Query complexity lower bounds for local list-decoding and hard-core predicates (even for small rate and huge lists)

Noga Ron-Zewi* Ronen Shaltiel† Nithin Varma‡

University of Haifa, Israel

Abstract

A binary code $\mathrm{Enc} : \{0,1\}^k \to \{0,1\}^n$ is $(\frac{1}{2}-\epsilon, L)$-*list decodable* if for every $w \in \{0,1\}^n$, there exists a set $\mathrm{List}(w)$ of size at most L, containing all messages $m \in \{0,1\}^k$ such that the relative Hamming distance between $\mathrm{Enc}(m)$ and w is at most $\frac{1}{2}-\epsilon$. A q-query *local list-decoder* is a randomized procedure that when given oracle access to a string w, makes at most q oracle calls, and for every message $m \in \mathrm{List}(w)$, with high probability, there exists $j \in [L]$ such that for every $i \in [k]$, with high probability, $\mathrm{Dec}^w(i,j) = m_i$.

We prove lower bounds on q, that apply even if L is huge (say $L = 2^{k^{0.9}}$) and the rate of Enc is small (meaning that $n \geq 2^k$):

- For $\epsilon = 1/k^\nu$ for some constant $\nu > 0$, we prove a lower bound of $q = \Omega(\frac{\log(1/\delta)}{\epsilon^2})$, where δ is the error probability of the local list-decoder. This bound is tight as there is a matching upper bound by Goldreich and Levin (STOC 1989) of $q = O(\frac{\log(1/\delta)}{\epsilon^2})$ for the Hadamard code (which has $n = 2^k$). This bound extends an earlier work of Grinberg, Shaltiel and Viola (FOCS 2018) which only works if $n \leq 2^{k^\nu}$ and the number of coins tossed by Dec is small (and therefore does not apply to the Hadamard code, or other codes with low rate).
- For smaller ϵ, we prove a lower bound of roughly $q = \Omega(\frac{1}{\sqrt{\epsilon}})$. To the best of our knowledge, this is the first lower bound on the number of queries of local list-decoders that gives $q \geq k$ for small ϵ.

Local list-decoders with small ϵ form the key component in the celebrated theorem of Goldreich and Levin that extracts a *hard-core predicate* from a one-way function. We show that black-box proofs *cannot* improve the Goldreich-Levin theorem and produce a hard-core predicate that is hard to predict with probability $\frac{1}{2} + \frac{1}{\ell^{\omega(1)}}$ when provided with a one-way function $f : \{0,1\}^\ell \to \{0,1\}^\ell$, such that circuits of size $\mathrm{poly}(\ell)$ cannot invert f with probability $\rho = 1/2^{\sqrt{\ell}}$ (or even $\rho = 1/2^{\Omega(\ell)}$). This limitation applies to any proof by black-box reduction (even if the reduction is allowed to use nonuniformity and has oracle access to f).

*noga@cs.haifa.ac.il

†ronen@cs.haifa.ac.il

‡nvarma@bu.edu

1 Introduction

We prove limitations on local list-decoding algorithms and on reductions establishing hard-core predicates.

1.1 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 e.g., [Gur06] for a survey. In this paper, we will be interested in list-decoding of binary codes.

Definition 1.1 (List-decodable code)**.** *For a function* $\mathrm{Enc} : \{0,1\}^k \to \{0,1\}^n$*, and* $w \in \{0,1\}^n$*, we define*

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

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). 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 1.2 (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 1.3 (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.*

[1] 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, $\mathsf{dist}(x,y) = |\,\{i \in [n] : x_i \neq y_i\}\,|/n$.

- $j \in [L]$ *: 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* $m \in \text{List}_\alpha^{\text{Enc}}(w)$*, with probability at least* $2/3$ *over choosing a uniform string* r^{shared}*, there exists* $j \in [L]$ *such that the procedure*

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

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

Remark 1.4 (On the generality of Definition 1.3)**.** *The goal of this paper is to prove lower bounds on local list-decoders, and so, making local list-decoders as general as possible, makes our results stronger. We now comment on the generality of Definition 1.3.*

- *In Definition 1.3 we do not require that* $L = |\text{List}_\alpha^{\text{Enc}}(w)|$*, and allow the local list-decoder to use a larger* L*. This means that on a given* w*, there may be many choices of* $j \in [L]$ *such that the procedure* $P_{w,j,r^{\text{shared}}}(i,r) = \text{Dec}^w(i,j,r^{\text{shared}},r)$ *locally computes messages* $m \notin \text{List}_\alpha^{\text{Enc}}(w)$*.*
- *In Definition 1.3 we do not place any restriction on the number of random coins used by the local list-decoder, making the task of local list-decoding easier.*
- *We allow* Dec *to make* adaptive *queries to its oracle.*
- *We are only interested in the total number of queries made by the combination of the two steps. It should be noted that w.l.o.g., a local list-decoder can defer all its queries to the second step (namely, after it receives the input* i*), and so, this definition captures local list-decoding algorithms which make queries to the oracle at both steps.*
- *To the best of our knowledge, all known local list-decoders in the literature are of the form presented in Definition 1.3.*

1.1.1 Lower bounds on the query complexity of local list-decoders

In this paper we prove lower bounds on the number of queries q of $(\frac{1}{2}-\epsilon, L, q, \delta)$-local list-decoders. Our goal is to show that the number of queries q has to be large, when ϵ is *small*. Our lower bounds apply even if the size of the list L is huge and approaches 2^k (note that a local list-decoder can trivially achieve $L = 2^k$ with a list of all messages). Our lower bounds also apply even if the rate of the code is very small, and $n \geq 2^k$.

We remark that this parameter regime is very different than the one studied in lower bounds on the number of queries of local decoders for *uniquely* decodable codes (that is, for $L = 1$). By the Plotkin bound, uniquely decodable codes cannot have $\epsilon < \frac{1}{4}$, and so, the main focus in uniquely decodable codes is to show that local decoders for codes with "good rate" and "large" $\epsilon = \Omega(1)$, must make many queries. In contrast, we are interested in the case where ϵ is small, and want to prove lower bounds that apply to huge lists and small rate.

Lower bounds for $\epsilon \geq 1/k^{\Omega(1)}$**.** Our first result is a tight lower bound of $q = \Omega(\frac{\log(1/\delta)}{\epsilon^2})$ on the number of queries, assuming ϵ is sufficiently large compared to $1/k$.

Theorem 1.5 (Tight lower bounds for $\epsilon \geq 1/k^{\Omega(1)}$)**.** *There exists a universal constant* $\nu > 0$ *such that for any* $L \leq 2^{k^{0.9}}$*,* $\epsilon \in (k^{-\nu}, \frac{1}{4})$*, and* $\delta \in (k^{-\nu}, \frac{1}{3})$*, we have that every* $(\frac{1}{2}-\epsilon, L, q, \delta)$*-local list-decoder for* $\text{Enc} : \{0,1\}^k \to \{0,1\}^n$ *must have* $q = \Omega(\frac{\log(1/\delta)}{\epsilon^2})$*.*

Theorem 1.5 is tight in the sense that the Hadamard code (which has length $n = 2^k$) has $(\frac{1}{2} - \epsilon, L = O(1/\epsilon^2), q = O(\frac{\log(1/\delta)}{\epsilon^2}), \delta)$ local list-decoders [GL89]. In fact, the Hadamard code was the motivation for this research, and is a running example in this paper.

Our results show that even if we allow list sizes L which approach 2^k, it is impossible to reduce the number of queries for the Hadamard code. Our results also show, that even if we are willing to allow very small rate $n \geq 2^k$, and huge list sizes, it is impossible to have codes whose local list-decoders make fewer queries than the local list-decoders for the Hadamard code.

Comparison to previous work. Theorem 1.5 improves and extends an earlier work by Grinberg, Shaltiel and Viola [GSV18] that gave the same bound of $q = \Omega(\frac{\log(1/\delta)}{\epsilon^2})$ for a more limited parameter regime: Specifically, in [GSV18], for the lower bound to hold, it is also required that $n \leq 2^{k^\nu}$, for some constant $\nu > 0$, and that the total number of coins tossed by the local list-decoder is less than $k^\nu - \log L$.[2] We stress that because of these two limitations, the lower bounds of [GSV18] do not apply to the Hadamard code and other low rate codes.

Extensions to large alphabet and erasures. The scenario that we consider in Theorem 1.5 has *binary* alphabet, and decoding from *errors*. We remark that in the case of large alphabets, or decoding from erasures, there are local list-decoders which achieve $q = O(\frac{\log(1/\delta)}{\epsilon})$ (which is smaller than what is possible for binary alphabet and decoding from errors), as was shown for the case of Hadamard codes in [RRZV18]. Our results extend to give a matching lower bound of $q = \Omega(\frac{\log(1/\delta)}{\epsilon})$ for decoding from erasures (for any alphabet size), and also the same lower bound on decoding from errors for any alphabet size. The exact details are deferred to the final version.

Lower bounds for $\epsilon < 1/k$. The best bound on q that Theorem 1.5 (as well as the aforementioned lower bounds of [GSV18]) can give is $q \geq k^{\Omega(1)}$. The next theorem shows that even for small $\epsilon \ll 1/k$, we can obtain a lower bound on q which is polynomial in $1/\epsilon$.

Theorem 1.6 (Lower bounds for small ϵ)**.** *There exist universal constants $\beta, c_1, c_2 > 0$ such that for every $L \leq \beta \cdot 2^k$, $\delta < \frac{1}{3}$ and $\epsilon \geq \frac{\beta}{\sqrt{n}}$ we have that every $(\frac{1}{2} - \epsilon, L, q, \delta)$-local list-decoder for* $\mathrm{Enc} : \{0,1\}^k \to \{0,1\}^n$ *must have $q \geq \frac{1}{c_1 \log(k) \cdot \sqrt{\epsilon}} - c_2 \log L$.*

Note that for sufficiently small $\epsilon = 1/(\log k)^{\omega(1)}$, we get $q = \Omega(\frac{1}{\epsilon^{1/2-o(1)}})$. It follows that together, Theorems 1.5 and Theorem 1.6 give a lower bound of $q = \Omega(\frac{1}{\epsilon^{1/2-o(1)}})$ that applies to every choice of $\epsilon \geq \Omega(\frac{1}{\sqrt{n}})$. To the best of our knowledge, Theorem 1.6 is the first lower bound on local list-decoders that is able to prove a lower bound of $q \geq k$ (and note that this is what we should expect when $\epsilon < \frac{1}{k}$). We also remark that the requirement that ϵ is not too small compared to n (as is made in Theorem 1.6) is required (as we cannot prove lower bounds on the number of queries in case $\epsilon < \frac{1}{n}$).

Goldreich and Levin [GL89] showed that locally list-decodable codes with small $\epsilon \ll 1/k$ can be used to give constructions of hard-core predicates. We explain this connection in the next section.

1.2 Hard-core predicates

The celebrated Goldreich-Levin theorem [GL89] considers the following scenario: There is a computational task where the required output is *non-Boolean* and is hard to compute on average. We would like to obtain a

[2] The work of [GSV18] is concerned with proving lower bounds on the number of queries of "nonuniform reductions for hardness amplification" [Vio06, SV10, AS11, GSV18]. As explained in [Vio06, SV10, AS11, GSV18] such lower bounds translate into lower bounds on local list-decoders, by "trading" the random coins of a local list-decoder for "nonuniform advice" for the reduction, and proving a lower bound on the number of queries made by the reduction.

hard-core predicate, which is a *Boolean* value that is hard to compute on average.

The Goldreich-Levin theorem gives a solution to this problem, and in retrospect, the theorem can also be viewed as a $(\frac{1}{2}-\epsilon, L^{\mathrm{Had}}=O(\frac{1}{\epsilon^2}), q^{\mathrm{Had}}=O(\frac{k}{\epsilon^2}), \delta=\frac{1}{2k})$-local list-decoder for the Hadamard code, defined by: $\mathrm{Enc}^{\mathrm{Had}}:\{0,1\}^k \to \{0,1\}^{n=2^k}$, where for every $r \in \{0,1\}^k$,

$$\mathrm{Enc}^{\mathrm{Had}}(x)_r = \left(\sum_{i\in[k]} x_i \cdot r_i\right) \mod 2.$$

In retrospect, the Goldreich-Levin theorem can also be seen as showing that *any* locally list-decodable code with suitable parameters can be used to produce hard-core predicates.

We consider two scenarios for the Goldreich-Levin theorem depending on whether we want to extract a hard-core bit from a function $g:\{0,1\}^\ell \to \{0,1\}^\ell$ that is *hard to compute* on a random input, or to extract a hard-core bit from a one-way function $f:\{0,1\}^\ell \to \{0,1\}^\ell$ that is *hard to invert* on a random output.

1.2.1 Functions that are hard to compute

Here the goal is to transform a *non-Boolean function* g that is hard to compute on a random input, into a *predicate* g^{pred} that is hard to compute on a random input. More precisely:

- *Assumption:* There is a non-Boolean function that is hard to compute with probability ρ.

  Namely, a function $g:\{0,1\}^\ell \to \{0,1\}^\ell$ such that for every circuit C of size s,

  $$\Pr_{x\leftarrow U_\ell}[C(x)=g(x)] \le \rho.^3$$

- *Conclusion:* There is a predicate $g^{\mathrm{pred}}:\{0,1\}^{\ell'} \to \{0,1\}$ that is hard to compute with probability $\frac{1}{2}+\epsilon$.

  Namely, for every circuit C' of size s',

  $$\Pr_{x\leftarrow U_{\ell'}}[C'(x)=g^{\mathrm{pred}}(x)] \le \frac{1}{2}+\epsilon.$$

- *Requirements:* The goal is to show that for every g, there exists a function g^{pred} with as small an ϵ as possible, without significant losses in the other parameters (meaning that s' is not much smaller than s, and ℓ' is not much larger than ℓ).

The Goldreich-Levin theorem for this setting can be expressed as follows.

Theorem 1.7 (Goldreich-Levin for functions that are hard to compute [GL89])**.** *For every function* $g:\{0,1\}^\ell \to \{0,1\}^\ell$*, define* $g^{\mathrm{pred}}:\{0,1\}^{\ell'=2\ell} \to \{0,1\}$ *by* $g^{\mathrm{pred}}(x,r)=\mathrm{Enc}^{\mathrm{Had}}(x)_r$*, and* $\rho=\frac{\epsilon}{2\cdot L^{\mathrm{Had}}}=\mathrm{poly}(\epsilon)$*. If for every circuit* C *of size* s*,*

$$\Pr_{x\leftarrow U_\ell}[C(x)=g(x)] \le \rho,$$

then for every circuit C' *of size* $s'=\frac{s}{q^{\mathrm{Had}}\cdot\mathrm{poly}(\ell)}=s\cdot\mathrm{poly}(\frac{\epsilon}{\ell})$*,*

$$\Pr_{x\leftarrow U_{2\ell}}[C'(x)=g^{\mathrm{pred}}(x)] \le \frac{1}{2}+\epsilon.$$

[3] We use U_ℓ to denote the uniform distribution on ℓ bits.

The Hadamard code can be replaced by any locally list-decodable code with list size L for decoding from radius $\frac{1}{2} - \epsilon$, with q queries for $\delta = 1/(2k)$. For such a code (assuming also that the local list-decoder can be computed efficiently) one gets the same behavior. Specifically, if the initial function is sufficiently hard and $\rho = \frac{\epsilon}{2L}$, then the Boolean target function is hard to compute, up to $\frac{1}{2} + \epsilon$ for circuits of size roughly $s' = s/q$.

Is it possible to improve the Goldreich-Levin theorem for $\rho \ll 1/s$? Suppose that we are given a function $g : \{0,1\}^\ell \to \{0,1\}^\ell$ that is hard to compute for circuits of size $s = \mathrm{poly}(\ell)$, with success say $\rho = 1/2^{\sqrt{\ell}}$. When applying Theorem 1.7, we gain nothing compared to the case that $\rho = 1/\mathrm{poly}(\ell)$. In both cases, we can obtain $\epsilon = 1/\mathrm{poly}(\ell)$, but not smaller! (Since otherwise $s' = s \cdot \mathrm{poly}(\epsilon/\ell)$ is smaller than 1 and the result is meaningless).

This is disappointing, as we may have expected to obtain $\epsilon \approx \rho = 1/2^{\sqrt{\ell}}$, or at least, to gain over the much weaker assumption that $\rho = 1/\mathrm{poly}(\ell)$. This leads to the following open problem:

Open problem 1.8 (Improve Goldreich-Levin for functions that are hard to compute)**.** *Suppose we are given a function $g : \{0,1\}^\ell \to \{0,1\}^\ell$ such that circuits C of size $s = \mathrm{poly}(\ell)$ cannot compute g with success $\rho = 1/2^{\sqrt{\ell}}$. Is it possible to convert g into a predicate with hardness $\frac{1}{2} + \epsilon$ for $\epsilon = 1/\ell^{\omega(1)}$?*

This is not possible to achieve using the Hadamard code, because the number of queries is $q \geq 1/\epsilon$, and Theorem 1.7 requires $s \geq s' \cdot q \geq q \geq 1/\epsilon$, which dictates that $\epsilon \geq 1/s$.

Note that when ρ is small, we can afford list-decodable codes with huge list sizes of $L \approx 1/\rho$. Motivated by this observation, we can ask the following question:

> Is it possible to solve this open problem by substituting the Hadamard code with a better code? Specifically, is it possible for local list-decoders to have $q = \frac{1}{\epsilon^{o(1)}}$ if allowed to use *huge* lists of size say $2^{\sqrt{k}}$, approaching the trivial bound 2^k? (Note that in the Hadamard code, the list size used is $\mathrm{poly}(1/\epsilon) = \mathrm{poly}(k)$ which is exponentially smaller).

We show, in Theorem 1.6, that it is impossible to solve the open problem by replacing the Hadamard code with a different locally list-decodable code.

The natural next question is whether we can use other techniques (not necessarily local list-decoding) to achieve the goal stated above. In this paper, we show that this cannot be done by *black-box techniques*:

Informal Theorem 1.9 (Black-box impossibility result for functions that are hard to compute)**.** *If $\rho \geq \frac{1}{2^{\ell/3}}$, $s = 2^{o(\ell)}$ is larger than some fixed polynomial in ℓ, and $\epsilon = \frac{1}{s^{\omega(1)}}$, then there does not exist a map that converts a function g into a function g^{pred} together with a black-box reduction showing that g^{pred} is a hard-core predicate for g.*

The parameters achieved in Theorem 1.9 rule out black-box proofs in which $\epsilon = \frac{1}{s^{\omega(1)}}$, not only for $s = \mathrm{poly}(\ell)$ and $\rho = 2^{-\sqrt{\ell}}$ (as in Open problem 1.8) but also for $\rho = 2^{-\Omega(\ell)}$, and allowing much larger s as long as $s = 2^{o(\ell)}$.

The precise statement of Theorem 1.9 is stated in Theorem 4.2, and the precise model is explained in Section 4.1.

To the best of our knowledge, this is the first result of this kind, that shows black-box impossibility results for Open problem 1.8. Moreover, we believe that the model that we introduce in Section 4.1 is very general and captures all known black-box techniques. In particular, our model (which we view as a conceptual contribution) allows the reduction to introduce nonuniformity when converting an adversary C' that breaks g^{pred} into an adversary C that breaks g.

1.2.2 Functions that are hard to invert

Here the goal is to transform a one-way function f into a new one-way function f^{newOWF} and a *predicate* f^{pred} such that it is hard to compute $f^{\text{pred}}(x)$ given $f^{\text{newOWF}}(x)$. More precisely:

- *Assumption:* There is a one-way function that is hard to invert with probability ρ.

  Namely, a function $f : \{0,1\}^\ell \to \{0,1\}^\ell$ such that for every circuit C of size s,

  $$\Pr_{x \leftarrow U_\ell}[C(f(x)) \in f^{-1}(f(x))] \leq \rho.$$

- *Conclusion:* There is a one-way function $f^{\text{newOWF}} : \{0,1\}^{\ell'} \to \{0,1\}^{\ell'}$, and a predicate $f^{\text{pred}} : \{0,1\}^{\ell'} \to \{0,1\}$, such that it is hard to predict $f^{\text{pred}}(x)$ with advantage $\frac{1}{2} + \epsilon$, when given access to $f^{\text{newOWF}}(x)$.

  Namely, for every circuit C' of size s',

  $$\Pr_{x \leftarrow U_{\ell'}}[C'(f^{\text{newOWF}}(x)) = f^{\text{pred}}(x)] \leq \frac{1}{2} + \epsilon.$$

- The goal is to show that for every f, there exist functions $f^{\text{newOWF}}, f^{\text{pred}}$ with as small ϵ as possible, without significant losses in the other parameters (meaning that: s' is not much smaller than s, and ℓ' is not much larger than ℓ).

