵ)**.** *If* $(\mathsf{Con}, \mathsf{Red})$ *is a* hard-function $\Rightarrow$ $(\frac{1}{2}+\epsilon)$-hard-function *proof with parameters* $\ell, \ell', a, \rho = 1, \rho' = \frac{1}{2}+\epsilon$*, satisfying* $a \leq \frac{2^\ell}{10}$ *and* $\ell' \geq \log \frac{1}{\epsilon^2} + \Omega(1)$ *then* Red *must make at least* $q = \Omega(\frac{1}{\epsilon})$ *queries.*

Theorem 4.2 (Lower bound in terms of ℓ)**.** *If* $(\mathsf{Con}, \mathsf{Red})$ *is a* hard-function $\Rightarrow$ $(\frac{1}{2}+\epsilon)$-hard-function *proof with parameters* $\ell, \ell', a, \rho = 1, \rho' = \frac{1}{2}+\epsilon$*, satisfying* $\epsilon \leq \frac{1}{10}$*, and* $\ell' \geq \log \frac{1}{\epsilon^2} + \Omega(1)$ *then* Red *must make at least* $q \geq \frac{\ell - \log(2a)}{3}$ *queries*

A quantitatively better lower bound of $q = \Omega(\frac{\ell}{\epsilon^2})$ was proven by Grinberg, Shaltiel and Viola [GSV18] for the case that $a \leq 2^{\nu\cdot\ell}$ for some constant $\nu > 0$. Theorems 4.1 and Theorem 4.2 achieve a smaller bound on q, but apply in the extreme high-end (where $a = 2^{(1-o(1))\cdot\ell}$) all the way up to $a = \frac{2^\ell}{10}$. This is especially significant in the case of Theorem 4.1 which (as we explained in detail in Section 1.3.4) can be used to show limitations on the PRG composition of Chen and Tell [CT21b].

Roadmap for this section. Both Theorem 4.1 and Theorem 4.2 will be proven by first connecting a black-box reduction to a depth 3 circuit for a version of the "coin problem". This connection is stated and proven in Section 4.1. The proofs of Theorem 4.1 and Theorem 4.2 show that such a depth 3 circuit must have large q. These proofs are given in Section 4.2 and Section 4.3.

4.1 Reductions as depth 3 circuits

The proof of Theorem 4.1 and Theorem 4.2 rely on the following lemma, which is inspired by an argument of Applebaum et al. [AASY16] (see also [RSV21]) and relates black-box reductions to constant depth circuits.

Lemma 4.3 (Reduction to circuit)**.** *Let* $(\mathsf{Con}, \mathsf{Red})$ *be a black-box* hard-function $\Rightarrow$ $(\frac{1}{2}+\epsilon)$-hard-function *proof with parameters* $\ell, \ell', a, \rho = 1, \rho' = \frac{1}{2}+\epsilon$*, such that* Red *makes* q *queries. For every* $\alpha \in \{0,1\}^a$*, there exists a* q*-CNF* C^α *with* $2^{q+\ell}$ *clauses, over* $n = 2^{\ell'}$ *variables, such that for* $C = \bigvee_{\alpha\in\{0,1\}^a} C^\alpha$*, we have that:*

- $\Pr[C(U_n)] \leq 2^{-(2^\ell - a)}$.
- *For every* $x \in \{0,1\}^n$ *such that* $\text{weight}(x) \leq (\frac{1}{2} - \epsilon) \cdot n$, $C(x) = 1$.

Proof. We will view a string z of length $n = 2^{\ell'}$, as a function $z \in \mathcal{F}_{\ell',1}$ and vice-versa, by $z(y) = z_y$. The proof of [AASY16] (and the proof presented here) makes use of an idea originating in [Vio06, SV10] (and credited to Madhu Sudan) that the reduction must succeed if given oracle access to $\mathsf{Con}_f \oplus z$ for a string z with $\text{weight}(z) \leq n \cdot (\frac{1}{2} - \epsilon)$ (as such an oracle $(\frac{1}{2} + \epsilon)$-hard-function-breaks Con_f), but cannot succeed when given oracle access to $\mathsf{Con}_f \oplus z$ for $z \leftarrow U_n$ (as z "wipes out" the information in Con_f). This will translate into the two conditions in the lemma. Details follow:

For every $f \in \mathcal{F}_{\ell,1}$, $\alpha \in \{0,1\}^a$ and $x \in \{0,1\}^\ell$, we define $C_{f,\alpha,x} : \{0,1\}^n \to \{0,1\}$ as follows:

$$C_{f,\alpha,x}(z) = 1 \text{ iff } \mathsf{Red}^{\mathsf{Con}_f \oplus z}(x, \alpha) = f(x).$$

We have that Red makes at most q queries, and this implies that for every f and α, the function $C_{f,\alpha}$ can be computed by a decision tree with height q. Therefore, it can be computed by a q-CNF with 2^q clauses. For every $f \in \mathcal{F}_{\ell,1}$, and $\alpha \in \{0,1\}^a$ we define $C_{f,\alpha} : \{0,1\}^n \to \{0,1\}$ as follows:

$$C_{f,\alpha}(z) = 1 \text{ iff } \forall x \in \{0,1\}^\ell, \mathsf{Red}^{\mathsf{Con}_f \oplus z}(x, \alpha) = f(x).$$

By definition for every f and α, $C_{f,\alpha}(x) = \bigwedge_{x \in \{0,1\}^\ell} C_{f,\alpha,x}$. This means that $C_{f,\alpha}$ is an AND of 2^ℓ q-CNFs with 2^q clauses, and overall, it can be computed by a q-CNF with $2^{q+\ell}$ clauses. For every $f \in \mathcal{F}_{\ell,1}$, we define $C_f : \{0,1\}^n \to \{0,1\}$ as follows:

$$C_{f,\alpha}(z) = 1 \text{ iff } \exists \alpha \in \{0,1\}^a, \text{ s.t. } \forall x \in \{0,1\}^\ell, \mathsf{Red}^{\mathsf{Con}_f \oplus z}(x, \alpha) = f(x).$$

This gives that for every $f \in \mathcal{F}_{\ell,1}$, C_f is an OR of 2^a q-CNFs with $2^{q+\ell}$ clauses.

Furthermore, by the definition of black-box $\mathsf{hard\text{-}function} \Rightarrow (\frac{1}{2} + \epsilon)$-$\mathsf{hard\text{-}function}$ proof, for every $f \in \mathcal{F}_{\ell,1}$, if $\text{weight}(z) \leq (\frac{1}{2} - \epsilon) \cdot n$, then the function $\mathsf{Con}_f \oplus z$, $\frac{1}{2} + \epsilon$-hard-function-breaks Con_f. This in turn implies (by Definition 1.7) that there exists $\alpha \in \{0,1\}^a$ such that for every $x \in \{0,1\}^\ell$, $\mathsf{Red}^{\mathsf{Con}_f \oplus z}(x) = f(x)$, meaning that $C_f(z) = 1$. This means that for every choice of $f \in \mathcal{F}_{\ell,1}$, C_f satisfies the second item.

