Electronic Colloquium on Computational Complexity, Report No. 16 (2011)

Lower bounds on the query complexity of non-uniform and adaptive reductions showing hardness amplification

Sergei Artemenko
University of Haifa

Ronen Shaltiel*
University of Haifa

February 7, 2011

Abstract

Hardness amplification results show that for every function f there exists a function $Amp(f)$ such that the following holds: if every circuit of size s computes f correctly on at most a $1-\delta$ fraction of inputs, then every circuit of size s' computes $Amp(f)$ correctly on at most a $1/2+\epsilon$ fraction of inputs. All hardness amplification results in the literature suffer from "size loss" meaning that $s' \leq \epsilon \cdot s$. In this paper we show that proofs using "non-uniform reductions" must suffer from size loss. To the best of our knowledge, all proofs in the literature are by non-uniform reductions. Our result is the first lower bound that applies to non-uniform reductions that are *adaptive*.

A reduction is an oracle circuit $R^{(\cdot)}$ such that when given oracle access to any function D that computes $Amp(f)$ correctly on a $1/2+\epsilon$ fraction of inputs, R^D computes f correctly on a $1-\delta$ fraction of inputs. A *non-uniform* reduction is allowed to also receive a short advice string α that may depend on both f and D in an arbitrary way. It is known that reductions showing hardness amplification must be non-uniform. A reduction is *non-adaptive* if it makes non-adaptive queries to its oracle. Shaltiel and Viola (STOC 2008) showed lower bounds on the number of queries made by non-uniform reductions that are *non-adaptive*. We show that every non-uniform reduction must make at least $\Omega(1/\epsilon)$ queries to its oracle (even if the reduction is *adaptive*). This implies that proofs by non-uniform reductions must suffer from size loss.

We also prove the same lower bounds on the number of queries of non-uniform and adaptive reductions that are allowed to rely on arbitrary specific properties of the function f. Previous limitations on reductions were proven for "function-generic" hardness amplification, in which the non-uniform reduction needs to work for every function f and therefore cannot rely on specific properties of the function.

*This research was supported by BSF grant 2004329 and ISF grant 686/07.

ISSN 1433-8092

1 Introduction

1.1 Background on hardness amplification

Hardness amplification results transform functions that are hard on the worst case (or sometimes mildly hard on average) into functions that are very hard on average. These results play an important role in computational complexity and cryptography. There are many results of this kind in the literature depending on the precise interpretation of "hard". In this paper we focus on hardness against Boolean circuits and use the following notation.

Definition 1.1. *Let* $g : \{0,1\}^n \to \{0,1\}^\ell$.

- *Let* $C : \{0,1\}^n \to \{0,1\}^\ell$. *We say that* C *has* agreement p *with* g *if* $\Pr_{X \leftarrow U_n}[C(X) = g(X)] \geq p$.
- *Let* $C : \{0,1\}^n \to \{0,1\}^\ell \cup \{\perp\}$. *We say that* C *has* errorless agreement p *with* g *if* C *has agreement* p *with* g *and for every* $x \in \{0,1\}^n$, *if* $C(x) \neq \perp$ *then* $C(x) = g(x)$.
- *We say that* g *is* p-hard *for* size s *if no circuit* C *of size* s *has agreement* p *with* g. *We say that* g *is* p-hard *for* errorless size s *if no circuit* C *of size* s *has errorless agreement* p *with* g.

Typical hardness amplification results start from a function $f : \{0,1\}^k \to \{0,1\}$ that is p-hard for size s and show that some function $g : \{0,1\}^n \to \{0,1\}^\ell$ is p'-hard for size s'. (The reader should think of k, n, p, p', s, s' and ℓ as parameters). These results "amplify hardness" in the sense that p' is typically much smaller than p (meaning that g is harder on average than f). We now briefly survey some of the literature on hardness amplification.

Worst-case to average-case. Here $p = 1$ (meaning that f is hard on the worst case for circuits of size s), $\ell = 1$ (meaning that g is Boolean), and $p' = 1/2 + \epsilon$ for a small parameter ϵ (meaning that circuits of size s' have advantage at most ϵ over random guessing when attempting to compute g). Many such results appear in the literature [Lip91, BFNW93, IW97, IW98, STV01, TV07, GGH$^+$07] see [Tre04] for a survey article.

Mildly-average-case to average case. This setup is similar to the one above except that $p = 1 - \delta$ for some small parameter δ (meaning that f is mildly average-case hard for circuits of size s). In other words, the setup of worst-case to average-case above can be seen as a special case in which $\delta < 1/2^k$. An extensively studied special case is Yao's XOR-Lemma in which $g(x_1, \ldots, x_t) = f(x_1) \oplus \ldots \oplus f(x_t)$ [Lev87, Imp95, IW97, KS03, Tre03] see [GNW95] for a survey article. Other examples are [O'D04, HVV06, Tre05, GK08]

Non-Boolean target function. The two setups mentioned above can also be considered when the target function $g : \{0,1\}^n \to \{0,1\}^\ell$ is not Boolean. In the Boolean case we set $p' = 1/2 + \epsilon$ as it is trivial to have agreement of $1/2$. We typically consider $\ell > \log(1/\epsilon)$ and set $p' = \epsilon$. Namely, it is required that no circuit of size s' has agreement ϵ with g. An extensively studied special case is direct-product theorems in which $g(x_1, \ldots, x_t) = (f(x_1), \ldots, f(x_t))$ [Imp95, IW97, GNW95, GG11, IJK09a, IJK09b, IJKW10].

Errorless amplification. The three notions above are also studied when the circuits attempting to compute f and g are errorless [BS07, Wat10].

We are interested in proving lower bounds on hardness amplification results. We want our lower bounds to hold for all the settings mentioned above. For this purpose we will focus on a specific setting (which we refer to as "basic hardness amplification") that is implied by all the settings mentioned above.

Basic hardness amplification. Let $\epsilon, \delta > 0$ and $\ell \geq 1$ be parameters. The *basic* hardness amplification task is to show that if f is $(1-\delta)$-hard for size s then g is ϵ-hard for *errorless* size s'. Stated in the contra-positive, the basic hardness amplification task is to show that if there exists a circuit D of size s' that has errorless agreement $p' = \epsilon$ with g then there exists a circuit C of size s that has agreement $p = 1 - \delta$ with f.

It is easy to see that basic hardness amplification is indeed implied by all the settings considered above.[1] Therefore, lower bounds on basic hardness amplification immediately apply to all the aforementioned settings. We make this statement more precise in Section 1.2.

Generic hardness amplification and error-correcting codes. Most of the hardness amplification results in the literature are *function-generic*, meaning that they provide a map Amp mapping functions $f : \{0,1\}^k \to \{0,1\}$ into functions $g = Amp(f)$ where $g : \{0,1\}^n \to \{0,1\}^\ell$ and show that for every f that is p-hard for size s, the function $g = Amp(f)$ is p'-hard for size s'. In contrast, a *function-specific* hardness amplification result uses specific functions f, g and the proof of the hardness amplification result is allowed to use specific properties of these functions. examples of function-specific hardness amplification are [Lip91, IW98, TV07, Tre03, Tre05].

It is known that function-generic hardness amplification from worst-case to strongly average-case is closely related to (locally) list-deocodable codes [STV01]. We elaborate on this relationship in Section 1.4.

Size loss in hardness amplification. A common disadvantage of all hardness amplification results surveyed above is that when starting from a function that is hard for circuits of size s, one obtains a function that is hard for circuits of smaller size $s' \leq \epsilon \cdot s$. This is a major disadvantage as it means that if one starts from a function that is hard for size s, existing results cannot produce a function that is $(1/2+\epsilon)$-hard for $\epsilon < 1/s$. It is natural to ask whether such a loss is necessary. In order to make this question precise, we to consider formal models for proofs of hardness amplification results.

1.2 Non-uniform reductions for hardness amplification

We are interested in proving impossibility results on proofs for hardness amplification and therefore consider the weakest variant of hardness amplification (which is *basic* hardness amplification). The notion that we study in this paper is that of "non-uniform" reductions. As explained in Section 1.3, this notion (defined below) captures the proofs of almost all hardness amplification results in the literature.