The Goldreich-Levin theorem for this setting can be expressed as follows.

Theorem 1.10 (Goldreich-Levin for functions that are hard to invert)**.** *For a function $f : \{0,1\}^\ell \to \{0,1\}^\ell$, define $f^{\text{newOWF}} : \{0,1\}^{2\ell} \to \{0,1\}^{2\ell}$ by $f^{\text{newOWF}}(x,r) = (f(x), r)$, $f^{\text{pred}} : \{0,1\}^{2\ell} \to \{0,1\}$ by $f^{\text{pred}}(x,r) = \text{Enc}^{\text{Had}}(x)_r$, and $\rho = \frac{\epsilon}{2 \cdot L^{\text{Had}}} = \text{poly}(\epsilon)$. If for every circuit C of size s,*

$$\Pr_{x \leftarrow U_\ell}[C(f(x)) \in f^{-1}(f(x))] \leq \rho,$$

then for every circuit C' of size $s' = \frac{s}{q^{\text{Had}} \cdot \text{poly}(\ell)} = s \cdot \text{poly}(\frac{\epsilon}{\ell})$,

$$\Pr_{x \leftarrow U_{2\ell}}[C'((f^{\text{newOWF}}(x))) = f^{\text{pred}}(x)] \leq \frac{1}{2} + \epsilon.$$

Remark 1.11. *The problem of obtaining a hard-core predicate for one-way functions, is interesting only if an unbounded adversary $\phi : \{0,1\}^{\ell'} \to \{0,1\}$ can predict $f^{\text{pred}}(x)$ when given $f^{\text{newOWF}}(x)$ as input. If this is not required, then one can take $\ell' = \ell + 1$, $f^{\text{pred}}(x) = x_1$, and $f^{\text{newOWF}}(x_1, \ldots, x_{n+1}) = f(x_2, \ldots, x_{n+1})$. However, this is trivial, and is not useful in applications. Therefore, when considering this problem, we will assume that there exists such a $\phi : \{0,1\}^{\ell'} \to \{0,1\}$.*

A natural example is the case where the original one-way function f and the constructed function f^{newOWF} are one-way permutations. In fact, in the case that f, $f^{\text{newOWF}}(x)$ are permutations, the setup of "functions that are hard to invert" can be seen as a special case of the setup of functions that are "hard to compute" by taking $g = f^{-1}$, and $g^{\text{pred}}(y) = f^{\text{pred}}((f^{\text{newOWF}})^{-1}(y))$.

We point out that, in this setting, the circuit C' that is trying to invert f (that is, to compute g) has an advantage over its counterpart in the setup of "functions that are hard to compute": It can use the efficient algorithm that computes the "forward direction" of f, when trying to invert f. In terms of g, this means that the circuit C' can compute g^{-1} for free. This distinction is explained in Section 4.2.

Is it possible to improve the Goldreich-Levin theorem for $\rho \ll 1/s$? The same problem that we saw with functions that are hard to compute, also shows up in the setup of functions that are hard to invert. Suppose that we are given a function $f : \{0,1\}^\ell \to \{0,1\}^\ell$ that is hard to invert for circuits of size $s = \mathrm{poly}(\ell)$ with success, say, $\rho = 1/2^{\sqrt{\ell}}$. When applying Theorem 1.10, we gain nothing compared to the case that $\rho = 1/\mathrm{poly}(\ell)$. In both cases, we can obtain $\epsilon = 1/\mathrm{poly}(\ell)$, but not smaller! This is expressed in the next open problem:

Open problem 1.12 (Improve Goldreich-Levin for functions that are hard to invert)**.** *If we are given a one-way function $f : \{0,1\}^\ell \to \{0,1\}^\ell$ such that circuits C of size $s = \mathrm{poly}(\ell)$ cannot invert f with success $\rho = 1/2^{\sqrt{\ell}}$. Is it possible to obtain a hard-core predicate f^{pred} with hardness $\frac{1}{2} + \epsilon$ for $\epsilon = 1/\ell^{\omega(1)}$ for some choice of one-way function f^{newOWF}?*

In this paper, we show that this cannot be done by *black-box techniques*. The formulation of Theorem 1.13 below, is very similar to that of Theorem 1.9 with some small modification in the parameters.

Informal Theorem 1.13 (Black-box impossibility result for functions that are hard to invert)**.** *If $\rho \geq 2^{-\ell/5}$, $s = 2^{o(\ell)}$ is larger than some fixed polynomial in ℓ, and $\epsilon = \frac{1}{s^{\omega(1)}}$ then there does not exist a map that converts a function f into functions $f^{\mathrm{newOWF}}, f^{\mathrm{pred}}$ together with a black-box reduction showing that f^{pred} is a hard-core predicate for f^{newOWF}.*

The precise statement of Theorem 1.13 is stated in Theorem 4.12, and the precise model is explained in Section 4.2.

To the best of our knowledge, this is the first result of this kind, that shows black-box impossibility results for open problem 1.12. Moreover, we believe that the model that we introduce in Section 4.2 is very general, and captures all known black-box techniques. In particular, our model (which we view as a conceptual contribution) allows the reduction to compute the easy direction of the function f, and to introduce nonuniformity when converting an adversary C' that breaks f^{pred} into an adversary C that breaks f.

1.3 Techniques

Our approach builds on earlier work for proving lower bounds on the number of queries of reductions for hardness amplification [Vio06, SV10, GSV18]. In this section, we give a high level overview of the arguments used to prove our main theorems.

1.3.1 Local list-decoders on random noisy codewords

Following [Vio06, SV10, GSV18], we will consider a scenario which we refer to as "random noisy codewords" in which a uniformly chosen message m is encoded, and the encoding is corrupted by a binary symmetric channel.

Definition 1.14 (Binary symmetric channels)**.** *Let BSC_p^n be the experiment in which a string $Z \in \{0,1\}^n$ is sampled, where $Z = Z_1, \ldots, Z_n$ is composed of i.i.d. bits, such that for every $i \in [n]$, $\Pr[Z_i = 1] = p$.*

Definition 1.15 (Random noisy codewords)**.** *Given a function $\mathrm{Enc} : \{0,1\}^k \to \{0,1\}^n$ and $p > 0$ we consider the following experiment (which we denote by $\mathrm{RNSY}_p^{\mathrm{Enc}}$):*

- *A message $m \leftarrow \{0,1\}^k$ is chosen uniformly.*
- *A noise string $z \leftarrow \mathrm{BSC}_p^n$ is chosen from a binary symmetric channel.*
- *We define $w = \mathrm{Enc}(m) \oplus z$.*

We use $(m, z, w) \leftarrow \mathrm{RNSY}_p^{\mathrm{Enc}}$ *to denote* m, z, w *which are sampled by this experiment. We omit* Enc *if it is clear from the context.*

Our goal is to prove lower bounds on the number of queries q of a $(\frac{1}{2} - \epsilon, L, q, \delta)$-local list-decoder Dec for a code $\mathrm{Enc} : \{0,1\}^k \to \{0,1\}^n$. For this purpose, we will consider the experiment RNSY_p for the values $p = \frac{1}{2} - 2\epsilon$ and $p = \frac{1}{2}$.

For $p = \frac{1}{2} - 2\epsilon$, and $(m, z, w) \leftarrow \mathrm{RNSY}_{\frac{1}{2}-\epsilon}$, by a Chernoff bound, the Hamming weight of z is, with very high probability, less than $\frac{1}{2} - \epsilon$. This implies that $\mathsf{dist}(w, \mathrm{Enc}(m)) \leq \frac{1}{2} - \epsilon$, meaning that $m \in \mathrm{List}^{\mathrm{Enc}}_{\frac{1}{2}-\epsilon}(w)$. It follows that there must exist $j \in [L]$ such that when given input j, and oracle access to w, Dec recovers the message m.

For $p = \frac{1}{2}$, and $(m, z, w) \leftarrow \mathrm{RNSY}_{\frac{1}{2}}$, the string z is uniformly distributed and independent of m. This means that $w = \mathrm{Enc}(m) \oplus z$ is uniformly distributed and independent of m. Consequently, when Dec is given oracle access to w, there is no information in w about the message m, and so, for every $j \in [L]$, the probability that Dec recovers m when given input j and oracle access to w is exponentially small.

Loosely speaking, this means that Dec can be used to distinguish $\mathrm{BSC}^n_{\frac{1}{2}-2\epsilon}$ from $\mathrm{BSC}^n_{\frac{1}{2}}$. It is known that distinguishing these two distributions requires many queries. We state this informally below, and a formal statement appears in Lemma 2.2.

Informal Theorem 1.16. *Any function* $T : \{0,1\}^q \to \{0,1\}$ *that distinguishes* $\mathrm{BSC}^q_{\frac{1}{2}-2\epsilon}$ *from* $\mathrm{BSC}^q_{\frac{1}{2}}$ *with advantage* δ*, must have* $q = \Omega(\frac{\log(1/\delta)}{\epsilon^2})$.

Thus, in order to prove a tight lower bound of $q = \Omega(\frac{\log(1/\delta)}{\epsilon^2})$, it is sufficient to show how to convert a $(\frac{1}{2} - \epsilon, L, q, \delta)$-local list-decoder Dec, into a function T that distinguishes $\mathrm{BSC}^q_{\frac{1}{2}-2\epsilon}$ from $\mathrm{BSC}^q_{\frac{1}{2}}$ with advantage δ. Note that we can allow T to be a "randomized procedure" that tosses coins, as by an averaging argument, such a randomized procedure can be turned into a deterministic procedure.

1.3.2 Warmup: the case of unique decoding

Let us consider the case that $L = 1$ (that is unique decoding). We stress that this case is uninteresting, as by the Plotkin bound, it is impossible for nontrivial codes to be uniquely decodable for $\epsilon < \frac{1}{4}$, and so, there are no local decoders for $L = 1$ and $\epsilon < \frac{1}{4}$, regardless of the number of queries. Nevertheless, this case will serve as a warmup for the approach we use later.

Our goal is to convert Dec into a randomized procedure $T : \{0,1\}^q \to \{0,1\}$ that distinguishes $\mathrm{BSC}^q_{\frac{1}{2}-2\epsilon}$ from $\mathrm{BSC}^q_{\frac{1}{2}}$. The procedure T will work as follows: On input $x \leftarrow \{0,1\}^q$, we choose $m \leftarrow \{0,1\}^k$, and $i \leftarrow [k]$. We then run Dec on input i, and when Dec makes its t'th query $\ell_t \in [n]$ to the oracle, we answer it by $\mathrm{Enc}(m)_{\ell_t} \oplus x_t$. That is, we answer as if Dec is run with input i and oracle access to $w = \mathrm{Enc}(m) \oplus z$, for z chosen from a binary symmetric channel. The final output of T is whether Dec reproduced m_i. This procedure T simulates $\mathrm{Dec}^w(i)$, and therefore distinguishes $\mathrm{BSC}^q_{\frac{1}{2}-2\epsilon}$ from $\mathrm{BSC}^q_{\frac{1}{2}}$, implying the desired lower bound.

Both Theorem 1.5 and Theorem 1.6 will follow by modifying the basic approach to handle $L > 1$. In the remainder of this section, we give a high level overview of the methods that we use. The formal section of this paper does not build on this high level overview, and readers can skip this high level overview and go directly to the formal section if they wish.

1.3.3 Reducing to the coin problem for AC^0

We start with explaining the approach of proving Theorem 1.6. Consider a randomized procedure C that on input $z \in \{0,1\}^n$, chooses $m \leftarrow \{0,1\}^k$ and prepares $w = \mathrm{Enc}(m) \oplus z$. The procedure then computes $\mathrm{Dec}^w(i,j)$ for all choices of $i \in [k]$ and $j \in [L]$ and accepts if there exists a $j \in [L]$ such that with index j, Dec^w recovers m. By the same rationale as in Section 1.3.2, C distinguishes $\mathrm{BSC}^n_{\frac{1}{2}-2\epsilon}$ from $\mathrm{BSC}^n_{\frac{1}{2}}$. This does not seem helpful, because C receives n input bits, and we cannot use Theorem 1.16 to get a lower bound on q.

Inspired by a lower bound on the size of nondeterministic reductions for hardness amplification due to Applebaum et al. [AASY16], we make the following observation: The procedure C can be seen as $k \cdot L$ computations (one for each choice of $i \in [k]$ and $j \in [L]$) such that:

- These $k \cdot L$ computations can be run *in parallel*.
- Once these computations are made, the final answer $C(z)$ is computed by a constant-depth circuit.
- Each of the $k \cdot L$ computations makes q queries into z, and therefore can be simulated by a size $O(q \cdot 2^q)$ circuit of depth 2.

Overall, this means that we can implement C by a circuit of size $s = \mathrm{poly}(k, L, 2^q)$ and constant depth. (In fact, a careful implementation gives depth 3).

This is useful because there are lower bounds showing that small constant-depth circuits cannot solve the "coin problem". Specifically, by the results of Cohen, Ganor and Raz [CGR14] circuits of size s and depth d cannot distinguish $\mathrm{BSC}^n_{\frac{1}{2}-2\epsilon}$ from $\mathrm{BSC}^n_{\frac{1}{2}}$ with constant advantage, unless $s \geq \exp(\Omega(\frac{1}{\epsilon^{d-1}}))$.[4] This gives the bound stated in Theorem 1.6.

We find it surprising that an *information theoretic* lower bound on the number of queries of local list-decoders is proven by considering concepts like constant-depth circuits from *circuit complexity*.

Extending the argument to lower bounds on hard-core predicates. It turns out that this argument is quite versatile, and this is the approach that we use to prove Theorems 1.9 and 1.13. Loosely speaking, in these theorems, we want to prove a lower bound on the number of queries made by a reduction that, when receiving oracle access to an adversary that breaks the hard-core predicate, is able to compute (or invert) the original function too well. Such lower bounds imply that such reductions do not produce small circuits when used in black-box proofs for hard-core predicates.

We will prove such lower bounds by showing that a reduction that makes q queries can be used to construct a circuit of size $s \approx 2^q$ and constant depth that solves the coin problem. Interestingly, this argument crucially relies on the fact that constant-depth circuits *can* distinguish BSC^n_ϵ from $\mathrm{BSC}^n_{2\epsilon}$ with size $\mathrm{poly}(n/\epsilon)$ which follows from the classical results of Ajtai on constant depth circuits for approximate majority [Ajt83].[5]

1.3.4 Conditioning on a good j

A disadvantage of the approach based on the coin problem is that at best, it can give lower bounds of $q = \Omega(1/\sqrt{\epsilon})$, and cannot give tight lower bounds of the form $q = \Omega(\frac{\log(1/\delta)}{\epsilon^2})$. In order to achieve such a bound

[4] These results of [CGR14] improve upon earlier work of Shaltiel and Viola [SV10] that gave slightly worse bound. These results are tight as shown by Limaye et al. [LSS+19] (that also extended the lower bound to hold for circuits that are also allowed to use parity gates).

[5] The proof of Theorem 1.13 uses an additional versatility of the argument (which we express in the terminology of codes): The argument works even if the individual procedures that are run in parallel are allowed to have some limited access to the message m, as long as this does not enable them to recover m. This property is used to handle reductions in a cryptographic setup, where reductions have access to the easy direction of a one-way function.

(as is the case in Theorem 1.5) we will try to reduce to Theorem 1.16 which *does* give a tight bound in case ϵ is not too small.

Our approach builds on the earlier work of Grinberg, Shaltiel and Viola [GSV18] that we surveyed in Section 1.1.1. When given a $(\frac{1}{2}-\epsilon, L, q, \delta)$-local list-decoder Dec, we say that an index $j \in [L]$ is *decoding*, if in the experiment $(m, z, w) \in \mathrm{RNSY}_{\frac{1}{2}-2\epsilon}$, when Dec is given oracle access to w and input j, then with probability $1 - 10\delta$ over $i \in [k]$, we have that $\mathrm{Dec}^w(i, j)$ recovers m_i.

We use a careful averaging argument to show that there exists an index $j' \in [L]$, and a fixed choice of the random coins of Dec, such that j' is decoding with probability at least $\Omega(1/L)$. We then consider the experiment $\mathrm{RNSY}'_{\frac{1}{2}-2\epsilon}$ in which we choose $(m, z, w) \leftarrow \mathrm{RNSY}_{\frac{1}{2}-2\epsilon}$ *conditioned* on the event $\{j' \text{ is decoding}\}$.

We have made progress, because in the experiment $\mathrm{RNSY}'_{\frac{1}{2}-2\epsilon}$ there exists a unique j' that is decoding, and so, when we implement the strategy explained in Section 1.3.2 we only need to consider this *single* j', which intuitively means that our scenario is similar to the warmup scenario of unique decoding described in Section 1.3.2.

The catch is that when choosing $(m, z, w) \leftarrow \mathrm{RNSY}'_{\frac{1}{2}-2\epsilon}$, we no longer have that z is distributed like $\mathrm{BSC}^n_{\frac{1}{2}-2\epsilon}$ (as the distribution of z may be skewed by conditioning on the event that j' is decoding).

Shaltiel and Viola [SV10] (and later work [GSV18, Sha20]) developed tools to handle this scenario. Loosely speaking, using these tools, it is possible to show that a large number of messages m are "useful" in the sense that there exists an event A_m such that if we consider (m, z, w) that are chosen from $\mathrm{RNSY}'_{\frac{1}{2}-2\epsilon}$ *conditioned* on A_m, then there exists a subset $B(m) \subseteq [n]$ of small size b, such that $z_{B(m)}$ is fixed, and $z_{[n]\setminus B(m)}$ is distributed like $\mathrm{BSC}^{n-b}_{\frac{1}{2}-2\epsilon}$.

If the number of possible choices for sets $B(m)$ is small, then by the pigeon-hole principle, there exists a fixed choice B that is good for a large number of useful messages m. This can be used to imitate the argument we used in the warmup, and prove a lower bound.[6]

Extending the argument to the case of small rate. A difficulty, that prevented [GSV18] from allowing length as large as $n = 2^k$, is that $B(m)$ is a subset of $[n]$, and so, even if $b = |B| = 1$, the number of possible choices for such sets is at least n. For the pigeon-hole principle argument above, we need that the number of messages (that is 2^k) is much larger than the number of possible choices for $B(m)$ (which is at least n). This means that one can only handle n which is sufficiently smaller than 2^k, and this approach cannot apply to codes with small rate (such as the Hadamard code).

We show how to solve this problem, and prove lower bounds for small rate codes. From a high level, our approach can be explained as follows: We consider the distribution of $B(m) = \{Y_1(m) < \ldots < Y_b(m)\}$ for a uniformly chosen useful m. We first show that if all the Y_j's have large min-entropy, then it is possible to prove a lower bound on q by reducing to Theorem 1.16 (the details of this are explained in the actual proof).

If on the other hand, one of the Y_j's has low min-entropy, then we will restrict our attention to a subset of useful messages on which Y_j is fixed. Loosely speaking, this reduces b by one, while not reducing the number of useful messages by too much (because the low min-entropy condition says that the amount of information that Y_j carries on m is small). In this trench warfare, in every iteration, we lose a fraction of useful messages, for the sake of decreasing b by one. Thus, eventually, we either reach the situation that all the Y_j's have large min-entropy, in which case we are done, or we reach the situation where B is fixed for all messages which we can also handle by the above.

We can withstand the losses and eventually win if ϵ is sufficiently larger than $1/\sqrt{k}$.

[6] Loosely speaking, this is because for good messages, in the conditioned experiment, z is distributed like $\mathrm{BSC}_{\frac{1}{2}-2\epsilon}$ (except that some bits of z are fixed as a function of m). Furthermore, as there are many good messages, the local list-decoder does not have enough information to correctly recover the message when given oracle access to $\mathrm{Enc}(m) \oplus \mathrm{BSC}^n_{\frac{1}{2}} = \mathrm{BSC}^n_{\frac{1}{2}}$.

1.4 More related work

Lower bounds on the number of queries of local decoders for *uniquely* decodable codes. In this paper, we prove lower bounds on the number of queries of local list-decoders. There is a long line of work that is concerned with proving lower bounds on the number of queries of uniquely decodable codes. As we have explained in Section 1.1.1, the parameter regime considered in the setting of uniquely decodable codes is very different than the parameter regime we consider here [Yek12].

Lower bounds on nonuniform black-box reductions for hardness amplification. A problem that is closely related to proving lower bounds on the number of queries of local list-decoders is the problem of proving lower bounds on the number of queries of nonuniform black-box reductions for hardness amplification. We have already discussed this line of work [Vio06, SV10, AS11, GSV18, Sha20] in Section 1.1.1.