We will now show that there exists an $f \in \mathcal{F}_{\ell,1}$ such that C_f satisfies the first item. For a uniformly chosen $z \leftarrow U_n$, we have that for every $f \in \mathcal{F}_{\ell,1}$, $\mathsf{Con}_f \oplus z$ is distributed uniformly over $\{0,1\}^n$, and contains no information about f. It follows that:

$$\begin{aligned}
\Pr_{f \leftarrow \mathcal{F}_{\ell,1}, z \leftarrow U_n}[C_f(z) = 1] &= \Pr_{F \leftarrow \mathcal{F}_{\ell,1}, z \leftarrow U_n}[\exists \alpha \in \{0,1\}^a, \text{ s.t. } \forall x \in \{0,1\}^\ell, \mathsf{Red}^{\mathsf{Con}_f \oplus z}(x, \alpha) = f(x)] \\
&= \Pr_{f \leftarrow \mathcal{F}_{\ell,1}, z \leftarrow U_n}[\exists \alpha \in \{0,1\}^a, \text{ s.t. } \forall x \in \{0,1\}^\ell, \mathsf{Red}^{z}(x, \alpha) = f(x)] \\
&\leq Pr_{f \leftarrow \mathcal{F}_{\ell,1}}[\exists \alpha \in \{0,1\}^a, \text{ s.t. } \forall x \in \{0,1\}^\ell, \mathsf{Red}^{z^*}(x, \alpha) = f(x)],
\end{aligned}$$

for some $z^* \in \{0,1\}^n$ which maximizes the success probability. We can therefore continue and obtain that:

$$\begin{aligned}
\Pr_{f \leftarrow \mathcal{F}_{\ell,1}, z \leftarrow U_n}[C_f(z) = 1] &\leq \Pr_{F \leftarrow \mathcal{F}_{\ell,1}}[\exists \alpha \in \{0,1\}^a, \text{ s.t. } \forall x \in \{0,1\}^\ell, \mathsf{Red}^{z^*}(x, \alpha) = f(x),] \\
&\leq \sum_{\alpha \in \{0,1\}^a} \Pr_{f \leftarrow \mathcal{F}_{\ell,1}}[\forall x \in \{0,1\}^\ell, \mathsf{Red}^{z^*}(x, \alpha) = f(x)] \\
&\leq 2^a \cdot \frac{1}{2^{2^\ell}},
\end{aligned}$$

where the penultimate inequality follows by a union bound, and the last inequality follows because for every $x \in \{0,1\}^\ell$,

$$\Pr_{f \leftarrow \mathcal{F}_{\ell,1}}[\mathsf{Red}^{z^*}(x,\alpha) = f(x)] = \frac{1}{2},$$

and these events are independent for the 2^ℓ choices of $x \in \{0,1\}^\ell$. Finally, by averaging, we conclude that there exists $f \in \mathcal{F}_{\ell,1}$ such that:

$$\Pr_{z \leftarrow U_n}[C_f(z) = 1] \leq 2^{-(2^\ell - a)}.$$

The final function C will be this function C_f which indeed satisfies the properties in the conclusion of the lemma. □

4.2 Proof of Theorem 4.1

In this section we prove Theorem 4.1. We split the proof into two parts, specified in Lemma 4.4 and Lemma 4.5 below. This splitting will allow us to use Lemma 4.5 in the application to lower bounds on local list-decoding in Section 5.

Lemma 4.4. *Assume the conditions of Theorem 4.1. For every $\alpha \in \{0,1\}^a$, there exists a q-CNF C^α over $n = 2^{\ell'}$ variables, such that for $C = \bigvee_{\alpha \in \{0,1\}^a} C^\alpha$, and $p = \frac{\epsilon}{10}$:*

- $\Pr[C(U_n) = 1] \leq 2^{-(2^\ell - a)}$.
- $\Pr_{\rho \leftarrow \mathsf{R}^n_p}[\Pr_{x \leftarrow U^{1/3}_{n \cdot p}}[C(\mathrm{Fill}_\rho(x)) = 1] \geq 0.99] \geq 0.99$.

The next lemma shows that under (a weak form) of the conclusion of Lemma 4.4, if a is slightly smaller than 2^ℓ then $q = \Omega(\frac{1}{\epsilon})$.

Lemma 4.5. *Let C be a circuit such that $C = \bigvee_{\alpha \in \{0,1\}^a} C^\alpha$, where for each $\alpha \in \{0,1\}^a$, C^α is a q-CNF over n variables. Let $p = \frac{\epsilon}{10}$ and assume that:*

- $\Pr[C(U_n) = 1] \leq 2^{-(2^\ell - a)}$.
- $\Pr_{\rho \leftarrow \mathsf{R}^n_p}\left[\Pr_{x \leftarrow U^{1/3}_{n \cdot p}}[C(\mathrm{Fill}_\rho(x)) = 1] \geq 0.01\right] \geq 0.01$.
- $a \leq \frac{2^\ell}{10}$.

Then $q \geq \frac{1000}{\epsilon}$.

Together, Lemma 4.4 and Lemma 4.5 imply Theorem 4.1. The proofs of Lemma 4.4 and Lemma 4.5 are given in Sections 4.2.1 and 4.2.2.

4.2.1 Proof of Lemma 4.4

We first apply Lemma 4.3 and obtain that for every $\alpha \in \{0,1\}^a$, there exists a q-CNF C^α over $n = 2^{\ell'}$ variables, such that:

- $\Pr[C^\alpha(U_n) = 1] \leq 2^{-(2^\ell - a)}$.
- For every $x \in \{0,1\}^n$ such that $\mathrm{weight}(x) \leq (\frac{1}{2} - \epsilon) \cdot n$, there exists $\alpha \in \{0,1\}^a$ such that $C^\alpha(x) = 1$.

When choosing $\rho \leftarrow \mathsf{R}_p^n$ we expect that $\frac{1}{n} \cdot |\{i : \rho(i) = 1\}| = \frac{1}{2} - \frac{p}{2}$. Therefore, by a Chernoff bound:

$$\Pr_{\rho \leftarrow \mathsf{R}_p^n}[\frac{1}{n} \cdot |\{i : \rho(i) = 1\}| \geq \frac{1}{2} - 0.49 \cdot p] \leq 2^{-\Omega(p^2 \cdot n)},$$

which is smaller than 0.01, by the assumption that $\ell' \geq \log \frac{1}{\epsilon^2} + \Omega(1)$, and $n = 2^{\ell'}$. If this event occurs, then for every $x \in \{0,1\}^{n \cdot p}$,

$$\text{weight}(\text{Fill}_\rho(x)) \leq (\frac{1}{2} - 0.49 \cdot p) \cdot n + \text{weight}(x).$$

This means that if $\text{weight}(x) \leq 0.34 \cdot p \cdot n$, then

$$\text{weight}(\text{Fill}_\rho(x)) \leq n \cdot (\frac{1}{2} - (0.49 - 0.34) \cdot p) \leq n \cdot (\frac{1}{2} - \epsilon),$$

By our assumption that $p \leq \frac{\epsilon}{10}$. Applying a Chernoff bound, we conclude that:

$$\Pr_{x \leftarrow U_{n \cdot p}^{1/3}}[\text{weight}(x) \geq 0.34 \cdot p \cdot n] \leq 2^{-\Omega(pn)},$$

which is smaller than 0.01 by the assumption that $\ell' \geq \log \frac{1}{\epsilon^2} + \Omega(1)$, and $n = 2^{\ell'}$.