Definition 1.2 (non-uniform reduction)**.** *Let $f : \{0,1\}^k \to \{0,1\}$ and $g : \{0,1\}^n \to \{0,1\}^\ell$ be functions. Let ϵ, δ and a be parameters. A* non-uniform reduction showing basic hardness amplification *(for f, g, ϵ, δ and a) is an oracle circuit $R^{(\cdot)}$ which takes two inputs $x \in \{0,1\}^k$ and $\alpha \in \{0,1\}^a$. It is required that for*

[1] Note that the basic hardness amplification task is trivially implied by all the settings above in case that g is non-boolean. In case g is Boolean, if there exists a circuit D of size s' that has errorless agreement ϵ with g then we can easily convert this circuit into a circuit D of size $s' + O(1)$ that has agreement $1/2 + \epsilon/2$ with g. Given input x, circuit D applies circuit D on x and outputs the same value if it is not '$\perp$', and a fixed bit $b \in \{0,1\}$ otherwise. It is easy to see that there exists a choice of b for which D has agreement $1/2 + \epsilon/2$ with g.

every function $D : \{0,1\}^n \to \{0,1\}^\ell \cup \{\perp\}$ *that has errorless agreement* ϵ *with* g*, there exists a string* $\alpha \in \{0,1\}^a$ *(which we call an "advice string") such that the function* $C(x) = R^D(x, \alpha)$ *has agreement* $1 - \delta$ *with* f*.*

We say that R *is* semi-uniform *if* $a = 0$ *(in which case* R *does not receive an advice string* α*). The* size *of the reduction is the size of the oracle circuit* $R^{(\cdot)}$*. We say that* R *makes at most* q *queries if for every choice of oracle* D *and inputs* $x \in \{0,1\}^k$*,* $\alpha \in \{0,1\}^a$*, reduction* $R^D(x, \alpha)$ *makes at most* q *queries to its oracle. We say that* R *is* non-adaptive *if for every choice of oracle and inputs,* R *makes non-adaptive queries to its oracle.*

In the discussion below we explain the choices made in Definition 1.2.

Usefulness of non-uniform reductions. We first note that a non-uniform reduction indeed implies a basic hardness amplification result in the following sense: If there exists a circuit D of size s' that has errorless agreement ϵ with g then we have that $C(x) = R^D(x, \alpha)$ has agreement $1 - \delta$ with f, and furthermore, C can be implemented by a circuit of size $s = r + a + q \cdot s'$ where r is the size of R and q is the number of queries made by R. It follows that the number of queries q made by the reduction is the dominant factor in the ratio between s and s'. In other words, if we show that every reduction R must use at least $q = \Omega(1/\epsilon)$ queries, then we get that $s = \Omega(s'/\epsilon)$ which gives that the size loss is $s' = O(s \cdot \epsilon)$.

What is non-uniform in this reduction? Reduction R has two sources of non-uniformity: First, R is a circuit and therefore may be hardwired with non-uniform advice (that may depend on f). Note that this is the case even for semi-uniform reductions. The second (and more interesting) source of non-uniformity is the advice string α. It is important to stress that the order of quantifiers in the definition above allows α to depend on the choice of D (in addition to the choice of f). This is in contrast to the non-uniformity of R that is fixed in advance and does not depend on D.

Lower bounds for semi-uniform reductions. We now illustrate the difference between semi-uniform reductions and general non-uniform reductions. It is not hard to show that semi-uniform reductions have to use $q = \Omega(1/\epsilon)$ queries. This follows by a folklore argument (attributed to Steven Rudich in [GNW95]). Consider a probability distribution over oracles which is uniformly distributed over all functions D that have errorless agreement ϵ with g. A semi-uniform reduction that makes $q = o(1/\epsilon)$ has probability $1 - o(1)$ to see only '$\perp$' on its q queries. Therefore, such a reduction cannot expect to get meaningful information from its oracle, and can be used to construct a small circuit (with no oracle) that has agreement $1 - \delta - o(1)$ with f. This shows that the existence of a reduction R unconditionally implies that f is not $(1 - \delta - o(1))$-hard. We explain this argument in more detail in Section 2.1.

We stress that the argument above critically depends on the fact that R is semi-uniform. A non-uniform reduction is allowed to receive an advice string α that is a function of D. Such an advice string can encode queries $y \in \{0,1\}^n$ such that $D(y) \neq \perp$. While this does not seem to help R in having large agreement with f, the argument of Rudich no longer applies. As we point out next, semi-uniform reductions are rare exceptions in the literature on hardness amplification, and the main contribution of this paper is developing techniques to extend Rudich's argument for *non-uniform* and *adaptive* reductions.

Non-uniform reductions for other settings of hardness amplification. Definition 1.2 is tailored for basic hardness amplification. However, the same reasoning can be used to define all the hardness amplification setups surveyed in Section 1.1. More precisely, we define the notion of "non-uniform reduction showing mildly-average-case to average-case hardness amplification" similarly by replacing the requirement that "D has errorless agreement ϵ with g" with the requirement that "D has agreement p with g" where $p = 1/2 + \epsilon$ in case $\ell = 1$ and $p = \epsilon$ in case $\ell > 1$. The discussion above about usefulness of non-uniform reductions trivially applies to this setting as well. Moreover, it trivially follows that a non-uniform reduction showing mildly-average-case to average-case hardness amplification implies a non-uniform reduction showing basic hardness amplification with essentially same parameters. As a consequence proving a lower bound of $q = \Omega(1/\epsilon)$ on the number of queries used by reductions showing basic hardness amplification entails the same lower bound in all the settings described in Section 1.1.

Function-generic hardness amplification. Definition 1.2 considers *specific* functions f, g. Most of the hardness amplification results in the literature are *function genereic* in the following sense:

Definition 1.3 (function-generic hardness amplification)**.** *Let ϵ, δ, a and ℓ be parameters. A* function-generic *reduction showing basic hardness amplification (for parameters ϵ, δ, a and ℓ) is a pair (Amp, R) where Amp is a map from functions $f : \{0,1\}^k \to \{0,1\}$ to functions $Amp(f) : \{0,1\}^n \to \{0,1\}^\ell$, and for every function $f : \{0,1\}^k \to \{0,1\}$, $R^{(\cdot)}$ is a non-uniform reduction showing basic hardness amplification for $f, g = Amp(f), \epsilon, \delta$ and a.*

We use definition 1.3 to also define the analogous notion for mildly-average-case to average-case hardness amplification. For the special case of Boolean mildly-average-case to average-case hardness amplification Definition 1.3 is equivalent to the notion of "black-box hardness amplification" defined in [SV10]. It is known that function-generic hardness amplification is equivalent to certain variants of list-decodable error-correcting codes. We elaborate on this connection in Section 1.4.

1.3 Our results

Function-generic hardness amplification. The vast majority of hardness amplification in the literature are function-generic reductions showing worst-case to average-case hardness amplification (or mildly-average-case to average-case hardness amplification). To the best of our knowledge, all the proofs in the literature are captured by Definition 1.3. Moreover, by the aforementioned connection to error-correcting codes (which we explain in Section 1.4), the reductions in these settings cannot be semi-uniform. Consequently, Rudich's argument does not apply for showing lower bounds on these reductions. Theorem 1.4 below proves lower bounds on the number of queries made by function-generic reductions showing basic hardness amplification.

Theorem 1.4 (main theorem for function-generic reductions)**.** *There exists a constant $c > 1$ such that the following holds. Let $k, n, \ell, \epsilon, \delta, r$ and a be parameters such that $a, \frac{1}{\epsilon}, \frac{1}{\delta}, n, r \le 2^{k/c}$ and $\delta \ge 2/3$. Let (Amp, R) be a function-generic reduction showing basic hardness amplification (for $f, g, \epsilon, \delta, \ell$ and a) and assume that R is of size r. Then, R makes at least $\frac{1}{100\epsilon}$ queries.*

We first note that the requirements of Theorem 1.4 capture the interesting regime of parameters: The requirement that $a, n, r \le 2^{k/c}$ is natural as reductions that do not fulfil this restriction produce exponential size circuits and are not useful for proving meaningful hardness amplification results. We remark that the requirement on r can in fact be removed from Theorem 1.4 as explained in the proof. We also stress that the constant $2/3$ can be replaced by any constant greater than $1/2$.

The bound in Theorem 1.4 is tight in the sense that there are function-generic reductions showing basic hardness amplification which for $\delta = \Omega(1)$ make $O(1/\epsilon)$ queries [GNW95, IJKW10, Wat10]. (In fact, some of these reductions are for showing non-Boolean mildly-average-case to average-case hardness amplification). For general δ, these reductions make $O(\frac{\log(1/\delta)}{\epsilon})$ queries. We can improve the bound in Theorem 1.4

to $\Omega(\frac{\log(1/\delta)}{\epsilon})$ which is tight for every δ. However, we only know how to do this in the special case where the reduction is *non-adaptive*.

By the previous discussion on the relationship between reductions showing various notions of hardness amplification it follows that Theorem 1.4 applies also for Boolean mildly-average-case to average-case amplification and gives the same lower bound of $\Omega(1/\epsilon)$ on the number of queries. In this setup the best known upper bounds [Imp95, KS03] make $O(\frac{\log(1/\delta)}{\epsilon^2})$ queries. A matching lower bound of $\Omega(\frac{\log(1/\delta)}{\epsilon^2})$ was given in [SV10] for the special case where the reduction R is *non-adaptive*. The argument in [SV10] heavily relies on the non-adaptivity of the reduction. The main contribution of this paper is developing techniques to handle reductions that are both *non-uniform* and *adaptive*, and Theorem 1.4 is the first bound on such general reductions (of any kind). Most reductions in the literature are non-adaptive, however there are some examples in the literature of adaptive reductions for hardness amplification and related tasks [SU05, GGH$^+$07].

Finally, we remark that the technique of [SV10] (which is different than the one used in this paper) can be adapted to the setting of basic hardness amplification (as observed in [Wat10]) showing our aforementioned lower bounds for the special case where the reduction is *non-adaptive*.