Lower bounds on such reductions can be translated to lower bounds on local list-decoders (as long as the number of coins tossed by the local list-decoders is small). We remark that for the purpose of hardness amplification, it does not make sense to take codes with small rate (namely, codes with $n = 2^{k^{\Omega(1)}}$). The focus of Theorem 1.5 is to handle such codes.

Additionally, when using codes for hardness amplification, it does not make sense to take $\epsilon < 1/k$ (or even $\epsilon < 1/\sqrt{k}$). In contrast, the parameter regime considered in Theorem 1.6 focuses on small ϵ.

Other improvements of the Goldreich-Levin theorem. In this paper, we are interested in whether the Goldreich-Levin theorem can be improved. Specifically, we are interested in improvements where, when the original function has hardness $\rho = 2^{-\Omega(\ell)}$ for polynomial size circuits, then the hard-core predicate has hardness $\frac{1}{2} + \epsilon$ for $\epsilon = \ell^{-\omega(1)}$. We remark that there are other aspects of the Goldreich-Levin theorem that one may want to improve.

- When given an initial non-Boolean function on ℓ bits, the Goldreich-Levin theorem produces a hard-core predicate on $\ell' = 2\ell$ bits. It is possible to make ℓ' smaller (specifically, $\ell' = \ell + O(\log(1/\epsilon))$ by using other locally list-decodable codes instead of Hadamard. Our limitations apply to *any* construction (even one that is not based on codes) and in particular also for such improvements.
- It is sometimes desirable to produce many hard-core bits (instead of the single hard-core bit) that is obtained by a hard-core predicate. This can be achieved by using "extractor codes" with a suitable local list-decoding algorithm. The reader is referred to [TZ04] for more details. Once again, our limitations obviously apply also for the case of producing many hard-core bits.

Organization of the paper

Our results on local list-decoders (and the proofs of Theorem 1.5 and Theorem 1.6) are presented in Section 3. Our results on hard-core predicates appear in Section 4 (which includes a precise description of the model and formal restatements of Theorem 1.9 and Theorem 1.13).

2 Preliminaries

Relative Hamming weight and distance: For a string $x \in \{0,1\}^n$, we use $\mathrm{weight}(x)$ to denote the relative Hamming weight of x, namely $\mathrm{weight}(x) = |\{i : x_i = 1\}|$.

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 $\mathsf{dist}(x,y) = |\{i \in [n] : x_i \neq y_i\}|/n$.

2.1 Random variables

Notation for random variables: We use U_n to denote the uniform distribution on $\{0,1\}^n$. Given a distribution D, we use $x \leftarrow D$ to denote the experiment in which x is chosen according to D. For a set S we also use $x \leftarrow S$ to denote the experiment in which x is chosen uniformly from S. When we write $x_1 \leftarrow D_1, x_2 \leftarrow D_2$ we mean that the two experiments $x_1 \leftarrow D_1$ and $x_2 \leftarrow D_2$ are independent. If X is a random variable, and D is a distribution, then expressions of the form $\Pr_{y \leftarrow D}[\cdot]$ where the event involves both X and y, are in a probability space where the experiments producing X and $y \leftarrow D$ are independent.

Min-entropy: For a discrete random variable X over $\{0,1\}^n$ we define:

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

We will also use the following lemma.

Lemma 2.1. *Suppose that M is a distribution over $\{0,1\}^k$ that is uniform over a subset S of size 2^r for $r \geq k - k^{0.99}$. If k is sufficiently large, then for every function $D : [k] \to \{0,1\}$, we have that:*

$$\Pr_{m \leftarrow M, i \leftarrow [k]}[D(i) = m_i] \leq 0.5001.$$

The proof of Lemma 2.1 appears in Appendix A.

2.2 The number of queries needed to distinguish $\mathrm{BSC}^n_{\frac{1}{2}-\epsilon}$ from $\mathrm{BSC}^n_{\frac{1}{2}}$

The following lemma by Shaltiel and Viola [SV10] is a formal restatement of Informal Theorem 1.16.

Lemma 2.2 ([Vio06, SV10])**.** *For every $\epsilon, \delta > 0$, such that $\delta < 0.4$, if $T : \{0,1\}^q \to \{0,1\}$ satisfies:*

- $\Pr[T(\mathrm{BSC}^q_{\frac{1}{2}-\epsilon}) = 1] \geq 1 - \delta$.
- $\Pr[T(\mathrm{BSC}^q_{\frac{1}{2}}) = 1] \leq 0.51$.

Then, $q = \Omega\left(\frac{\log \frac{1}{\delta}}{\epsilon^2}\right)$.

2.3 Constant depth circuits, approximate majority, and the coin problem

As is standard in complexity theory, when discussing circuits, we consider circuits over the standard set of gates $\{\mathrm{AND}, \mathrm{OR}, \mathrm{NOT}\}$. We use the convention that the size of a circuit is the the number of gates and wires. With this convention, a circuit C that on input $x \in \{0,1\}^n$, outputs x_1, has size $O(1)$. If we mention the depth of the circuit, then we mean that AND,OR gates have unbounded fan-in, and otherwise these gates have fan-in 2.

Constant depth circuits for approximate majority: We use the following classical result by Ajtai showing that constant depth circuits can compute approximate majority:

Theorem 2.3 ([Ajt83])**.** *There exists a constant c such that for every two constants $0 \leq p < P < 1$, and every sufficiently large n, there exists a circuit C of size n^c and depth c such that for every $x \in \{0,1\}^n$:*

- *If* $\mathrm{weight}(x) \geq P$ *then* $C(x) = 1$.
- *if* $\mathrm{weight}(x) \leq p$ *then* $C(x) = 0$.

Lower bounds for the coin problem: We use lower bounds on the size of constant depth circuits for the "coin problem". A sequence of works by [SV10, Aar10, CGR14, LSS$^+$19] gives such lower bounds, and the statement below is due to Aaronson [Aar10] and Cohen, Ganor and Raz [CGR14] (and was improved by Limaye et al. [LSS$^+$19] to also hold for circuits that are allowed to use PARITY gates of unbounded fan-in).

Theorem 2.4 ([Aar10, CGR14, LSS$^+$19])**.** *Suppose $C : \{0,1\}^n \to \{0,1\}$ is a circuit of depth d satisfying:*

- $\Pr_{z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}-\epsilon}}[C(z) = 1] \geq 0.9$,
- $\Pr_{z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}+\epsilon}}[C(z) = 0] \leq 0.1$.

Then, C must have size at least $\exp(\Omega\left(d \cdot (1/\epsilon)^{\frac{1}{d-1}}\right))$.

For our purposes, we prefer to replace the distributions $\mathrm{BSC}^n_{\frac{1}{2}-\epsilon}$ and $\mathrm{BSC}^n_{\frac{1}{2}+\epsilon}$, by $\mathrm{BSC}^n_{\frac{1}{2}-\epsilon}$ and $\mathrm{BSC}^n_{\frac{1}{2}}$ (as is the case in Lemma 2.2). The next corollary shows that the results of Theorem 2.4 imply a similar bound when comparing $\mathrm{BSC}^n_{\frac{1}{2}-\epsilon}$ to $\mathrm{BSC}^n_{\frac{1}{2}}$.[7]

Corollary 2.5. *Suppose $C : \{0,1\}^n \to \{0,1\}$ is a circuit of depth d satisfying:*

- $\Pr_{z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}-\epsilon}}[C(z) = 1] \geq 0.99$,
- $\Pr_{z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}}}[C(z) = 0] \leq 0.01$.

Then, C must have size at least $\exp(\Omega\left(d \cdot (1/\epsilon)^{\frac{1}{d-1}}\right))$.

For completeness, we show that Corollary 2.5 follows from Theorem 2.4 in Appendix B.

3 Query complexity lower bounds for local list-decoding

In this section we prove Theorem 1.5 and Theorem 1.6 and provide lower bounds on the query complexity of local list-decoders. In Section 3.1, we introduce a relaxed concept that we call "approximate local list-decoders on noisy random codewords" (ARLLD) in which the local list-decoder is only required to recover a random message that was corrupted by a binary symmetric channel (meaning that the original message appears in the list of messages that are locally computed by the local list-decoder).

We then use a careful averaging argument to show that any local list-decoder (LLD) can be converted into an ARLLD with roughly the same parameters, and furthermore, the obtained ARLLD is *deterministic*.

This means that when proving Theorem 1.5 and Theorem 1.6 it is sufficient to consider ARLLDs, and these proofs appear in Sections 3.2 and 3.3 respectively.

3.1 Definition of approximate local list-decoders on noisy random codewords

Our goal is to prove lower bounds on the number of queries q of $(\frac{1}{2} - \epsilon, L, q, \delta)$-local list-decoders. We will show that it is sufficient to consider local list-decoders that need to perform an easier task. More specifically, we relax the task of a local list-decoder in the following ways:

[7] We remark that bounds for the latter choice of distributions were proven by Shaltiel and Viola [SV10], but we prefer to rely on the subsequent bounds of [Aar10, CGR14, LSS$^+$19], which are tighter and lead to a larger constant in the exponent of $\frac{1}{\epsilon}$ in Theorem 1.6.

- The local list-decoder does not need to succeed on every $w \in \{0,1\}^n$, but only with not too small probability over a "random noisy codeword" which is sampled by encoding a uniformly chosen message m, and hitting $\mathrm{Enc}(m)$ with the noise generated by a binary symmetric channel, to obtain a word w. It is required that with not too small probability over the choice of the message m and the random noise, there exists a j such that the local decoder with oracle access to w, and input j, recovers m.
- The local decoder is *approximate* and is not required to recover m_i correctly on every $i \in [k]$. Instead, it is allowed to err on a δ fraction of i's.

This makes the task of the decoder easier. It turns out that with this relaxation, random coins are not very helpful to the local list-decoder, and so, it is sufficient to consider deterministic local list-decoders (which do not have access to random coins). This is captured in the following definition.

Definition 3.1 (Approximate local list-decoder on noisy random codewords)**.** *Let* $\mathrm{Enc} : \{0,1\}^k \to \{0,1\}^n$ *be a function, and* $\epsilon < \frac{1}{4}$*. A* $(\frac{1}{2} - \epsilon, L, q, \delta)$*-approximate* RNSY *local list-decoder (ARLLD) for* Enc *is a* deterministic *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.*

It is required that, with probability at least $1/3$ *over choosing* $(m, z, w) \leftarrow \mathrm{RNSY}^{\mathrm{Enc}}_{\frac{1}{2}-2\epsilon}$*, there exists* $j \in [L]$ *such that*

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

We stress again that Dec is deterministic and the probability in Definition 3.1 is taken over the choice of a uniform random coordinate $i \in [k]$.

The following proposition shows that in order to prove lower bounds on local list-decoders (LLDs, Definition 1.3), it is sufficient to prove lower bounds on approximate RNSY local list-decoders (ARLLDs, Definition 3.1).

Proposition 3.2 (LLD implies ARLLD)**.** *There exists a universal constant* $a > 1$ *such that for every* $a \cdot \sqrt{\frac{1}{n}} \leq \epsilon < \frac{1}{4}$*, if there exists an* $(\frac{1}{2} - \epsilon, L, q, \delta)$*-local list-decoder for a function* $\mathrm{Enc} : \{0,1\}^k \to \{0,1\}^n$ *then there also exists an* $(\frac{1}{2} - \epsilon, L, q, 10 \cdot \delta)$*-approximate* RNSY *local list-decoder for* Enc*.*

Proof. Within this proof, in order to avoid clutter, we use RNSY to denote $\mathrm{RNSY}^{\mathrm{Enc}}_{\frac{1}{2}-2\epsilon}$. Let Dec denote an LLD for Enc. For $(m, z, w) \leftarrow \mathrm{RNSY}$, by a Chernoff bound, for $\gamma = 2^{-\Omega(\epsilon^2 \cdot n)}$, with probability $1 - \gamma$, we have that $\mathsf{dist}(\mathrm{Enc}(m), w) \leq \frac{1}{2} - \epsilon$, meaning that $m \in \mathrm{List}^{\mathrm{Enc}}_{\frac{1}{2}-\epsilon}(w)$. 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 $P_{w,j,r^{\mathrm{shared}}}(i, r) = \mathrm{Dec}^w(i, j, r^{\mathrm{shared}}, r)$ locally computes m with error δ.

Let E_1 be the experiment in which $(m, z, w) \leftarrow \mathrm{RNSY}$ and let 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 $\hat{r}^{\mathrm{shared}}$ such that:

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

Let S denote the set of triplets (m, z, w) in the support of RNSY for which the event above occurs. For every such triplet, 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 triplet $(m, z, w) \in S$, produces such a $j \in [L]$. This means that:

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

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

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

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

since $P_{w,f((m,z,w)),\hat{r}^{shared}}$ 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:

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

By Markov's inequality:

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

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

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

Which gives that:

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

where the second inequality follows because by our requirements on ϵ, we can choose a so that $2/3 - \gamma > \frac{1}{2}$. Thus, the oracle procedure $\overline{\mathrm{Dec}}^{(\cdot)}$ is a $(\frac{1}{2} - \epsilon, L, q, 10 \cdot \delta)$-ARLLD as required. □

By Proposition 3.2 in order to prove our main theorems on local list-decoders, it is sufficient to prove them for approximate local list-decoders.

3.2 Proof of Theorem 1.5

We use the following definition.

Definition 3.3. *Given a string $w \in \{0,1\}^n$, a subset of coordinates $B = \{h_1 < \ldots < h_b\} \subseteq [n]$ of size b, and a string $v \in \{0,1\}^B$, we let $\mathrm{Fix}_{B\to v}(w) \in \{0,1\}^n$ denote the string that is obtained from w by fixing the bits in B to the corresponding values in v. That is,*

$$(\mathrm{Fix}_{B\to v}(w))_\ell = \begin{cases} v(h_i), & \exists i \text{ s.t. } \ell = h_i \\ w_\ell, & \ell \notin B \end{cases}$$

The lower bound will follow from the following lemma.

Lemma 3.4. *There exists a universal constant $\nu > 0$ such that the following holds for any $L \leq 2^{k^{0.9}}$, $\epsilon, \delta \geq k^{-\nu}$, and $q \leq \frac{\log(1/\delta)}{\epsilon^2}$. Let* Dec *be a $(\frac{1}{2} - \epsilon, q, L, \delta)$-ARLLD for* $\mathrm{Enc} : \{0,1\}^k \to \{0,1\}^n$. *Then there exist $m' \in \{0,1\}^k$, $i' \in [k]$, $j' \in [L]$, a subset $B \subseteq [n]$, and a string $v \in \{0,1\}^B$ such that:*

1. $\Pr_{z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}-2\epsilon}} \left[\mathrm{Dec}^{\mathrm{Fix}_{B \to v}(\mathrm{Enc}(m') \oplus z)}(i', j') = m'_{i'} \right] \geq 1 - 200\delta.$
2. $\Pr_{z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}}} \left[\mathrm{Dec}^{\mathrm{Fix}_{B \to v}(\mathrm{Enc}(m') \oplus z)}(i', j') = m'_{i'} \right] \leq 0.51.$

Proof of Theorem 1.5. Consider a $(\frac{1}{2} - \epsilon, L, q, \delta)$-LLD for $\mathrm{Enc} : \{0,1\}^k \to \{0,1\}^n$. By assumption that $\delta < \frac{1}{3}$, we can further assume that $\delta < 0.0002$, since, if otherwise, we can get to the desired error probability by amplification, at the loss of only a constant factor in the query complexity. We may further assume that $q \leq \frac{\log(1/\delta)}{\epsilon^2}$, otherwise we are done.

Applying Proposition 3.2, we get that there exists a $(\frac{1}{2} - \epsilon, L, q, \delta')$-ARLLD for Enc, where $\delta' < 0.002$. Applying Lemma 3.4 to this decoder, we can see that Dec, when given oracle access to $\mathrm{Fix}_{B \to v}(\mathrm{Enc}(m') \oplus z)$ and inputs i', j' makes q queries and outputs $m_{i'}$, (1) with probability at least $1 - 200\delta' = 0.6$ if $z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}-2\epsilon}$ and (2) with probability at most 0.51 if $z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}}$. Finally, applying Lemma 2.2 here, completes the proof of Theorem 1.5. □

We will prove Lemma 3.4 using the probabilistic method. The main technical part of the proof is the following lemma.

Lemma 3.5. *There exists a universal constant $\nu > 0$ such that the following holds for any $L \leq 2^{k^{0.9}}$, $\epsilon, \delta \geq k^{-\nu}$, and $q \leq \frac{\log(1/\delta)}{\epsilon^2}$. Suppose that* Dec *is a $(\frac{1}{2} - \epsilon, q, L, \delta)$-ARLLD for* $\mathrm{Enc} : \{0,1\}^k \to \{0,1\}^n$. *Then there exist:*

- *$j' \in [L]$,*
- *Functions B, v that given $m \in \{0,1\}^k$ produce a set $B(m) \subseteq [n]$ of size $b = O(\frac{q \cdot \log L}{\delta})$ and $v(m) \in \{0,1\}^{B(m)}$, respectively,*
- *A distribution* MU *over $\{0,1\}^k$,*

such that if we use WBV_p to denote the experiment in which:

- *A message $m \in \{0,1\}^k$ is chosen by $m \leftarrow \mathrm{MU}$.*
- *A noise string z is chosen by $z \leftarrow \mathrm{BSC}^n_p$.*
- *A word w is obtained by $\mathrm{Fix}_{B(m) \to v(m)}(\mathrm{Enc}(m) \oplus z)$.*

We have that:

1. $\Pr_{(m,z,w) \leftarrow \mathrm{WBV}_{\frac{1}{2}-2\epsilon}, i \leftarrow [k]}[\mathrm{Dec}^w(i, j') = m_i] \geq 1 - 2\delta.$
2. $\Pr_{(m,z,w) \leftarrow \mathrm{WBV}_{\frac{1}{2}}, i \leftarrow [k]}[\mathrm{Dec}^w(i, j') = m_i] \leq 0.501.$

Lemma 3.4 follows from Lemma 3.5 by a Markov argument as follows.

Proof of Lemma 3.4. By applying Markov's inequality to the first and second conditions in Lemma 3.5, we have:

$$\Pr_{m\leftarrow \mathrm{MU}, i\leftarrow[k]}\left[\Pr_{z\leftarrow \mathrm{BSC}^n_{\frac{1}{2}-2\epsilon}}[\mathrm{Dec}^{\mathrm{Fix}_{B(m)\to v(m)}(\mathrm{Enc}(m)\oplus z)}(i,j')\neq m_i] > 200\delta\right] < \frac{1}{100},$$

and

$$\Pr_{m\leftarrow \mathrm{MU}, i\leftarrow[k]}\left[\Pr_{z\leftarrow \mathrm{BSC}^n_{\frac{1}{2}}}[\mathrm{Dec}^{\mathrm{Fix}_{B(m)\to v(m)}(\mathrm{Enc}(m)\oplus z)}(i,j')= m_i] > 0.51\right] < \frac{0.501}{0.51} < 0.985.$$

Hence, by the union bound, it follows that there exists $m' \in \{0,1\}^k, i' \in [k]$ such that:

$$\Pr_{z\leftarrow \mathrm{BSC}^n_{\frac{1}{2}-2\epsilon}}[\mathrm{Dec}^{\mathrm{Fix}_{B(m')\to v(m')}(\mathrm{Enc}(m')\oplus z)}(i',j')= m'_{i'}] \geq 1-200\delta,$$
$$\Pr_{z\leftarrow \mathrm{BSC}^n_{\frac{1}{2}}}[\mathrm{Dec}^{\mathrm{Fix}_{B(m')\to v(m')}(\mathrm{Enc}(m')\oplus z)}(i',j')= m'_{i'}] \leq 0.51.$$

Lemma 3.4 follows. □

3.2.1 Proof of the first item of Lemma 3.5

We are given a $(\frac{1}{2}-\epsilon, q, L, \delta)$-ARLLD Dec for $\mathrm{Enc}: \{0,1\}^k \to \{0,1\}^n$. To avoid clutter, we will omit Enc in $\mathrm{RNSY}^{\mathrm{Enc}}_p$ in this section. We start with a couple of useful definitions.

Definition 3.6. *We say that an element $j \in [L]$ is decoding for m, w if*

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

The definition of ARLLD says that with probability at least $1/3$ over choosing $(m,z,w) \leftarrow \mathrm{RNSY}_{\frac{1}{2}-2\epsilon}$, there exists a $j \in [L]$ that is decoding for m, w. By averaging over the L choices of j, it follows that, there exists a $j' \in [L]$ such that with probability $1/(3L)$ over choosing $(m,z,w) \leftarrow \mathrm{RNSY}_{\frac{1}{2}-2\epsilon}$, this fixed j' is decoding for m, w. This is stated in the next claim.

