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 28, 2012

Abstract

Hardness amplification results show that for every Boolean function f there exists a Boolean 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 such 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 that may depend on both f and D in an arbitrary way. The well known connection between hardness amplification and list-decodable error-correcting codes implies that reductions showing hardness amplification cannot be uniform for $\delta, \epsilon < 1/4$. A reduction is *non-adaptive* if it makes non-adaptive queries to its oracle. Shaltiel and Viola (SICOMP 2010) 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 consider the case where the amplified function $Amp(f)$ is not Boolean. In this setting, the desired conclusion of hardness amplification is that every circuit of size s' computes $Amp(f)$ correctly on at most an ϵ fraction of inputs. Our results also apply in this setting, showing that size loss of $s' \leq \epsilon \cdot s$ is necessary for non-uniform reductions. This is in contrast to the results of Shaltiel and Viola which do not hold in this setting.

We also consider a setting where the hardness amplification result is not required to work for every function f. Instead, it is only required that $Amp(f)$ is hard for functions f in some class (e.g., functions in NP, functions with certain properties, or at the extreme, a fixed specific function). This allows the proof of correctness of the reduction R to exploit specific properties of the function f, and was used to provide hardness amplification results beating the coding theoretic lower bounds in some respects. Nevertheless, we show that even in this case, size loss of $s' \leq \epsilon \cdot s$ is necessary for non-uniform reductions.

*A preliminary version of this paper appeared in RANDOM 2011.

†This research was supported by BSF grants 2004329 and 2010120, ISF grants 686/07 and 864/11, and ERC starting grant 279559.

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 assumed to be p-hard for size s and show that some related 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).

Distinction between function-generic and function-specific amplification. 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* $f : \{0,1\}^k \to \{0,1\}$ 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 may be stated for specific functions f, g. This means that the proof of the hardness amplification result is allowed to use specific properties of these functions.

There is also a range "in between", where the hardness amplification result provides a map Amp (as in function-generic amplification) but the hardness amplification result does not apply to *every* function $f : \{0,1\}^k \to \{0,1\}$. Instead, it holds only for functions f from some specific family of functions [Lip91, IW01, TV07, Tre03, Tre05]. Such families may be complexity classes like NP,PSPACE or families capturing a certain property of functions like "random self reducibility".

1.2 A brief survey of typical settings of hardness amplification

We now briefly survey typical choices of parameters in hardness amplification results.

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 on a random input). Many such results are known [Lip91, BFNW93, IW97, IW01, STV01, TV07, Tre04, GGH$^+$07].

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 particular, the setup of worst-case to average-case above can be seen as a special case of this setup 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, GNW11, IW97, KS03, Tre03]. 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 $1/2$. For general ℓ, it is natural to set $p' = 1/2^\ell + \epsilon$. In the literature, one typically considers $\ell \gg \log(1/\epsilon)$ for which $1/2^\ell + \epsilon \approx \epsilon$ and it is therefore typical to 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, GNW11, 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, Wat11].

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 $s' \leq \epsilon \cdot s$. (In fact, for Boolean target function all known results have $s' \leq \epsilon^2 \cdot s$.)

This loss is a major disadvantage as it means that if one starts from a function that is hard for size s, existing results cannot achieve $\epsilon < 1/s$. It is natural to ask whether such a loss is necessary. In this paper we give a positive answer showing that a size loss of $s' \leq O(\epsilon \cdot s)$ is unavoidable for a large family of proof techniques.

1.3 Basic hardness amplification - a unified framework

We are interested in proving limitations on hardness amplification results. We want our limitations 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"). We define the notion of basic hardness amplification so that it is implied by all the settings mentioned above. This means that limitations on basic hardness amplification translate into limitations on all the settings above.

Basic hardness amplification. Let $f : \{0,1\}^k \to \{0,1\}$ and $g : \{0,1\}^n \to \{0,1\}^\ell$ be functions. Let $\epsilon, \delta > 0$ 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.

We now explain why basic hardness amplification is implied by the settings above. We start by comparing basic hardness amplification to mildly-average-case to average-case hardness amplification. Recall that the definition of the latter notion uses a different threshold p' depending on whether or not g is Boolean. If g is non-Boolean then $p' = \epsilon$ (as in basic hardness amplification) and so basic hardness amplification trivially follows from mildly-average-case to average-case amplification. If g is Boolean then $p' = 1/2 + \epsilon$ (which is different than basic hardness amplification in which $p' = \epsilon$). However, in basic hardness amplification we

consider *errorless* agreement, and it is easy to show that if there exists a circuit that has errorless agreement ϵ with g, then there exists a circuit (of essentially the same size) that has agreement $1/2 + \epsilon/2$ with g. This argument (that is explained in Section 1.4) shows that basic hardness amplification also follows in this case.

We now compare basic hardness amplification to errorless amplification. For this purpose we consider the contra-positive statements. The basic hardness amplification task is to show that if there exists a circuit D that has *errorless* agreement $p' = \epsilon$ with g, then there exists a circuit C that has agreement $p = 1 - \delta$ with f. However, in contrast to errorless amplification, it is not required that the agreement of C is errorless. Therefore, basic hardness amplification is implied by (and in fact weaker than) errorless amplification.

1.4 Non-uniform reductions for hardness amplification

Our goal is to show that certain proof techniques showing hardness amplification cannot avoid size loss (and can only give $s' \leq \epsilon \cdot s$). We will focus on the setting of basic hardness amplification (and will later argue that this implies that size loss cannot be avoided in other settings as well).

The proof techniques that we study are "non-uniform reductions". As we explain later in Section 1.5, this notion captures the proofs of almost all hardness amplification results in the literature. We start with a precise definition of non-uniform reductions.

Definition 1.2 (non-uniform reduction)**.** *Let $f : \{0,1\}^k \to \{0,1\}$, $g : \{0,1\}^n \to \{0,1\}^\ell$ be functions. Let ϵ, δ and be parameters.*

- *A* semi-uniform reduction showing basic hardness amplification *(for f, g, ϵ, δ) is an oracle circuit $R^{(\cdot)}$ that receives an input $x \in \{0,1\}^k$. It is required that for every function $D : \{0,1\}^n \to \{0,1\}^\ell \cup \{\perp\}$ that has errorless agreement ϵ with g, the function $C(x) = R^D(x)$ has agreement $1 - \delta$ with f.*

- *Let a be an integer. 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 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.*

In particular, a semi-uniform reduction is a special case of non-uniform reductions in which the "advice length" a is zero. 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 $x \in \{0,1\}^k$ and $\alpha \in \{0,1\}^a$, the 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.

Non-uniform reductions give hardness amplification. 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 there exists $\alpha \in \{0,1\}^a$ such that the function $C(x) = R^D(x, \alpha)$ has agreement $1 - \delta$ with f. 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. Summing up, we indeed get a basic hardness amplification result for f, g, ϵ, δ.

The number of queries governs the size loss. By the discussion above, the number of queries q made by the reduction is the dominant factor in the ratio between s and s'. More precisely, 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 yields that the size loss is $s' = O(s \cdot \epsilon)$.

What is non-uniform in this reduction? Let us first consider semi-uniform reductions. Such a reduction R is an oracle circuit, and is allowed to be hardwired with non-uniform advice that may depend on the choice of f, g. A semi-uniform reduction is "black-box" in the sense that it receives black-box access to its oracle.