Function-specific hardness amplification. In contrast to function-generic reductions, non-uniform reductions for specific functions f, g (as defined in Definition 1.2) are allowed to depend on the choice of functions f, g and their particular properties. It is therefore harder to show lower bounds against such reductions. Moreover, as we now explain, we cannot expect to prove that for every function f, g, every non-uniform reduction R showing basic hardness amplification must use $\Omega(1/\epsilon)$ queries. This is because if f is a function such that there exists a small circuit C that has agreement $1-\delta$ with f, then there exists a trivial non-uniform reduction R that makes *no queries* as reduction R can ignores its oracle and set $R^{(\cdot)}(x) = C(x)$. Consequently, the best result that we can hope for in this setting is of the form: for every functions f, g and every non-uniform reduction $R^{(\cdot)}$ for f, g, if R makes $o(1/\epsilon)$ queries then there exists a circuit C (with no oracle) of size comparable to that of R that has agreement almost $1-\delta$ with f. Theorem 1.5 stated below is of this form.

Theorem 1.5 (main theorem for function-specific reductions)**.** *Let ϵ, δ and a be parameters. Let $f : \{0,1\}^k \to \{0,1\}$ and $g : \{0,1\}^n \to \{0,1\}^\ell$ be functions. Let $R^{(\cdot)}$ be a non-uniform reduction for f, g, ϵ, δ and a. If R is of size r and makes q queries then for every $\rho \geq 10\epsilon q$ there exists a circuit C of size $r + poly(a, q, n, 1/\rho)$ that has agreement $1-\delta-\rho$ with f.*

Theorem 1.5 says that if $q = o(1/\epsilon)$ then the mere existence of reduction R implies the existence of a circuit C that has agreement $1-\delta-o(1)$ with f. This can be interpreted as a lower bound on the number of queries in the following sense: Reductions making $o(1/\epsilon)$ queries are not useful as their existence implies that the hardness amplification assumption does not hold.

Function-specific hardness amplification in the literature. Function-specific hardness amplification results are quite rare. One motivation for developing such results is that function-specific reductions can bypass the coding theoretic objection and be semi-uniform (or even completely uniform). Examples are the reductions in [IW98, TV07, Tre03, Tre05]. Theorem 1.5 shows that such reductions must make $\Omega(1/\epsilon)$ queries even if they are non-uniform.

In the function-specific setting there are few examples in the literature of reductions for tasks related to hardness amplification that have proofs not captured by Definition 1.2. It was pointed out in [GTS07] that the techniques of [GSTS07, Ats06] (that show some worst-case to average-case reduction for NP) are not *black-box* in a sense that we now explain. Semi-uniform reductions are black-box in the sense that R has only black-box access to D. Non-uniform reductions allow R to also get some short advice string α about D. Note that there is no requirement that α is generated using black-box access to D (and this is why we refrain from using the term "black-box" when referring to non-uniform reductions). However, even non-uniform reductions make no assumption about the oracle D and are required to perform for every function D (even if D is not computable by a small circuit). The reductions used in [GSTS07, Ats06] are only guaranteed to perform in case D is efficient, and are therefore not captured by Definition 1.2. The reader is referred to [GTS07, GV08] for a discussion on such reductions.

1.4 Hardness amplification and error-correcting codes

It was pointed out in [STV01] that hardness amplification is closely related to error-correcting codes. We now explain this relationship using our terminology. For this purpose, we identify a function $f : \{0,1\}^k \rightarrow \{0,1\}$ with its truth table which is a string $f \in \{0,1\}^K$ for $K = 2^k$.

Definition 1.6 (List-decodable codes). *A map $Enc : \{0,1\}^K \rightarrow \{0,1\}^N$ is (ϵ, A)-list-decodable if for every $D \in \{0,1\}^N$, there is a list of at most A strings $f \in \{0,1\}^K$ such that D has agreement $1/2 + \epsilon$ with $Enc(f)$. Enc is uniquely-decodable if $A = 1$.*

Let $K = 2^k$ and let δ be a parameter. Local decoders (for uniquely-decodable codes) are randomized oracle procedures $Dec^{(\cdot)}$ which when given oracle access to D and input $x \in \{0,1\}^k$, returns $f(x)$ with probability $1 - \delta$. In the case of list-decodable codes, the local decoder Dec also receives a second input α which is the index in the list. This leads to the following definition.

Definition 1.7 (Local list-decoder). *Let $Enc : \{0,1\}^K \rightarrow \{0,1\}^N$ be (ϵ, A)-list-decodable. A local list-decoder with list-size A' and error δ for Enc is a randomized oracle procedure $Dec^{(\cdot)}$ such that for every $D \in \{0,1\}^N$, and every f in the list of D, there exists an $1 \le \alpha \le A'$ such that for every $x \in \{0,1\}^k$, $\Pr[Dec^D(x, \alpha) = f(x)] \ge 1 - \delta$ where the probability is over the internal coin tosses of Dec.*

The following lemma shows that local list-decoding implies function generic hardness amplification. It follows that our lower bounds on function-generic hardness amplification also apply (with the same parameters) for local list-decoders (even if they make adaptive queries).

Lemma 1.8 (Local list-decoders imply function-generic hardness amplification). *Let $Enc : \{0,1\}^{2^k} \rightarrow \{0,1\}^{2^n}$ be $(\epsilon, 2^{a'})$-list-decodable and let Dec be a local list decoder for Enc with list size $2^{a'}$ and error δ, and assume that Dec makes at most q queries and tosses at most t coins. Then, there is a function-generic reduction showing mildly-average-case to average-case amplification for k, n, ϵ, δ with $\ell = 1$ and $a = a' + t$, and furthermore the reduction makes q queries.*

Proof. (of Lemma 1.8) Let Enc be $(\epsilon, 2^{a'})$-list-decodable and let Dec be a local list-decoder for Enc with list size $2^{a'}$ and error δ. Let $D \in \{0,1\}^n$. By an averaging argument, for every f in the list of D, there exists a fixing $\beta \in \{0,1\}^t$ for the coin tosses of Dec and $1 \le \alpha \le 2^{a'}$ such that $Dec^D(\cdot, \alpha)$ has agreement $1 - \delta$ with f when its coins are fixed to β. We define $Amp = Enc$ and $D^{(\cdot)}(x; (\alpha, \beta)) = Dec^{(\cdot)}(x, \alpha)$ using β as coins.[2] □

It is interesting to note that even in the special case of unique decoding, Lemma 1.8 gives a function-generic that is non-uniform. The following corollary is obtained by applying Theorem 1.4.

[2] Note that the argument above applies even if we use a less restrictive notion of local list-decoders in which the requirement made in Definition 1.7 that "for every $x \in \{0,1\}^k$..." is replaced by "for a $(1-\delta)$-fraction of $x \in \{0,1\}^k$..." and then the reduction is for $\delta' = 2\delta$. Thus, our lower bounds apply even in this more general setting.

Corollary 1.9 (Lower bounds on number of queries of local list-decoders)**.** *There exists a constant $c > 1$ such that the following holds. Let $Enc : \{0,1\}^{2^k} \to \{0,1\}^{2^n}$ be $(\epsilon, 2^{a'})$-list-decodable and let Dec be a local list decoder for Enc with list size $2^{a'}$ and error δ, and assume that Dec tosses at most t coins. If $a', \frac{1}{\epsilon}, \frac{1}{\delta}, n, t \leq 2^{k/c}$ then Dec makes at least $1/100\epsilon$ queries.*

We remark that the main question in locally-decodable codes is how many queries are needed for uniquely-decodable codes with constant rate. In our terminology, this corresponds to constant ϵ and δ and our results are interesting for a different regime of parameters.

Decoding from erasures. The lower bound of Theorem 1.4 holds even for basic hardness amplification. The corresponding coding-theoretic setting is that of list-decoding from erasures. More precisely, in Definition 1.6 we can allow D to have errorless agreement ϵ with $Enc(f)$ (rather than agreement $1/2 + \epsilon$ with $Enc(f)$). In coding theoretic terminology this corresponds to a noisy channel that corrupts $Enc(f)$ by erasing a $1 - \epsilon$ fraction of the symbols (by replacing them with the special symbol '$\perp$') and keeping the remaining symbols unchanged. Corollary 1.9 applies in this setting even when allowing list-decoding.

1.5 Related work

We have already surveyed many results on hardness amplification. We now survey some relevant previous work regarding limitations on proof techniques for hardness amplification. We focus on such previous work that is relevant to this paper and the reader is referred to [SV10] for a more comprehensive survey.

The complexity of reductions showing hardness amplification was studied in [SV10, GR08]. Both papers show that function-generic reductions for mildly-average-case to average-case hardness amplification cannot be computed by small constant depth circuits if ϵ is small. Both results fail to rule out general reductions. The result of [GR08] rules out *adaptive* reductions but only if they use very low non-uniformity (meaning that $a = O(\log(1/\epsilon)) \ll k$. The result of [SV10] rules out non-uniform reductions with large non-uniformity (allowing $a = 2^{\Omega(k)}$) but only if they are *non-adaptive*. As mentioned earlier, our results extend previous lower bounds on the number of queries that were proven in [SV10] for *non-adaptive* reductions. This suggests that our techniques may be useful in extending the result of [SV10] regarding constant depth circuits to *adaptive* reductions. We stress however, that we are studying reductions showing *basic* hardness amplification and there are such reductions in the literature that can be computed by small constant depth circuits [IJKW10].