Claim 3.7. *There exists $j' \in [L]$ such that with probability at least $1/(3L)$ over choosing $(m,z,w) \leftarrow \mathrm{RNSY}_{\frac{1}{2}-2\epsilon}$, j' is decoding for m, w.*

Definition 3.8. *We say that a message $m \in \{0,1\}^k$ is* useful *if*

$$\Pr_{z\leftarrow \mathrm{BSC}^n_{\frac{1}{2}-2\epsilon}}[j' \textit{ is decoding for } m, \mathrm{Enc}(m)\oplus z] \geq \frac{1}{6L}.$$

It follows that:

Claim 3.9. *There are at least $2^k/(6L)$ useful messages.*

Proof. Otherwise, when choosing $m \leftarrow \{0,1\}^k, z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}-2\epsilon}$ and setting $w = \mathrm{Enc}(m) \oplus z$ (as is done for $(m, z, w) \leftarrow \mathrm{RNSY}_{\frac{1}{2}-2\epsilon}$):

$$\begin{aligned}\Pr[j' \text{ is decoding for } m, w] &\leq \Pr[m \text{ is useful}] + \Pr[j' \text{ is decoding for } m, w | m \text{ is not useful}] \\ &< \frac{1}{6L} + \frac{1}{6L} = \frac{1}{3L},\end{aligned}$$

which contradicts Claim 3.7. □

Definition 3.10. *Given a string $w \in \{0,1\}^n$ and a set $B \subseteq [n]$, we denote by $w(B) \in \{0,1\}^B$, the substring of w restricted to the coordinates in B.*

Definition 3.11. *For a random variable W over $\{0,1\}^n$, a set $B \subseteq [n]$ and $v \in \{0,1\}^B$, such that $\Pr[W_h = v(h) \ \forall h \in B] > 0$, we define the probability distribution $\mathrm{Cond}_{B\to v}(W)$ to be $(W | W_h = v(h) \ \forall h \in B)$.*

Remark 3.12. *For a random variable W over $\{0,1\}^n$, it is important to distinguish $\mathrm{Fix}_{B\to v}(W)$ from $\mathrm{Cond}_{B\to v}(W)$. The former means that we sample $w \leftarrow W$ and replace the content of w in the indices in B by the corresponding values taken from v. The latter is only defined if W is a random variable for which the event $\{W_h = v(h) \ \forall h \in B\}$ can occur, and for such a variable, $\mathrm{Cond}_{B\to v}(W)$ is obtained by* conditioning *the random variable W on the event $\{W_h = v(h) \ \forall h \in B\}$. In particular, this conditioning may mean that when restricting $\mathrm{Cond}_{B\to v}(W)$ and W to indices that are not in B, we may get different distributions. This is in contrast to $\mathrm{Fix}_{B\to v}(W)$ where by definition, restricting $\mathrm{Fix}_{B\to v}(W)$ and W to indices that are not in B, gives the same distribution.*

A useful observation is that if W is a sequence of n independent bit variables, then for every B, v, $\mathrm{Cond}_{B\to v}(W) = \mathrm{Fix}_{B\to v}(W)$.

We shall use the following lemma from [Sha20], which improves a similar lemma (with more conditions) that was proven in [GSV18].

Lemma 3.13 ([Sha20])**.** *Let W be a probability distribution over $\{0,1\}^n$, let $A \subseteq \{0,1\}^n$ be an event such that $\Pr[W \in A] \geq 2^{-a}$, and let $W' = (W | W \in A)$. For every $\eta > 0$, there exists a set $B \subseteq [n]$ of size $b = O(qa/\eta)$, and $v \in \{0,1\}^B$ such that for every oracle procedure $D^{\cdot}$ that makes q queries:*

$$|\Pr[D^{\mathrm{Cond}_{B\to v}(W)} = 1] - \Pr[D^{\mathrm{Cond}_{B\to v}(W')} = 1]| \leq \eta.$$

We now explain why this lemma is useful. Note that if we start with some distribution W over $\{0,1\}^n$, then after conditioning on the event $\{W \in A\}$, the bits in the obtained distribution $W' = (W \in A)$ may become correlated. The Lemma says that there exists a set $B \subseteq [n]$ and $v \in \{0,1\}^B$ such that if we *further condition* both W and W' on the event $\{(W | W_h = v(h) \ \forall h \in B)\}$, to obtain the distributions $\mathrm{Cond}_{B\to v}(W)$ and $\mathrm{Cond}_{B\to v}(W')$, then these two distributions are "similar" in the sense that a procedure D that makes few oracle calls, cannot significantly distinguish between them.

This is useful because if $W = \mathrm{BSC}^n_p$, then W is a sequence of independent bits, and so, $\mathrm{Cond}_{B\to v}(W) = \mathrm{Fix}_{B\to v}(W)$. Namely, a distribution in which the bits in B are fixed, and the bits outside of B are independent and distributed like BSC^{n-b}_p. Loosely speaking, this means that as long as we do not mind to condition on the event $\{(W | W_h = v(h) \ \forall h \in B)\}$, then in order to understand how D behaves when given oracle to W' it is sufficient to understand how it behaves when given oracle access to W.

Definition 3.14. *For a message m we denote by $\mathrm{NSY}(m)$ the distribution over $\{0,1\}^n$ obtained by choosing $z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}-2\epsilon}$, and setting $w = \mathrm{Enc}(m) \oplus z$. We use $\mathrm{NSY}'(m)$ to denote the distribution in which $w \leftarrow \mathrm{NSY}(m)$ conditioned on the event $\{j' \text{ is decoding for } m, w\}$.*

Using the above Lemma 3.13 we obtain the following.

Claim 3.15. *For every useful $m \in \{0,1\}^k$ there exist a set $B(m) \subseteq [n]$ of size $b = O(q \cdot (\log L)/\delta)$, and $v(m) \in \{0,1\}^{B(m)}$ such that for every $i \in [k]$:*

$$|\Pr[\mathrm{Dec}^{\mathrm{Cond}_{B(m)\to v(m)}(\mathrm{NSY}(m))}(i,j') = 1] - \Pr[\mathrm{Dec}^{\mathrm{Cond}_{B(m)\to v(m)}(\mathrm{NSY}'(m))}(i,j') = 1]| \leq \delta.$$

Proof. Apply Lemma 3.13 with W being the distribution $\mathrm{NSY}(m)$, A being the event that j' is decoding for m, w, where $w \leftarrow \mathrm{NSY}(m)$, and $D = \mathrm{Dec}^{(\cdot)}(i,j')$, for any $i \in [k]$. Note that indeed, under this setting we have that

$$\Pr[W \in A] = \Pr_{z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}-2\epsilon}}[j' \text{ is decoding for } m, \mathrm{Enc}(m) \oplus z] \geq \frac{1}{6L},$$

by assumption that m is useful, and $W' = \mathrm{NSY}'(m)$. □

Next observe that by the definition of usefulness, we have that:

Claim 3.16. *For every useful $m \in \{0,1\}^k$,*

$$\Pr_{i \leftarrow [k]}[\mathrm{Dec}^{\mathrm{Cond}_{B(m)\to v(m)}(\mathrm{NSY}'(m))}(i,j') = m_i] \geq 1 - \delta.$$

By Claim 3.15 for every fixed i, $\mathrm{Dec}^{(\cdot)}(i,j')$ cannot distinguish between the oracles in Claim 3.15 with advantage larger than δ. It follows that it cannot do this when $i \leftarrow [k]$ is chosen at random, which gives:

Claim 3.17. *For every useful $m \in \{0,1\}^k$,*

$$\Pr_{i \leftarrow [k]}[\mathrm{Dec}^{\mathrm{Cond}_{B(m)\to v(m)}(\mathrm{NSY}(m))}(i,j') = m_i] \geq 1 - 2 \cdot \delta.$$

Moreover, by definition $\mathrm{NSY}(m)$ is composed of n independent bit random variables, and so,

$$\mathrm{Cond}_{B(m)\to v(m)}(\mathrm{NSY}(m)) = \mathrm{Fix}_{B(m)\to v(m)}(\mathrm{NSY}(m)).$$

As Claim 3.17 is true for every useful m, it is also true for every probability distribution MU over useful messages m. This is stated below.

Claim 3.18. *For any distribution* MU *over useful messages,*

$$\Pr_{m \leftarrow \mathrm{MU}, i \leftarrow [k]}[\mathrm{Dec}^{\mathrm{Fix}_{B(m)\to v(m)}(\mathrm{NSY}(M))}(i,j') = M_i] \geq 1 - 2 \cdot \delta.$$

Let MU denote the uniform distribution over useful messages. Note that for any choice of distribution MU, the experiment in which we choose $(m,z,w) \leftarrow \mathrm{WBV}_{\frac{1}{2}-2\epsilon}$ and consider the pair (m,w) is by definition identical to the experiment in which we choose $m \leftarrow \mathrm{MU}$ and set $w = \mathrm{Fix}_{B(m)\to v(m)}(\mathrm{NSY}(m))$.

It follows that for our choices of $j', B(\cdot), v(\cdot)$, every distribution MU over useful messages satisfies the first item of Lemma 3.5. This is summarized in the claim below.

Claim 3.19. *There exist:*

- *$j' \in [L]$,*
- *Functions B, v that given $m \in \{0,1\}^k$ produce a set $B(m) \subseteq [n]$ of size $b = O(\frac{q \cdot \log L}{\delta})$ and $v(m) \in \{0,1\}^{B(m)}$, respectively,*

such that for every distribution MU *over useful messages,*

$$\Pr_{(m,z,w) \leftarrow \mathrm{WBV}_{\frac{1}{2}-2\epsilon}, i \leftarrow [k]}[\mathrm{Dec}^{w}(i,j') = m_i] \geq 1 - 2\delta.$$

3.2.2 Proof of the second item of Lemma 3.5

Let $j', b, B(\cdot), v(\cdot)$ be as in Claim 3.19. In order to complete the proof of Lemma 3.5 we need to show that for these choices, there exists a distribution MU over useful messages such that:

$$\Pr_{(m,z,w)\leftarrow \mathrm{WBV}_{\frac{1}{2}}, i\leftarrow[k]}[\mathrm{Dec}^w(i,j') = m_i] \leq 0.501.$$

For $p = \frac{1}{2}$, BSC_p^n is a uniformly chosen string of length n. It follows that for every $m \in \{0,1\}^k$, the distributions $\mathrm{Enc}(m) \oplus \mathrm{BSC}_{\frac{1}{2}}^n$ and $\mathrm{BSC}_{\frac{1}{2}}^n$ are identical (as the uniform string $\mathrm{BSC}_{\frac{1}{2}}^n$ masks out $\mathrm{Enc}(m)$).

This means that for every choice of distribution MU, the pair $(m,w) \leftarrow \mathrm{WBV}_{\frac{1}{2}}$ is distributed exactly like a pair $(m, \mathrm{Fix}_{B(m)\to v(m)}(z))$ where $m \leftarrow \mathrm{MU}, z \leftarrow \mathrm{BSC}_{\frac{1}{2}}^n$. It follows that in order to complete the proof of Lemma 3.5 it is sufficient to prove the following lemma:

Lemma 3.20. *There exists a universal constant $\nu > 0$ such that the following holds for any $L \leq 2^{k^{0.9}}$, $\epsilon, \delta \geq k^{-\nu}$, and $q \leq \frac{\log(1/\delta)}{\epsilon^2}$. There exists a distribution MU over useful messages such that for every oracle procedure $D^{(\cdot)}(i)$ that makes at most q queries to its oracle it holds that:*

$$\Pr_{m\leftarrow\mathrm{MU}, z\leftarrow\mathrm{BSC}_{\frac{1}{2}}^n, i\leftarrow[k]}[D^{\mathrm{Fix}_{B(m)\to v(m)}(z)}(i) = m_i] \leq 0.501.$$

Lemma 3.20 implies Lemma 3.5 by setting $D(\cdot)$ to be $\mathrm{Dec}(\cdot, j')$.

We will denote the elements of $B(m)$ by:

$$B(m) = \{h_1(m) < \ldots < h_b(m)\}.$$

In order to prove Lemma 3.20, we will prove the following claim.

Claim 3.21. *Let S be a subset of useful messages, such that $|S| \geq 2^{k-k^{0.99}}$, and let MD be the uniform distribution over S. Let b be an integer, and let B, v be functions that given $m \in S$ produce a set $B(m) \subseteq [n]$ of size b and $v(m) \in \{0,1\}^{B(m)}$, respectively. If there exists an oracle procedure $D^{(\cdot)}$ that makes at most q queries such that:*

$$\Pr_{m\leftarrow\mathrm{MD}, z\leftarrow\mathrm{BSC}_{\frac{1}{2}}^n, i\leftarrow[k]}[D^{\mathrm{Fix}_{B(m)\to v(m)}(z)}(i) = m_i] > 0.501,$$

then there exist:

- *A subset $\bar{S} \subseteq S$ such that $\frac{|\bar{S}|}{|S|} \geq 2^{-(t+1)}$, where $t = 11 + \log b + \log q$.*
- *An index $j \in [b]$, a codeword index $h' \in [n]$, and a value $v' \in \{0,1\}$ such that for every message $m \in \bar{S}$, the j-th codeword index in $B(m)$ is $h_j(m) = h'$ and the value of the corresponding coordinate in v is $(v(m))(h') = v'$.*

We first show that Claim 3.21 implies Lemma 3.20.

Proof of Lemma 3.20. Our goal is to find a distribution MU over useful messages so that

$$\Pr_{m\leftarrow\mathrm{MU}, z\leftarrow\mathrm{BSC}_{\frac{1}{2}}^n, i\leftarrow[k]}[D^{\mathrm{Fix}_{B(m)\to v(m)}(z)}(i) = m_i] \leq 0.501 \tag{1}$$

for every q-query oracle procedure $D^{(\cdot)}(i)$.

To this end, we observe that if it is the case that $B(m) = B(m')$ and $v(m) = v(m')$ for all $m, m' \leftarrow \mathrm{MU}$, then $\mathrm{Fix}_{B(m)\to v(m)}(z)$ does not convey any information on $m \leftarrow \mathrm{MU}$. Thus, if MU satisfies this property, and is uniform over a set of size at least $2^{k-k^{0.99}}$, then by Lemma 2.1, any oracle algorithm D does not satisfy (1), irrespective of the number of queries that D makes. To use this observation, we apply Claim 3.21 repeatedly till we either reach a distribution MU that satisfies (1), in which case we are done, or we reach a distribution MU which satisfies that $B(m) = B(m')$ and $v(m) = v(m')$ for all $m, m' \leftarrow \mathrm{MU}$.

More specifically, we initialize MU with the uniform distribution on the set of all useful messages. While there exist a q-query oracle procedure D which does not satisfy (1) and $m, m' \leftarrow \mathrm{MU}$ such that $B(m) \neq B(m')$ or $v(m) \neq v(m')$, then by Claim 3.21, assuming that MU is supported on a set of size at least $2^{k-k^{0.99}}$, there exist a subset $\bar{S} \subseteq S$ such that $\frac{|\bar{S}|}{|S|} \geq 2^{-(t+1)}$, and an index $j \in [b]$, a codeword index $h' \in [n]$, and a value $v' \in \{0,1\}$ such that for every message $m \in \bar{S}$, the j-th codeword index in $B(m)$ is $h_j(m) = h'$ and the value of the corresponding coordinate in v is $(v(m))(h') = v'$. We thus set MU to be the uniform distribution over the messages in $\bar{S}$, which fixes one of the positions in $B(\cdot)$ and the corresponding value in $v(\cdot)$ for all messages sampled from MU.

Repeatedly applying the above argument, we eventually either reach a distribution MU which satisfies (1) for any q-query oracle procedure D, or we reach a distribution MU so that $B(m) = B(m')$ and $v(m) = v(m')$ for all $m, m' \leftarrow \mathrm{MU}$. In the former case we are clearly done, while in the latter case, by Lemma 2.1, it suffices to show that when the process terminates, MU is distributed uniformly over a set of size at least $2^{k-k^{0.99}}$ (as in this case (1) holds for any oracle procedure D, irrespective of the number of queries it makes).

To see that the above condition holds, note that the total number of iterations is at most b, since in each iteration at least one of the b indices in $B(\cdot)$ is fixed. Also recall that by Claim 3.9, there are at least $2^k/(6L)$ useful messages. Consequently, when the process terminates, the number of messages in the support of MU is at least

$$\begin{aligned}
\frac{2^k}{6L} \cdot 2^{-(t+1)b} &= \frac{2^k}{6L} \cdot \left(2^{12} \cdot b \cdot q\right)^{-b} \\
&\geq \frac{2^k}{L} \cdot \left(\frac{\delta}{q \log L}\right)^{O(q(\log L)/\delta)} \\
&\geq \frac{2^k}{L} \cdot \left(\frac{\epsilon\delta}{\log L}\right)^{O(\log(1/\delta)(\log L)/(\delta\epsilon^2))} \\
&= 2^k \cdot \exp\left(-(\log L \log\log L) \cdot \mathrm{poly}(1/\delta, 1/\epsilon)\right),
\end{aligned}$$

where the first equality follows recalling that $t = 11 + \log b + \log q$ by Claim 3.21, the second inequality follows recalling that $b = O(q \cdot (\log L)/\delta)$ by Claim 3.19, and the third inequality follows by assumption that $q \leq \log(1/\delta)/\epsilon^2$. Finally, note that by choosing a sufficiently small constant $\nu > 0$, and recalling our assumption that $L \leq 2^{k^{0.9}}$ and $\epsilon, \delta \geq k^{-\nu}$, we can guarantee that the above expression is at least $2^{k-k^{0.99}}$. This concludes the proof of the lemma. □

Claim 3.21 will follow from the next two claims:

Claim 3.22. *Suppose that* MD *is a uniform distribution over a set S of size at least $2^{k-k^{0.99}}$, and that for every $j \in [b]$, $H_\infty(h_j(\mathrm{MD})) \geq t$ for $t = 11 + \log b + \log q$. Then*

$$\Pr_{m\leftarrow \mathrm{MD}, z\leftarrow \mathrm{BSC}^n_{\frac{1}{2}}, i\leftarrow[k]}\left[D^{\mathrm{Fix}_{B(m)\to v(m)}(z)}(i) = m_i\right] \leq 0.501.$$

Proof. Let E_B denote the event that $D^{\mathrm{Fix}_{B(m)\to v(m)}(z)}(i)$ makes a query into $B(m)$, and let $\tilde{E}_B$ denote the event that $D^z(i)$ makes a query into $B(m)$. Then we have that when choosing $m \leftarrow \mathrm{MD}$,$z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}}$, and $i \leftarrow [k]$,

$$\Pr\left[D^{\mathrm{Fix}_{B(m)\to v(m)}(z)}(i) = m_i\right] \leq \Pr\left[\left(D^{\mathrm{Fix}_{B(m)\to v(m)}(z)}(i) = m_i\right) \cap \neg E_B\right] + \Pr[E_B].$$

To bound the right-hand term, we first claim that $\Pr[E_B] = \Pr[\tilde{E}_B]$. To see this, note that for any fixed m, i, the set of strings z on which $D^{\mathrm{Fix}_{B(m)\to v(m)}(z)}(i)$ makes a query into $B(m)$ is identical the set of strings z on which $D^z(i)$ makes a query into $B(m)$, since the location of the queries made before the first query to $B(m)$ are the same for the oracles $\mathrm{Fix}_{B(m)\to v(m)}(z)$ and z. Thus, to bound the right-hand term, it suffices to bound the probability of the event $\tilde{E}_B$.

To bound the probability that $\tilde{E}_B$ occurs, we this time fix the string z and the index i, and note that this determines the query pattern of $D^z(i)$. Next we recall our assumption that for $m \leftarrow \mathrm{MD}$, $H_\infty(h_j(m)) \geq t$ for all $j \in [b]$. Thus the probability, over $m \leftarrow \mathrm{MD}$ (noting that this choice of m is independent of the fixing of z, i), that a specific query of $D^z(i)$ is to a particular point in $B(m)$ is at most 2^{-t}. Hence, by a union bound, the probability that the event $\tilde{E}_B$ occurs is at most $q \cdot b \cdot 2^{-t}$. Finally, by our setting of $t = 11 + \log b + \log q$, we have that this probability is at most 2^{-11}. We conclude that the right-hand term satisfies

$$\Pr[E_B] \leq 2^{-11}.$$

To bound the left-hand term, we once more claim that

$$\Pr\left[\left(D^{\mathrm{Fix}_{B(m)\to v(m)}(z)}(i) = m_i\right) \cap \neg E_B\right] = \Pr\left[(D^z(i) = m_i) \cap \neg \tilde{E}_B\right].$$

Once more, this follows since when fixing m, i, the set of strings z on which $D^{\mathrm{Fix}_{B(m)\to v(m)}(z)}(i)$ does not make a query into $B(m)$ is identical the set of strings z on which $D^z(i)$ does not make a query into $B(m)$, and fixing each such string z induces the same query pattern and values for both $D^z(i)$ and $\mathrm{Fix}_{B(m)\to v(m)}(z)$. Thus we conclude that for fixed m, i, the set of strings z which lead to the event $\left(D^{\mathrm{Fix}_{B(m)\to v(m)}(z)}(i) = m_i\right) \cap \neg E_B$ is identical to the set of strings z that lead to the event $(D^z(i) = m_i) \cap \neg \tilde{E}_B$, and so the probabilities are the same. It thus suffices to bound the probability of the event $(D^z(i) = m_i) \cap \neg \tilde{E}_B$.