Our main focus in this paper is on general non-uniform reductions. A general non-uniform reduction also receives a second input $\alpha \in \{0,1\}^a$. The order of quantifiers in Definition 1.2 allows α to *depend* on the function D supplied to R as an oracle. There is no requirement that α can be efficiently computed using black-box access to D (and this is why we refrain from using the term "black-box" when considering non-uniform reductions).

Extending the notion of non-uniform reductions to 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 settings surveyed in Section 1.2. We will show that non-uniform reductions showing basic hardness amplification must suffer from size loss of $s' = O(\epsilon \cdot s)$, and as we now explain, this implies that non-uniform reductions for other settings also suffer from the same size loss.

More precisely, we define the notion of "non-uniform reduction showing mildly-average-case to average-case hardness amplification" by replacing the requirement that "D has errorless agreement ϵ with g" in Definition 1.2 with the requirement that "D has agreement p with g" where $p = 1/2 + \epsilon$ in case $\ell = 1$ and $p = \epsilon$ in case $\ell > 1$. It is easy to observe that such reductions can indeed be used to produce mildly-average-case to average-case hardness amplification results (by the same logic used for basic hardness amplification). Moreover, as in the case of basic hardness amplification, a non-uniform reduction showing mildly-average-case to average-case hardness amplification that makes $q = \Omega(1/\epsilon)$ queries gives a mildly-average-case to average-case hardness amplification result in which $s' = O(\epsilon \cdot s)$.

We also observe that a non-uniform reduction showing mildly-average-case to average-case hardness amplification can be easily transformed into a non-uniform reduction showing basic hardness amplification with essentially the same parameters.[1] Consequently, 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

[1]For completeness, we now present this simple argument. Let $f : \{0,1\}^k \to \{0,1\}$ and $g : \{0,1\}^n \to \{0,1\}^\ell$ be some functions and let $R^{(\cdot)}$ be a reduction showing mildly-average-case to average-case hardness amplification for f, g and some parameters ϵ, δ and a using q queries. We will construct a non-uniform reduction R' showing basic hardness amplification with related parameters. Recall that the threshold p' used in the definition of mildly-average-case to average-case hardness amplification is set differently depending on whether or not g is Boolean. If $\ell > 1$ (meaning that g is non-Boolean) then $p' = \epsilon$ in both settings and we can simply use R as R'. If $\ell = 1$ (meaning that g is Boolean) we will construct a non-uniform reduction R' showing basic hardness amplification for the related parameters $2\epsilon, \delta, a+1$, and using the same number of queries as R. The issue that we need to address is that the reduction R' expects to receive oracle access to a function D' that has errorless agreement 2ϵ with g, while we are given a reduction R that expects to receive oracle access to a function D that has agreement $1/2 + \epsilon$ with g. We construct R' as follows: We interpret an advice string $\alpha' \in \{0,1\}^{a+1}$ for R' as a pair (α, b) where $\alpha \in \{0,1\}^a$ and $b \in \{0,1\}$. Let $D' : \{0,1\}^n \to \{0,1\}^\ell \cup \{\perp\}$ be some function that has errorless agreement 2ϵ with g. Upon receiving inputs x, α' and oracle access to D', the reduction R' uses oracle access to D' in order to emulate oracle access to a function $D : \{0,1\}^n \to \{0,1\}^\ell$ defined as follows: $D(y)$ is set to $D'(y)$ if $D'(y) \neq \perp$, and to b otherwise. Note that for every function D' that has errorless agreement 2ϵ with g, there exists a $b \in \{0,1\}$ for which D has agreement $1/2 + \epsilon$ with g. On oracle D', and inputs $x, (\alpha, b)$, R' simulates $R^D(x, \alpha)$ (and note that each query to D can be simulated by one query to D'). It is guaranteed that there exists an $\alpha \in \{0,1\}^a$ for which $h(x) = R^D(x, \alpha)$ has agreement $1 - \delta$ with f, and therefore, there exists an advice string $(\alpha, b) \in \{0,1\}^{a+1}$ on which

the settings described in Section 1.2, and shows that hardness amplification results (in any setting) that are proven by non-uniform reductions must have $s' = O(\epsilon \cdot s)$. This allows us to focus on non-uniform reductions for basic hardness amplification in the remainder of the paper.

Reductions showing function-generic hardness amplification. Definition 1.2 considers *specific* functions f, g. Most of the hardness amplification results in the literature are *function-generic* in the sense explained in Section 1.1. In the definition below we extend the notion of non-uniform reduction to the function-generic case.

Definition 1.3 (Reductions showing 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.*

In Definition 1.3 we require that there exists a single non-uniform reduction R that is good for every function f. Note however, that by the definition, when applied for a function f and an oracle D, the non-uniform reduction R receives an advice string α that may depend on both f and D.[2]

We can use Definition 1.3 to also define the analogous function-generic notion for mildly-average-case to average-case hardness amplification. We remark that 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 (see e.g., [STV01]) that function-generic hardness amplification is equivalent to certain variants of list-decodable error-correcting codes. Loosely speaking, the map Amp can be seen as "encoding" the (truth table of) f into the (truth table of) $Amp(f)$, and the non-uniform reduction R can be seen as a "list-decoding" procedure in the following sense: Given a (truth table) D that has sufficiently high agreement with $Amp(f)$, R produces a list of 2^a candidates where one of them has agreement $1 - \delta$ with f (where the candidates are $R^D(\cdot, \alpha)$ for $\alpha \in \{0,1\}^a$). In particular, semi-uniform reductions (in which $a = 0$) correspond to the special case of "unique-decoding". We elaborate on this connection in Section 3.

Why should we consider 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 [GNW11]) that we explain later on. This argument, however, does not apply to general non-uniform reductions. The main contribution of this paper is extending Rudich's argument so that it applies to *non-uniform reductions* even if they are allowed to be *adaptive*.

We stress that semi-uniform reductions are rare exceptions in the literature on hardness amplification. In fact, most of the reductions given in the literature are non-uniform and quite a few of them use "large values" of non-uniformity a. Consequently, if we want to show limitations of "existing proof techniques" we need to allow reductions to be non-uniform. Moreover, as mentioned above, reductions showing function-generic hardness amplification are equivalent to certain list-decodable error-correcting codes. This connection rules

$(R')^{D'}(x, (\alpha, b))$ has agreement $1 - \delta$ with f.

[2] To emphasize this point we remark that an alternative approach to define a function-generic reduction is to allow a different reduction R_f for every function f. This seemingly stronger notion of a reduction is captured by Definition 1.3 in the following sense: If we slightly increase the "advice length" a, we can capture the seemingly stronger notion by Definition 1.3. This is because when applied for a function f and an oracle D, the non-uniform reduction R can also expect to receive the specific circuit R_f as an additional advice.

out the existence of semi-uniform reductions in many hardness amplification settings. Consequently, the use of non-uniform reductions rather than semi-uniform reductions (or in coding terminology, of "list-decoding" rather than "unique decoding") is inevitable in many settings (see e.g., [TV07] and the discussion in Section 3).

1.5 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, the reductions in these settings cannot be semi-uniform in the "list-decoding regime" (that is for $\delta, \epsilon < 1/4$). Theorem 1.4 below asserts 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 $a, \frac{1}{\epsilon}, n, r \leq 2^{k/c}$ and $\delta \leq 1/3$. Let (Amp, R) be a function-generic reduction showing basic hardness amplification (for ϵ, δ, ℓ and a) and assume that R is of size r. Then, R makes at least $\frac{1}{100\epsilon}$ queries.*

We have stated Theorem 1.4 in a general form with many parameters. In typical hardness amplification results the parameter setting is $n = \text{poly}(k)$, $\epsilon = 1/k^b$ for some constant b (or sometimes slightly super constant b), $\delta \leq 1/3$, and $r, a = \text{poly}(k)$. Note that Theorem 1.4 holds for these choices. (In fact, the theorem holds even when $\text{poly}(k)$ is replaced by $2^{k/c}$ for some universal constant $c > 1$). We remark that the constant $1/3$ in Theorem 1.4 can be replaced by any constant smaller than $1/2$.