In this paper we are interested in the complexity of function-generic reductions showing hardness amplification. There is an orthogonal line of work [Vio05a, LTW08] that aims to show limitations on "fully-black-box constructions" of hardness amplifications. In our terminology, these are function-generic non-uniform reductions (Amp, R) with the restriction that there exists an oracle machine $M^{(\cdot)}$ called *construction* such that for every function f, $Amp(f)$ is implemented by M^f. The goal in this direction is to prove lower bounds on the complexity of M (which corresponds to encoding), whereas we focus on R (which corresponds to decoding).

There are many other results showing limitations on reductions for hardness amplification and related tasks in various settings. A partial list includes [FF93, TV07, BT06, RTV04, Vio05b, AGGM06, LTW07].

Organization of this paper. In Section 2 we give a high level overview of the proof. We present the formal proof in Section 3.

2 Overview of the technique

In this Section we give a high level overview of the proof. The reader may skip to the formal proof given in Section 3 at any time. The purpose of this section is to highlight the main ideas and choices made in the proof.

Our goal is to prove Theorem 1.5. Let us recall the setup. We are given functions $f : \{0,1\}^k \to \{0,1\}$, $g : \{0,1\}^n \to \{0,1\}^\ell$ and parameters ϵ, δ and a. We consider a non-uniform reduction $R^{(\cdot)}$ for f, g, ϵ, δ and a, and let r be the size of R and q be the number of queries. Let $\rho \geq 10\epsilon q$. We can assume that $q \leq 1/10\epsilon$ so that $\rho \leq 1$. Our goal is to show that there exists a circuit C of size $r + \text{poly}(a, q, n, 1/\rho)$ that has agreement $1-\delta-\rho$ with f. We remark that Theorem 1.4 easily follows from Theorem 1.5 as if we choose function f at random, it is unlikely that there is a small circuit with agreement $1 - \delta - o(1) \geq 2/3$ with f. This rules out function-generic reductions making $o(1/\epsilon)$ queries as by Theorem 1.5 the existence of a function-generic reduction implies the existence of such a circuit.

2.1 The case of semi-uniform reductions

As an appetizer, let us first consider the special case that R is semi-uniform which means that $a = 0$ and R does not get an advice string α. Let $D : \{0,1\}^n \to \{0,1\}^\ell \cup \{\perp\}$ be some function. We say that $y \in \{0,1\}^n$ *answers* (with respect to D) if $D(y) \neq \perp$. We say that $x \in \{0,1\}^k$ is *silent* (with respect to reduction R and function D) if no query asked by $R^D(x)$ answers. We consider the following probability space.

Definition 2.1 (Random oracle)**.** *Let $(V(y))_{y\in\{0,1\}^n}$ be a sequence of independent and identically distributed random variables, where for every $y \in \{0,1\}^n$, $V(y) = 1$ with probability 2ϵ and $V(y) = 0$ with probability $1 - 2\epsilon$. We view random variable V as a function $V : \{0,1\}^n \to \{0,1\}$ and define a random variable $D : \{0,1\}^n \to \{0,1\}^\ell \cup \{\perp\}$ by $D(y) = g(y)$ if $V(y) = 1$ and $D(y) = \perp$ if $V(y) = 0$.*

We will use this probability space throughout this section and all expressions involving probability or expectation refer to this space. By a Chernoff bound, with probability $1-2^{-\Omega(2^k)}$, D has errorless agreement ϵ with g which implies that $R^D(\cdot)$ has agreement $1 - \delta$ with f. For every $x \in \{0,1\}^k$ we define a random variable A_x indicating the event that x is silent with respect to D. We have that for every $x \in \{0,1\}^k$, $\mathbb{E}[A_x] = \Pr[A_x = 1] \geq (1 - 2\epsilon)^q \geq 1 - 2\epsilon q \geq 1 - \rho$. Let $A = \sum_{x\in\{0,1\}^k} A_x$ be the random variable counting the number of silent inputs. By linearity of expectation $\mathbb{E}[A] \geq 2^k \cdot (1 - \rho)$. By averaging, there exists a function $D' : \{0,1\}^n \to \{0,1\}^\ell \cup \{\perp\}$ which has errorless agreement ϵ with g and a $1 - \rho$ fraction of $x \in \{0,1\}^k$ are silent with respect to D'. We have that $R^{D'}(\cdot)$ has agreement $1 - \delta$ with f. Consider circuit C (with no oracle) which on input x simulates $R^{D'}(x)$ by answering all queries made to the oracle by $\perp$. It follows that C (which has size comparable to R) simulates $R^{D'}$ correctly on all silent inputs and therefore has agreement $1 - \delta - \rho$ with f. This concludes the proof in this case.

An advantage of the argument above is that it allows us to focus on individual $x \in \{0,1\}^k$ and analyze the probability that x is silent. We will try to maintain this feature in the general case.

2.2 Strategy for non-uniform reductions

We now consider non-uniform reductions that receive an advice string $\alpha \in \{0,1\}^a$. The definition of non-uniform reductions says that for every function D that has errorless agreement ϵ with g there exists $\alpha \in \{0,1\}^a$ such that $R^D(\cdot, \alpha)$ has agreement $1 - \delta$ with f. For every such D, let $\alpha(D)$ to be some advice string that is good for D. This defines a map α from oracles to advice strings.

Let us consider the probability space in Definition 2.1. We once again have that with probability $1 - 2^{-\Omega(2^k)}$, $R^D(\cdot, \alpha(D))$ has agreement $1-\delta$ with f. However, we cannot expect to show that there are many silent inputs. For all we know, $\alpha(D)$ may contain an encoding of a $y' \in \{0,1\}^n$ for which $D(y') \neq \perp$. This allows $R^D(x, \alpha(D))$ to ask a query that answers and note that this holds with probability one for every $x \in \{0,1\}^k$. Consequently, no inputs are silent for this reduction, and the previous argument fails.

Conditioning on a fixed advice string. We would like to return to the setup where R does not obtain advice about D. For this purpose, we note that there exists an advice string $\alpha' \in \{0,1\}^a$ such that $\Pr[\alpha(D) = \alpha'] \geq 2^{-a}$. Let E denote the event

$$E = \left\{\alpha(D) = \alpha'\right\} \cap \{D \text{ has errorless agreement } \epsilon \text{ with } g\}.$$

Note that $\Pr[E] \geq 2^{-(a+1)}$. We consider the probability space conditioned on the event E (which we refer to as the *conditioned space*). In the conditioned space, R uses the same advice string α' for every choice of oracle D. Thus, we can think of R as being hardwired with the advice string α' (meaning that R does not really receive advice about D in the conditioned space). The penalty in this approach is that the distribution over oracles D in the conditioned space is different than the distribution in the original space. More precisely, the variables $(V(y))_{y \in \{0,1\}^n}$ may become correlated, and individual variables $V(y)$ may be distributed differenlty than in the original space. We can hope to control these effects as the advice string is relatively short compared to the length of the truth table of V. Indeed, note that for the reduction R described above, the conditioned space can have some $y \in \{0,1\}^n$ on which the event $\{V(y) = 1\}$ holds with probability one, meaning that $D(y)$ always answers. However the number of such bad y is bounded by a. This suggests the following proof strategy.

Proof strategy for non-uniform reductions

- Given R and event E, identify a small set of "'bad queries" $B \subseteq \{0,1\}^n$ (where small means poly$(a, q, 1/\rho)$).
- Say that x is *almost silent* if all queries $y \notin B$ asked by $R^D(x, \alpha')$ do not answer. Show that for every $x \in \{0,1\}^k$, the probability (in the conditioned space) that x is almost-silent is at least $1-\rho$.
- It follows as before (by linearity of expectation and the probabilistic method) that there exists a function D' such that $R^{D'}(\cdot, \alpha')$ has agreement $1-\delta$ with f, and furthermore, a $1-\rho$ fraction of $x \in \{0,1\}^k$ are almost silent with respect to D'.
- Construct a circuit $C(x)$ that has agreement $1-\delta-\rho$ with f as follows: C is hardwired with B and the values $(D'(y))_{y \in B}$. On input x, C simulates $R^{D'}(x, \alpha')$ answering queries y to the oracle by $D'(y)$ if $y \in B$ and by '$\perp$' if $y \notin B$. Note that C correctly simulates $R^{D'}(\cdot, \alpha')$ on almost silent inputs, and that C can be implemented by a circuit of size $r + \text{poly}(a, q, n, 1/\rho)$ as required.

In the special case described above where $\alpha(D)$ encodes queries on which D answers, we can implement this strategy by simply setting B to be these queries. We next explain how to implement this strategy for non-adaptive reductions.