4.2.2 Proof of Lemma 4.5

We want to show that following a random restriction, each C^α is a low height decision tree.

Claim 4.6. *If $q < \frac{1000}{\epsilon}$ then there exists a restriction $\rho : [n] \to \{0,1,*\}$ with $\text{Free}(\rho) = n \cdot p$ such that:*

- *For every $\alpha \in \{0,1\}^a$, C_ρ^α is a decision tree of height a.*
- $\Pr[C_\rho(U_n) = 1] \leq 2^{-(2^\ell - a - 10)}$.
- $\Pr[C_\rho(U_n^{1/3}) = 1] \geq 0.01$.

Proof. We will show that when choosing $\rho \leftarrow \mathsf{R}_p^n$ there is a positive probability to obtain a ρ that satisfies all three items. This will follow by a union bound in which we show that the probability that each of the items does not hold is small.

For the first item, we note that by Theorem 2.3 for every $\alpha \in \{0,1\}^a$, the probability over $\rho \leftarrow \mathsf{R}_p^n$ that C_ρ^α does not have a decision tree of height a, is at most $(7 \cdot p \cdot q)^a \leq (7/100)^a$. By a union bound over the 2^a choices of $\alpha \in \{0,1\}^a$, the probability over $\rho \leftarrow \mathsf{R}_p^n$ that there exists $\alpha \in \{0,1\}^a$ such that C_ρ^α does not have a decision tree of height a, is at most 0.001.

For the second item, we recall that by Lemma 4.4 we have that: $\Pr[C(U_n) = 1] \leq 2^{-(2^\ell - a)}$. Therefore, by Markov's in equality we conclude that with probability at least 0.999 over $\rho \leftarrow \mathsf{R}_p^n$:

$$\Pr[C_\rho(U_n) = 1] \leq 1000 \cdot 2^{-(2^\ell - a)} \leq 2^{-(2^\ell - a - 10)}.$$

Finally, by the second item in Lemma 4.4 we have that with probability at least 0.01 over $\rho \leftarrow \mathsf{R}_p^n$:

$$\Pr[C_\rho(U_n^{1/3}) = 1] \geq 0.01$$

Overall, by a union bound with probability at least $1 - (0.99 + 0.001 + 0.001) > 0$ over $\rho \leftarrow \mathsf{R}_p^n$, we obtain a ρ that satisfies all three conditions. □

Let ρ be the restriction guaranteed by the claim. We can view C_ρ as $C_\rho = \bigvee_{\alpha\in\{0,1\}^a} C_\rho^\alpha$. This means that C_ρ is an OR of 2^a decision trees of height a. Each such decision tree can be replaced by an a-DNF with 2^a clauses, and overall, we get that C_ρ is an a-DNF with 2^{2a} clauses. This gives that:

Claim 4.7. *If $q < \frac{1000}{\epsilon}$ then there exists an a-DNF C' with $p \cdot n$ variables and 2^{2a} clauses such that:*

- $\Pr[C'(U_{p\cdot n}) = 1] \leq 2^{-(2^\ell - a - 10)}$.
- $\Pr[C'(U^{1/3}_{p\cdot n}) = 1] \geq 0.01$.

This means that one of the 2^{2a} clauses of C' must satisfy the following:

Claim 4.8. *If $q < \frac{1000}{\epsilon}$ then there exists a DNF clause $\bar{C}$ over $t \leq a$ literals such that:*

- $\Pr[\bar{C}(U_t) = 1] \leq 2^{-(2^\ell - a - 10)}$.
- $\Pr[\bar{C}(U_t^{1/3}) = 1] \geq \frac{0.01}{2^{2a}} \geq 2^{-(2a+10)}$.

We are now ready to prove Lemma 4.5. We assume for the purpose of contradiction that $q < \frac{1000}{\epsilon}$ and show that Claim 4.8 yields a contradiction. This is because a DNF clause over $t \leq a$ literals cannot distinguish U_t from $U_t^{1/3}$ with the parameters stated in Claim 4.8.

More precisely, as under $U_t^{1/3}$, every literal evaluates to one with probability that is upper bounded by $2/3$, we conclude that:

$$\Pr[\bar{C}(U_t^{1/3}) = 1] \leq \left(\frac{2}{3}\right)^t.$$

Under U_t, every literal evaluates to one with probability half. Therefore,

$$\Pr[\bar{C}(U_t) = 1] = \left(\frac{1}{2}\right)^t.$$

By Claim 4.8 it follows that:

$$\left(\frac{4}{3}\right)^t = \frac{\left(\frac{2}{3}\right)^t}{\left(\frac{1}{2}\right)^t} \geq \frac{2^{-(2a+10)}}{2^{-(2^\ell - a - 10)}} = 2^{2^\ell - 3a - 20}.$$

which gives that:

$$a \geq t \geq \frac{2^\ell - 3a - 20}{\log \frac{4}{3}} > \frac{2^\ell}{10},$$

using the assumption that $a \leq \frac{2^\ell}{10}$, and the lemma follows.

4.3 Proof of Theorem 4.2

In this section we prove Theorem 4.2. We first apply Lemma 4.3 and obtain that for every $\alpha \in \{0,1\}^a$, there exists a q-CNF C^α with $2^{q+\ell}$ clauses, over $n = 2^{\ell'}$ variables, such that for $C = \bigvee_{\alpha\in\{0,1\}^a} C^\alpha$, we have that:

- $\Pr[C(U_n) = 1] \leq 2^{-(2^\ell - a)}$.
- For every $x \in \{0,1\}^n$ such that $\text{weight}(x) \leq (\frac{1}{2} - \epsilon) \cdot n$, there exists $\alpha \in \{0,1\}^a$ such that $C^\alpha(x) = 1$.

As there are 2^a such CNFs, it follows that one of them satisfies:

Claim 4.9. *There exists $\alpha \in \{0,1\}^a$ such that:*

- $\Pr[C^\alpha(U_n) = 1] \leq 2^{-(2^\ell - a)}$.
- $\Pr[C^\alpha(U_n^{1/3}) = 1] \geq 2^{-(a+1)}$.

Proof. The first item follows because the top gate of C is an OR gate. For the second item, by a Chernoff bound, and our assumption that $\epsilon \leq \frac{1}{10}$:

$$\Pr_{x \leftarrow U_n^{1/3}}[\text{weight}(x) \leq (\frac{1}{2} - \epsilon) \cdot n] \leq 2^{-\Omega(n)} \leq \frac{1}{2},$$

meaning that:

$$\Pr_{x \leftarrow U_n^{1/3}}[C(x) = 1] \geq \frac{1}{2},$$

and by averaging, there exists $\alpha \in \{0,1\}^a$ such that:

$$\Pr_{x \leftarrow U_n^{1/3}}[C^\alpha(x) = 1] \geq \frac{1}{2 \cdot 2^a}.$$

□

Let $C_0 = C^\alpha$ that is guaranteed by Claim 4.9. We will start from C_0 and will iteratively apply the following lemma (which is proven in Section 4.3.1).