The requirements that $a, n, r \leq 2^{k/c}$ are natural in the sense that reductions in which one of these parameters is larger than 2^k are not useful to prove hardness amplification results. This is because that such reductions produce circuits of size larger than 2^k for the source function f (and such circuits exist trivially without needing to rely on the reduction). Nevertheless, we mention that the requirement that $r \leq 2^{k/c}$ in Theorem 1.4 is unnecessary and the theorem can be proven without it. In the formal section, we prove the theorem with the requirement, and later sketch the modifications needed in order to remove it.

Tightness of Theorem 1.4. 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 [GNW11, IJKW10, Wat11]. (In fact, some of these reductions are in a stronger setting, 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 believe that this is the right bound, but are unable to match it. We remark that it is possible to improve the bound in Theorem 1.4 to the "right bound" of $\Omega(\frac{\log(1/\delta)}{\epsilon})$ (which is tight for every δ) in the special case where the reduction is *non-adaptive*.

Comparison of Theorem 1.4 to [SV10]. Theorem 1.4 is stated for *basic hardness amplification*. Nevertheless, 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 a larger number of $O(\frac{\log(1/\delta)}{\epsilon^2})$ queries. The problem of giving a matching lower bound was considered in [SV10] and a matching lower bound is

shown for the special case of *non-adaptive* reductions. Using our terminology, the result of [SV10] is a bound of $\Omega(\frac{\log(1/\delta)}{\epsilon^2})$ on the number of queries of non-adaptive, non-uniform, function-generic reductions showing Boolean mildly-average-case to average-case amplification. This result is incomparable to Theorem 1.4. On the one hand it gives a stronger quantitative bound on the number of queries. On the other, it only handles *non-adaptive reductions*, and it only applies in the setting of *Boolean* mildly-average-case to average-case amplification. We stress once again that in the setting of *non-Boolean* mildly-average-case to average-case amplification, there exist non-uniform and non-adaptive reductions making only $O(\frac{\log(1/\delta)}{\epsilon})$ queries [Imp95, KS03].

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 [Wat11]) showing our aforementioned lower bounds for the special case where the reduction is *non-adaptive*.

Function-specific hardness amplification. Function-generic non-uniform reductions (as in Definition 1.3) need to handle every possible function $f : \{0,1\}^k \to \{0,1\}$. In contrast, non-uniform reductions for specific functions f, g (as defined in Definition 1.2) are only required to work for that specific pair f, g. 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 the reduction R can ignore 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 showing basic hardness amplification 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 $r + \text{poly}(a, q, n, 1/\rho)$ that has agreement $1 - \delta - \rho$ with f.*

Theorem 1.5 implies that if $q = o(1/\epsilon)$ (and setting $\rho = o(1)$) then the mere existence of the reduction R implies the existence of a small 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 assumption does not hold.

Note that Theorem 1.5 is only interesting when $q \leq 1/10\epsilon$ as otherwise the requirement that $\rho \geq 10\epsilon q$ implies that $\rho \geq 1$ and the conclusion of the theorem is meaningless.

We will later prove that Theorem 1.4 follows from Theorem 1.5. This will be done by applying Theorem 1.5 on a random function f. Loosely speaking, a random function is unlikely to have a small circuit that agrees with it on significantly more than half the inputs, and therefore, the existence of a function-generic reduction making $q = o(1/\epsilon)$ queries gives a contradiction, as by Theorem 1.5 it implies a small circuit that has high agreement with f. The precise details appear in Section 2.

Function-specific hardness amplification in the literature. Function-specific hardness amplification results are less common than function-generic results. One motivation for studying 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 [IW01, TV07, Tre03, Tre05] in which the reduction and the proof of its correctness relies on the fact that f has certain properties. Another example is in Cryptography where protocols are often constructed assuming the hardness of some *specific* function (e.g., factoring or discrete log) and properties of this function are used to improve either security or efficiency. Theorem 1.5 shows that in these settings, non-uniform reductions must make $\Omega(1/\epsilon)$ queries.

Reductions that are not captured by Definition 1.2. 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. Recall that there is no requirement that α is generated using black-box access to D. 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.

1.6 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 Boolean 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 $a = O(\log(1/\epsilon))$ which is much smaller than k in typical settings). 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]. Therefore, to attack the problem above we need techniques that distinguish between basic hardness amplification and Boolean mildly-average-case to average-case amplification.

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 the complexity of 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, LTW11].

1.7 Organization of this paper

In Section 2 we prove Theorems 1.4 and 1.5. In Section 3 we elaborate on the relationship between hardness amplification and error correcting codes and point out that our results translate into lower bounds on the query complexity of local decoders for list-decodable codes.

2 Proof of main theorems

In this section we prove Theorem 1.4 and Theorem 1.5. We start by proving Theorem 1.5 (and later show that Theorem 1.4 follows from Theorem 1.5). Let us start by recalling the setup of Theorem 1.5.

The setup of Theorem 1.5. 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 $\rho \geq 10\epsilon q$ be the parameter from the theorem statement. We can assume without loss of generality that $\rho \leq 1$ and therefore that $q \leq 1/10\epsilon$. We will use these choices in the remainder of Section 2. 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 difference of our overall approach from that of [SV10]. We stress that while our technique below relies on some of the ideas developed in [SV10], our overall approach is very different. The approach of [SV10] (which consider non-adaptive function-generic reductions showing mildly-average-case to average-case amplification) is to show that the existence of a "too good" function-generic reduction implies a "too good" statistical test that can distinguish between q independent fair coins and q independent biased coins. In contrast, our approach is to show that the existence of a "too good" function-specific reduction yields small circuits for the function f. We do not attempt to mimic the approach of [SV10], as it seems difficult to extend it to adaptive reductions.

A probability distribution over oracles. Let $v : \{0,1\}^n \to \{0,1\}$ be a function. We use v to define a function $D(v)$ where $D(v) : \{0,1\}^n \to \{0,1\}^\ell \cup \{\perp\}$ is defined as follows.

$$D(v)(y) = \begin{cases} \perp & v(y) = 0 \\ g(y) & v(y) = 1 \end{cases}$$

Throughout the proof we use the following probability space: The probability space consists of independent identically distributed random variables $V = (V(y))_{y \in \{0,1\}^n}$ 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 $V : \{0,1\}^n \to \{0,1\}$ and define the random variable $D = D(V)$. We will use this probability space throughout this section and all expressions involving probability or expectation refer to this space.

Rudich's argument and the case of semi-uniform reductions. We now point out that it is easy to prove Theorem 1.5 in case the reduction R is semi-uniform. The argument below is a folklore argument attributed to Steven Rudich in [GNW11]. Recall that R makes $q \leq 1/10\epsilon$ queries. It follows that on every input

$x \in \{0,1\}^k$, the reduction R is unlikely to receive an answer that is different from '$\perp$' from the oracle $D = D(V)$. It follows that the expected number of inputs x on which R gets an answer different from '$\perp$' from D is small. By the probabilistic method, there exists a fixed function $v : \{0,1\}^n \to \{0,1\}$ such that for almost all inputs $x \in \{0,1\}^k$, $R^{D(v)}(x)$ sees only '$\perp$' when querying the oracle. On every such input, the oracle is not helpful and so it is easy to simulate $R^{D(v)}(x)$ by a small circuit $C(x)$ (with no oracle) on all such inputs. We obtained a circuit $C(x)$ that has high agreement with $R^{D(v)}(x)$, and therefore high agreement with f. This yields the theorem.