To bound the probability that $(D^z(i) = m_i) \cap \neg \tilde{E}_B$ occurs, we note that it is at most the probability that $D^z(i) = m_i$ occurs. Recalling our assumption that MD is uniform over a set of size $2^{k-k^{0.99}}$, by Lemma 2.1, this latter probability is at most 0.5001. So the left-hand term satisfies that

$$\Pr\left[\left(D^{\mathrm{Fix}_{B(m)\to v(m)}(z)}(i) = m_i\right) \cap \neg E_B\right] \leq 0.5001.$$

Summing up the two probabilities, we get that

$$\Pr_{m\leftarrow \mathrm{MD}, z\leftarrow \mathrm{BSC}^n_{\frac{1}{2}}, i\leftarrow[k]}\left[D^{\mathrm{Fix}_{B(m)\to v(m)}(z)}(i) = m_i\right] \leq 0.5001 + 2^{-11} \leq 0.501,$$

which concludes the proof of the claim. □

Claim 3.23. *If there exists $j \in [b]$, such that $H_\infty(h_j(\mathrm{MD})) < t$ then there exist:*

- *A subset $\bar{S} \subseteq S$ such that $\frac{|\bar{S}|}{|S|} \geq 2^{-(t+1)}$.*

- *A codeword index $h' \in [n]$ and a value $v' \in \{0,1\}$ such that for every message $m \in \bar{S}$, the j-th index in $B(m)$ is $h_j(m) = h'$ and the value of the corresponding coordinate in v is $(v(m))(h') = v'$.*

Proof. Since $H_\infty(h_j(\mathrm{MD})) < t$, there must exist $h' \in [n]$ such that $\Pr[h_j(\mathrm{MD}) = h'] \geq 2^{-t}$. Let $v' \in \{0,1\}$ be such that $\Pr[(v(\mathrm{MD}))(h') = v' | h_j(\mathrm{MD}) = h'] \geq 1/2$. In other words, v' is the more probable value taken by $v(m)$ at the index h', conditioned on $m \leftarrow \mathrm{MD}$ satisfying that the j-th index in $B(m)$ is $h_j(m) = h'$. So we get that with probability at least $2^{-(t+1)}$, both events $h_j(\mathrm{MD}) = h'$ and $v(\mathrm{MD})(h') = v'$ hold. This event can be thought of in turn as a subset $\bar{S}$ of messages of density at least $2^{-(t+1)}$ inside S. □

3.3 Proof of Theorem 1.6

In this section we prove Theorem 1.6. The Theorem will follow by from next lemma.

Lemma 3.24. *There exist universal constants $\beta > 0$ and $c > 1$ such that if If* Dec *is a $(\frac{1}{2} - \epsilon, q, L, 1/2k)$-ARLLD for* Enc $: \{0,1\}^k \to \{0,1\}^n$*, and $L \leq \beta \cdot 2^k$, then setting $n' = cn$, there exists a circuit $C : \{0,1\}^{n'} \to \{0,1\}$ of size $O(L \cdot k \cdot 2^{2q})$ and depth 3, such that:*

- $\Pr_{z \leftarrow \mathrm{BSC}^{n'}_{\frac{1}{2}-2\epsilon}}[C(z) = 1] \geq 0.99$.
- $\Pr_{z \leftarrow \mathrm{BSC}^{n'}_{\frac{1}{2}}}[C(z) = 1] \leq 0.01$.

We first prove that Lemma 3.24 implies Theorem 1.6.

Proof of Theorem 1.6. Consider a $(\frac{1}{2} - \epsilon, q, L, \delta)$-LLD Dec for Enc $: \{0,1\}^k \to \{0,1\}^n$, where $\delta \leq 1/3$. 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 $(\frac{1}{2} - \epsilon, q' = O(q \log k), L, 1/20k)$-LLD for Enc $: \{0,1\}^k \to \{0,1\}^n$.

By our requirements on ϵ, we can use Proposition 3.2 to show that there exists a $(\frac{1}{2} - \epsilon, q', L, 1/2k)$-ARLLD for Enc $: \{0,1\}^k \to \{0,1\}^n$. By Lemma 3.24, there exists a circuit C of of size

$$s = O(L \cdot k \cdot 2^{2 \cdot q'}) = L \cdot k \cdot 2^{O(q \log k)}$$

and depth 3 that distinguishes $\mathrm{BSC}^{n'}_{\frac{1}{2}-2\epsilon}$ from $\mathrm{BSC}^{n'}_{\frac{1}{2}}$. By Corollary 2.5 such a circuit must have size:

$$s \geq \exp(\Omega\left((1/\epsilon)^{1/2}\right)).$$

This implies that

$$q \geq \frac{\Omega\left((1/\epsilon)^{1/2}\right) - \log L - \log k}{O(\log k)}.$$

□

In the remainder of this section we prove Lemma 3.24. Let Dec be a $(\frac{1}{2} - \epsilon, q, L, 1/2k)$-ARLLD for Enc $: \{0,1\}^k \to \{0,1\}^n$. We will construct the circuit C in the following sequence of claims:

Definition 3.25. *For every $i \in [k]$ and $j \in [L]$, let $A_{i,j} : \{0,1\}^n \to \{0,1\}$ be defined by: $A_{i,j}(w) = \mathrm{Dec}^w(i, j)$.*

Claim 3.26. *For every $i \in [k]$ and $j \in [L]$, there exist CNF circuits $A^T_{i,j} : \{0,1\}^n \to \{0,1\}$ and $A^F_{i,j} : \{0,1\}^n \to \{0,1\}$ of size $O(q \cdot 2^q)$ such that for every $w \in \{0,1\}^n$, $A^T_{i,j}(w) = A_{i,j}(w)$, and $A^F_{i,j}(w) = 1 - A_{i,j}(w)$.*

Proof. For fixed i, j, we can view the computations $A_{i,j}(w) = \mathrm{Dec}^w(i,j)$ as a depth q decision tree that makes queries to w. Such a decision tree can be simulated by a size $O(q \cdot 2^q)$ DNF, which is a disjunction over the at most 2^q accepting paths of the tree, where each path is a conjunction of q literals. This can also gives a CNF of the same size for $1 - A_{i,j}$. The same argument can be repeated for $1 - A_{i,j}$ giving a CNF of the same size for $A_{i,j}$. □

Definition 3.27. *For every $m \in \{0,1\}^k$ we define the circuit $C_m : \{0,1\}^n \to \{0,1\}$ that is hardwired with the message $m \in \{0,1\}^k$, and the encoding $\mathrm{Enc}(m)$. Given input $z \in \{0,1\}^n$, the circuit C_m acts as follows:*

- *Prepare $w = \mathrm{Enc}(m) \oplus z$.*
- *For every $i \in [k]$ and $j \in [L]$ compute $A_{i,j}(w)$, and compute $b_{i,j} \in \{0,1\}$ which answers whether $A_{i,j}(w) = m_i$.*
- *For every j, compute b_j which is the conjunction of $b_{1,j}, \ldots, b_{k,j}$.*
- *Compute the disjunction of $b_1, \ldots, b_L$ and output it.*

Claim 3.28. *For every $m \in \{0,1\}^k$ the circuit C_m can be implemented in size $O(k \cdot L \cdot 2^{2q})$ and depth 3. Furthermore, for every $m \in \{0,1\}^k$, and $z \in \{0,1\}^n$, $C_m(z) = 1$ iff there exists $j \in [L]$ such that for every $i \in [k]$, $\mathrm{Dec}^{\mathrm{Enc}(m) \oplus z}(i,j) = m_i$.*

Proof. It is immediate that the circuit C_m performs the task described in the claim. We now explain how to implement the circuit in small size and depth.

The string m is of length k. We note that when using $\mathrm{Enc}(m)$ to prepare w, we only need to have $\mathrm{Enc}(m)$ at coordinates ℓ such that there exists i, j such that $A_{i,j}(w)$ depends on the ℓ'th input. As each circuit $A^T_{i,j}$ is a circuit of size $O(q \cdot 2^q)$ it depends on at most $O(q \cdot 2^q)$ input bits. Thus, C_m only requires $O(k \cdot L \cdot q \cdot 2^q)$ bits of $\mathrm{Enc}(m)$. Overall, the size of the advice of C_m is $O(k \cdot L \cdot q \cdot 2^q)$.

Computing every bit of w amounts to at most one negation gate, and does not increase the depth. For every $i \in [k]$ and $j \in [L]$, we want to compute the bit $b_{i,j}$ which is one iff $A_{i,j}(w) = m_i$. Note that if $m_i = 1$ then $b_{i,j} = A^T_{i,j}(w)$ and if $m_i = 0$ then $b_{i,j} = A^F_{i,j}(w)$. As m_i is a fixed constant, this gives that for every $i \in [k]$ and $j \in [L]$, $b_{i,j}$ can be computed by a CNF of size $O(q \cdot 2^q)$ that is applied on the input z. For every $j \in [L]$, computing b_j is done using a single AND gate, and, since the top gate of the CNF computing b_i is also an AND gate, this does not increase the depth. Finally, computing the output adds a top OR gate. overall, the depth is 3 and the size is bounded by:

$$O(k \cdot L \cdot q \cdot 2^q) \leq O(k \cdot L \cdot 2^{2q}).$$ □

The definition of ARLLD immediately gives that:

Claim 3.29. $\Pr_{m \leftarrow \{0,1\}^k, z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}-2\epsilon}}[C_m(z) = 1] \geq \frac{1}{6}$.

On the other hand, we can show that:

Claim 3.30. $\Pr_{m \leftarrow \{0,1\}^k, z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}}}[C_m(z) = 1] \leq L \cdot 2^{-k}$.

Proof. The first step of $C_m(z)$ is to prepare $w = \mathrm{Enc}(m) \oplus z$. However, for $p = \frac{1}{2}$, and $z \leftarrow \mathrm{BSC}_p^n$, we have that $w = \mathrm{Enc}(m) \oplus z$ is uniformly chosen, and independent of m. This means that the bits $A_{i,j}(w)$ for $i \in [k]$ and $j \in [L]$ are independent of m. Consequently, for every $j \in [L]$, we have that:

$$\begin{aligned}\Pr_{m \leftarrow \{0,1\}^k, z \leftarrow \mathrm{BSC}_{\frac{1}{2}}^n, w = \mathrm{Enc}(m) \oplus z}[m = A_{1,j}(w) \circ \ldots \circ A_{k,j}(w)] &\leq \Pr_{m \leftarrow \{0,1\}^k, w \leftarrow \{0,1\}^n}[m = A_{1,j}(w) \circ \ldots \circ A_{k,j}(w)] \\ &\leq 2^{-k}\end{aligned}$$

By a union bound over all choices of $j \in [L]$, we have that:

$$\Pr_{m \leftarrow \{0,1\}^k, z \leftarrow \mathrm{BSC}_{\frac{1}{2}}^n, w = \mathrm{Enc}(m) \oplus z}[\exists j : \text{s.t. } m = A_{1,j}(w) \circ \ldots \circ A_{k,j}(w)] \leq L \cdot 2^{-k}$$

It follows that:

$$\Pr_{m \leftarrow \{0,1\}^k, z \leftarrow \mathrm{BSC}_{\frac{1}{2}}^n}[C_m(z) = 1] \leq L \cdot 2^{-k},$$

as required. □

We are finally ready to prove Lemma 3.24.

Proof of Lemma 3.24. By our choices, we have that $0 < L \cdot 2^{-k} \leq \beta < 1$. By applying Markov's inequality to Claims 3.29 and 3.30, we can see that:

$$\Pr_{m \leftarrow \{0,1\}^k}\left[\Pr_{z \leftarrow \mathrm{BSC}_{\frac{1}{2}-2\epsilon}^n}[C_m(z) \neq 1] > \frac{9}{10}\right] < \frac{5/6}{9/10} \leq \frac{95}{100}.$$

$$\Pr_{m \leftarrow \{0,1\}^k}\left[\Pr_{z \leftarrow \mathrm{BSC}_{\frac{1}{2}}^n}[C_m(z) = 1] > \sqrt{\beta}\right] < \sqrt{\beta}.$$

Therefore, for a sufficiently small constant $\beta > 0$, by a union bound, we get that there exists $m \in \{0,1\}^k$ satisfying both:

$$\begin{aligned}\Pr_{z \leftarrow \mathrm{BSC}_{\frac{1}{2}-2\epsilon}^n}[C_m(z) = 1] &\geq \frac{1}{10}, \\ \Pr_{z \leftarrow \mathrm{BSC}_{\frac{1}{2}}^n}[C_m(z) = 1] &\leq \sqrt{\beta},\end{aligned}$$

It is possible to amplify these thresholds to 0.99 and 0.01 as follows: Let c be a constant that we choose later, and let $n' = cn$. Consider a circuit $C : \{0,1\}^{n'} \to \{0,1\}$ that when receiving input $x \in \{0,1\}^{n'}$, treats it as c strings $x_1, \ldots, x_c \in \{0,1\}^n$. C will apply C_m on each of the c strings, and the final output is the OR of the results. As the top gate of C_m is an OR gate, adding an additional OR gate, does not increase the depth of the circuit. The size of the circuit increases by a constant factor.

We can view $z \leftarrow \mathrm{BSC}_p^{n'}$ as obtained by concatenating c strings $z_1, \ldots, z_c$, where each of them is sampled uniformly and independently at random from BSC_p^n. Hence, the event $C(z) = 1$ is identical to $\bigvee_{\ell \in [c]} C_m(z_\ell) = 1$. Therefore,

$$\Pr_{z \leftarrow \mathrm{BSC}_{\frac{1}{2}-2\epsilon}^{n'}}[C(z) = 1] \geq 1 - (\frac{9}{10})^c,$$

and

$$\Pr_{z \leftarrow \mathrm{BSC}_{\frac{1}{2}}^{n'}}[C(z) = 1] \leq c \cdot \sqrt{\beta}.$$

By choosing c to be sufficiently large, we can see that $1 - (\frac{9}{10})^c \geq 0.99$. By choosing β to be sufficiently small, we also have that $c\sqrt{\beta} \leq 0.01$. □

4 Limitations on black-box proofs for hard-core predicates

In this section, we present our results regarding the limitations on black-box proofs for hard-core predicate theorems. In Section 4.1, we state our results for functions that are hard to compute, give a formal restatement of Theorem 1.9, and prove the theorem. In Section 4.2, we state our results for functions that are hard to invert, give a formal restatement of Theorem 1.13, and prove the theorem.

4.1 The case of functions that are hard to compute

4.1.1 The model for black-box proofs

In this section, we state and explain our model for black-box proofs for hard core predicates, in the setting of functions that are hard to compute. The formal definition is given in Definition 4.1. Below, we provide a detailed explanation for the considerations made while coming up with the formal definition. The reader can skip directly to the formal definition if he wishes.

Explanation of the model: Recall that (as explained in Section 1.2.1) the Goldreich-Levin theorem (stated precisely in Theorem 1.7) has the following form:

- We are given an arbitrary hard function $g : \{0,1\}^\ell \to \{0,1\}^\ell$. (Intuitively, it is assumed that it is hard to compute g with success probability ρ).
- There is a specified construction that transforms g into a predicate $g^{\mathrm{pred}} : \{0,1\}^{\ell'} \to \{0,1\}$ for some ℓ' related to ℓ. (Intuitively, we will want to argue that g^{pred} is a hard-core predicate that is hard to compute with success $\frac{1}{2} + \epsilon$).

  We will model this construction as a map Con, which, given g produces g^{pred}. We place *no limitations* on the map Con (and, in particular, do not require that g^{pred} can be efficiently computed if g is). This only makes our results stronger.

  In the case of Theorem 1.7, we have that: $\mathrm{Con}(g) = g^{\mathrm{pred}}$ where $\ell' = 2\ell$ and we think of the ℓ'-bit long input of g^{pred} as two strings $x, r \in \{0,1\}^\ell$, setting:

$$g^{\mathrm{pred}}(x, r) = \mathrm{Enc}^{\mathrm{Had}}(g(x))_r = (\sum_{i \in [\ell]} g(x)_i \cdot r_i) \mod 2.$$

- We model the proof showing that g^{pred} is a hard-core predicate in the following way: The proof is a pair $(\mathrm{Con}, \mathrm{Red})$ where $\mathrm{Red}^{(\cdot)}$ is an oracle procedure, such that when $\mathrm{Red}^{(\cdot)}$ receives oracle access to an "adversary" $h : \{0,1\}^{\ell'} \to \{0,1\}$ that breaks the security of g^{pred}, we have that Red^h breaks the security of g. More precisely, we require that: for every $g : \{0,1\}^\ell \to \{0,1\}^\ell$ and for every $h : \{0,1\}^{\ell'} \to \{0,1\}$ such that:

$$\Pr_{x \leftarrow U_{\ell'}}[h(x) = g^{\mathrm{pred}}(x)] \geq \frac{1}{2} + \epsilon,$$

it holds that:

$$\Pr_{x \leftarrow U_\ell}[\mathrm{Red}^h(x) = g(x)] \geq \rho.$$

- In the actual definition, we will allow the reduction to have more power (which only makes our results stronger). As we are aiming to prove a result on circuits (which are allowed to use nonuniform advice) we will allow the reduction to receive an advice string α of length t, where, this advice string can depend on g and h. This leads to the following strengthening of the requirement above. Namely, we will require that: for every $g : \{0,1\}^\ell \to \{0,1\}^\ell$ and for every $h : \{0,1\}^{\ell'} \to \{0,1\}$, that:

  $$\Pr_{x \leftarrow U_{\ell'}}[h(x) = g^{\mathrm{pred}}(x)] \geq \frac{1}{2} + \epsilon,$$

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

  $$\Pr_{x \leftarrow U_\ell}[\mathrm{Red}^h(x, \alpha) = g(x)] \geq \rho.$$

  We remark that in many related settings (for example, "hardness amplification"; see [SV10, GSV18], for a discussion) known proofs by reduction *critically make use* of the ability to introduce nonuniformity, and so, we feel that when ruling out black-box proofs in scenarios involving circuits, it is necessary to consider nonuniform black-box reductions.
- We make no restrictions on the complexity of the procedure $\mathrm{Red}^{(\cdot)}$, except for requiring that it makes at most q queries to its oracle (for some parameter q). Our black-box impossibility results will follow from proving lower bounds on q.

Formal definition: We now give a formal definition of our model for black-box proofs for hard-core predicates.

Definition 4.1 (Nonuniform black-box proofs for hard-core predicates for hard-to-compute functions)**.** *A pair* $(\mathrm{Con}, \mathrm{Red})$ *is a nonuniform black-box proof for hard-core predicates for hard-to-compute functions with parameters* $\ell, \ell', \rho, \epsilon$*, that uses* q *queries, and* t *bits of advice if:*

- Con *is a* construction map *which given a function* $g : \{0,1\}^\ell \to \{0,1\}^\ell$*, produces a function* $\mathrm{Con}(g) = g^{\mathrm{pred}}$*, where* $g^{\mathrm{pred}} : \{0,1\}^{\ell'} \to \{0,1\}$*.*
- $\mathrm{Red}^{(\cdot)}$ *is a* reduction*, that is an oracle procedure that, given oracle access to a function* $h : \{0,1\}^{\ell'} \to \{0,1\}$*, makes at most* q *queries to its oracle.*

Furthermore, for every functions $g : \{0,1\}^\ell \to \{0,1\}^\ell$ *and* $h : \{0,1\}^{\ell'} \to \{0,1\}$ *such that:*

$$\Pr_{x \leftarrow U_{\ell'}}[h(x) = g^{\mathrm{pred}}(x)] \geq \frac{1}{2} + \epsilon,$$

there exists $\alpha \in \{0,1\}^t$*, such that:*

$$\Pr_{x \leftarrow U_\ell}[\mathrm{Red}^h(x, \alpha) = g(x)] \geq \rho.$$

The role of the number of queries, and black-box impossibility results: We now explain the role of the parameter q (that measures the number of queries made by Red) and why lower bounds on q translate into black-box impossibility results.