Lemma 4.10. *For every v-CNF D over m variables, if $\Pr[D(U_m^{1/3}) = 1] \geq 2^{-w}$ then there exists a restriction $\rho : [m] \to \{0,1,*\}$ with $\text{Free}(\rho) = m' = m - w \cdot 4^v$, such that:*

- *D_ρ is a $(v-1)$-CNF over m' variables.*
- $\Pr[D_\rho(U_{m'}) = 1] \leq \Pr[D(U_m) = 1] \cdot 2^{w \cdot 4^v}$.
- $\Pr[D_\rho(U_{m'}^{1/3}) = 1] \geq \Pr[D(U_m^{1/3}) = 1]$.

More precisely, at every step $i > 0$, we apply Lemma 4.10 with $w = a + 1$ on $D = C_{i-1}$, and set $C_i = D_\rho$. We use m_i to denote the input length of C_i. We will maintain the following invariant (and note that this indeed holds for $i = 0$).

Invariant: At step i the following hold:

- C_i is a $(q-i)$-CNF.
- $\Pr[C_i(U_{m_i}) = 1] \leq 2^{-\left(2^\ell - a - i \cdot (a+1) \cdot 4^q\right)}$.
- $\Pr[C_i(U_{m_i}^{1/3}) = 1] \geq 2^{-(a+1)}$.

It immediately follows that this invariant holds for every $i \leq q$. This means that at the conclusion of this process, for $i = q$ we have that:

- C_m is a constant.
- $\Pr[C_q(U_{m_q}) = 1] \leq 2^{-\left(2^\ell - a - q\cdot(a+1)\cdot 4^q\right)}$.
- $\Pr[C_q(U^{1/3}_{m_q}) = 1] \geq 2^{-(a+1)}$.

Therefore, by the third item, it must be that C_m is the constant one, and by the second item it must be that:

$$2^{-\left(2^\ell - a - q\cdot(a+1)\cdot 4^q\right)} \geq 1.$$

which implies that:

$$q \geq \frac{\ell - \log(a+1)}{3}.$$

4.3.1 Proof of Lemma 4.10

In this section, we prove Lemma 4.10. We first observe that if a v-CNF does not have many clauses that have disjoint variables, then there must be a small "cover" (namely, a set of variables such that each clause contains a variable from the cover).

Proposition 4.11. *In every v-CNF D, if there does not exist t clauses with disjoint variables, then there is a subset S of $s = (t-1)\cdot v$ variables, such that every clause of D contains a variable in S.*

Proof. We go over the clauses one by one, staring with an empty S. Every time the current clause does not contain a variable from S, we mark the current clause, and add its variables to S. Every time we mark a clause, its variables are disjoint from all variables of previously marked clauses. By our assumption, we mark at most $t-1$ clauses, and so when we conclude the size of S is at most $(t-1)\cdot v$. □

Lemma 4.12. *For every v-CNF D over m variables, if $\Pr[D(U^{1/3}_m) = 1] \geq 2^{-w}$ then there exists a set S of size $s = w \cdot 4^v$ variables such that every clause of D contains at least one variable in S.*

Proof. If there does not exists a set S with the required properties, then by Proposition 4.11, D has $t = \frac{s}{v}$ clauses that have disjoint variables. Under $U^{1/3}_m$, each literal of D has probability at least $1/3$ to evaluate to zero. This gives that each clause has probability at most $1 - \left(\frac{1}{3}\right)^v$ to evaluate to one. Consequently, the probability that t disjoint clauses evaluate to one is at most $\left(1 - \left(\frac{1}{3}\right)^v\right)^t$, implying that:

$$\Pr[D(U^{1/3}_m) = 1] \leq \left(1 - \left(\frac{1}{3}\right)^v\right)^t \leq e^{-3^{-v}\cdot t} < e^{-4^{-v}\cdot s} \leq 2^{-w}.$$

and we get a contradiction. □

We are ready to prove Lemma 4.10. By Lemma 4.12 there exists a set S of size $s = w \cdot 4^v$ variables such that every clause of D contains at least one variable in S. Let A be the set of all restrictions that fix the variables in S, while leaving the other variables free. More precisely,

$$A = \{\rho : [m] \to \{0, 1, *\} : \rho(i) \neq * \text{ iff } i \in S\}.$$

We will show that there exists a $\rho \in A$ that satisfies the guarantees of Lemma 4.10 with positive probability.

The first item of Lemma 4.10 holds for every $\rho \in A$. This is because every clause of D has a variable in S and following the restriction, each clause is over at most $v-1$ variables.

The second item of Lemma 4.10 also holds for every $\rho \in A$. For every choice $y \in \{0,1\}^{w \cdot 4^v}$ of the restricted bits in ρ, let $E_{\rho,y}$ denote the event $\left\{X|_{[n]\setminus \mathrm{Select}(\rho)} = y\right\}$. For every $\rho \in A$ we have that:

$$\begin{aligned}\Pr[D_\rho[D(U_{m'}) = 1] &= \Pr_{X \leftarrow U_m}[D(X) = 1 | E_{\rho,y}] \\ &\leq \frac{\Pr_{X \leftarrow U_m}[D(X) = 1]}{\Pr_{X \leftarrow U_m}[E_{\rho,y}]} \\ &= \Pr[D(U_m) = 1] \cdot 2^{w \cdot 4^v}.\end{aligned}$$

The third item follows because by averaging, there exists $\rho \in A$ and $y \in \{0,1\}^{w \cdot 4^v}$ such that $\mathrm{Assign}(\rho) = y$ and:

$$\Pr_{X \leftarrow U_m^{1/3}}[D(X) = 1 | E_{\rho,y}] \geq \Pr_{X \leftarrow U_m^{1/3}}[D(X) = 1],$$

For this choice of ρ and y, we have that

$$\begin{aligned}\Pr[D_\rho(U_{m'}^{1/3}) = 1] &= \Pr_{X \leftarrow U_m^{1/3}}[D(X) = 1 | X \in E_{\rho,y}] \\ &\geq \Pr_{X \leftarrow U_m^{1/3}}[D(X) = 1] \\ &= \Pr[D(U_{m'}^{1/3}) = 1].\end{aligned}$$

This concludes the proof of Lemma 4.10.

5 Improved lower bounds for local list-decoding algorithms

In this section we show that the techniques developed in Section 4 can be used to get an improved lower bounds on the number of queries of local list-decoding algorithms. In Section 5.1 we review the definition of locally list-decodable codes, and local decoding algorithms. In Section 5.2 we state our improved bound and compare it to previous work. In Section 5.3 we prove the improved bound.

5.1 Definition of locally list-decodable codes

List-decodable codes are a natural extension of (uniquely decodable) error-correcting codes, as it allows (list) decoding for error regimes where unique decoding is impossible. This is an extensively studied area; see [Gur06] for a survey. In this paper, we will be interested in list-decoding of binary codes.