However, we want to consider reductions that are not necessarily semi-uniform and such reductions are given an advice string α. This advice string may depend on the oracle D and it can allow R to ask queries that are answered by a value different from '$\perp$'. For example, the advice string may include a $y \in \{0,1\}^n$ on which $D(y) \neq \perp$. While this does not seem to help R in getting large agreement with f, the argument of Rudich no longer applies. Our high level approach is to try and extend Rudich's argument so that we can handle general non-uniform and adaptive reductions.

Existence of a good advice string. Our first step is to try and in some sense "fix" the advice string of the reduction so that it "becomes semi-uniform".

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

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

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

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

Note that $\Pr[E] \geq 2^{-a} - 2^{-\Omega(2^k)} \geq 2^{-(a+1)}$. We view the event E as a subset of all functions $v : \{0,1\}^n \to \{0,1\}$ and note that we have that for every $v \in E$, the function $h(x) = R^{D(v)}(x, \alpha')$ has agreement $1 - \delta$ with f. Loosely speaking, we have that conditioned on the event $\{V \in E\}$, R is essentially semi-uniform (in the sense that the advice string α' does not depend on D).

An information theoretic lemma. Following the previous discussion, we want to understand how are the random variables $(V(y))_{y \in \{0,1\}^n}$ distributed when conditioned on the event $\{V \in E\}$. For this purpose we will use the following lemma by [Raz98].

Lemma 2.1. *[Raz98] Let $Z = (Z_1, \ldots, Z_N)$ be independent random variables over some finite set S. Let a and η be parameters and let $E \subseteq S^N$ be a set such that $\Pr[Z \in E] \geq 2^{-a}$. There exists a set $B \subseteq \{1, \ldots, N\}$ of size $O(a/\eta^2)$ such that for every $y \in \{1, \ldots, N\} \setminus B$, the two distributions Z_y and $(Z_y | Z \in E)$ are η-close.*[3]

In our setup, the lemma says that if we pick $\eta = \epsilon$ and apply the lemma on the random variables $(V(y))_{y \in \{0,1\}^n}$, then there exists a small set $B \subseteq \{0,1\}^n$ of "bad queries" such that for every $y \notin B$, $\Pr[V(y) = 1 | V \in E] \leq \Pr[V(y) = 1] + \eta \leq 3\epsilon$. This means that even conditioned on the event $\{V \in E\}$, except for few bad queries $y \in B$, the oracle D is likely to answer a good query $y \notin B$ by '$\perp$'. Loosely speaking, we can now hope to bound the "amount of information" that R gets from its oracle D by arguing that on "good queries", R is not likely to get meaningful information, and there are only few bad queries.

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

Remark 2.2 (Difficulty of handling adaptive reductions)**.** *This rationale is indeed suitable for lower bounds on non-adaptive reductions. In such reductions, every input $x \in \{0,1\}^k$, defines queries $y_1^x, \ldots, y_q^x \in \{0,1\}^n$ that are asked by the reduction $R^D(x, \alpha')$ (and these queries are constants that do not depend on D). Therefore, we can indeed use the lemma and a union bound to show that for every $x \in \{0,1\}^k$,*

$$\Pr[\exists i : y_i^x \notin B \text{ and } V(y_i^x) = 1 | V \in E] \leq \sum_{1 \leq i \leq q} \Pr[y_i^x \notin B \text{ and } V(y_i^x) = 1 | V \in E] \leq q \cdot 3\epsilon \leq \rho.$$

This turns out to be sufficient to handle non-adaptive reductions. Loosely speaking, this is because conditioned on $\{V \in E\}$, we expect that on a $1-\rho$ fraction of inputs $x \in \{0,1\}^k$, all queries $y \notin B$ are answered by '$\perp$'. Note that on any such input x we can simulate $R^D(x, \alpha')$ by a small circuit $C(x)$ (with no oracle) if we hardwire C with α', $(D(y))_{y \in B}$ and B. We explain this argument more precisely in Section 2.4.

Jumping ahead we mention that this rationale is problematic in case the reduction R is adaptive. This is because an adaptive reduction may first query the oracle D on bad queries $y \in B$ and use the answers on bad queries (and in particular whether or not bad queries answer by '$\perp$') to gain additional information on the random variable $(V|V \in E)$. This additional information may allow the adaptive reduction R to ask queries $y \notin B$ on which the probability that $V(y) = 1$ (and therefore $D(y) \neq \perp$) is large. This means that we can't hope to claim that the reduction R does not gain information from asking queries that are not in B. We give an example of such a reduction in Section 2.4.

2.1 A canonical execution of reductions

We now proceed with the proof. Let us start with some notation. Let $v : \{0,1\}^n \to \{0,1\}$ be some function. For every $x \in \{0,1\}^k$ and $1 \leq i \leq q$ we define $Q_i^x(v) \in \{0,1\}^n$ to be the i'th query asked by $R^{D(v)}(x, \alpha')$. Note that as R is allowed to be adaptive, these queries may depend on $D(v)$ and therefore on v. This means that in our probability space, $Q_1^x(V), \ldots, Q_q^x(V)$ are random variables that may depend on V.

We consider a mental experiment which we call "the canonical execution" of $R^{D(v)}(x, \alpha')$. Loosely speaking, in the mental experiment we simulate the "real execution" $R^{D(v)}(x, \alpha')$ while sometimes replacing the answers of the oracle $D(v)$ by '$\perp$'. Analyzing the mental experiment will be helpful in understanding the real execution. We now give the precise definition.

Definition of the canonical execution. Let $v : \{0,1\}^n \to \{0,1\}$ be some function. 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 $\leq i$.

In the canonical execution, when R asks the i'th query to its oracle, we answer it according to the following *canonical rule*: If the query is in $\bar{B}_i$ then we answer it the same way $D(v)$ answers it. However, if the query is not in $\bar{B}_i$ we answer it by '$\perp$' regardless of the answer of $D(v)$.

More precisely, given sets $B_1, \ldots, B_q$ and for every $x \in \{0,1\}^k$, we define a sequence of queries $W_1^x(v), \ldots, W_q^x(v) \in \{0,1\}^n$ as follows: We simulate $R^{D(v)}(x, \alpha')$ until it asks its first query $Q_1^x(v)$ and set $W_1^x(v) = Q_1^x(v)$. We answer this query by the canonical rule above and give the answer back to R. Having received the answer, R computes its next query which we denote by $W_2^x(v)$. (Note that $W_2^x(v)$ may be different than $Q_2^x(v)$ in case R is adaptive). We answer this question by the canonical rule above, and continue this process to define $W_1^x(v), \ldots, W_q^x(v)$ as well as the final output of the reduction R at the end of the execution (which we denote by $R_{\mathsf{C}}^{D(v)}(x, \alpha')$).