For this purpose, it is illustrative to examine the argument showing that nonuniform black-box proofs yield hard-core predicates: When given a pair $(\mathrm{Con}, \mathrm{Red})$ that is a nonuniform black-box proof for hard-core predicates for hard-to-compute functions with parameters $\ell, \ell', \rho, \epsilon$, that uses q queries, and t bits of advice, we obtain that for any function $g : \{0,1\}^\ell \to \{0,1\}^\ell$, if there exists a circuit $C' : \{0,1\}^{\ell'} \to \{0,1\}$ of size s' such that:

$$\Pr_{x \leftarrow U_{\ell'}}[C'(x) = g^{\mathrm{pred}}(x)] \geq \frac{1}{2} + \epsilon,$$

then there exists $\alpha \in \{0,1\}^t$, such that:

$$\Pr_{x \leftarrow U_\ell}[\mathrm{Red}^{C'}(x, \alpha) = g(x)] \geq \rho.$$

Note that if the reduction Red can be implemented by a circuit of size r, then the circuit $C(x) = \mathrm{Red}^{C'}(x, \alpha)$ is a circuit of size:

$$s = r + t + q \cdot s'$$

that computes g with success probability ρ.

It follows that in a black-box proof, with q queries, and t bits of advice, we get a hard-core theorem that needs to assume that the original function g has hardness against circuits of size s, for:

$$s \geq q + t.$$

4.1.2 Precise statements of limitations

Our main result on black-box proofs for hard-core predicates in the setting of functions that are hard to compute is the following theorem.

Theorem 4.2. *There exists a universal constant $\beta > 0$ such that for every sufficiently large ℓ and ℓ' we have that if $(\mathrm{Con}, \mathrm{Red})$ is a nonuniform black-box proof for hard-core predicates for hard-to-compute functions with parameters $\ell, \ell', \rho, \epsilon$, that uses q queries, and t bits of advice, and furthermore $\epsilon \geq \frac{1}{2^{\ell'/3}}$, $t \leq 2^{\ell/3}$ and $\rho \geq \frac{1}{2^{\ell/3}}$, then*

$$q \geq \Omega(\frac{1}{\epsilon^\beta}) - O(t + \ell).$$

We now explain why Theorem 4.2 implies the informal statement made in Theorem 1.9. Recall that in Section 4.1.1 we explained that when using a nonuniform black-box proof to obtain a hard-core predicate, we get a hard-core predicate theorem in which $s \geq q + t$.

Theorem 4.2 implies that for $s > \ell^{2/\beta}$ it is impossible for such a proof to establish $\epsilon = 1/s^{\frac{2}{\beta}}$ (even if ρ is very small). This follows as otherwise, using the fact that $s \geq q + t \geq t$, we get that:

$$q \geq \Omega(\frac{1}{\epsilon^\beta}) - O(t + \ell) \geq \Omega(s^2) - O(t) > s,$$

which is a contradiction to $s \geq q + t \geq q$. In particular, the parameter setting considered in Theorem 1.9, in which $s = 2^{o(\ell)}$ and $\epsilon = \frac{1}{s^{\omega(1)}}$, is impossible to achieve.

4.1.3 Proof of Theorem 4.2

Theorem 4.2 will follow from the next lemma, showing that a proof with small q can be transformed into a small constant depth circuit for the coin problem.

Lemma 4.3. *There exists a universal constant d such that for every sufficiently large ℓ and ℓ' we have that if $(\mathrm{Con}, \mathrm{Red})$ is a nonuniform black-box proof for hard-core predicates for hard-to-compute functions with parameters $\ell, \ell', \rho, \epsilon$, that uses q queries, and t bits of advice, and furthermore $\epsilon \geq \frac{1}{2^{\ell'/3}}$, $t \leq 2^{\ell/3}$ and $\rho \geq \frac{1}{2^{\ell/3}}$ then there exists a circuit C of size $s = \mathrm{poly}(2^q, 2^\ell, 2^t)$ and depth d such that:*

- $\Pr_{z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}-2\epsilon}}[C(z) = 1] \geq 0.99$.
- $\Pr_{z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}}}[C(z) = 1] \leq 0.01$.

We first show that Theorem 4.2 follows from Lemma 4.3.

Proof of Theorem 4.2. The theorem follows directly from Lemma 4.3 and Corollary 2.5, which give that:

$$s = \mathrm{poly}(2^q, 2^\ell, 2^t) \geq \exp(\Omega(d \cdot (1/\epsilon)^{1/(d-1)})),$$

The statement of Theorem 4.2 follows by taking the logarithm on both sides and setting $\beta = 1/(d-1)$. □

In the remainder of this section we prove Lemma 4.3. Let $(\mathrm{Con}, \mathrm{Red})$ be a nonuniform black-box proof for hard-core predicates for hard-to-compute functions with parameters $\ell, \ell', \rho, \epsilon$, that uses q queries, and t bits of advice. Throughout this section we assume that the requirements made in Lemma 4.3 are met.

We will identify functions $h : \{0,1\}^\ell \to \{0,1\}$ with strings $h \in \{0,1\}^{2^\ell}$. More precisely, we fix some ordering on strings $x \in \{0,1\}^\ell$ and then, the value of string h at position x is the function h applied on x. We will use h to denote these two objects (both the function and the string) and this means that a function h can be given as an argument to a function that receives strings of length 2^ℓ.

High level description of the proof of Lemma 4.3. The proof will use the same structure as the proof of Lemma 3.24, which was the main technical lemma in the proof of Theorem 1.6. Loosely speaking, the reduction Red plays the role of a local list-decoder, the function g plays the role of the message, the construction map Con plays the role of the encoder, so that the function g^{pred} (viewed as a $2^{\ell'}$ bit long string) plays the role of the encoding of the message.

Imitating the approach used in the proof of Lemma 3.24, we will consider the function (or string) $h \in \{0,1\}^{2^{\ell'}}$ defined by $h = g^{\mathrm{pred}} \oplus z$ where $z \leftarrow \mathrm{BSC}^{2^{\ell'}}_p$ for $p = \frac{1}{2}$ and $p = \frac{1}{2} - 2\epsilon$. We will try to show that when the function g is chosen uniformly, then for $p = \frac{1}{2} - 2\epsilon$, Red^h has to succeed, and for $p = \frac{1}{2}$, Red^h cannot succeed. We will then leverage this difference to produce a constant depth circuit of size roughly 2^q that distinguishes $\mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}-2\epsilon}$ from $\mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}}$.

However, there are some complications. When the reduction Red^h succeeds for $p = \frac{1}{2} - 2\epsilon$, we only have that there exists an α with which it computes g with success ρ (which is extremely small) and is not much larger than the success probability of Red^h for $p = \frac{1}{2}$ (which we will show is less than $\rho/10$).

At first glance, this seems like a problem, as in general, in order to distinguish success probability $a + \rho/10$ from $a + \rho$ for an arbitrary value of $a \in [0,1]$, constant depth circuits need to have size that is $2^{(1/\rho)^{\Omega(1)}}$ which is much too large for our purposes. Fortunately, for $a = 0$ (that is for the task of distinguishing success probability $\rho/10$ from ρ) it is possible to distinguish with circuits of size $\mathrm{poly}(1/\rho)$. The formal statement of this is given in Lemma 4.4.

We now return to the formal proof, starting with the following lemma.

Lemma 4.4. *There exists a universal constant d, such that every n, ρ there exists a circuit $D^n_\rho : \{0,1\}^n \to \{0,1\}$ of size $\mathrm{poly}(n/\rho)$ and depth d, such that for every $x \in \{0,1\}^n$:*

- *If* $\mathrm{weight}(x) \geq \rho$ *then* $D^n_\rho(x) = 1$.
- *If* $\mathrm{weight}(x) \leq \rho/10$ *then* $D^n_\rho(x) = 0$.

We note that by the lower bound of Razborov and Smolensky [Raz87, Smo87] small constant-depth circuits cannot compute the majority function. Nevertheless, by the results of Ajtai [Ajt83] (stated in Theorem 2.3) small constant-depth circuits can compute *approximate majority*. That is, they can distinguish between strings with relative Hamming weight $\geq P$ from strings with relative Hamming weight $\leq p$ whenever $p < P$ are constants. The proof of the lemma uses circuits for approximate majority.

Proof of Lemma 4.4. We first construct a distribution over circuits that achieves the goal. Let $a > 1$ be a constant that we choose later. Let $n' = \frac{a}{\rho}$. Let us consider the experiment E_1 in which a uniform multi-set S of $[n]$ of size n' is chosen uniformly. (That is $i_1, \ldots, i_{n'}$ are chosen independently and uniformly from $[n]$ and S is the multi-set $\{i_1, \ldots, i_{n'}\}$). Note that for every $x \in \{0,1\}^n$,

- If $\mathrm{weight}(x) \geq \rho$ then $\mathbb{E}_{S \leftarrow E_1}[\sum_{i \in S} x_i] \geq a$.
- If $\mathrm{weight}(x) \leq \rho/10$ then $\mathbb{E}_{S \leftarrow E_1}[\sum_{i \in S} x_i] \leq a/10$.

For every multi-set $S \subseteq [n]$ of size n' we consider the circuit $C_S : \{0,1\}^n \to \{0,1\}$ that works as follows:

- For every choice of $a/5$ elements $j_1, \ldots, j_{a/5}$ in S, compute the disjunction of $x_{j_1}, \ldots, x_{j_{a/5}}$.
- Compute the conjunction of the $(n')^{a/5}$ bits from the previous item.

This gives that there exists a constant c, that depends on a such that C_S is a circuit of size $\mathrm{poly}(1/\rho)$ and depth 2. Furthermore $C_S(x)$ answers one iff $\sum_{i \in S} x_i \geq a/5$.

By a (multiplicative) Chernoff bound,[8] it follows that there exists a universal constant $\eta > 0$ such that for a sufficiently large constant a, for every $x \in \{0,1\}^n$:

- If $\mathrm{weight}(x) \geq \rho$ then $\Pr_{S \leftarrow E_1}[\frac{1}{n'} \cdot \sum_{i \in S} x_i \leq a/5] \leq 2^{-\eta a} \leq 1/3$ which implies $\Pr_{S \leftarrow E_1}[C_S(x) = 1] \geq 2/3$.
- If $\mathrm{weight}(x) \leq \rho/10$ then $\Pr_{S \leftarrow E_1}[\frac{1}{n'} \cdot \sum_{i \in S} x_i \geq a/5] \leq 2^{-\eta a} \leq 1/3$ which implies $\Pr_{S \leftarrow E_1}[C_S(x) = 1] \leq 1/3$.

By taking $t = O(n)$ independent copies of C_S and computing approximate majority, we can reduce the error probability from $1/3$ to 2^{-2n}, and apply Adleman's argument to obtain a single circuit of size $n \cdot \mathrm{poly}(1/\rho)$ and constant depth. Details follow:

For $t = O(n)$ multi-sets $S_1, \ldots, S_t \subseteq [n]$ of size n', we consider the circuit $C_{S_1,\ldots,S_t}(x)$ which for every $i \in [t]$ computes $b_i = C_{S_i}(x)$ and then computes approximate majority (with parameters $p = 0.49$ and $P = 0.51$) on the string $b = b_1, \ldots, b_t$. Note that each such circuit has constant depth and size $\mathrm{poly}(n/\rho)$. Let E_2 denote the experiment in which the t sets $S_1, \ldots, S_t$ are chosen independently, where each $S_i \subseteq [n]$ is a uniformly chosen multi-set of size n'. By a Chernoff bound, it follows that for every $x \in \{0,1\}^n$:

- If $\mathrm{weight}(x) \geq \rho$ then $\Pr_{S_1,\ldots,S_t \leftarrow E_2}[C_{S_1,\ldots,S_t}(x) = 1] \geq 1 - 2^{-2n}$.
- If $\mathrm{weight}(x) \leq \rho/10$ then $\Pr_{S_1,\ldots,S_t \leftarrow E_2}[C_{S_1,\ldots,S_t}(x) = 1] \leq 2^{-2n}$.

[8] Specifically, we mean the following version of the Chernoff bound: If X is the sum of n independent variables $X_1, \ldots, X_n \in [0,1]$, and $\mathbb{E}(X) = \mu$ then for every $0 \leq \delta \leq 1$, $\Pr[|X - \mu| \geq \delta\mu] \leq 2e^{-\frac{\delta^2 \cdot \mu}{3}}$.

By a union bound over all 2^n choices of $x \in \{0,1\}^n$ we obtain that there exist sets $S'_1, \ldots, S'_t$ such that setting $D^\rho_n = C_{S'_1,\ldots,S'_t}$, we obtain the circuit guaranteed in the statement of the theorem. □

We will prove Lemma 4.3 using the following sequence of claims. The overall structure of the argument is similar to the proof of Lemma 3.4.

Claim 4.5. *For every $x \in \{0,1\}^\ell$ and every $\alpha \in \{0,1\}^t$, there exists a circuit of size $\ell \cdot q \cdot 2^q$ and depth 2, $A_{x,\alpha} : \{0,1\}^{2^{\ell'}} \to \{0,1\}^\ell$ such that for every $h : \{0,1\}^{\ell'} \to \{0,1\}$,*

$$A_{x,\alpha}(h) = \mathrm{Red}^h(x,\alpha).$$

Proof. For every $x \in \{0,1\}^\ell$ and every $\alpha \in \{0,1\}^t$, the function $A_{x,\alpha}(h)$ can be computed by a depth q decision tree, that has outputs of length ℓ bits. Each output bit of this function can be computed by a DNF of size $O(q \cdot 2^q)$ and overall, the function can be computed by a depth 2 circuit of size $\ell \cdot q \cdot 2^q$ as required. □

Claim 4.6. *There exists a universal constant d such that for every $g : \{0,1\}^\ell \to \{0,1\}^\ell$, there exists a circuit $C_g : \{0,1\}^{2^{\ell'}} \to \{0,1\}$ of size $\mathrm{poly}(2^q, 2^\ell, 2^t)$ and depth d such that the following holds for every $z \in \{0,1\}^{2^{\ell'}}$:*

- *If there exists $\alpha \in \{0,1\}^t$ such that $\Pr_{x \leftarrow U_\ell}[\mathrm{Red}^{g^{\mathrm{pred}} \oplus z}(x,\alpha) = g(x)] \geq \rho$ then $C_g(z) = 1$.*
- *If for all $\alpha \in \{0,1\}^t$, $\Pr_{x \leftarrow U_\ell}[\mathrm{Red}^{g^{\mathrm{pred}} \oplus z}(x,\alpha) = g(x)] \leq \rho/10$ then $C_g(z) = 0$.*

Proof. The circuit C_g will be hardwired with g and $g^{\mathrm{pred}} = \mathrm{Con}(g)$. Upon receiving an input $z \in \{0,1\}^{2^{\ell'}}$ it will act as follows:

- Prepare $w = g^{\mathrm{pred}} \oplus z$. (Here we think of g^{pred}, w, z as a strings in $\{0,1\}^{2^{\ell'}}$.
- For every $x \in \{0,1\}^\ell$ and $\alpha \in \{0,1\}^t$ compute $A_{x,\alpha}(w)$, and compute: $b_{x,\alpha} \in \{0,1\}$ defined by:

$$b_{x,\alpha} = \begin{cases} 0 & A_{x,\alpha}(w) \neq g(x) \\ 1 & A_{x,\alpha}(w) = g(x) \end{cases}$$

- For every $\alpha \in \{0,1\}^t$, let v_α denote the 2^ℓ bit long concatenation of all bits $(b_{x,\alpha})_{x \in \{0,1\}^\ell}$ (fixing some order on $x \in \{0,1\}^\ell$), and compute
$$b_\alpha = D^{2^\ell}_\rho(v_\alpha),$$
where $D^{2^\ell}_\rho$ is the circuit guaranteed in Lemma 4.4.
- Compute the disjunction of the 2^t bits $(b_\alpha)_{\alpha \in \{0,1\}^t}$ and output it.

It is immediate that the circuit C_g performs the task specified in the lemma. We now explain how to implement the circuit in small size and depth.

The function g can be described using $\ell \cdot 2^\ell$ bits. We note that when using the string g^{pred} to prepare w, we only need to have g_{pred} at coordinates $y \in \{0,1\}^{\ell'}$ such that there exists x, α such that $A_{x,\alpha}(w)$ depends on w_y. As each circuit $A_{x,\alpha}$ is a circuit of size $O(\ell \cdot q \cdot 2^q)$ it depends on at most $O(\ell \cdot q \cdot 2^q)$ input bits. Thus, going over all choices of $x \in \{0,1\}^\ell$ and $\alpha \in \{0,1\}^t$, C_g only requires $O(2^t \cdot 2^\ell \cdot \ell \cdot q \cdot 2^q)$ bits of g^{pred}. Overall, the size of the advice of C_g is $O(2^t \cdot 2^{2\ell} \cdot 2^{2q})$. The circuit C_g is constant depth by construction, and its size is indeed:

$$\mathrm{poly}(2^q, 2^\ell, 2^t, 1/\rho) = \mathrm{poly}(2^q, 2^\ell, 2^t),$$

since, by the requirement on ρ we have that $\rho \geq 2^{-\ell}$. □

We will consider the case where $g : \{0,1\}^\ell \to \{0,1\}^\ell$ is a uniformly chosen function, and will analyze the behavior of C_g on $z \leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}-2\epsilon}$ and on $z \leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}}$.

Definition 4.7. *Let F_ℓ denote the set of all functions $g : \{0,1\}^\ell \to \{0,1\}^\ell$.*

Claim 4.8. $\Pr_{g \leftarrow F_\ell, z \leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}-2\epsilon}}[C_g(z) = 1] \geq 0.999$.

Proof. Imagine that $g \leftarrow F_\ell$ is already chosen and fixed, and let $g^{\mathrm{pred}} = \mathrm{Con}(g)$. By a Chernoff bound, with probability $1 - 2^{-\frac{1}{3}\cdot\epsilon^2\cdot 2^{\ell'}}$ over $z \leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}-2\epsilon}$, we have that the Hamming weight of z is at most $\frac{1}{2} - \epsilon$. This probability is larger than 0.999 by the requirement that $\epsilon \geq \frac{1}{2^{\ell'/3}}$ and that ℓ' is sufficiently large. Whenever this event occurs, we have that for $h = g^{\mathrm{pred}} \oplus z$, it holds that:

$$\Pr_{x \leftarrow U_{\ell'}}[h(x) = g^{\mathrm{pred}}(x)] \geq \frac{1}{2} + \epsilon.$$

Therefore, by Definition 4.1 there exists $\alpha \in \{0,1\}^t$, such that:

$$\Pr_{x \leftarrow U_\ell}[\mathrm{Red}^h(x, \alpha) = g(x)] \geq \rho.$$

By Claim 4.6 it follows that when whenever this occurs, $C_g(z) = 1$, and the claim follows. □

On the other hand, we can show that:

Claim 4.9. $\Pr_{g \leftarrow F_\ell, z \leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}}}[C_g(z) = 1] \leq 0.001$.

Proof. The first step of $C_g(z)$ is to prepare $w = g^{\mathrm{pred}} \oplus z$. However, for $p = \frac{1}{2}$, and $g \leftarrow F_\ell, z \leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}}$, we have that $w = g^{\mathrm{pred}} \oplus z$ is uniformly chosen, and independent of g. This means that the bits $A_{x,\alpha}(w)$ for $x \in \{0,1\}^\ell$ and $\alpha \in \{0,1\}^t$ are independent of g. It follows that for every $\alpha \in \{0,1\}^t$, we have that when choosing $g \leftarrow F_\ell, w \leftarrow \{0,1\}^{2^{\ell'}}$, and considering the function $v_\alpha \in F_\ell$ defined by $v_\alpha(x) = (A_{x,\alpha}(w))$, we have that v_α is independent of g. Therefore, the probability that v_α agrees with g in a $\rho' = \rho/10$ fraction of inputs $x \in \{0,1\}^\ell$ is at most:

$$\binom{2^\ell}{\rho' \cdot 2^\ell} \cdot \frac{1}{2^{\rho' \cdot \ell \cdot 2^\ell}} \leq \left(\frac{e}{\rho}\right)^{\rho' \cdot 2^\ell} \cdot \left(\frac{1}{2^\ell}\right)^{\rho' \cdot 2^\ell} \leq \frac{1}{2^{\ell/2}},$$

where the first inequality follows from $\binom{n}{k} \leq (\frac{en}{k})^k$ and the second inequality follows by our requirements on $\rho \geq 2^{-\ell/3}$.