Definition 5.1 (List-decodable code)**.** *For a function* $\mathrm{Enc} : \{0,1\}^k \to \{0,1\}^n$*, and* $w \in \{0,1\}^n$*, we define*

$$\mathrm{List}_\alpha^{\mathrm{Enc}}(w) = \left\{m \in \{0,1\}^k : \mathsf{dist}(\mathrm{Enc}(m), w) \leq \alpha\right\}.$$

We say that Enc *is* (α, L)*-list-decodable if for every* $w \in \{0,1\}^n$*,* $|\mathrm{List}_\alpha^{\mathrm{Enc}}(w)| \leq L$*.*

The task of *algorithmic* list-decoding is to produce the list $\mathrm{List}_{\alpha}^{\mathrm{Enc}}(w)$ on input $w \in \{0,1\}^n$.

Local unique decoding algorithms are algorithms that given an index $i \in [k]$, make few oracle queries to w, and reproduce the bit m_i (with high probability over the choice of their random coins), where m denotes the unique codeword close to w. This notion of *local decoding* has many connections and applications in computer science and mathematics [Yek12].

We will be interested in *local* list-decoding algorithms that receive oracle access to a received word $w \in \{0,1\}^n$, as well as inputs $i \in [k]$ and $j \in [L]$. We will require that for every $m \in \mathrm{List}_{\alpha}^{\mathrm{Enc}}(w)$, with high probability, there exists a $j \in [L]$ such that for every $i \in [k]$, when Dec receives oracle access to w and inputs i, j, it produces m_i with high probability over its choice of random coins. This motivates the next definition.

Definition 5.2 (Randomized local computation)**.** *We say that a procedure $P(i,r)$ locally computes a string $m \in \{0,1\}^k$ with error δ, if for every $i \in [k]$, $\Pr[P(i,R) = m_i] \geq 1-\delta$ (where the probability is over a uniform choice of the "string of random coins" R).*

The definition of local list-decoders considers an algorithmic scenario that works in two steps:

- At the first step (which can be thought of as a preprocessing step) the local list-decoder Dec is given oracle access to w and an index $j \in [L]$. It tosses random coins (which we denote by r^{shared}).
- At the second step, the decoder receives the additional index $i \in [k]$, and tosses additional coins r.
- It is required that for every $w \in \{0,1\}^n$ and $m \in \mathrm{List}_{\alpha}^{\mathrm{Enc}}(w)$, with probability $2/3$ over the choice of the shared coins r^{shared}, there exists $j \in [L]$ such that when the local list-decoder receives j, it locally computes m (using its "non-shared" coins r). The definition uses two types of random coins because the coins r^{shared} are "shared" between different choices of $i \in [k]$ and allow different i's to "coordinate". The coins r, are chosen independently for different choices of $i \in [k]$.

This is formally stated in the next definition.

Definition 5.3 (Local list-decoder)**.** *Let* $\mathrm{Enc} : \{0,1\}^k \to \{0,1\}^n$ *be a function. An* (α, L, q, δ)*-local list-decoder (LLD) for* Enc *is an oracle procedure* $\mathrm{Dec}^{(\cdot)}$ *that receives oracle access to a word* $w \in \{0,1\}^n$*, and makes at most* q *calls to the oracle. The procedure* Dec *also receives inputs:*

- $i \in [k]$ *: The index of the symbol that it needs to decode.*
- $j \in [L]$ *: An index to the list.*
- *Two strings* r^{shared}, r *that are used as random coins.*

It is required that for every $w \in \{0,1\}^n$*, and for every* $m \in \mathrm{List}_{\alpha}^{\mathrm{Enc}}(w)$*, with probability at least* $2/3$ *over choosing a uniform string* r^{shared}*, there exists* $j \in [L]$ *such that the procedure*

$$P_{w,j,r^{\mathrm{shared}}}(i,r) = \mathrm{Dec}^{w}(i, j, r^{\mathrm{shared}}, r)$$

locally computes m *with error* δ*. If we omit* δ*, then we mean* $\delta = 1/3$*.*

See [RSV21] for a discussion on the generality of this definition, and on past work in this area.

5.2 Our Results

Ron-Zewi, Shaltiel and Varma [RSV21] showed lower bounds on the number of queries of local list decoders. We use our improved techniques to prove the following theorem.

Theorem 5.4 (Improved lower bounds for small ϵ)**.** *There exist constants $c_1, c_2 > 1$ such that for every $L \leq 2^{\frac{k}{20}}$, $\delta < \frac{1}{3}$, $n \geq \frac{c_2}{\epsilon^2}$, and every $(\frac{1}{2}-\epsilon, L, q, \delta)$-local list-decoder for $\mathrm{Enc} : \{0,1\}^k \to \{0,1\}^n$, we have that $q \geq \frac{1}{c_1 \cdot \log(k) \cdot \epsilon}$.*

The previous bounds of [RSV21] come in two forms:

- For $L \leq 2^{k^{0.9}}$ and $\epsilon \geq \frac{1}{k^{\Omega(1)}}$, [RSV21] obtain tight bounds (up to constants), showing that: $q = \Omega(\frac{\log(1/\delta)}{\epsilon^2})$.
- If these conditions are not met (and in particular, if ϵ is smaller than $1/k$), [RSV21] obtain weaker bounds, showing that if $\delta < \frac{1}{3}$, $q = \Omega(\frac{1}{\sqrt{\epsilon} \cdot \log k}) - O(\log L)$.

Theorem 5.4 improves upon the second item above (although it does not match the optimal bound of the first item). More specifically, Theorem 5.4 replaces $\sqrt{\epsilon}$ with ϵ, and does not have the additive term of "$-O(\log L)$".

Not surprisingly, these improvements directly correspond to "the extreme high-end". Making this analogy, (namely, setting $L = 2^a$, and $k = 2^\ell$) we have that for $L = 2^{k^{1-o(1)}}$, previous work does not give any bound if $\epsilon \geq \frac{1}{k^{1.9}}$, and in particular for $\epsilon \approx \frac{1}{k}$. In contrast, Theorem 5.4 gives a bound that is $\Omega(\frac{1}{\log k \cdot \epsilon})$, which is not far from the known upper bound of $O(\frac{1}{\epsilon^2})$, and is polynomially related to the upper bound for $\epsilon \leq \frac{1}{\log^2 k}$.

5.3 Proof of Theorem 5.4

In this section we prove Theorem 5.4. The proof uses a similar approach as that of [RSV21] (which is in turn based on [AASY16]) to transform an LLD into a depth 3 circuit for a certain problem. (We make stronger requirements on the circuit, and therefore, need to redo the reduction, taking care to obtain these stronger requirements). Once we obtain a depth 3 circuit, we use Lemma 4.5 to show improved lower bounds. Details follow:

We assume that $L \leq 2^{\frac{k}{20}}$, and $n \geq \frac{c_2}{\epsilon^2}$. Our goal is to prove lower bounds on the number of queries q of $(\frac{1}{2}-\epsilon, L, q, \delta)$-local list-decoders for $\delta < \frac{1}{3}$. It is possible to amplify the error probability δ from $1/3$ to $1/20k$ as follows: After choosing the random string r^{shared}, we choose $e = O(\log k)$ independent uniform strings $r_1, \ldots, r_e$, and apply $\mathrm{Dec}^{(\cdot)}(i, j, r_\ell, r^{\mathrm{shared}})$ for all choices of $\ell \in [e]$. We then output the majority vote of the individual e outputs. It is standard that this gives a $(\frac{1}{2}-\epsilon, q' = O(q \cdot \log k), L, 1/20k)$-LLD for $\mathrm{Enc} : \{0,1\}^k \to \{0,1\}^n$.