Loosely speaking, the canonical execution describes an idealized situation in which queries that are considered good, are answered by '$\perp$'. Our proof will try to relate the real execution and the canonical execution. We first make a few observations:

- As we already pointed out, if R is adaptive, the canonical queries may be different than the real queries.
- In the canonical execution the same query $y \in \{0,1\}^n$ may be answered differently at different steps. For example if $y \notin \bar{B}_i$ and $y \in B_{i+1}$ then the query y is answered by '$\perp$' at step i, and by $D(v)(y)$ (which can be different from '$\perp$') at step $i+1$.
- The definition of $W^x_{i+1}(v)$ depends only on the sets $B_1, \ldots, B_i$ (and does not depend on the choice of sets $B_{i+1}, \ldots, B_q$). This allows us to use $W^x_1(v), \ldots, W^x_{i+1}(v)$ when defining the set B_{i+1}.

2.2 Roadmap of the proof

Recall that our goal is to show that there exists a small circuit $C(x)$ that has agreement $1-\delta-\rho$ with f. We have that for every $v \in E$, the "real execution" $h(x) = R^{D(v)}(x, \alpha')$ has agreement $1-\delta$ with f. Thus, to complete the proof, it is sufficient to show that there exists $v \in E$ for which there exists a small circuit $C(x)$ (that may depend on v and α') such that for $(1-\rho)\cdot 2^k$ inputs $x \in \{0,1\}^k$, $C(x) = R^{D(v)}(x, \alpha')$. Summing up, we want a construct a small circuit that simulates the real execution on almost all inputs.

We first observe that it is easy to simulate the canonical execution by a circuit $C(x)$ that is small in case the sets $B_1, \ldots, B_q$ are small. Loosely speaking, this is because to simulate the canonical execution we do not need to know the answers to queries that are good. The precise statement appears in the next lemma.

Lemma 2.3. *Let $v : \{0,1\}^n \to \{0,1\}$ be some function such that $v \in E$ and let $B_1, \ldots, B_q$ be sets of size at most b that are used to define the canonical execution. There exists a circuit $C(x)$ such that for every $x \in \{0,1\}^k$, $C(x) = R_{\mathbf{C}}^{D(v)}(x, \alpha')$ and furthermore the size of C is at most $r + a + poly(n, q, b)$.*

Proof. The circuit C that we construct will use the circuit R (which is of size r) and will be hardwired with:

- The string α' (which is of length a).
- The sets $B_1, \ldots, B_q$ (which can be encoded using qnb bits).
- The values $(D(v)(y))_{y \in \bar{B}_q}$ (which can be encoded using $O(qb)$ bits).

On input x, the circuit C will simulate the canonical execution of $R_{\mathbf{C}}^{D(v)}(x, \alpha')$. Note that to answer the i'th query $W^x_i(v)$ of the canonical execution it is sufficient to know whether $W^x_i(v) \in \bar{B}_i$, and the value of $D(v)$ on $\bar{B}_i$. Thus, there exists a circuit of size $r + \text{poly}(n, q, b)$ which performs this simulation. □

Recall however, that our goal is to simulate the real execution and not the canonical execution. For this purpose we want to relate the two executions and make the following definition. Let $v : \{0,1\}^n \to \{0,1\}$ be some function. For $x \in \{0,1\}^k$ and $1 \le i \le q$ we define:

$$A^x_i(v) = \begin{cases} 1 & v(W^x_i(v)) = 1 \text{ and } W^x_i(v) \notin \bar{B}_i \\ 0 & \text{otherwise.} \end{cases}$$

In words, $A^x_i(v)$ is one iff the i'th query asked in the canonical execution is good, and yet $D(v)$ does not answer it by '$\perp$'. The reason we are interested in this notion is the following observation. Note that if $A^x_1(v) = 0$, then the answer given by the canonical rule on the first query coincides with the real answer and in particular $W^x_2(v) = Q^x_2(v)$. This motivates the following definition.

Definition of canonically silent inputs. Let $v : \{0,1\}^n \to \{0,1\}$. We say that $x \in \{0,1\}^k$ is *canonically silent* for v if

$$\sum_{1 \le i \le q} A_i^x(v) = 0$$

We now observe that on a canonically silent input, the real execution coincides with the canonical execution.

Lemma 2.4. *Let $v : \{0,1\}^n \to \{0,1\}$ be some function such that $v \in E$ and let $B_1, \ldots, B_q$ be sets that are used to define the canonical execution. If x is canonically silent for v then*

$$R_{\mathcal{C}}^{D(v)}(x, \alpha') = R^{D(v)}(x, \alpha').$$

Proof. Let $x \in \{0,1\}^k$ be canonically silent for v. We will show that for every $1 \le i \le q$, $Q_i^x(v) = W_i^x(v)$ and that each such query is answered by the same answer in the two executions. We will prove this by induction on i. The base case is $i = 1$ and 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 first 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 $D(v)(W_1^x(v)) = \perp$. It follows that in both cases the answers coincide.[4] Therefore, the next query is the same in both executions and we have that $Q_2^x(v) = W_2^x(v)$.

The same reasoning can be used to show that if the two executions coincide in the first i steps, then the fact that $A_{i+1}^x(v) = 0$ implies that they continue to coincide in step $i+1$. This means that the two executions coincide until the end and in particular that the output of R is identical in the two executions. □

By the previous discussion, Theorem 1.5 will follow if we can find a function $v \in E$ and small sets $B_1, \ldots, B_q$ on which the number of canonically silent inputs is at least $(1-\rho) \cdot 2^k$. This is the main technical part of the proof and is summarized in the next lemma.

Lemma 2.5. *There exists $v : \{0,1\}^n \to \{0,1\}$ such that $v \in E$ and there exist sets $B_1, \ldots, B_q \subseteq \{0,1\}^n$ such that*

- *For every $1 \le i \le q$, $|B_i| = O(\frac{aq^3}{\rho^2}) = \text{poly}(a, q, 1/\rho)$.*
- *The number of inputs $x \in \{0,1\}^k$ that are canonically silent for v is at least $(1-\rho) \cdot 2^k$.*

In Section 2.3 we formally verify that Theorems 1.4 and 1.5 follow from Lemma 2.5. The remainder of this section is devoted to proving Lemma 2.5. The proof is by the probabilistic method. We will show that with positive probability over choosing V, we obtain a function with the required properties. It is instructive to first consider the easier special case in which R is non-adaptive. We give this argument in Section 2.4. The reader may also skip directly to the proof of the general case given in Section 2.5.

[4] A subtle point is that it may be the case that $W_1^x(v)$ is not in $\bar{B}_1$ but is in B_2. This happens if this query is considered good at step 1 and bad at step 2. Nevertheless, the fact that $A_1^x(v) = 0$ implies that on this query $D(v)$ answers '$\perp$' regardless of whether it becomes bad later on.

2.3 Proof that Theorems 1.4 and 1.5 follow from Lemma 2.5

Proof of Theorem 1.5. By Lemma 2.5 there exist $v \in E$ and sets $B_1, \ldots, B_q$ such that for every $1 \leq i \leq q$, $|B_i| \leq b$ for $b = O(\frac{aq^3}{\rho^2}) = \text{poly}(a, q, 1/\rho)$. As $v \in E$, we have that $h(x) = R^{D(v)}(x, \alpha')$ has agreement $1 - \delta$ with f. By Lemma 2.5 the number of inputs $x \in \{0,1\}^k$ that are canonically silent for v is at least $(1 - \rho) \cdot 2^k$. Therefore, by Lemma 2.4 the canonical execution $h_{\mathbf{C}}(x) = R_{\mathbf{C}}^{D(v)}(x, \alpha')$ has agreement $1 - \rho$ with the real execution $h(x)$. By Lemma 2.3 there exists a circuit $C(x)$ of size $r + a + \text{poly}(n, b, q) = r + \text{poly}(a, n, q, 1/\rho)$ that simulates the canonical execution $h_{\mathbf{C}}(x)$. It follows that C has agreement $1 - \rho$ with the real execution $h(x)$ and therefore C has agreement $1 - \delta - \rho$ with f as required.