For fixed $w \in \{0,1\}^{2^{\ell'}}$, the condition that v_α agrees with g in a $\rho/10$ fraction of inputs $x \in \{0,1\}^\ell$, can also be phrased as:

$$\Pr_{x \leftarrow U_\ell}[\mathrm{Red}^w(x, \alpha) = g(x)] \geq \rho/10.$$

It follows that for every $\alpha \in \{0,1\}^t$, we have that with probability $1 - 2^{-\ell/2}$ over the choice of $g \leftarrow F_\ell, z \leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}}$ we have that:

$$\Pr_{x \leftarrow U_\ell}[\mathrm{Red}^{g^{\mathrm{pred}} \oplus z}(x, \alpha) = g(x)] < \rho/10,$$

By a union bound over all 2^t choices of $\alpha \in \{0,1\}^t$, we have that with probability $1 - 2^t \cdot 2^{-\ell/2}$ over the choice of $g \leftarrow F_\ell, z \leftarrow \mathrm{BSC}_{\frac{1}{2}}^{2^{\ell'}}$ we have that for all $\alpha \in \{0,1\}^t$,

$$\Pr_{x \leftarrow U_\ell}[\mathrm{Red}^{g^{\mathrm{pred}} \oplus z}(x, \alpha) = g(x)] < \rho/10,$$

which by Claim 4.6 implies that $C_g(z) = 0$. Consequently, in order to complete the proof of the claim, it remains to verify that:

$$2^t \cdot 2^{-\ell/3} \leq 0.001.$$

This follows by the requirements that $t \leq 2^{\ell/3}$. □

We are finally ready to prove Lemma 4.3.

Proof of Lemma 4.3. From Claims 4.8 and 4.9, it follows that:

$$\Pr_{g \leftarrow F_\ell}\left[\Pr_{z \leftarrow \mathrm{BSC}_{\frac{1}{2}-2\epsilon}^{2^{\ell'}}}[C_g(z) = 0] > 0.01\right] \leq 0.1,$$

$$\Pr_{g \leftarrow F_\ell}\left[\Pr_{z \leftarrow \mathrm{BSC}_{\frac{1}{2}}^{2^{\ell'}}}[C_g(z) = 1] > 0.01\right] \leq 0.1.$$

Thus, by a union bound, there exists $g : \{0,1\}^\ell \to \{0,1\}^\ell$ such that

$$\Pr_{z \leftarrow \mathrm{BSC}_{\frac{1}{2}-2\epsilon}^{2^{\ell'}}}[C_g(z) = 1] \geq 0.99,$$

$$\Pr_{z \leftarrow \mathrm{BSC}_{\frac{1}{2}}^{2^{\ell'}}}[C_g(z) = 1] \leq 0.01.$$

This completes the proof of the lemma, as C_g satisfies all the other requirements as well. □

4.2 The case of functions that are hard to invert

4.2.1 The model for black-box proofs

In this section we state and explain our model for black-box proofs for hard core predicates, in the setting of functions that are hard to invert. The precise formal definition is given in concise form in Definition 4.10. Below, we provide a detailed explanation for the considerations made in the formal definition. The reader can skip directly to the formal definition if he wishes.

This setting is very similar to the case of functions that are hard to compute, but there are several key differences that we explain below.

Explanation of the model: Recall that (as explained in Section 1.2.2) the Goldreich-Levin theorem (stated precisely in Theorem 1.10) has the following form:

- We are given an arbitrary function $f : \{0,1\}^\ell \to \{0,1\}^\ell$. (Intuitively, it is assumed that f is a one-way function, meaning that it is hard to invert f with success probability ρ)

- There is a specified construction that transforms f into two functions: A "new one-way function" $f^{\mathrm{newOWF}} : \{0,1\}^{\ell'} \to \{0,1\}^{\ell'}$ and a predicate $f^{\mathrm{pred}} : \{0,1\}^{\ell'} \to \{0,1\}$ for some ℓ' related to ℓ. (Intuitively, we will want to argue that f^{pred} is a hard-core predicate such that for $x \leftarrow U_{\ell'}$, $f^{\mathrm{pred}}(x)$ is hard to compute with success $\frac{1}{2} + \epsilon$ when given $f^{\mathrm{newOWF}}(x)$).

  We will model this construction as a map Con which given f produces a pair of functions $(f^{\mathrm{newOWF}}, f^{\mathrm{pred}})$. Once again, we place *no limitations* on the map Con (and in particular do not require that $f^{\mathrm{newOWF}}, f^{\mathrm{pred}}$ can be efficiently computed if f is). This only makes our results stronger.

  In the case of Theorem 1.10, we have that: $\mathrm{Con}(f) = (f^{\mathrm{newOWF}}, f^{\mathrm{pred}})$ where $\ell' = 2\ell$ and we think of ℓ' bit long the input of g^{pred} as two strings $x, r \in \{0,1\}^{\ell}$, setting:

$$f^{\mathrm{newOWF}}(x,r) = (f(x), r), \text{and}$$

$$f^{\mathrm{pred}}(x,r) = \mathrm{Enc}^{\mathrm{Had}}(x)_r = (\sum_{i \in [\ell]} x_i \cdot r_i) \mod 2.$$

- We model the proof showing that f^{pred} is a hard-core predicate, in the following way: The proof is a pair $(\mathrm{Con}, \mathrm{Red})$ where $\mathrm{Red}^{(\cdot)}$ is an oracle procedure, such that when $\mathrm{Red}^{(\cdot)}$ receives oracle access to an "adversary" $h : \{0,1\}^{\ell'} \to \{0,1\}$ that breaks the security of f^{pred}, we have that Red^{h} breaks the security of f.

  It is illustrative to consider the case where f, f^{newOWF} are *permutations*, and with this choice, the model we have introduced so far is identical to the one considered in Section 4.1 if we set $g = f^{-1}$.

  In the setup of functions that are *hard to invert*, a reduction Red can potentially want to *compute the function* f (as we are implicitly assuming that f is efficiently computable). Many reductions in the cryptographic literature (e.g. from one-way functions to pseudorandom generators) critically rely on this ability, and so, if we want to handle a general case, we should allow the reduction Red to also *receive oracle access to* f, allowing it to compute f on chosen values, if it wants to.

  This means that in the actual definition, $\mathrm{Red}^{(\cdot,\cdot)}$ is an oracle procedure with *two* oracles: It receives oracle access both to h and to f. More precisely, we require that: for every $f : \{0,1\}^{\ell} \to \{0,1\}^{\ell}$ and for every $h : \{0,1\}^{\ell'} \to \{0,1\}$, that:

$$\Pr_{x \leftarrow U_{\ell'}}[h(f^{\mathrm{newOWF}}(x)) = f^{\mathrm{pred}}(x)] \geq \frac{1}{2} + \epsilon,$$

  It holds that:

$$\Pr_{x \leftarrow U_{\ell}}[\mathrm{Red}^{h,f}(f(x)) \in f^{-1}(f(x))] \geq \rho.^{9}$$

  Note that this means that even in the case that f, f^{newOWF} are permutations, reductions (in the setting for functions that are hard to invert) are more powerful then reductions (in the setting of functions that are hard to compute) for the function $g = f^{-1}$, and indeed, it is more difficult to prove impossibility results for the case of functions that are hard to invert.

- As explained in the case of functions that are hard to compute, we are once again aiming to prove a result for circuits (which are allowed to use nonuniform advice) and as in the case of functions that are hard to compute, we will allow the reduction to receive an advice string α of length t. (Intuitively, this advice

[9] In fact, we should also allow h to be an oracle procedure $h^{(\cdot)}$ that receives oracle access to f. However, as we want to prove lower bounds on black-box proofs, we choose not to do that, as the lower bounds that we prove obviously also rule out this case. This can be interpretted as saying that the choice of h that we use in our lower bound, does not make calls to f.

string can depend on f and h). This leads to the following strengthening of the requirement above. Namely, we will require that: for every $f : \{0,1\}^\ell \to \{0,1\}^\ell$ and for every $h : \{0,1\}^{\ell'} \to \{0,1\}$, that:

$$\Pr_{x \leftarrow U_{\ell'}}[h(f^{\text{newOWF}}(x)) = f^{\text{pred}}(x)] \geq \frac{1}{2} + \epsilon,$$

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

$$\Pr_{x \leftarrow U_\ell}[\text{Red}^{h,f}(f(x)) \in f^{-1}(f(x))] \geq \rho.$$

- Once again, we make no restrictions on the complexity of the procedure $\text{Red}^{(\cdot,\cdot)}$ except for requiring that it makes at most q queries to each of its two oracles (for some parameter q). Our black-box impossibility results will follow from proving lower bounds on q.

Formal definition: Following this discussion, we now give a formal definition.

Definition 4.10 (nonuniform black-box proof for hard-core predicates for hard to invert functions)**.** *A pair* (Con, Red) *is a* nonuniform black-box proof *for* hard-core predicates for hard to invert functions *with parameters* $\ell, \ell', \rho, \epsilon$*, that uses* q queries*, and* t bits of advice *if:*

- Con *is a* construction map *which given a function* $f : \{0,1\}^\ell \to \{0,1\}^\ell$*, produces two functions* $\text{Con}(f) = (f^{\text{newOWF}}, f^{\text{pred}})$ *such that* $f^{\text{newOWF}} : \{0,1\}^{\ell'} \to \{0,1\}^{\ell'}$ *and* $f^{\text{pred}} : \{0,1\}^{\ell'} \to \{0,1\}$*.*
- $\text{Red}^{(\cdot,\cdot)}$ *is a* reduction*, that is an oracle procedure that given oracle access to functions* $h : \{0,1\}^{\ell'} \to \{0,1\}$*, and* $f : \{0,1\}^\ell \to \{0,1\}$*, makes at most* q *queries to each of its two oracles.*

Furthermore, for every functions $f : \{0,1\}^\ell \to \{0,1\}^\ell$ *and* $h : \{0,1\}^{\ell'} \to \{0,1\}$ *such that:*

$$\Pr_{x \leftarrow U_{\ell'}}[h(f^{\text{newOWF}}(x)) = f^{\text{pred}}(x)] \geq \frac{1}{2} + \epsilon,$$

there exists $\alpha \in \{0,1\}^t$*, such that:*

$$\Pr_{x \leftarrow U_\ell}[\text{Red}^{h,f}(f(x), \alpha) \in f^{-1}(f(x))] \geq \rho.$$

Avoiding trivial constructions: We now explain that it is possible to have black-box proofs that are *trivial* and provide hard-core predicates that are hard because of trivial reasons. We need to avoid such trivial constructions if we want to prove interesting limitations.

Specifically, it is easy to obtain hard-core predicates that are hard because information on $f^{\text{pred}}(x)$ is not present in $f^{\text{newOWF}}(x)$. Indeed, consider the construction $\text{Con}(f) = (f^{\text{newOWF}}, f^{\text{pred}})$ with:

$$f^{\text{pred}}(x) = x_1, \text{and}$$

$$f^{\text{newOWF}}(x) = x_2, \ldots, x_\ell.$$

In this case, f^{pred} is a hardcore-predicate, because it is impossible (even for unbounded adversaries) to compute $f^{\text{pred}}(x)$ when given $f^{\text{newOWF}}(x)$. This means that for this construction map, there is a reduction that makes $q = 0$ queries. Consequently, in order to prove lower bounds, we need to avoid such trivial (and uninteresting) construction maps, and require that for every f, there exists a function $\phi_f : \{0,1\}^{\ell'} \to \{0,1\}$ such that for every $x \in \{0,1\}^{\ell'}$, $\phi_f(f^{\text{newOWF}}(x)) = f^{\text{pred}}(x)$, meaning that there is information on $f^{\text{pred}}(x)$ in $f^{\text{newOWF}}(x)$.

An additional case of a trivial construction that we want to avoid, is the case in which

$$H_\infty(f^{\text{newOWF}}(U_{\ell'})) < \log(1/\rho).$$

Such a construction is uninteresting, because in such a case, if we define the function $\psi_f : \{0,1\}^{\ell'} \to \{0,1\}^{\ell'}$ to output the constant $x \in \{0,1\}^{\ell}$ such that

$$\Pr[f^{\text{newOWF}}(U_{\ell'}) = f^{\text{newOWF}}(x)] \geq \rho,$$

(and note that such an x exists if $H_\infty(f^{\text{newOWF}}(U_{\ell'})) < \log(1/\rho)$) then we get that there is a constant function ψ_f such that ψ_f inverts the function f^{newOWF} with probability ρ. Such a construction is uninteresting because in that case f^{newOWF} is obviously not a one-way function.

This leads to the following characterization of nontrivial construction maps, in which we require that Con avoids these two trivial examples.

Definition 4.11 (nontrivial construction map)**.** *We say that a construction map* $\text{Con}(f) = (f^{\text{newOWF}}, f^{\text{pred}})$ *is* ρ-*trivial if it satisfies the following two requirements:*

- *For every* f*, the functions* $(f^{\text{newOWF}}, f^{\text{pred}})$ *produced by* $\text{Con}(f)$ *are such that there exists a function* $\phi_f : \{0,1\}^{\ell'} \to \{0,1\}$ *such that for every* $x \in \{0,1\}^{\ell'}$*,* $\phi_f(f^{\text{newOWF}}(x)) = f^{\text{pred}}(x)$*.*
- $H_\infty(f^{\text{newOWF}}(U_{\ell'})) \geq \log(1/\rho)$*.*

We say that a pair (Con, Red) *is* ρ-*nontrivial, if* Con *is not trivial.*

The role of the number of queries, and black-box impossibility results: We now explain the role of the parameter q (that measures the number of queries made by Red) and why lower bounds on q translate into black-box impossibility results. This explanation is similar to the one given in Section 4.1 (with the modifications explained above).

For this purpose, it is illustrative to examine the argument showing that nonuniform black-box proofs yield hard-core predicates: When given a pair (Con, Red) that is a nonuniform black-box proof for hard-core predicates for hard to invert functions with parameters $\ell, \ell', \rho, \epsilon$, that uses q queries, and t bits of advice, we obtain that for any function $f : \{0,1\}^{\ell} \to \{0,1\}^{\ell}$, if there exists a circuit $C' : \{0,1\}^{\ell'} \to \{0,1\}$ of size s' such that:

$$\Pr_{x \leftarrow U_{\ell'}}[C'(f^{\text{newOWF}}(x)) = f^{\text{pred}}(x)] \geq \frac{1}{2} + \epsilon,$$

then there exists $\alpha \in \{0,1\}^t$, such that:

$$\Pr_{x \leftarrow U_{\ell}}[\text{Red}^{C',f}(f(x), \alpha) \in f^{-1}(f((x))] \geq \rho.$$

Note that if the reduction Red can be implemented by a circuit of size r, and the function f can be computed by a circuit of size m, then the circuit $C(y) = \text{Red}^{C',f}(y, \alpha)$ is a circuit of size:

$$s = r + t + q \cdot m + q \cdot s'$$

that inverts f with success probability ρ.

It follows that in a black-box proof, with q queries, and t bits of advice, we get a hard-core theorem that needs to assume that the original function f cannot be inverted by circuits of size s, for:

$$s \geq q + t.$$

4.2.2 Precise statements of limitations

Our main result on black-box proofs for hard-core predicates in the setting of functions that are hard to invert is the following theorem.

Theorem 4.12. *There exists a universal constant $\beta > 0$ such that for every sufficiently large ℓ and ℓ' we have that if $(\mathrm{Con}, \mathrm{Red})$ is a ρ-nontrivial nonuniform black-box proof for hard-core predicates for hard to invert functions with parameters $\ell, \ell', \rho, \epsilon$, that uses q queries, and t bits of advice, and furthermore $t \leq 2^{\ell/5}$, $\rho \geq \frac{1}{2^{\ell/5}}$ and $\rho \leq \beta \cdot \epsilon^2$, then*

$$q \geq \Omega(\frac{1}{\epsilon^\beta}) - O(t + \ell).$$

We now explain why Theorem 4.12 implies the informal statement made in Theorem 1.13. This explanation is essentially identical to the one following Theorem 4.12

Recall that in Section 4.2.1 we explained that when using a nonuniform black-box proof to obtain hard-core predicates, we get a hard-core predicate theorem in which $s \geq q + t$.

Theorem 4.12 implies that for $s > \ell^{2/\beta}$ it is impossible for such a proof to establish $\epsilon = 1/s^{\frac{2}{\beta}}$ (even if ρ is very small). This follows as otherwise, using the fact that $s \geq q + t \geq t$, we get that:

$$q \geq \Omega(\frac{1}{\epsilon^\beta}) - O(t + \ell) \geq \Omega(s^2) - O(t) > s,$$

which is a contradiction to $s \geq q + t \geq q$. In particular, the parameter setting considered in Theorem 1.13, in which $s = 2^{o(\ell)}$ and $\epsilon = \frac{1}{s^{\omega(1)}}$, is impossible to achieve.

4.2.3 Proof of Theorem 4.12

The proof of Theorem 4.12 is similar in structure to the proof of Theorem 4.2 with three main differences:

- Rather than choosing the initial function uniformly from F_ℓ (the set of all functions from ℓ bits to ℓ bits) we will restrict the choice to permutations. This is helpful because for a permutation f, the function f^{-1} is well defined, and inverting f (that is producing an element in $f^{-1}(f(x))$ when given $f(x)$) can be thought of as computing f^{-1} that is producing x on input $f(x)$.
- A more significant difference, is that as explained in detail in Section 4.2.1, in the setup of functions that are hard to invert, the reduction Red has oracle access to f (in addition to oracle access to h). This means that it is no longer the case that the answer of the reduction on inputs x, α and oracle h is *determined* by h, x, α (as the answer depends on f). Therefore, it is not the case that there exists circuits $A_{x,\alpha}(h)$ that simulate the reduction (as we argued in Claim 4.5) and we need to be more careful when showing that for every function f, there exists a circuit $C_f(z)$ that is analogous to the circuit guaranteed in Claim 4.6.

  Furthermore, as Red gets oracle access to f, we can no longer claim that when h is independent of f, then Red has no information on f. Instead, use results by Gennaro and Trevisan [GT00] showing that an oracle circuit that makes a subexponential number of queries to a random permutation f, cannot invert f with high probability.
- Unlike the case of Section 4.1, the distribution that is given as input to h is not necessarily uniform. More precisely, The distribution of $f^{\mathrm{newOWF}}(U_{\ell'})$ (on which h needs to predict the hard-core predicate) is not necessarily uniform. By the nontriviality condition in Definition 4.11 we have that this distribution has high min-entropy, and we need to adjust the argument to hold with this weaker requirement.

Theorem 4.12 will follow from the next lemma.

Lemma 4.13. *There exists a universal constant d, such that for every sufficiently large ℓ and ℓ' we have that if $(\mathrm{Con}, \mathrm{Red})$ is a ρ-nontrivial nonuniform black-box proof for hard-core predicates for hard to invert functions with parameters $\ell, \ell', \rho, \epsilon$, that uses q queries, and t bits of advice, and furthermore $t \leq 2^{\ell/5}$, $\rho \geq \frac{1}{2^{\ell/5}}$ and $\rho \leq \frac{\epsilon^2}{d}$, then there exists a circuit C of size $s = \mathrm{poly}(2^q, 2^\ell, 2^t)$ and depth d such that:*

- $\Pr_{z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}-2\epsilon}}[C(z) = 1] \geq 0.99$.
- $\Pr_{z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}}}[C(z) = 1] \leq 0.01$.

Once again, just like in the previous section, by reduction to the coin problem, Theorem 4.12 follows from Lemma 4.13.

Proof of Theorem 4.12. The theorem follows directly from Lemma 4.13 and Corollary 2.5, which give that:

$$s = \mathrm{poly}(2^q, 2^\ell, 2^t) \geq \exp(\Omega(d \cdot (1/\epsilon)^{1/(d-1)})),$$

The statement of Theorem 4.2 follows by taking the logarithm on both sides and setting $\beta = 1/(d-1)$. □

In the remainder of this section we prove Lemma 4.13. Let $(\mathrm{Con}, \mathrm{Red})$ be a nontrivial nonuniform black-box proof for hard-core predicates for hard to invert functions with parameters $\ell, \ell', \rho, \epsilon$, that uses q queries, and t bits of advice. Throughout this section we assume that the requirements made in Lemma 4.13 are met.

We will prove Lemma 4.13 using the following sequence of claims. The proof uses the same structure as the proof of Lemma 4.3 however, the reduction is now more powerful as it has oracle access to the function f, and the setting is more general as the distribution $f^{\mathrm{newOWF}}(U_\ell)$ (on which the oracle is judged) is not necessarily uniform. In this setting, there is no direct analog Claim 4.5, and instead we prove an analog of Claim 4.14 directly.