2.3 The case of non-uniform reductions that are non-adaptive

We now consider the special case where the non-uniform reduction R is non-adaptive. We use an approach developed in [SV10] for handling non-adaptive reductions in the related setting of Boolean mildly-average-case to average-case hardness amplification. The approach relies on the following simple information theoretic lemma from [SV10]. (It is explained in [SV10] that this Lemma can be seen as a generalization of a Lemma from [Raz98] and that it also follows from the technique of [EIRS01]).

Lemma 2.2. *Let $L \subseteq \{0,1\}^n$ and let $(V(y))_{y \in L}$ be independent random variables. Let a, q and η be parameters and let E be an event such that $\Pr[E] \geq 2^{-a}$. There exists a set $B \subseteq L$ such that $|B| = O(aq/\eta^2)$ such that for every $y_1, \ldots, y_q \in L \setminus B$, the distribution $(V(y_1), \ldots, V(y_q))$ is η-close to the distribution $((V(y_1), \ldots, V(y_q))|E)$.*[3]

We are planning to implement the strategy of Section 2.2. Lemma 2.2 (applied with $L = \{0,1\}^n$ and $\eta = \rho/2$) gives a way to define a set B. We are left with showing that for every $x \in \{0,1\}^k$, the probability (in the conditioned space) that x is almost silent is at least $1 - \rho$. Indeed, for every $x \in \{0,1\}^k$ the non-adaptive reduction R defines specific queries $y_1, \ldots, y_q$ to its oracle (and by the non-adaptivity of R these queries are fixed as a function of x). Assume w.l.o.g. that the last $0 \leq t \leq q$ queries are in B. In the original probability space the probability that $y_1, \ldots, y_{q-t}$ don't answer is simply

$$\Pr[V(y_1) = 0 \wedge \ldots \wedge V(y_{q-t}) = 0] = (1 - 2\epsilon)^{(q-t)} \geq 1 - \rho/2.$$

By Lemma 2.2 we have that in the conditioned space:

$$\Pr[V(y_1) = 0 \wedge \ldots \wedge V(y_{q-t}) = 0|E] \geq \Pr[V(y_1) = 0 \wedge \ldots \wedge V(y_{q-t}) = 0] - \rho/2 \geq 1 - \rho$$

meaning that x is almost silent with probability $1 - \rho$ in the conditioned space. This concludes the proof by the strategy outlined in Section 2.2.

2.4 The case non-uniform reductions that are adaptive

A counterexample to the strategy of Section 2.2. The proof strategy we devised in Section 2.2 fails for adaptive reductions in the following sense. There exists an oracle procedure R that makes $O(n)$ queries, and an event E that has probability at least 2^{-a} (for $a = O(n \log(1/\epsilon)$ which is reasonable for proving hardness amplification results) with the following properties: No matter how we choose a set $B \subseteq \{0,1\}^n$ of size $o(2^n)$ of "bad queries", for every input x, with probability $1 - o(1)$ over the conditioned space, $R^D(x)$ asks a query y that answers and is not in B. This means that we cannot hope to show that x is almost-silent with high probability as required by the strategy of Section 2.2.

We now sketch this counterexample. Fix some distinct $y_1, \ldots, y_n \in \{0,1\}^n$ and $z_1, \ldots, z_n \in \{0,1\}^n$. For every $1 \leq i \leq n$, we interpret the pair $(V(y_i), V(z_i))$ as a bit P_i using "von-Neumann's extractor". More precisely, P_i is defined to be $V(y_i)$ in case $y_i \neq z_i$ and is undefined otherwise. If the sequence $P_1, \ldots, P_n$ is defined, then we can view it as a string $P \in \{0,1\}^n$. We define event E as follows:

$$E = \{\forall i : V(y_i) \neq V(z_i)\} \cap \{V(P) = 1\}.$$

Note that conditioned on E, the random variable P is defined with probability one. Furthermore, random variable P is uniformly distributed conditioned on E.

[3]Two distributions P, Q over the same domain are ϵ-close if for every event A, $|\Pr_P[A] - \Pr_Q[A]| \leq \epsilon$.

An adaptive procedure R with $O(n)$ queries can easily find a query that answers conditioned on E. (In fact, the procedure R described next uses only "two levels of adaptivity"). Procedure R first queries oracle D at $y_1, \ldots, y_n$ and computes P. It then queries D at P and note that query P always answers conditioned on E.

However, note that no matter how we choose $B \subseteq \{0,1\}^n$ of size $o(2^n)$, P that is uniformly distributed in $\{0,1\}^n$ conditioned on E is unlikely to land in B.

The main technical contribution of this paper is developing an approach to handle adaptive reductions. We now describe some of the high-level ideas. The precise details appear in Section 3.

Further conditioning. We start by modifying the strategy of Section 2.2. Instead of performing the analysis in the conditioned space (that is conditioned on E), we choose some event $E' \subseteq E$ and perform the analysis conditioned on E'. We refer to this probability space as the "further conditioned space". Note that $\alpha(D)$ is fixed conditioned on any event $E' \subseteq E$ which means that we can apply the strategy of Section 2.2 replacing E with any event $E' \subseteq E$. To make this approach less abstract, consider the event E from the counterexample above. We consider the event $E' = E \cap \{\forall i : V(y_i) = 1 \wedge V(z_i) = 0\}$. This gives that P is fixed conditioned on E' and we can hope to mark P as a bad query and implement the strategy of Section 2.2. We remark that when applying Lemma 2.2 on E we obtain the set $B = \{y_1, \ldots, y_n, z_1, \ldots, z_n\}$. Which means that Lemma 2.2 "correctly identifies" queries $y_1, \ldots, y_n$ and $z_1, \ldots, z_n$ as "problematic". This suggests the following iterative further conditioning strategy:

Iterative further conditioning

- Given R and event E, identify a small set of "'bad queries" $B \subseteq \{0,1\}^n$ using Lemma 2.2.
- Let $E' = E \cap \{\forall y \in B : V(y) = c_y\}$ where $\{c_y\}_{y \in B}$ are some constants.
- Set $E \leftarrow E'$ and repeat.

We first observe that applying two steps of this strategy "correctly handles" the event E of the counterexample in the following sense: In the first step, Lemma 2.2 identifies the set $B_1 = \{y_1, \ldots, y_n, z_1, \ldots, z_n\}$ and fixes V on these queries to obtain event E_1 in which P is fixed to some string p. In the second step, Lemma 2.2 identifies the set $B_2 = \{p\}$ and we mark it as bad. Overall, after two steps we mark the queries in $B = B_1 \cup B_2$ as bad. Recall that the strategy of Section 2.2 requires that we bound the probability that x is almost silent for every input x. Having fixed P to p and marked p as bad, we can indeed bound this probability in the further conditioned space.

In general we expect to make q iterations of further conditioning (one for each "level of adaptivity") and and it is easy to extend the counterexample to show that this is indeed necessary.

A canonical execution. In the i'th iteration of further conditioning we identify a set B of bad queries using Lemma 2.2. We would like to also use the Lemma to argue that for every input $x \in \{0,1\}^k$, the query Q_i^x asked by $R^D(x, \alpha')$ at the i'th step is unlikely to answer if it is good. However, Q_i^x is a random variable that depends on V and Lemma 2.2 can only argue about fixed queries y. We handle this problem by considering a mental experiment which we refer to as the *canonical execution* of $R^D(x, \alpha')$. In this experiment we simulate $R^D(x, \alpha')$ and whenever R asks a query $y \in \{0,1\}^n$, we check whether y is considered bad at this point. If it does, we answer the query by $D(y)$ (as in a real execution). However, if it doesn't, we answer the query by '$\perp$' (regardless of $D(y)$). In the further conditioned space, the i'th query W_i^x of the canonical execution is a fixed constant y that depends on x but not on V. This is because the value of V was fixed on all bad queries during the further conditioning pocess. Thus, we know in advance what are the queries that will be asked in the canonical execution and this allows us to perform the strategy of Section 2.2 on the canonical execution. To complete the argument we show that if D does not answer on all good queries in the canonical execution of $R^D(x, \alpha')$, then D also does not answer on all good queries in the real execution (or in other words that x is almost silent). This allows us to implement a variant of the strategy of Section 2.2. Loosely speaking, the approach above can be summarized as saying that in *basic* hardness amplification, even adaptive reductions cannot expect to find queries that answer, unless their advice string points them to such queries.

Maintaining probabilities during the further conditioning process. A technical problem is that each iteration of further conditioning changes the probability space. If we bound the probability of some event A conditioned on event E_i and later further condition to $E_{i+1} \subseteq E_i$, then we no longer have control over the probability of event A conditioned on the new space E_{i+1}. Because of this problem, the proof presented in Section 3 is somewhat cumbersome and employs the analysis of the iterative further conditioning process in a carefully chosen order.

3 Proof of main theorems

3.1 Preparations

In this section we prove Theorem 1.4 and Theorem 1.5. We start by proving Theorem 1.5 (as Theorem 1.4 easily follows from Theorem 1.5). Let us start by recalling the setup.