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 $a, \frac{1}{\epsilon}, n, r \leq 2^{k/c}$ for a constant $c > 1$ that we determine later and let $\delta \leq 1/3$. Let (Amp, R) be a function-generic reduction showing basic hardness amplification (for ϵ, δ, ℓ 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 $1 - \delta - \rho \geq 1 - 1/3 - 1/10$ with f (and note that the agreement is a constant that is strictly larger than $1/2$). This is a contradiction for a sufficiently large constant $c > 1$, as a standard calculation shows that a random function is likely to not have such agreement with circuits of size $2^{o(k)}$.

We remark that any function f that cannot be approximated by circuits of size $2^{o(k)}$ can be used to show the failure of the function-generic reduction. In particular, the "counterexample function" f need not depend on the choice of Amp or R although the statement of Theorem 1.4 allows it to.

We also 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 significantly larger than $1/2$ with f. Note that if it exists, a 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.

2.4 Proof of Lemma 2.5 in the special case that R is non-adaptive

Consider the case that R is non-adaptive. This means that the queries asked on an input x and advice string α' do not depend on the oracle. More formally, for every function $v : \{0,1\}^n \to \{0,1\}$ such that $v \in E$, every $x \in \{0,1\}^k$ and every $1 \leq i \leq q$, there exists $y_i^x \in \{0,1\}^n$ (that does not depend on v) such that $Q_i^x(v) = W_i^x(v) = y_i^x$. We apply Lemma 2.1 on the independent random variables $(V(y))_{y \in \{0,1\}^n}$, the event E, and setting $\eta = \rho/10q \geq \epsilon$. We indeed have that $\Pr[V \in E] \geq 2^{-(a+1)}$ and therefore by the lemma there exists a set $B \subseteq \{0,1\}^n$ of size at most $O(a/\eta^2) = poly(a, q, 1/\rho)$ such that for $y \notin B$, $(V(y)|V \in E)$ is η-close to $V(y)$. In particular, for $y \notin B$,

$$\Pr[V(y) = 1 | V \in E] \leq \Pr[V(y) = 1] + \eta \leq 2\epsilon + \eta \leq 3\eta \leq \rho/q.$$

We set $B_1 = B_2 = \ldots = B_q = B$. (This means that in case R is non-adaptive we can simplify the definition of the canonical execution and do not need to distinguish between bad queries at different steps.) Having defined the sets $B_1, \ldots, B_q$ the canonical execution is now completely defined.

For every $x \in \{0,1\}^k$, we define $S^x(v)$ as follows:

$$S^x(v) = \begin{cases} 1 & x \text{ is canonically silent for } v \\ 0 & \text{otherwise.} \end{cases}$$

We also define $S(v) = \sum_{x \in \{0,1\}^k} S^x(v)$ to be the number of inputs that are canonically silent at v. We want to estimate $\mathbb{E}[S(V)|V \in E]$. For this purpose we consider some $1 \leq i \leq q$ and note that:

$$\begin{aligned}\Pr[A_i^x(V) = 1 | V \in E] &= \Pr[V(W_i^x(V)) = 1 \text{ and } W_i^x(V) \notin \bar{B}_i | V \in E] \\ &= \Pr[V(y_i^x) = 1 \text{ and } y_i^x \notin B | V \in E] \leq \rho/q.\end{aligned}$$

Therefore, by a union bound we have that:

$$\Pr[S^x(V) = 0 | V \in E] = \Pr[\sum_{1 \leq i \leq q} A_i^x(V) \neq 0 | V \in E] \leq \sum_{1 \leq i \leq q} \Pr[A_i^x(V) = 1 | V \in E] \leq q \cdot (\rho/q) = \rho.$$

Therefore, by linearity of expectation:

$$\mathbb{E}[S(V)|V \in E] = \sum_{x \in \{0,1\}^k} \mathbb{E}[S^x(V)|V \in E] = \sum_{x \in \{0,1\}^k} \Pr[S^x(V) = 1 | V \in E] \geq 2^k \cdot (1 - \rho).$$

By the probabilistic method we can conclude that there exists $v \in E$ such that $S(v) \geq 2^k \cdot (1 - \rho)$, which means that the number of canonically silent inputs for v is as required.

Why this approach does not directly extend to the adaptive case. Let us first quickly summarize the approach above. In the case that R is non-adaptive we used Lemma 2.1 on event E to obtain a small set B of bad queries. We set $B_1 = B_2 = \ldots = B_q = B$ to be the set of bad queries. We were then able to argue that for every input x, conditioned on the event $\{V \in E\}$ it is unlikely that $R^D(x, \alpha')$ asks a query $Q_i^x(V) \notin B$ such that $V(Q_i^x(V)) = 1$.

We now observe that this is not necessarily true in case that R is adaptive. More precisely, we observe that there exists an oracle procedure R that makes n queries, and an event E such that $\Pr[V \in E] \geq 2^{-\Omega(n \log(1/\epsilon))}$ such that for every set $B \subseteq \{0,1\}^n$ of size $o(2^n)$ (and in particular to the set chosen by Lemma 2.1 which is much smaller), for every input x, with high probability conditioned on $\{V \in E\}$, R asks a query $Q_i^x(V) \notin B$ such that $V(Q_i^x(V)) = 1$. In particular, we cannot hope that the canonical execution and the real execution coincide on many inputs.

We now sketch this example. Fix some distinct $y_1, \ldots, y_{n-1}, z_1, \ldots, z_{n-1} \in \{0,1\}^n$ and assume that n is large enough so that all these strings start with zero. We define event $A = \{\forall i : V(y_i) \neq V(z_i)\}$. We interpret the sequence $P = (V(y_1), \ldots, V(y_{n-1}))$ as an $n-1$ bit string. Note that $(P|V \in A)$ is uniformly distributed over $\{0,1\}^{n-1}$. This is because before conditioning, for every i the two events $\{V(y_i) = 0, V(z_i) = 1\}$ and $\{V(y_i) = 1, V(z_i) = 0\}$ are equally likely. (This is the same observation that is made in the analysis of the so called "von-Neumann extractor".) We now consider the event $E = A \cap \{V(1 \circ P) = 1\}$ (where $1 \circ P$ refers to the n bit string that is the concatenation of '1' and P). Note that conditioned on E, P is uniformly distributed over $\{0,1\}^{n-1}$ and with probability one, $V(1 \circ P) = 1$. The adaptive procedure R described next makes n queries. On every input x, the procedure R first queries oracle D at $y_1, \ldots, y_{n-1}$ and computes P. It then queries D at $1 \circ P$ and note that $\Pr[V(1 \circ P) = 1 | V \in E] = 1$. Yet, for every set B, $\Pr[1 \circ P \in B | V \in E] \leq |B|/2^{n-1}$. This indeed means that for any small set B, conditioned on $\{V \in E\}$, R is very likely to ask a query that is not in B, and yet D does not answer '$\perp$' on this query.

Modifications needed for the adaptive case. The main technical contribution of this paper is developing an approach to handle adaptive reductions. An inspection of the example above suggests how to extend the proof technique to adaptive reductions. When we apply Lemma 2.1 on the event E above we obtain the set $B_1 = \{y_1, \ldots, y_{n-1}, z_1, \ldots, z_{n-1}\}$. This gives hope as the lemma "correctly identifies" that this set of queries is special. Consider the most likely outcome z of the random variable $V(B_1) = (V(y))_{y \in B_1}$ and let $E_1 = E \cap \{V(B_1) = z\}$.