Claim 4.14. *There exists a universal constant d such that for every permutation $f : \{0,1\}^\ell \to \{0,1\}^\ell$, there exists a circuit $C_f : \{0,1\}^{2^{\ell'}} \to \{0,1\}$ of size $\mathrm{poly}(2^q, 2^\ell, 2^t)$ and depth d such that the following holds for every $z \in \{0,1\}^{2^{\ell'}}$:*

- *If there exists $\alpha \in \{0,1\}^t$ such that $\Pr_{x \leftarrow U_\ell}[\mathrm{Red}^{\phi_f \oplus z, f}(f(x), \alpha) = x] \geq \rho$ then $C_f(z) = 1$.*
- *If for all $\alpha \in \{0,1\}^t$, $\Pr_{x \leftarrow U_\ell}[\mathrm{Red}^{\phi_f \oplus z, f}(f(x), \alpha) = x] \leq \rho/10$ then $C_f(z) = 0$.*

Proof. The circuit C_f will be hardwired with f and ϕ_f (where ϕ_f is the function whose existence is guaranteed for f by the nontriviality condition in Definition 4.11). Upon receiving an input $z \in \{0,1\}^{2^{\ell'}}$ it will act as follows:

- Prepare $w = \phi_f \oplus z$. (Here we think of ϕ_f, z, w as a string in $\{0,1\}^{2^{\ell'}}$.
- For every $x \in \{0,1\}^\ell$ and $\alpha \in \{0,1\}^t$ compute $\mathrm{Red}^{w,f}(f(x), \alpha)$, and compute $b_{x,\alpha} \in \{0,1\}$ defined by:
$$b_{x,\alpha} = \begin{cases} 0 & \mathrm{Red}^{w,f}(f(x), \alpha)) \neq x \\ 1 & \mathrm{Red}^{w,f}(f(x), \alpha)) = x \end{cases}$$
- For every $\alpha \in \{0,1\}^t$, let v_α denote the 2^ℓ bit long concatenation of all bits $(b_{x,\alpha})_{x \in \{0,1\}^\ell}$ (fixing some order on $x \in \{0,1\}^\ell$), and compute
$$b_\alpha = D_\rho^{2^\ell}(v_\alpha),$$
where $D_\rho^{2^\ell}$ is the circuit guaranteed in Lemma 4.4.

- Compute the disjunction of the 2^t bits $(b_\alpha)_{\alpha\in\{0,1\}^t}$ and output it.

It is immediate that the circuit C_f performs the task specified in the lemma. We now explain how to implement the circuit in small size and depth.

The string f can be described by $\ell \cdot 2^\ell$ bits. We note that when using the string ϕ_f to prepare w, we only need to have g_{pred} at coordinates $y \in \{0,1\}^{\ell'}$ such that there exists x, α such that $\mathrm{Red}^{w,f} f(x), \alpha$ depends on w_y. As on every pair $(f(x), \alpha)$ the reduction $\mathrm{Red}^{w,f} f(x), \alpha$ makes at most q queries to its oracle, it can depend on at most 2^q choices of $y \in \{0,1\}^{\ell'}$. Thus, going over all choices of $x \in \{0,1\}^\ell$ and $\alpha \in \{0,1\}^t$, C_g only requires $O(2^t \cdot 2^\ell \cdot 2^q)$ bits of ϕ_f. Note that any query that the reduction makes to its oracle f, is a constant that C_f has hardwired (because C_f is hardwired with f). Overall, the size of the advice of C_f is $O(2^t \cdot 2^\ell \cdot 2^q)$. The circuit C_f is constant depth by construction, and its size is indeed:

$$\mathrm{poly}(2^q, 2^\ell, 2^t, 1/\rho) = \mathrm{poly}(2^q, 2^\ell, 2^t),$$

by the requirement that $\rho \geq 2^{-\ell}$. □

We will consider the case where $f : \{0,1\}^\ell \to \{0,1\}^\ell$ is a uniformly chosen permutation, and will analyze the behavior of C_f on $z \leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}-2\epsilon}$ and on $z \leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}}$.

Definition 4.15. *Let Π_ℓ denote the set of all permutations $f : \{0,1\}^\ell \to \{0,1\}^\ell$.*

Claim 4.16. $\Pr_{f\leftarrow\Pi_\ell, z\leftarrow\mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}-2\epsilon}}[C_f(z) = 1] \geq 0.999$.

Proof. Imagine that $f \leftarrow \Pi_\ell$ is already chosen and fixed. Let $(f^{\mathrm{newOWF}}, f^{\mathrm{pred}}) = \mathrm{Con}(f)$, and let $\phi_f : \{0,1\}^{\ell'} \to \{0,1\}$ be the function guaranteed by the fact that Con is nontrivial. We now consider the additional experiment of choosing $z \leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}-2\epsilon}$. Let $h : \{0,1\}^{\ell'} \to \{0,1\}$ be defined by $h = \phi_f \oplus z$. Our goal is to show that with probability at least 0.999 over choosing $z \leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}-2\epsilon}$, we have that:

$$\Pr_{x\leftarrow U_{\ell'}}[h(f^{\mathrm{newOWF}}(x)) = f^{\mathrm{pred}}(x)] \geq \frac{1}{2} + \epsilon. \tag{2}$$

This is because whenever f, z satisfy the condition above, then by the properties of Red, we have that there exists $\alpha \in \{0,1\}^t$ such that:

$$\Pr_{x\leftarrow U_\ell}[\mathrm{Red}^{h,f}(f(x), \alpha) = x] \geq \rho.$$

which in turn by Claim 4.14 implies that $C_f(z) = 1$.

By definition, for every $x \in \{0,1\}^{\ell'}$, $\phi_f(f^{\mathrm{newOWF}}(x)) = f^{\mathrm{pred}}(x)$. Consequently, the event

$$\left\{h(f^{\mathrm{newOWF}}(x)) = f^{\mathrm{pred}}(x)\right\}$$

that appears in (2) can be expressed as

$$\left\{h(f^{\mathrm{newOWF}}(x)) = \phi_f(f^{\mathrm{newOWF}}(x))\right\}.$$

As $h = \phi_f \oplus z$, this event can also be expressed as

$$\left\{z(f^{\mathrm{newOWF}}(x)) = 0\right\}.$$

Thus, in order to prove the claim, it is sufficient to prove that with probability at least 0.999 over choosing $z \leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}-2\epsilon}$, we have that:

$$\Pr_{x \leftarrow U_{\ell'}}[z(f^{\mathrm{newOWF}}(x)) = 0] \geq \frac{1}{2} + \epsilon.$$

Note that $Y = f^{\mathrm{newOWF}}(U_{\ell'})$ is not necessarily uniform. For every $y \in \{0,1\}^{\ell'}$, we define $p_y = \Pr[Y = y]$. By the nontriviality of Con we have that $H_\infty(Y) \geq \log(1/\rho)$, which means that for every $y \in \{0,1\}^{\ell'}$, $p_y \leq \rho$. Thus, in order to conclude the proof, it is sufficient to show that:

$$\Pr_{z \leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}-2\epsilon}}[\sum_{y \in \{0,1\}^{\ell'}} p_y \cdot z_y < \frac{1}{2} - \epsilon] \geq 0.999.$$

(This can be thought of as a "weighted version" of Hamming weight in which the p_y are not all the same). When $z \leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}-2\epsilon}$, the random $2^{\ell'}$ random variables $x_y = p_y \cdot z_y$ (one for each choice of $y \in \{0,1\}^{\ell'}$) are independent, and lie in the interval $[0, \rho]$. We have that:

$$\mathbb{E}_{z \leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}-2\epsilon}}[\sum_{y \in \{0,1\}^{\ell'}} p_y \cdot z_y] = \frac{1}{2} - 2\epsilon.$$

We can apply a Chernoff bound to bound the probability of deviation from the expectation and obtain that:

$$\Pr_{z \leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}-2\epsilon}}[\sum_{y \in \{0,1\}^{\ell'}} p_y \cdot z_y < \frac{1}{2} - \epsilon] \leq e^{-\Omega(\frac{\epsilon^2}{\rho})},$$

where the last equality uses $\epsilon < \frac{1}{4}$ which we can assume w.l.o.g. By our requirement that ρ is sufficiently smaller than $\epsilon^2/100$, we get that the probability is indeed larger than 0.999. □

On the other hand, we can show that:

Claim 4.17. $\Pr_{f \leftarrow \Pi_\ell, z \leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}}}[C_f(z) = 1] \leq 0.001$.

In order to prove Claim 4.17 we will use the following result by Gennaro and Trevisan [GT00]:[10]

Theorem 4.18 ([GT00])**.** *For sufficiently large ℓ, for every oracle procedure $P^{(\cdot)}$ that makes at most $2^{\ell/5}$ queries to its oracle, and accepts inputs $x \in \{0,1\}^\ell$ and $\alpha \in \{0,1\}^t$ for $t \leq 2^{\ell/5}$, it holds that:*

$$\Pr_{f \leftarrow \Pi_\ell}[\exists \alpha \in \{0,1\}^t : \Pr_{x \leftarrow U_\ell}[P^f(f(x), \alpha) = x] \geq 2^{-\ell/5}] \leq 2^{-2^{\ell/2}}$$

We now prove Claim 4.17

Proof. (of Claim 4.17) The first step of $C_f(z)$ is to prepare $w = \phi_f \oplus z$. However, for $p = \frac{1}{2}$, and $f \leftarrow \Pi_\ell, z \leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}}$, we have that $w = \phi_f \oplus z$ is uniformly chosen, and independent of f.

[10] In [GT00] the theorem is stated for P which is an oracle circuit of size $s = 2^{\ell/5}$ which implies the statement that we give as the number of circuits of size s is larger than $2^t \leq 2^{2^{\ell/5}}$.

Therefore, for any choice of advice $\alpha \in \{0,1\}^t$ oracle access to w does not help the reduction $\mathrm{Red}(\cdot,\alpha)$ to invert f. More precisely, Let $P^{(\cdot)}(x,\alpha)$ be an implementation of $\mathrm{Red}^{(\cdot,\cdot)}$ where whenever Red makes a query to its first oracle h, the query is answered by a fresh uniform random bit. We have that:

$$\begin{aligned}
&\Pr_{f\leftarrow\Pi_\ell, z\leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}}}[C_f(z)=1] \\
\leq &\Pr_{f\leftarrow\Pi_\ell, z\leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}}}[\exists \alpha\in\{0,1\}^t : \Pr_{x\leftarrow U_\ell}[\mathrm{Red}^{\phi_f\oplus z, f}(f(x),\alpha)=x] \geq 2^{-\ell/5}] \\
\leq &\Pr_{f\leftarrow\Pi_\ell, w\leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}}}[\exists \alpha\in\{0,1\}^t : \Pr_{x\leftarrow U_\ell}[\mathrm{Red}^{w, f}(f(x),\alpha)=x] \geq 2^{-\ell/5}] \\
\leq &\Pr_{f\leftarrow\Pi_\ell}[\exists \alpha\in\{0,1\}^t : \Pr_{x\leftarrow U_\ell}[P^f(f(x),\alpha)=x] \geq 2^{-\ell/5}] \\
\leq &\, 2^{-2^{\ell/2}} \leq 0.001,
\end{aligned}$$

where the first inequality follows from Claim 4.14 and where penultimate inequality follows from Theorem 4.18. □

The proof of Lemma 4.13 now follows in exactly from Claims 4.16 and 4.17 in exactly the same way as in the end of the previous section. Specifically:

Proof of Lemma 4.13. From Claims 4.16 and 4.17, it follows that:

$$\begin{aligned}
\Pr_{f\leftarrow\Pi_\ell}\left[\Pr_{z\leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}-2\epsilon}}[C_f(z)=0] > 0.01\right] \leq 0.1, \\
\Pr_{f\leftarrow\Pi_\ell}\left[\Pr_{z\leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}}}[C_f(z)=1] > 0.01\right] \leq 0.1.
\end{aligned}$$

Thus, by a union bound, there exists $f:\{0,1\}^\ell \to \{0,1\}^\ell$ such that

$$\begin{aligned}
\Pr_{z\leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}-2\epsilon}}[C_f(z)=1] \geq 0.99, \\
\Pr_{z\leftarrow \mathrm{BSC}^{2^{\ell'}}_{\frac{1}{2}}}[C_f(z)=1] \leq 0.01.
\end{aligned}$$

This completes the proof of the lemma, as C_f satisfies all the other requirements as well. □

Conclusion and open problems

Unlike Theorem 1.5 (that handles large ϵ), Theorem 1.6 (that handles small ϵ) does not achieve a bound of $q = \Omega(\frac{\log(1/\delta)}{\epsilon^2})$, and only achieves a bound of $\Omega(\frac{1}{\sqrt{\epsilon}})$. A natural open problem is to improve the bound on q for small ϵ to match the bound for large ϵ.

In the case of large ϵ, Theorem 1.5 can be extended to handle local list-decoding from *erasures*, and gives a lower bound of $q = \Omega(\frac{\log(1/\delta)}{\epsilon})$ on the number of queries of local list-decoders that decode from a $1-\epsilon$ fraction of erasures. We do not see how to extend the proof of Theorem 1.6 to erasures.

The model of black-box proofs that we introduce in Section 4 is quite general, and to the best of our knowledge, covers all known proofs in the literature on hard-core predicates for general one-way functions. Is it possible to circumvent the black-box limitations and answer open problems 1.8 and 1.12 for *specific candidates* for one-way functions?

More generally, is it possible to come up with non-black-box techniques that circumvent the limitations?

Acknowledgment

We are grateful to Ilan Newman for participating in early stages of this research and for many helpful discussions.

Noga Ron-Zewi was partially supported by ISF grant 735/20. Ronen Shaltiel was partially supported by ISF grant 1628/17. Nithin Varma was partially supported by ISF grant 497/17, and Israel PBC Fellowship for Outstanding Postdoctoral Researchers from India and China.

References

[Aar10] Scott Aaronson. BQP and the polynomial hierarchy. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 141–150. ACM, 2010.

[AASY16] Benny Applebaum, Sergei Artemenko, Ronen Shaltiel, and Guang Yang. Incompressible functions, relative-error extractors, and the power of nondeterministic reductions. *Comput. Complex.*, 25(2):349–418, 2016.

[Ajt83] Miklós Ajtai. Σ^1_1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24(1):1–48, 1983.

[AS11] Sergei Artemenko and Ronen Shaltiel. Lower bounds on the query complexity of non-uniform and adaptive reductions showing hardness amplification. In Leslie Ann Goldberg, Klaus Jansen, R. Ravi, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 14th International Workshop, APPROX 2011, and 15th International Workshop, RANDOM 2011, Princeton, NJ, USA, August 17-19, 2011. Proceedings*, volume 6845 of *Lecture Notes in Computer Science*, pages 377–388. Springer, 2011.

[CGR14] Gil Cohen, Anat Ganor, and Ran Raz. Two sides of the coin problem. In Klaus Jansen, José D. P. Rolim, Nikhil R. Devanur, and Cristopher Moore, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2014, September 4-6, 2014, Barcelona, Spain*, volume 28 of *LIPIcs*, pages 618–629. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2014.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In David S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 25–32. ACM, 1989.

[GSV18] Aryeh Grinberg, Ronen Shaltiel, and Emanuele Viola. Indistinguishability by adaptive procedures with advice, and lower bounds on hardness amplification proofs. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 956–966. IEEE Computer Society, 2018.

[GT00] Rosario Gennaro and Luca Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*, pages 305–313. IEEE Computer Society, 2000.

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

[LSS+19] Nutan Limaye, Karteek Sreenivasaiah, Srikanth Srinivasan, Utkarsh Tripathi, and S. Venkitesh. A fixed-depth size-hierarchy theorem for $AC^0[\oplus]$ via the coin problem. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 442–453. ACM, 2019.

[Raz87] Alexander Razborov. Lower bounds on the dimension of schemes of bounded depth in a complete basis containing the logical addition function. *Akademiya Nauk SSSR. Matematicheskie Zametki*, 41(4):598–607, 1987. English translation in Mathematical Notes of the Academy of Sci. of the USSR, 41(4):333-338, 1987.

[RRZV18] Sofya Raskhodnikova, Noga Ron-Zewi, and Nithin Varma. Erasures versus errors in local decoding and property testing. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:195, 2018.

[Sha20] Ronen Shaltiel. Is it possible to improve Yao's XOR lemma using reductions that exploit the efficiency of their oracle? In Jaroslaw Byrka and Raghu Meka, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2020, August 17-19, 2020, Virtual Conference*, volume 176 of *LIPIcs*, pages 10:1–10:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pages 77–82, 1987.

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

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

[Vio06] Emanuele Viola. The complexity of hardness amplification and derandomization, 2006.

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

A Proof of Lemma 2.1

In this proof we use the following notation. For two distributions X, Y over the $\{0,1\}^n$ we say that they are ϵ-close if for every event $A \subseteq \{0,1\}^n$, $|\Pr[X \in A] - \Pr[Y \in A]| \leq \epsilon$. We will also use Shannon's entropy which we denote by $H(X)$, and the following statement of Pinsker's lemma:

Lemma A.1 (Pinsker's lemma)**.** *If X is a distribution over $\{0,1\}^n$ and $H(X) \geq n - \epsilon$ then, X is $\sqrt{\epsilon}$-close to U_n.*

Proof of lemma 2.1. By the requirements on M, we have that $H(M) \geq k - k^{0.99}$. The Shannon entropy function satisfies $H(M_1) + \ldots + H(M_k) \geq H(M_1, \ldots, M_k)$ and therefore:

$$H(M_1) + \ldots + H(M_k) \geq k - k^{0.99}$$

It follows that:

$$\mathbb{E}_{i \leftarrow [k]} H(M_i) = 1 - k^{-0.01}.$$

By Markov's inequality, for every c, the fraction of $i \in [k]$ such that $H(M_i) < 1 - c \cdot k^{-0.01}$ is less than $1/c$. By Pinsker's lemma, for every i such that $H(M_i) \geq 1 - c \cdot k^{-0.01}$, we have that M_i is $\sqrt{c \cdot k^{-0.01}}$-close to U_1. Therefore,

$$\Pr_{m \leftarrow M, i \leftarrow [k]}[D(i) = m_i] \leq \frac{1}{2} + \sqrt{c \cdot k^{-0.01}} + 1/c \leq 0.5001,$$

for a sufficiently large constant c. □

B Proof of Corollary 2.5

The proof is by reduction to Theorem 2.4.

of Corollary 2.5. For $\epsilon' = \Theta(\epsilon)$, given a circuit C that satisfies:

- $\Pr_{z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}-\epsilon'}}[C(z) = 1] \geq 0.99$,
- $\Pr_{z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}}}[C(z) = 0] \leq 0.01$.

We will show the existence of a circuit C' that satisfies:

- $\Pr_{z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}-\epsilon}}[C'(z) = 1] \geq 0.9$,
- $\Pr_{z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}+\epsilon}}[C'(z) = 0] \leq 0.1$.

We will start by constructing a randomized circuit C' which upon receiving input $x \in \{0,1\}^n$, for every $i \in [n]$ independently, C' replaces input bit x_i by zero with probability $p = \frac{2\epsilon}{1+2\epsilon}$, let x'_i denote the obtained bit, and let $C'(x) = C(x')$. The choice of p is made so that for every i:

- If $x \leftarrow \mathrm{BSC}^n_{\frac{1}{2}+\epsilon}$ then $x' \leftarrow \mathrm{BSC}^n_{\frac{1}{2}}$.
- if $x \leftarrow \mathrm{BSC}^n_{\frac{1}{2}-\epsilon}$ then $x' \leftarrow \mathrm{BSC}^n_{\frac{1}{2}-\epsilon'}$ for $\epsilon' = \epsilon + \epsilon \cdot \frac{1-2\epsilon}{1+2\epsilon} = \Theta(\epsilon)$.

It follows that if C distinguishes between $\mathrm{BSC}^n_{\frac{1}{2}-\epsilon'}$ and $\mathrm{BSC}^n_{\frac{1}{2}}$ then C' satisfies:

- $\Pr_{z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}-\epsilon}}[C'(z) = 1] \geq 0.99$,

- $\Pr_{z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}+\epsilon}}[C'(z) = 0] \leq 0.01$.

This gives that:

$$\Pr_{z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}-\epsilon}}[C'(z) = 1] - \Pr_{z \leftarrow \mathrm{BSC}^n_{\frac{1}{2}+\epsilon}}[C'(z) = 0] \geq 0.98,$$

where the probability in the expressions above is also over the randomness of C'. Therefore, there exists a fixing of the random coins of C' which achieves this gap of 0.98 and, hence, satisfies the requirements on C' (this follows because for numbers $0 \leq p \leq P < 1$ that satisfy $P - p \geq 0.98$, it holds that: $P \geq 0.9$ and $p \leq 0.1$). We note that the size and depth of C' are bounded by the size and depth of C respectively, and the corollary follows. $\square$