We plan to reduce to Lemma 4.5, and set $p = \frac{\epsilon}{10}$ (as in that lemma). Our next step is to fix the random coins of the decoder, and obtain a *deterministic* decoder $\overline{Dec}$ which succeeds in a specific experiment in which: a message $m \leftarrow \{0,1\}^k$ is chosen uniformly, and a "noise string" z is chosen by choosing $\rho \leftarrow \mathsf{R}_p^n$ and filling the unrestricted variables with $U_{p \cdot n}^{1/3}$. Finally, a "received word" w is obtained by $w = \mathrm{Enc}(m) \oplus z$. This is defined formally below:

Definition 5.5 (The experiment RNSY)**.** *We consider the following experiment (which we denote by* RNSY*): The experiment works in two steps. The first step (denoted* RNSY^1*) works as follows:*

- *A message $m \leftarrow \{0,1\}^k$ is chosen uniformly.*

- *A restriction* $\rho \leftarrow R_p^n$ *is chosen uniformly.*

The second step is defined for a fixed $m \in \{0,1\}^k$ *and* $\rho \in R_p^n$*. It is denoted by* $\mathrm{RNSY}^2(m,\rho)$ *and works as follows:*

- *A string* z *is chosen from* $U_{p\cdot n}^{1/3}$*.*
- *We define* $x = \mathrm{Fill}_\rho(z)$*.*
- *We define* $w = \mathrm{Enc}(m) \oplus x$*.*

We use $(m,\rho,z,w) \leftarrow \mathrm{RNSY}$ *to denote* m,ρ,z,w *which are sampled by the two steps of this experiment.*

The next lemma fixes the coins of the local-decoder, making it deterministic. It is similar in spirit to Proposition 3.1 in [RSV21], except that the experiment RNSY that we use is different and more complicated than the one used in [RSV21].

Lemma 5.6. *There exists a constant* $c > 1$ *such that if* $n \geq \frac{c}{\epsilon^2}$ *and* Dec *is a* $(\frac{1}{2}-\epsilon, L, q, \delta)$*-local list-decoder for a function* $\mathrm{Enc} : \{0,1\}^k \to \{0,1\}^n$ *then there exists an oracle procedure* $\overline{Dec}^{(\cdot)}(i,j)$ *receiving* $i \in [k]$*,* $j \in [L]$*, and making* q *queries to a string* $w \in \{0,1\}^n$*, such that with probability at least* 0.51 *over choosing* $(m,\rho) \leftarrow \mathrm{RNSY}^1$*, we have that with probability at least* 0.01 *over choosing* $(z,w) \leftarrow \mathrm{RNSY}^2(m,\rho)$*, there exists* $j \in [L]$ *such that*

$$\Pr_{i \leftarrow [k]}[\overline{Dec}^w(i,j) = m_i] \geq 1 - 10 \cdot \delta.$$

Proof. Let Dec denote an LLD for Enc. When choosing $\rho \leftarrow \mathsf{R}_p^n$ we expect that $\frac{1}{n} \cdot |\{i : \rho(i) = 1\}| = \frac{1}{2} - \frac{p}{2}$. Therefore, by a Chernoff bound:

$$\Pr_{\rho \leftarrow \mathsf{R}_p^n}[\frac{1}{n} \cdot |\{i : \rho(i) = 1\}| \geq \frac{1}{2} - 0.49 \cdot p] \leq 2^{-\Omega(p^2 \cdot n)},$$

which is smaller than 0.01, by our assumption that n is sufficiently larger than $1/\epsilon^2$. If this event occurs, then for every $z \in \{0,1\}^{n \cdot p}$,

$$\mathrm{weight}(\mathrm{Fill}_\rho(z)) \leq (\frac{1}{2} - 0.49 \cdot p) \cdot n + \mathrm{weight}(z).$$

This means that if $\mathrm{weight}(z) \leq 0.34 \cdot p \cdot n$, then

$$\mathrm{weight}(\mathrm{Fill}_\rho(z)) \leq n \cdot (\frac{1}{2} - (0.49 - 0.34) \cdot p) \leq n \cdot (\frac{1}{2} - \epsilon),$$

By our assumption that $p \leq \frac{\epsilon}{10}$. Applying a Chernoff bound, we conclude that:

$$\Pr_{z \leftarrow U_{n\cdot p}^{1/3}}[\mathrm{weight}(z) \geq 0.34 \cdot p \cdot n] \leq 2^{-\Omega(pn)},$$

which is smaller than 0.01 by the assumption that n is sufficiently larger than $1/\epsilon^2$.

Overall, this means that for $\gamma = 0.02$, with probability at least $1-\gamma$ over choosing $(m,\rho,z,w) \leftarrow \mathrm{RNSY}$, we have that $\mathsf{dist}(\mathrm{Enc}(m), w) \leq \frac{1}{2} - \epsilon$, meaning that $m \in \mathrm{List}_{\frac{1}{2}-\epsilon}^{\mathrm{Enc}}(w)$. By the definition of

LLD, this gives that whenever this occurs, with probability at least $2/3$ over the choice of r^{shared}, there exists $j \in [L]$ such that the procedure $P_{w,j,r^{\text{shared}}}(i,r) = \text{Dec}^w(i,j,r^{\text{shared}},r)$ locally computes m with error δ.

Let E_1 be the experiment in which $(m,\rho,z,w) \leftarrow \text{RNSY}$ and r^{shared} be an independent uniform string. It follows that:

$$\Pr_{E_1}[\exists j \in [L] : P_{w,j,r^{shared}} \text{ locally computes } m \text{ with error } \delta] \geq \frac{2}{3} - \gamma.$$

By averaging, there exists a fixed string $\hat{r}^{\text{shared}}$ such that:

$$\Pr_{\text{RNSY}}[\exists j \in [L] : P_{w,j,\hat{r}^{shared}} \text{ locally computes } m \text{ with error } \delta] \geq \frac{2}{3} - \gamma.$$

Let S denote the set of quadruples (m,ρ,z,w) in the support of RNSY for which the event above occurs. For every such quadruple, we have that there exists a $j \in [L]$ for which $P_{w,j,\hat{r}^{shared}}$ locally computes m with error δ. Let f be a mapping that given a quadruple $(m,\rho,z,w) \in S$, produces such a $j \in [L]$. This means that:

$$\Pr_{\text{RNSY}}[P_{w,f(m,\rho,z,w),\hat{r}^{shared}} \text{ locally computes } m \text{ with error } \delta] \geq \frac{2}{3} - \gamma.$$

Let RNSY' be the experiment in which $(m,\rho,z,w) \leftarrow (\text{RNSY} \,|(m,\rho,z,w) \in S)$. Namely, we choose (m,ρ,z,w) from the experiment RNSY, conditioned on the event that $(m,\rho,z,w) \in S$.