The setup. We are given functions $f : \{0,1\}^k \to \{0,1\}$, $g : \{0,1\}^n \to \{0,1\}^\ell$ and parameters ϵ, δ and a. We consider a non-uniform reduction $R^{(\cdot)}$ for f, g, ϵ, δ and a, and let r be the size of R and q be the number of queries. Let $\rho \geq 10\epsilon q$. Our goal is to show that there exists a circuit C of size $r + \text{poly}(a, q, n, 1/\rho)$ that has agreement $1 - \delta - \rho$ with f.

The map $\alpha(D)$. Let α be a map that for every D that has ϵ errorless agreement with g, assigns an advice string $\alpha(D) \in \{0,1\}^a$ such that $R^D(x, \alpha)$ has agreement $1 - \delta$ with f. Such a map exists by Definition 1.2.

Some notation. For a function $V : \{0,1\}^n \to \{0,1\}$ and a set $B \subseteq \{0,1\}^n$ we define $V(B) = (V(y))_{y \in B}$. We view $V(B)$ as an element in $\{0,1\}^B$.

The probability space. We use the probability space of Definition 2.1 which we now specify using more precise notation. The probability space consists of independent identically distributed random variables $(V(y))_{y \in \{0,1\}^n}$ where for each $y \in \{0,1\}^n$, $V(y) = 1$ with probability 2ϵ and $V(y) = 0$ with probability $1 - 2\epsilon$. We view random variable V as a function $V : \{0,1\}^n \to \{0,1\}$. We define a random variable $D : \{0,1\}^n \to \{0,1\}^\ell \cup \{\perp\}$ by $D(y) = g(y)$ if $V(y) = 1$ and $D(y) = \perp$ if $V(y) = 0$.

We will use this probability space throughout this section and all expressions involving probability or expectation refer to this space. The probability space is defined over the set $S = \{0,1\}^{\{0,1\}^n}$ of all functions $V : \{0,1\}^n \to \{0,1\}$. Events in this probability space are subsets $E \subseteq S$. A random variable A is a map from S to some set. For a (fixed) function $V' : \{0,1\}^n \to \{0,1\}$, we can think of V' as a "point" in S and let $A[V']$ denote the value of the map A when applied on V'. Thus, for example, $D[V']$ denotes the function obtained when the point in the probability space is V'.

The event E. By a Chernoff bound we have that

$$\Pr[D \text{ has errorless agreement } \epsilon \text{ with } g] \geq 1 - 2^{-\Omega(2^{-k})}.$$

There exists a string $\alpha' \in \{0,1\}^a$ such that $\Pr[\alpha(D) = \alpha'] \geq 2^{-a}$. We define

$$E = \{\alpha(D) = \alpha'\} \cap \{D \text{ has errorless agreement } \epsilon \text{ with } g\}.$$

Note that $\Pr[E] \geq 2^{-a} - 2^{-\Omega(2^k)} \geq 2^{-(a+1)}$ and $\alpha(D)$ is fixed to α' in the event E.

3.2 Real and canonical executions

The real execution. For every $x \in \{0,1\}^k$ and $1 \leq i \leq q$ we define the random variable Q_i^x to be the i'th query asked by $R^D(x, \alpha')$. We refer to $Q_1^x, \ldots, Q_q^x$ as the "real queries". Note that as R is adaptive, these queries depend on D.

The canonical execution. We now define a different concept of "canonical queries" $W_1^x, \ldots, W_q^x$ as follows. Let $B_1, \ldots, B_q$ be subsets of $\{0,1\}^n$ that we determine later. (We think of B_i as a set of "bad queries at stage i"). For every $1 \leq i \leq q$ we define $\bar{B}_i = \bigcup_{1 \leq j \leq i} B_j$ to be the set of all queries marked as "bad" at stage $\leq i$. We also define $B = \bar{B}_q$ to be the set of all queries marked as "bad".

For every $x \in \{0,1\}^k$, we define the "canonical execution" of $R^D(x, \alpha')$ as follows: We simulate $R^D(x, \alpha)$ and when the simulation asks its i'th query (denoted by W_i^x, we answer it by the following "canonical rule": We answer the query by '$\perp$' if $y \notin \bar{B}_i$ and by $D(W_i^x)$ otherwise. More precisely, the first canonical query is $W_1^x = Q_1^x$. At every step $i \geq 1$, the canonical execution answers query W_i^x by the canonical rule above. This answer is then used by $R^D(x, \alpha')$ to determine its next query W_{i+1}^x, and this iterative process determines $W_1^x, \ldots, W_q^x$.

Note that as R is adaptive, the queries $W_1^x, \ldots, W_q^x$ that are queried in the canonical execution may differ from the "real queries" $Q_1^x, \ldots, Q_q^x$ because the answers supplied in the canonical execution may differ from those of D.
For every $x \in \{0,1\}^k$ and $1 \leq i \leq q$ we also define the following random variables:

- P_i^x is an indicator random variable indicating the event $\{V(Q_i^x) = 1 \wedge Q_i^x \notin \bar{B}_i\}$.
- A_i^x is an indicator random variable indicating the event $\{V(W_i^x) = 1 \wedge W_i^x \notin \bar{B}_i\}$.

Roadmap: the strategy of the proof. We now explain the intuition behind the definitions above and sketch the argument for the proof below. The reader can safely skip this paragraph and go directly to the formal proof if he wishes.

The reduction R is adaptive, and therefore the queries $Q_1^x, \ldots, Q_q^x$ made in the real execution on input x may depend on the answers of D. The advantage of the canonical execution is that the queries $W_1^x, \ldots, W_q^x$ made in the canonical execution on input x only depend on the answers of D to queries in B and we refer to those queries as "bad queries". We stress that the set B is fixed, and does not depend on x or D. Thus, we can simulate the canonical execution on all inputs $x \in \{0,1\}^k$ without access to oracle D if we know the answers to bad queries. This means that we can construct a circuit C (with no oracle) that simulates the canonical execution on all inputs by hardwiring C with the answers of D to bad queries. The size of the circuit C depends on the size of $|B|$ and is small if $|B|$ is small.

We stress however, that we are interested in simulating the real execution and not the canonical execution. We will say that x is *almost silent* if $\sum_{1 \le i \le q} P_i^x = 0$ and *canonically almost silent* if $\sum_{1 \le i \le q} A_i^x = 0$. We first observe that if x is canonically almost silent, then the answers supplied by the canonical rule coincide with the answers of D. This means that the canonical execution coincides with the real execution and in particular that x is almost silent. It follows that on a canonically almost silent x, the circuit C described above correctly simulates the real execution of $R^D(x, \alpha')$.

We will use the probabilistic method to show that there exist sets $B_1, \ldots, B_q$ such that their union B is small, and furthermore there exists a (fixed) function $V' \in E$ such that for V' and the oracle $D[V']$ determined from it, a $1-\rho$ fraction of inputs $x \in \{0,1\}^k$ are canonically almost silent. The conclusion is that the circuit C defined above (that has no oracle) correctly simulates the real execution of $R^{D[V']}(x, \alpha')$ on a $1-\rho$ fraction of inputs $x \in \{0,1\}^k$ and therefore C has agreement $1-\delta-\rho$ with f.

The main technical lemma. We now continue with the formal presentation of the proof. Let $V' : \{0,1\}^n \to \{0,1\}$ be some function and let $B_1, \ldots, B_q$ be some subsets of $\{0,1\}^n$. We say that an input $x \in \{0,1\}^k$ is *almost silent* if $\sum_{1 \le i \le q} P_i^x[V'] = 0$. We say that an input $x \in \{0,1\}^k$ is *canonically almost silent* if $\sum_{1 \le i \le q} A_i^x[V'] = 0$. We use the probabilistic method to prove the following lemma (which is the main technical lemma in the proof).

Lemma 3.1. *There exists $V' : \{0,1\}^n \to \{0,1\}$ such that $V' \in E$ and sets $B_1, \ldots, B_q \subseteq \{0,1\}^n$ such that*

- $|B| = poly(a, q, 1/\rho)$.
- *The number of canonically almost silent inputs $x \in \{0,1\}^k$ is at least $(1-\rho) \cdot 2^k$.*

We prove Lemma 3.1 in Section 3.5. We now show that Theorem 1.5 and Theorem 1.4 follow from Lemma 3.1.

3.3 Proof of Theorem 1.5

Let V' and $B_1, \ldots, B_q$ be the function and sets guaranteed by Lemma 3.1. We first observe that:

Lemma 3.2. *Every canonically almost silent $x \in \{0,1\}^k$ is also almost silent.*

Proof. Let $x \in \{0,1\}^k$ be canonically almost silent. We will show that for every $1 \le i \le q$, $Q_i^x[V'] = W_i^x[V']$. Note that this implies that for every $1 \le i \le q$, $P_i^x[V'] = A_i^x[V']$ and therefore x is also almost silent. We have that $Q_1^x[V'] = W_1^x[V']$ by definition. We know that $A_1^x[V'] = 0$ and we now observe that this implies that the query $Q_1^x[V']$ is answered in the same way in both the canonical execution and the real execution. This follows by the following case analysis. If $W_1^x[V'] \in \bar{B}_1$ then the canonical execution answers in the same way as the real execution by definition. If $W_1^x[V'] \notin \bar{B}_1$ then by definition, the canonical execution answers it by '$\perp$'. However, as $A_1^x[V'] = 0$ we have that $V'(W_1^x[V']) = 0$ which means that $D[V'](W_i^x[V']) = \perp$. It follows that in both cases the answers coincide.[4] Therefore, the next query is the same in both executions and we have that $Q_2^x[V'] = W_2^x[V']$. We can continue this reasoning and conclude by induction for every $1 \le i \le q$, $Q_i^x[V'] = W_i^x[V']$. □

[4] A subtle point is that it may be the case that $W_i^x[V']$ is not in $\bar{B}_1$ but is in B. This happens if this query is considered good at step 1 and bad later on. Nevertheless, the fact that $A_1^x[V'] = 0$ implies that this query does not answer regardless of whether it becomes bad later on.