Recall that we are planning to use the probabilistic method to show the existence of a function $v \in E$ with some nice properties. Since $E_1 \subseteq E$, we may as well perform the probabilistic analysis in the probability space $(V|V \in E_1)$ rather than $(V|V \in E)$. The advantage is that conditioned on E_1, P is fixed to some constant value p. Furthermore, if we apply Lemma 2.1 with event E_1 we obtain a set $B_2 \subseteq \{0,1\}^n$ that contains $1 \circ p$. At this point we can decide that the set of bad queries is $B = B_1 \cup B_2$, and note that conditioned on E_1, the reduction R is "handled correctly" in the sense that it does not ask a good query y on which V answers one.

Summing up, we were able to perform the analysis conditioned on some event $E_1 \subseteq E$. We note that in the example above the reduction R uses two "levels of adaptivity". In case R makes q levels of adaptivity, we will need to make q iterations of the process above. In each iteration we will use Lemma 2.1 to identify a new set of bad queries B_i and will "further condition" the probability space to some event $E_i \subseteq E_{i-1}$ by fixing the bad queries.

The actual proof given in the next section applies this high level idea. A difficulty that arises is that in the end, we want to perform the analysis conditioned on event E_q (which is the final event in the "iterative further conditioning process"). However, the guarantees that we get from Lemma 2.1 on intermediate E_i's are not necessarily maintained in E_q.

2.5 Proof of Lemma 2.5

Let us start with 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 sometimes view $v(B)$ as an element in $\{0,1\}^B$, namely a string of length $|B|$ that is defined by picking some order on the set B.

The first step towards proving Lemma 2.5 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, and note that $\bar{B}_0 = \emptyset$.

We have already defined sets $B_1, \ldots, B_i$ and therefore the functions $W_1^x(v), \ldots, W_{i+1}^x(v)$ and $A_1^x(v), \ldots, A_i^x(v)$ are already well defined (even though we did not yet define sets $B_{i+1}, \ldots, B_q$). We can use these functions to define random variables $W_1^x, \ldots, W_{i+1}^x$ by $W_j^x = W_j^x(V)$ as well as $A_1^x, \ldots, A_i^x$ by $A_j^x = A_j^x(V)$.
We assume that the following invariant holds at step i:

- $|B_i| = O(\frac{aq^3}{\rho^2})$ where the hidden constant does not depend on i. (Note that this holds for $i = 0$.)
- There exists a fixed $b_i \in \{0,1\}^{\bar{B}_i}$ such that $E_i \subseteq \left\{V(\bar{B}_i) = b_i\right\}$. (Note that this vacuously holds for $i = 0$ as $\bar{B}_0 = \emptyset$ and therefore the event $\left\{V(\bar{B}_0) = b_0\right\}$ is the entire probability space.)
- $\Pr[E_i | V(\bar{B}_i) = b_i] \geq 2^{-(a+1+i)}$. (Note that this holds for $i = 0$ as $\Pr[E_0] \geq 2^{-(a+1)}$).

- For every $1 \leq j \leq i$, $\Pr[\sum_{x\in\{0,1\}^k} A^x_j \leq \frac{\rho\cdot 2^k}{q} | V \in E_i] = 1$. (Note that this holds vacuously for $i = 0$.)

We next 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 2.5.

Obtaining the set B_{i+1} and event E_{i+1} that meet the invariant. We are planning to apply Lemma 2.1 to obtain the set B_{i+1}. We now state the choices with which we will apply the Lemma. Let $L = \{0,1\}^n \setminus \bar{B}_i$ be the set of queries that we did not yet mark as "bad". Let $Z = V(L)$ and $N = |L|$. We view Z as a sequence of N random variables defined by $(V(y))_{y\in L}$ and note that these N variables are independent as required by Lemma 2.1. We are planning to use E_i to play the role of E from the lemma. Note that E_i is an event in our probability space, meaning that it is some subset of functions $v : \{0,1\}^n \to \{0,1\}$. For the lemma, we need to view E_i as a subset $\hat{E}_i$ of $\{0,1\}^N$. For this purpose we define $\hat{E}_i = \{v(L) : v \in E_i\}$. To apply the lemma on Z with event $\hat{E}_i$ we need to check that $\Pr[Z \in \hat{E}_i]$ is large. Towards this goal, we first note that:

$$\Pr[V \in E_i | V(\bar{B}_i) = b_i] = \frac{\Pr[V \in E_i \cap V(\bar{B}_i) = b_i]}{\Pr[V(\bar{B}_i) = b_i]} = \frac{\Pr[V(L) \in \hat{E}_i \cap V(\bar{B}_i) = b_i]}{\Pr[V(\bar{B}_i) = b_i]}$$

$$= \frac{\Pr[V(L) \in \hat{E}_i] \cdot \Pr[V(\bar{B}_i) = b_i]}{\Pr[V(\bar{B}_i) = b_i]} = \Pr[V(L) \in \hat{E}_i].$$

We now use the computation above to verify that $\Pr[Z \in \hat{E}_i]$ is large:

$$\Pr[Z \in \hat{E}_i] = \Pr[V(L) \in \hat{E}_i] = \Pr[V \in E_i | V(\bar{B}_i) = b_i] = \Pr[E_i | V(\bar{B}_i) = b_i] \geq 2^{-(a+1+i)} \geq 2^{-(a+1+q)}.$$

We can now apply Lemma 2.1 setting $\eta = \rho/10q$ and note that $\eta \geq \epsilon$ by the requirement on ρ in Theorem 1.5. Let $B_{i+1} \subseteq L \subseteq \{0,1\}^n$ be the set obtained from Lemma 2.1. We have that

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

Thus, B_{i+1} indeed meets the size requirement of the invariant. By the lemma, for every $y \in L \setminus B_{i+1}$ we have that $(Z(y)|Z \in \hat{E}_i)$ is η-close to $Z(y)$. Note that $y \in L \setminus B_{i+1}$ iff $y \in \{0,1\}^n \setminus \bar{B}_{i+1}$. We use this to conclude that for every $y \notin \bar{B}_{i+1}$,

$$\Pr[V(y) = 1 | V \in E_i] = \Pr[Z(y) = 1 | Z \in \hat{E}_i] \leq \Pr[Z(y) = 1] + \eta \leq 2\epsilon + \eta \leq 3\eta$$

where the first equality follows because for $y \notin \bar{B}_{i+1}$, $(V(y)|V \in E_i)$ is distributed like $(Z(y)|Z \in \hat{E}_i)$, and the second inequality follows by the consequence of Lemma 2.1.

We now observe that conditioned on the event E_i, for every $x \in \{0,1\}^k$, the random variable W^x_{i+1} is fixed to some constant $y^x \in \{0,1\}^n$. (Or more formally, for every $x \in \{0,1\}^k$ there exists $y^x \in \{0,1\}^n$ such that $\Pr[W^x_{i+1} = y^x | E_i] = 1$.) This is because $E_i \subseteq \left\{V(\bar{B}_i) = b_i\right\}$ which means that all answers of D 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 $x, \bar{B}_i$ and $D(\bar{B}_i)$.