Let E_2 be the experiment in which we choose independently a random string r, $i \leftarrow [k]$ and $(m,\rho,z,w) \leftarrow \text{RNSY}'$. We obtain that:

$$\Pr_{E_2}[\text{Dec}^w(i, f(m,\rho,z,w), \hat{r}^{shared}, r) = m_i] \geq 1 - \delta,$$

since $P_{w,f(m,\rho,z,w),\hat{r}^{shared}}$ computes correctly each coordinate m_i with probability at least $1-\delta$ over the choice of r.

By averaging, there exists a fixed string $\hat{r}$ such that:

$$\Pr_{(m,\rho,z,w)\leftarrow\text{RNSY}',i\leftarrow[k]}[\text{Dec}^w(i, f(m,\rho,z,w), \hat{r}^{shared}, \hat{r}) = m_i] \geq 1 - \delta.$$

By Markov's inequality:

$$\Pr_{(m,\rho,z,w)\leftarrow\text{RNSY}'}\left[\Pr_{i\leftarrow[k]}[\text{Dec}^w(i, f(m,\rho,z,w), \hat{r}^{shared}, \hat{r}) \neq m_i] \geq 10\delta\right] \leq \frac{1}{10}.$$

Let $\overline{\text{Dec}}^w(i,j) = \text{Dec}^w(i,j,\hat{r}^{shared},\hat{r})$. We obtain that:

$$\Pr_{(m,\rho,z,w)\leftarrow\text{RNSY}'}[\Pr_{i\leftarrow[k]}[\overline{\text{Dec}}^w(i, f(m,\rho,z,w)) = m_i] > 1 - 10\delta] > \frac{9}{10}.$$

Which gives that:

$$\Pr_{(m,\rho,z,w)\leftarrow\text{RNSY}}\left[\Pr_{i\leftarrow[k]}[\overline{\text{Dec}}^w(i, f(m,\rho,z,w)) = m_i] > 1 - 10\delta\right] > \left(\frac{2}{3} - \gamma\right) \cdot \frac{9}{10} > 0.55.$$

Let A denote the event:

$$A = \left\{ \Pr_{i \leftarrow [k]} [\overline{\mathrm{Dec}}^{w}(i, f(m, \rho, z, w)) = m_i] > 1 - 10\delta \right\},$$

so that we have that:

$$\Pr_{(m,\rho,z,w) \leftarrow \mathrm{RNSY}}[A] > 0.55.$$

By the definition of the two steps of RNSY, and an averaging argument this gives that:

$$\Pr_{(m,\rho) \leftarrow \mathrm{RNSY}^1} \left[\Pr_{(z,w) \leftarrow \mathrm{RNSY}^2(m,w)} [A] > 0.51] > 0.01 \right].$$

This concludes the proof. □

Applying Lemma 5.6 on the $(\frac{1}{2} - \epsilon, q' = O(q \log k), L, 1/20k)$-LLD that we have previously obtained, gives the following corollary:

Claim 5.7. *There exists an oracle procedure $\overline{Dec}^{(\cdot)}(i, j)$ receiving $i \in [k]$, $j \in [L]$, and making $q' = O(q \cdot \log k)$ queries to a string $w \in \{0,1\}^n$, such that with probability at least 0.51 over choosing $(m, \rho) \leftarrow \mathrm{RNSY}^1$, we have that with probability at least 0.01 over choosing $(z, w) \leftarrow \mathrm{RNSY}^2(m, \rho)$, there exists $j \in [L]$ such that for every $i \leftarrow [k]$, $\overline{Dec}^{w}(i, j) = m_i$.*

Proof. This follows directly from Lemma 5.6, noticing that a statement of the form:

$$\Pr_{i \leftarrow [k]} [\overline{Dec}^{w}(i, j) = m_i] \geq 1 - \frac{1}{2k},$$

implies that for every $i \leftarrow [k]$, $\overline{Dec}^{w}(i, j) = m_i$. □

We can use this (in a similar way to Lemma 4.3) to argue that:

Claim 5.8. *Let $a = \log L + 100$ and let $\ell = \log k$. For every $\alpha \in \{0,1\}^a$, there exists a q-CNF C^α over n variables, such that for $C = \bigvee_{\alpha \in \{0,1\}^a} C^\alpha$, and $p = \frac{\epsilon}{10}$:*

- $\Pr[C(U_n) = 1] \leq 2^{-(2^\ell - a)}$.
- $\Pr_{\rho \leftarrow \mathsf{R}_p^n}[\Pr_{z \leftarrow U_{n \cdot p}^{1/3}}[C(\mathrm{Fill}_\rho(z)) = 1] \geq 0.01] \geq 0.01$.

Proof. The proof is essentially identical to that of Lemma 4.3. More precisely, the message $m \in \{0,1\}^k$ plays the role of $f \in \{0,1\}^{2^\ell}$, the index $i \in [k]$ plays the role of $x \in \{0,1\}^\ell$, and the index $j \in [L]$, plays the role of $\alpha \in \{0,1\}^a$. The argument in the proof of Lemma 4.3 shows that for every $m \in \{0,1\}^k$, there exists a circuit C_m of the required form such that:

- $\Pr_{m \leftarrow \{0,1\}^k, x \leftarrow \{0,1\}^n}[C_m(x) = 1] \leq 2^{-(2^\ell - a)}$.
- With probability at least 0.51 over choosing $(m, \rho) \leftarrow \mathrm{RNSY}^1$, we have that with probability at least 0.01 over choosing $(z, w) \leftarrow \mathrm{RNSY}^2(m, \rho)$, we have that $C_m(x) = 1$, for $x = \mathrm{Fill}_\rho(z)$ (as in experiment RNSY).

By applying Markov's inequality on each one of the two items, we can obtain that:

- $\Pr_{m \leftarrow \{0,1\}^k}\left[\Pr_{x \leftarrow \{0,1\}^n}[C_m(x)=1] \leq 2^{-(2^\ell - a - 100)}\right] \geq 1 - 2^{-100}$.
- With probability at least 0.1 over choosing $m \leftarrow \{0,1\}^k$, we have that with probability at least 0.01 over choosing $\rho \leftarrow \mathsf{R}_p^n$, we have that with probability at least 0.01 over choosing $(z,w) \leftarrow \mathrm{RNSY}^2(m,\rho)$, we have that $C_m(x)=1$, for $x = \mathrm{Fill}_\rho(z)$.

This allows to do a union bound, and obtain that there exists $m \in \{0,1\}^k$ such that setting $C = C_m$, meets the conclusion of Claim 5.8. □

We are finally ready to prove Theorem 5.4. Using Lemma 4.5 we conclude that assuming $a + 100 \leq \frac{2^\ell}{10}$ (which for sufficiently large k, follows by our assumption that $L \leq 2^{k/20}$) we get that $q' \geq \frac{1000}{\epsilon}$ which gives that $q = \Omega(\frac{1}{\epsilon \cdot \log k})$, as required.

6 Conclusion and Open Problems

The most interesting open problem is Open Problem 1.5. We hope that Theorem 1.11 may help to point us to new constructions.

However, it is possible that the answer to Open Problem 1.9 is negative, showing that black-box proofs cannot be used to solve Open Problem 1.5. Is this the case? If it is, can we show this?