It follows that the number of almost silent inputs $x \in \{0,1\}^k$ is at least $(1-\rho) \cdot 2^k$. We now define a circuit C as follows: C is hardwired with α', the sets $B_1, \ldots, B_q$ (that can be encoded by a bit string of length $|B| \cdot (n+q)$) and $(D[V'](B)$ (which can be encoded as a string of length $|B|$). On input $x \in \{0,1\}^k$, C simulates $R^{D[V']}(x, \alpha')$ as follows. When R makes its i'th query $y \in \{0,1\}^n$ to its oracle, C supplies the answer according to the canonical rule. That is, C supplies answer '$\perp$' if $y \notin \bar{B}_i$, and C supplies answer $D[V'](y)$ otherwise. Note that $C(x)$ correctly simulates $R^{D[V']}(x, \alpha')$ on every x that is almost silent. (This is because on such inputs, C answers all queries in the same way as $D[V']$).

We have that $V' \in E$ and therefore $R^{D[V']}(\cdot, \alpha')$ has agreement $1-\delta$ with f. Circuit C has agreement $1-\rho$ with $R^{D[V']}(\cdot, \alpha')$ and therefore C has agreement $1-\delta-\rho$ with f. The size of C is bounded by $r + a + O(|B| \cdot n) + \text{poly}(n, q) = r + \text{poly}(a, q, 1/\rho, n)$ as required This completes the proof of Theorem 1.5.

3.4 Proof of Theorem 1.4

Theorem 1.4 easily follows from Theorem 1.5. Let $k, n, \ell, \epsilon, \delta, r$ and a be parameters such that $a, \frac{1}{\epsilon}, \frac{1}{\delta}, n, r \leq 2^{k/c}$ for a constant $c > 1$ that we determine later and let $\delta \geq 2/3$. Let (Amp, R) be a function-generic reduction showing basic hardness amplification (for $f, g, \epsilon, \delta, \ell$ and a) and assume that R is of size r. Then, by theorem 1.5, if R makes $q \leq \frac{1}{100\epsilon}$ queries, we can set $\rho = 10\epsilon q \leq 1/10$ and have that for every function f, there exists a circuit C of size $r + \text{poly}(a, q, 1/\rho, n) = 2^{O(k/c)}$ that has agreement $1-\delta-\rho \geq 99/100$ with f. This is a contradiction for a sufficiently large constant $c > 1$, as a standard calculation shows that random function is likely to not have such agreement with circuits of size $2^{o(k)}$.

We remark that we can use a more careful argument to get a contradiction without requiring that $r \leq 2^{k/c}$. This is because a random function f is not likely to have a string of length $2^{o(k)}$ that describes a function C that has agreement $99/100$ with f. Note that if it exists, the reduction R can be used to describe any function by a string of length $\text{poly}(a, q, 1/\rho, n)$ and we obtain the same contradiction.

3.5 Proof of Lemma 3.1

The first step towards proving Lemma 3.1 is to define sets $B_1, \ldots, B_q$. We will do this by an iterative process which "further conditions" the probability space to smaller events.

Iterative further conditioning. We now describe an iterative process that defines a sequence of events $E_0, \ldots, E_q$ and sets $B_0, \ldots, B_q \subseteq \{0,1\}^n$. Let $E_0 = E$ and $B_0 = \emptyset$. Let $i \geq 0$ and assume that we already defined E_i, B_i (note that this holds for $i = 0$). Recall that $\bar{B}_i = \bigcup_{1 \leq j \leq i} B_j$ is the union of the sets we defined so far.

Note that for every $x \in \{0,1\}^k$, and $1 \leq j \leq i$ the definition of A_j^x and W_{j+1}^x only depend on the choice of sets $B_1, \ldots, B_j$. Thus, the random variables $A_1^x, \ldots, A_i^x$ and $W_1^x, \ldots, W_{i+1}^x$ are well defined at this point (even though we did not yet define the sets $B_{i+1}, \ldots, B_q$). We will maintain the following invariant throughout the iterative process.

- $|B_i| = O(\frac{aq^3}{\rho^2})$ (where the hidden constant does not depend on i).
- For every $1 \leq j < i$, $\Pr[\sum_{x \in \{0,1\}^k} A_j^x \leq \frac{\rho \cdot 2^k}{q} | E_i] = 1$. (Note that this holds vacuously for $i = 0$).
- There exist a fixed $v_i \in \{0,1\}^{\bar{B}_i}$ such that $E_i \subseteq \{V(\bar{B}_i) = v_i\}$. (Note that this vacuously holds for $i = 0$ as the event $\{V(\bar{B}_0) = v_0\}$ is the entire probability space).

- $\Pr[E_i | V(\bar{B}_i) = v_i] \geq 2^{-(a+1+i)}$. (Note that this holds for $i = 0$ as $\Pr[E_0] \geq 2^{-(a+1)}$).

We now show that for every $i \geq 0$ we can define an event $E_{i+1} \subseteq E_i$ and a set $B_{i+1} \subseteq \{0,1\}^n$ that maintain the invariant for $i+1$. By iteratively repeating this process we define events $E_0, \ldots, E_q$ and sets $B_0, \ldots, B_q$ that maintain the invariant for $i = q$ and these will be used to prove Lemma 3.1.

Obtaining the event E_{i+1} and set B_{i+1}. Let $L = \{0,1\}^n \setminus \bar{B}_i$ be the set of queries that we did not yet mark as "bad". Note that $V(L)$ has the same distribution as $(V(L) | V(\bar{B}_i) = v_i)$. (This is because $(V(y))_{y \in \{0,1\}^n}$ are independent). We apply Lemma 2.2 with the following choices: E_i plays the role of E, q is set to one, and $\eta = \rho/10q \geq \epsilon$ (where the inequality follows from the requirement on ρ in Theorem 1.5). Let B_{i+1} be the set obtained from Lemma 2.2. We have that

$$|B_{i+1}| = O((a+i+1)/\eta^2) = O((a+q)/\eta^2) = O(aq^3/\rho^2)$$

using the fact that $i \leq q$ and the definition of η. Thus, B_{i+1} indeed maintains the invariant. Moreover, for every $y \in L \setminus B_{i+1}$, $(V(y) | V(\bar{B}_i) = v_i)$ is η-close to $(V(y) | V(\bar{B}_i) = v_i) \wedge E_i)) = (V(y) | E_i))$ (where the equality follows as $E_i \subseteq \{V(\bar{B}_i) = v_i\}$).

Note that for every $x \in \{0,1\}^k$, W^x_{i+1} (which is already defined at this point) is fixed to some constant $y^x \in \{0,1\}^n$ in the event E_i. This is because $E_i \subseteq \{V(\bar{B}_i) = v_i\}$ which means that all answers to queries in $\bar{B}_i$ are fixed, and recall that the queries $W^x_1, \ldots, W^x_{i+1}$ of the canonical execution are completely determined by x and $V(\bar{B}_i)$. By Lemma 2.2, for every $y \in L \setminus B_{i+1}$ (which is equivalent to saying that $y \notin \bar{B}_{i+1}$) we have that:

$$\Pr[V(y) = 1 | E_i] \leq \Pr[V(y) = 1 | V(\bar{B}_i) = v_i] + \eta \leq 2\epsilon + \eta \leq 3\eta$$

where the inequality follows because $(V(y) | V(\bar{B}_i) = v_i)$ is distributed like $V(y)$ that has probability 2ϵ to be one. It follows that for every $x \in \{0,1\}^k$:

$$\mathbb{E}[A^x_{i+1} | E_i] = \Pr[A^x_{i+1} = 1 | E_i] = \Pr[V(W^x_{i+1}) = 1 \wedge W^x_{i+1} \notin \bar{B}_{i+1} | E_i]$$
$$= \Pr[V(y^x) = 1 \wedge y^x \notin B_{i+1} | E_i] \leq 3\eta.$$

Thus, by linearity of expectation we have that:

$$\mathbb{E}[\sum_{x \in \{0,1\}^k} A^x_{i+1} | E_i] \leq 3\eta \cdot 2^k.$$

and by Markov's inequality:

$$\Pr[\sum_{x \in \{0,1\}^k} A^x_{i+1} > 6\eta \cdot 2^k | E_i] < 1/2$$

We now define event E'_i as follows:

$$E'_i = E_i \cap \left\{ \sum_{x \in \{0,1\}^k} A^x_{i+1} \leq 6\eta \cdot 2^k \right\}$$