We observe that having defined B_{i+1}, the random variable $A^x_{i+1} = A^x_{i+1}(V)$ is now defined for every $x \in \{0,1\}^k$. This is because the function $A^x_{i+1}(v)$ is defined in terms of $W^x_{i+1}(v)$ (which is already defined) and $\bar{B}_{i+1}$. By the consequences of our application of Lemma 2.1 we have that for every $x \in \{0,1\}^k$:

$$\mathbb{E}[A^x_{i+1} | V \in E_i] = \Pr[A^x_{i+1} = 1 | V \in E_i] = \Pr[V(W^x_{i+1}) = 1 \wedge W^x_{i+1} \notin \bar{B}_{i+1} | V \in E_i]$$

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

Thus, by linearity of expectation we have that:

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

and by Markov's inequality:

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

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

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

The event E_{i+1} (that we need to define) 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 the sum of A^x_{i+1} in the invariant. We have seen that $\Pr[E'_i | E_i] \geq 1/2$ and therefore

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

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

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

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

$$\begin{aligned} &\Pr[E_{i+1} | V(\bar{B}_{i+1}) = b_{i+1}] = \Pr[E'_i | V(\bar{B}_{i+1}) = b_{i+1}] \\ &= \Pr[E'_i | V(\bar{B}_i) = b_i \wedge V(B_{i+1}) = z] \geq 2^{-(a+1+(i+1))}. \end{aligned}$$

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 2.5. 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 for every $1 \leq i \leq q$, $|B_i| = O(aq^3/\rho^2)$ as required in Lemma 2.5. Let $v : \{0,1\}^n \to \{0,1\}$ be some function such that $v \in E_q \subseteq E$. We have that for every $1 \leq j \leq q$,

$$\sum_{x \in \{0,1\}^k} A^x_j(v) \leq \frac{\rho \cdot 2^k}{q}.$$

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 silent for v. This concludes the proof of the lemma.

3 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 $f:\{0,1\}^k\to\{0,1\}$ with its truth table which is a string $f\in\{0,1\}^K$ for $K=2^k$.

Definition 3.1 (List-decodable codes)**.** *A map $Enc:\{0,1\}^K\to\{0,1\}^N$ is (ϵ,A)-list-decodable if for every $D\in\{0,1\}^N$, the list of all strings $f\in\{0,1\}^K$ such that D has agreement $1/2+\epsilon$ with $Enc(f)$, has size at most A. Enc is uniquely-decodable if $A=1$.*

It is well known that a map cannot be uniquely decodable for $\epsilon<1/4$ unless K is tiny. 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 3.2 (Local list-decoder)**.** *Let $Enc:\{0,1\}^K\to\{0,1\}^N$ 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 $1\leq\alpha\leq A'$ such that for every $x\in\{0,1\}^k$, $\Pr[Dec^D(x,\alpha)=f(x)]\geq 1-\delta$ 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 3.3 (Local list-decoders imply function-generic hardness amplification)**.** *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 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. 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 $D\in\{0,1\}^{2^n}$. 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 $1\leq\alpha\leq 2^{a'}$ such that $Dec^D(\cdot,\alpha)$ has agreement $1-\delta$ with f when its coins are fixed to β. We define $Amp=Enc$ and $R^{(\cdot)}(x;(\alpha,\beta))=Dec^{(\cdot)}(x,\alpha)$ using β as coins.[5] □

[5]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 3.2 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.

It is interesting to note that even in the special case of unique decoding, Lemma 3.3 gives a function-generic reduction that is non-uniform. The following corollary is obtained by applying Theorem 1.4.

Corollary 3.4 (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 $a', \frac{1}{\epsilon}, n, t \leq 2^{k/c}$ 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 3.1 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 3.4 applies in this setting even when allowing list-decoding.

4 Conclusion and open problems

Our results rule out certain proof techniques for showing hardness amplification results with small "size loss". As we explain in Section 1.5, the framework of reductions that we study captures essentially all hardness amplification results in the literature. Nevertheless, it may be possible to bypass these limitations by developing alternative proof techniques. We remark that the techniques of [GSTS07, Ats06] are not captured in our framework (as explained in Section 1.5).

We now mention a few open problems (continuing the discussion of Section 1.6).

- Extend the results of [SV10, GR08] regarding "necessity of majority" to *adaptive* non-uniform reductions. More specifically, show that non-uniform and adaptive function-generic reductions for mildly-average-case to average-case hardness amplification cannot be computed by small constant depth circuits if ϵ is small.

- Extend the results of [SV10] regarding "number of queries" to *adaptive* reductions. More specifically, show that non-uniform and adaptive function-generic reductions for mildly-average-case to average-case hardness amplification must use $q = \Omega(\frac{\log(1/\delta)}{\epsilon^2})$ queries. (Note that a lower bound of $q = \Omega(1/\epsilon)$ follows from our results on basic hardness amplification.)

- Our results on basic hardness amplification give a lower bound of $q = \Omega(1/\epsilon)$ for $\delta \leq 1/3$. This meets the known upper bounds for constant δ. However, it seems that the right lower bound should be $q = \Omega(\frac{\log(1/\delta)}{\epsilon})$ and match the known upper bounds of [KS03]. We do not know how to show such a bound for non-uniform and adaptive reductions. We mention that the approach presented in this paper can be used to show such a lower bound for function-generic reductions that are non-adaptive. We also mention that such a lower bound also follows from the approach of [SV10] as observed in [Wat11].

Finally, the framework of function-specific reductions suggested in this paper captures more proof techniques than those captured in earlier work. It is natural to study the questions above (as well as related questions in the area) using this more general framework.

Acknowledgements

The second author is grateful to Oded Goldreich, Avi Wigderson and Emanuele Viola for many interesting discussions on hardness amplification. We also thank Oded Goldreich, Danny Gutfreund, Iftach Haitner and anonymous referees for helpful comments and suggestions.

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 *Proceedings of the 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.

[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 *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pages 440–449, 2007.

[GIL+90] O. Goldreich, R. Impagliazzo, L. A. Levin, R. Venkatesan, and D. Zuckerman. Security preserving amplification of hardness. In *Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science*, pages 318–326, 1990.

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

[GNW11] O. Goldreich, N. Nisan, and A. Wigderson. On Yao's XOR-lemma. In Oded Goldreich, editor, *Studies in Complexity and Cryptography*, volume 6650 of *Lecture Notes in Computer Science*, pages 273–301. Springer, 2011.

[GR08] D. Gutfreund and G. Rothblum. The complexity of local list decoding. In *Proceedings of the 12th Intl. Workshop on Randomization and Computation*, 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 *Proceedings of the 11th Intl. Workshop on Randomization and Computation*, pages 569–583, 2007.

[GV08] D. Gutfreund and S. P. Vadhan. Limitations of hardness vs. randomness under uniform reductions. In *Proceedings of the 12th Intl. Workshop on Randomization and Computation*, 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 *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science*, pages 538–545, 1995.

[IW97] R. Impagliazzo and A. Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 220–229, 1997.

[IW01] R. Impagliazzo and A. Wigderson. Randomness vs time: Derandomization under a uniform assumption. *J. Comput. Syst. Sci.*, 63(4):672–688, 2001.

[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.

[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.

[LTW11] C.-J. Lu, S.-C. Tsai, and H.-L. Wu. Complexity of hard-core set proofs. *Computational Complexity*, 20(1):145–171, 2011.

[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 *Proceedings of the 1st Theory of Cryptography Conference*, 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 *Proceedings of the 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.

[Wat11] T. Watson. Query complexity in errorless hardness amplification. In *Proceedings of the 15th Intl. Workshop on Randomization and Computation*, pages 688–699, 2011.