Easier problems towards showing a negative answer to Open Problem 1.9 are:

- Is it true that in any black-box hard-function $\Rightarrow$ PRG proof, the number of queries $q \geq m$ or even $q \geq m^2$? A positive answer will show that the cost of the hybrid argument (in terms of the number queries used by the reduction) is unavoidable in black-box hard-function $\Rightarrow$ PRG proofs.
- We don't know whether a super-constant number of queries is necessary for constant ϵ. Can we show a super-constant lower bound on the number of queries of a reduction for a black-box hard-function $\Rightarrow$ PRG proof?
- In fact, we don't even know to show a $q > 1$ lower bound for black-box ρ-hard-function $\Rightarrow$ ϵ-PRG proof for constant ϵ, and $\rho = \frac{1}{2} + \epsilon$. Can we show this?
- In all cases above, the question is open even for small values of a (that do not apply in the extreme high-end).

Another approach to hardness vs. randomness was very recently suggested by Chen and Tell [CT21a]. They use an assumption which is less standard, and incomparable to THE EXTREME HIGH-END HARDNESS ASSUMPTION to construct "target PRGs" which are weaker than PRGs, but suffice for fast derandomization of randomized algorithms. It is interesting to investigate the power of this approach. For more details on this approach and exciting recent developments in this area, see the survey paper [CT23].

Finally, another open problem is to further improve our lower bounds on the number of queries for reductions for black-box hardness amplification. More specifically, our improved lower bounds for hardness amplification apply in the extreme high-end, but are not tight. We only get $q \geq \max(\Omega(\ell), \Omega(\frac{1}{\epsilon}))$, whereas the known upper bounds give $q = O(\frac{\ell}{\epsilon^2})$. Can we prove a matching lower bound that applies in the extreme high-end? (namely, for $a = 2^{(1-o(1))\cdot\ell}$). Such lower bounds were given by [GSV18] for the high-end, namely for $a \leq 2^{\nu \cdot \ell}$ for a constant $\nu > 0$.

Acknowledgements

We are grateful to anonymous referees for very helpful comments and suggestions.

References

[AASY16] B. Applebaum, S. Artemenko, R. Shaltiel, and G. Yang. Incompressible functions, relative-error extractors, and the power of nondeterministic reductions. *Computational Complexity*, 25(2):349–418, 2016.

[ABK+06] E. Allender, H. Buhrman, M. Koucký, D. van Melkebeek, and D. Ronneburger. Power from random strings. *SIAM J. Comput.*, 35(6):1467–1493, 2006.

[AIKS16] S. Artemenko, R. Impagliazzo, V. Kabanets, and R. Shaltiel. Pseudorandomness when the odds are against you. In *31st Conference on Computational Complexity*, pages 9:1–9:35, 2016.

[AS14] 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.

[Bea94] P. Beame. A switching lemma primer. Technical Report UW-CSE-95-07-01, University of Washington, 1994.

[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.

[BM84] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal on Computing*, 13(4):850–864, November 1984.

[BSW03] B. Barak, R. Shaltiel, and A. Wigderson. Computational analogues of entropy. In *RANDOM-APPROX*, pages 200–215, 2003.

[CGI+16] M. L. Carmosino, J. Gao, R. Impagliazzo, I. Mihajlin, R. Paturi, and S. Schneider. Non-deterministic extensions of the strong exponential time hypothesis and consequences for non-reducibility. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, ITCS*, pages 261–270, 2016.

[CIKK16] M. L. Carmosino, R. Impagliazzo, V. Kabanets, and A. Kolokolova. Learning algorithms from natural proofs. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC*, volume 50 of *LIPIcs*, pages 10:1–10:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[CT21a] L. Chen and R. Tell. Hardness vs randomness, revised: Uniform, non-black-box, and instance-wise. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 125–136, 2021.

[CT21b] L. Chen and R. Tell. Simple and fast derandomization from very hard functions: eliminating randomness at almost no cost. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 283–291. ACM, 2021.

[CT23] L. Chen and R. Tell. New ways of studying the BPP = P conjecture. *Electron. Colloquium Comput. Complex.*, 2023.

[DMOZ20] D. Doron, D. Moshkovitz, J. Oh, and D. Zuckerman. Nearly optimal pseudorandomness from hardness. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 1057–1068. IEEE, 2020.

[GM84] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, April 1984. Preliminary version appeared in STOC' 82.

[GSV18] A. Grinberg, R. Shaltiel, and E. Viola. Indistinguishability by adaptive procedures with advice, and lower bounds on hardness amplification proofs. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 956–966, 2018.

[Gur06] V. Guruswami. Algorithmic results in list decoding. *Foundations and Trends in Theoretical Computer Science*, 2(2), 2006.

[GW14] O. Goldreich and A. Wigderson. On derandomizing algorithms that err extremely rarely. In *Symposium on Theory of Computing, STOC*, pages 109–118. ACM, 2014.

[Hås86] J. Håstad. Almost optimal lower bounds for small depth circuits. In *STOC*, pages 6–20, 1986.

[HILL99] 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.

[Hir18] S. Hirahara. Non-black-box worst-case to average-case reductions within NP. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 247–258, 2018.

[Imp95] R. Impagliazzo. Hard-core distributions for somewhat hard problems. In *FOCS*, pages 538–545, 1995.

[ISW99] R. Impagliazzo, R. Shaltiel, and A. Wigderson. Near-optimal conversion of hardness into pseudo-randomness. In *FOCS*, pages 181–190, 1999.

[ISW06] R. Impagliazzo, R. Shaltiel, and A. Wigderson. Reducing the seed length in the nisan-wigderson generator. *Combinatorica*, 26(6):647–681, 2006.

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

[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.

[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.

[RRV02] R. Raz, O. Reingold, and S. P. Vadhan. Extracting all the randomness and reducing the error in trevisan's extractors. *J. Comput. Syst. Sci.*, 65(1):97–128, 2002.

[RSV21] N. Ron-Zewi, R. Shaltiel, and Nithin Varma. Query complexity lower bounds for local list-decoding and hard-core predicates (even for small rate and huge lists). In *12th Innovations in Theoretical Computer Science Conference, ITCS*, volume 185 of *LIPIcs*, pages 33:1–33:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[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.

[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 pseudo-random 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.

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

[Tel21] R. Tell. How to find water in the ocean: A survey on quantified derandomization. *Electron. Colloquium Comput. Complex.*, page 120, 2021.

[Tre01] L. Trevisan. Extractors and pseudorandom generators. *Journal of the ACM*, 48, 2001.

[TZ04] A. Ta-Shma and D. Zuckerman. Extractor codes. *IEEE Trans. Information Theory*, 50(12):3015–3025, 2004.

[Uma03] C. Umans. Pseudo-random generators for all hardnesses. *Journal of Computer and System Sciences*, 67:419–440, 2003.

[Uma09] C. Umans. Reconstructive dispersers and hitting set generators. *Algorithmica*, 55(1):134–156, 2009.

[Vio05] E. Viola. The complexity of constructing pseudorandom generators from hard functions. *Computational Complexity*, 13(3-4):147–188, 2005.

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

[Yao82] A. C. Yao. Theory and applications of trapdoor functions (extended abstract). In *FOCS*, pages 80–91, 1982.

[Yek12] S. Yekhanin. Locally decodable codes. *Found. Trends Theor. Comput. Sci.*, 6(3):139–255, 2012.