As $\eta = \rho/10q$ we have that $6\eta \leq \rho/q$. By the definition of E'_i we have obtained that

$$\Pr[\sum_{x \in \{0,1\}^k} A^x_{i+1} \leq \frac{\rho \cdot 2^k}{q} | E'_i] = 1$$

Our final event E_{i+1} will be a subset of E'_i and therefore the event above will hold with probability one conditioned on E_{i+1} as well. This means that we indeed maintain the requirement on A_{i+1} in the invariant. We have that that $\Pr[E'_i|E_i] \geq 1/2$ and therefore

$$\Pr[E'_i|V(\bar{B}_i) = v_i] \geq \Pr[E_i|V(\bar{B}_i) = v_i] \cdot \frac{1}{2} \geq 2^{-(a+1+i+1)} = 2^{-(a+1+(i+1))}.$$

By an averaging argument there exists $z \in \{0,1\}^{B_{i+1}}$ for which

$$\Pr[E'_i|V(\bar{B}_i) = v_i \wedge V(B_{i+1}) = z] \geq \Pr[E'_i|V(\bar{B}_i) = v_i] \geq 2^{-(a+1+(i+1))}.$$

Let v_{i+1} denote the pair (v_i, z), so that event $\{V(\bar{B}_i) = v_i \wedge V(B_{i+1}) = z\}$ is the event $\{V(\bar{B}_{i+1}) = v_{i+1}\}$. We define $E_{i+1} = E'_i \cap \{V(B_{i+1}) = z\}$ so that $E_{i+1} \subseteq \{V(\bar{B}_{i+1}) = v_{i+1}\}$ maintains the invariant. We also verify that

$$\Pr[E_{i+1}|V(\bar{B}_{i+1}) = v_{i+1}] = \Pr[E'_i|V(\bar{B}_{i+1}) = v_{i+1}] = \Pr[E'_i|V(\bar{B}_i) = v_i \wedge V(B_{i+1}) = z] \geq 2^{-(a+1+(i+1))}.$$

At this point we have defined event E_{i+1} and set B_{i+1} and we already showed that they maintain the invariant. This completes the description of the iterative process.

Finishing up. We are now ready to prove Lemma 3.1. Applying the iterative process above yields sets $B_1, \ldots, B_q$ and an event $E_q \subseteq E$ with positive probability for which the invariant above holds. We have that $|B| = q \cdot O(aq^3/\rho^2) = O(aq^4/\rho^2) = \text{poly}(a, q, 1/\rho)$ as required in Lemma 3.1. Let $V' : \{0,1\}^n \to \{0,1\}$ be some function such that $V' \in E_q \subseteq E$. We have that for every $1 \leq j \leq q$,

$$\sum_{x \in \{0,1\}^k} A_j^x[V'] \leq \frac{\rho \cdot 2^k}{q}.$$

It follows that:

$$\sum_{1 \leq j \leq q} \sum_{x \in \{0,1\}^k} A_j^x[V'] \leq \rho \cdot 2^k$$

Therefore, there are at most $\rho \cdot 2^k$ inputs $x \in \{0,1\}^k$ for which $\sum_{1 \leq j \leq q} A_j^x[V'] \neq 0$. We conclude that there are at least $(1-\rho) \cdot 2^k$ inputs $x \in \{0,1\}^k$ for which $\sum_{1 \leq j \leq q} A_j^x[V'] = 0$ meaning that these inputs are canonically almost silent. This concludes the proof of the lemma.

References

[AGGM06] A. Akavia, O. Goldreich, S. Goldwasser, and D. Moshkovitz. On basing one-way functions on np-hardness. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 701–710, 2006.

[Ats06] A. Atserias. Distinguishing sat from polynomial-size circuits, through black-box queries. In *IEEE Conference on Computational Complexity*, pages 88–95, 2006.

[BFNW93] L. Babai, L. Fortnow, N. Nisan, and A. Wigderson. Bpp has subexponential time simulations unless exptime has publishable proofs. *Computational Complexity*, 3:307–318, 1993.

[BS07] A. Bogdanov and M. Safra. Hardness amplification for errorless heuristics. In *48th Annual IEEE Symposium on Foundations of Computer Science*, pages 418–426, 2007.

[BT06] A. Bogdanov and L. Trevisan. On worst-case to average-case reductions for np problems. *SIAM J. Comput.*, 36(4):1119–1159, 2006.

[EIRS01] J. Edmonds, R. Impagliazzo, S. Rudich, and J. Sgall. Communication complexity towards lower bounds on circuit depth. *Computational Complexity*, 10(3):210–246, 2001.

[FF93] J. Feigenbaum and L. Fortnow. Random-self-reducibility of complete sets. *SIAM J. Comput.*, 22(5):994–1005, 1993.

[GG11] P. Gopalan and V. Guruswami. Hardness amplification within np against deterministic algorithms. *J. Comput. Syst. Sci.*, 77(1):107–121, 2011.

[GGH$^+$07] S. Goldwasser, D. Gutfreund, A. Healy, T. Kaufman, and G. N. Rothblum. Verifying and decoding in constant depth. In *39th Annual ACM Symposium on Theory of Computing (STOC)*, pages 440–449, 2007.

[GK08] V. Guruswami and V. Kabanets. Hardness amplification via space-efficient direct products. *Computational Complexity*, 17(4):475–500, 2008.

[GNW95] O. Goldreich, N. Nisan, and A. Wigderson. On Yao's XOR lemma. Technical Report TR95–050, *Electronic Colloquium on Computational Complexity*, March 1995. www.eccc.uni-trier.de/.

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

[GSTS07] D. Gutfreund, R. Shaltiel, and A. Ta-Shma. If np languages are hard on the worst-case, then it is easy to find their hard instances. *Computational Complexity*, 16(4):412–441, 2007.

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

[GV08] D. Gutfreund and S. P. Vadhan. Limitations of hardness vs. randomness under uniform reductions. In *APPROX-RANDOM*, pages 469–482, 2008.

[HVV06] A. Healy, S. P. Vadhan, and E. Viola. Using nondeterminism to amplify hardness. *SIAM J. Comput.*, 35(4):903–931, 2006.

[IJK09a] R. Impagliazzo, R. Jaiswal, and V. Kabanets. Approximate list-decoding of direct product codes and uniform hardness amplification. *SIAM J. Comput.*, 39(2):564–605, 2009.

[IJK09b] R. Impagliazzo, R. Jaiswal, and V. Kabanets. Chernoff-type direct product theorems. *J. Cryptology*, 22(1):75–92, 2009.

[IJKW10] R. Impagliazzo, R. Jaiswal, V. Kabanets, and A. Wigderson. Uniform direct product theorems: Simplified, optimized, and derandomized. *SIAM J. Comput.*, 39(4):1637–1665, 2010.

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

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

[IW98] R. Impagliazzo and A. Wigderson. Randomness vs. time: De-randomization under a uniform assumption. In *39th Annual Symposium on Foundations of Computer Science*. IEEE, 1998.

[KS03] A. Klivans and R. A. Servedio. Boosting and hard-core sets. *Machine Learning*, 53(3):217–238, 2003.

[Lev87] L. A. Levin. One-way functions and pseudorandom generators. *Combinatorica*, 7(4):357–363, 1987.

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

[LTW07] C.-J. Lu, S.-C. Tsai, and H.-L. Wu. On the complexity of hard-core set constructions. In *Automata, Languages and Programming, 34th International Colloquium*, pages 183–194, 2007.

[LTW08] C.-J. Lu, S.-C. Tsai, and H.-L. Wu. On the complexity of hardness amplification. *IEEE Transactions on Information Theory*, 54(10):4575–4586, 2008.

[O'D04] R. O'Donnell. Hardness amplification within np. *J. Comput. Syst. Sci.*, 69(1):68–94, 2004.

[Raz98] R. Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998.

[RTV04] O. Reingold, L. Trevisan, and S. P. Vadhan. Notions of reducibility between cryptographic primitives. In *TCC*, pages 1–20, 2004.

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

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

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

[Tre03] L. Trevisan. List-decoding using the xor lemma. In *44th Symposium on Foundations of Computer Science*, pages 126–135, 2003.

[Tre04] L. Trevisan. Some applications of coding theory in computational complexity. In *Complexity of computations and proofs*, volume 13 of *Quad. Mat.*, pages 347–424. Dept. Math., Seconda Univ. Napoli, Caserta, 2004.

[Tre05] L. Trevisan. On uniform amplification of hardness in np. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 31–38, 2005.

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

[Vio05a] E. Viola. The complexity of constructing pseudorandom generators from hard functions. *Computational Complexity*, 13(3-4):147–188, 2005.

[Vio05b] E. Viola. On constructing parallel pseudorandom generators from one-way functions. In *IEEE Conference on Computational Complexity*, pages 183–197, 2005.

[Wat10] T. Watson. Query complexity in errorless hardness amplification. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:126, 2010.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de