

國立交通大學

資訊工程學系

博士論文



以黑箱法加強布林函數難度之複雜度

The Complexity of Black-Box Hardness Amplification

研究生：吳信龍

指導教授：蔡錫鈞 教授

中華民國九十六年四月

以黑箱法加強布林函數難度之複雜度
The Complexity of Black-Box Hardness Amplification

研究生：吳信龍

Student : Hsin-Lung Wu

指導教授：蔡錫鈞

Advisor : Shi-Chun Tsai

國立交通大學資訊學院
資訊工程學系
博士論文



A dissertation is submitted to
Department of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy
in
Computer Science
April 2007
Hsinchu, Taiwan, Republic of China

中華民國九十六年四月

以黑箱法加強布林函數難度之複雜度

學生：吳信龍

指導教授：蔡錫鈞

國立交通大學 資訊工程學系 博士班

摘 要

在高複雜度下，不同的布林函數難度與偽亂數性是等價的。然而，在 NP 複雜度下，它們之間的關係是很不清楚的。在本論文之前半部，我們建立 mild-hardness, average-case hardness 與偽亂數性在 NP 之下的等價性，並且說明上述概念與 worst-case hardness 之分野。我們主要有下列的結果：1、任何強黑箱式地從 worst-case hardness 加強至 average-case hardness 不可能在 NP 下實踐。2、任何強黑箱式地從 worst-case hardness 加強至 average-case hardness 需要多量額外之消息元 (advice bits)。3、如能在 NP 複雜度下，以弱黑箱式達成從 worst-case hardness 加強至 average-case hardness 之難度加強法，其意含一 NP 問題類之 average-case hardness。4、改進 Healy 等人之 NP 問題類之難度加強法之結果。在本論文之第二部份，我們探討難核構造之問題。我們主要有三項結果：1、任何強式黑箱式難核構造法均需要大量之詢問元 (query bits)。2、任何弱式黑箱式難核構造法均需多量之消息元 (advice bits)。3、弱式黑箱式難核構造法不可能在 $AC^0[p]$ 下達成。

The Complexity of Black-Box Hardness Amplification

Hsin-Lung Wu

April 17, 2007





Abstract

It is well-known that hardness and pseudorandomness are equivalent in high complexity class such as exponential time [NW94]. However, the relationship between various degrees of hardness and pseudorandomness is not clear in NP. In the first part of thesis, we widen the gap between worse-case hardness and average-case hardness while establishing the equivalence between average-case hardness and pseudorandomness in NP.

By using the method developed in [IL90] and [NW94], one can build the equivalence between average-case hardness and pseudorandomness within NP. On the other hand, the interplay between worse-case hardness and average-case hardness is closely related to the so-called hardness amplification which is the task of transforming a hard function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, with which any small circuit disagrees on δ fraction of the input, into a harder function f' , with which any small circuit disagrees on ε fraction of the input where $\varepsilon > \delta$. To separate worse-case hardness and average-case hardness in NP, we study the complexity of hardness amplification procedures. Our results include the following.

- No strongly black-box hardness amplification from hardness $(1 - \delta)/2$ to $(1 - \delta^k)/2$ can be realized in $\text{ATIME}(O(1), k^{o(1)})$. As a result, for $k = n^{\omega(1)}$, such hardness amplification cannot be carried out in NP. Therefore, such hardness amplification in general requires a high complexity.
- We show that even without any restriction on the complexity of the amplification procedure, such a strongly black-box hardness amplifica-

tion must be inherently non-uniform in the following sense. To guarantee the hardness of the resulting function f' , even against uniform machines, one has to start with a function f which is hard against non-uniform algorithms with $\Omega(k \log(1/\delta))$ bits of advice.

- From worst-case hardness to average-case hardness, we consider a weaker class of hardness amplifications called weakly black-box hardness amplification. First, we show that if an amplification procedure in **NP** can amplify hardness beyond a polynomial factor, then it must embed in itself a hard function computable in **NP**. As a result, it is impossible to have such a hardness amplification with hardness measured against **NP/poly**.
- We consider the task of transforming non-negligible hardness to average-case hardness for the complexity class **NP**. We show that if there is a balanced function in **NP** such that any circuit of size $s(n) = 2^{\Omega(n)}$ fails to compute it on a $1/\text{poly}(n)$ fraction of inputs, then there is a function in **NP** such that any circuit of size $s'(n)$ fails to compute it on a $1/2 - 1/s'(n)$ fraction of inputs, with $s'(n) = 2^{\Omega(n^{2/3})}$. This improves the result of Healy et al. (STOC'04), which only achieves $s'(n) = 2^{\Omega(n^{1/2})}$ for the case with $s(n) = 2^{\Omega(n)}$.

In the second part of this thesis, we study a fundamental result of Impagliazzo (*FOCS'95*) known as the hard-core set lemma. Consider any function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ which is “mildly-hard”, in the sense that any circuit of size s must disagree with f on some δ fraction of inputs. Then the hard-core lemma says that f must have a hard-core set H of density δ on which it is “extremely hard”, in the sense that any circuit of size $s' = O(s / (\frac{1}{\varepsilon^2} \log(\frac{1}{\varepsilon\delta})))$ must disagree with f on at least $(1 - \varepsilon)/2$ fraction of inputs from H .

There are three issues of the lemma which we would like to address: the loss of circuit size, the need of non-uniformity, and its inapplicability to a low complexity class. We introduce two models of hard-core set constructions,

a strongly black-box one and a weakly black-box one, and show that those issues are unavoidable in such models.

- We show that in any strongly black-box construction, one can only prove the hardness of a hard-core set for smaller circuits of size at most $s' = O(s/(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}))$.
- We show that any weakly black-box construction must be inherently non-uniform — to have a hard-core set for a class G of functions, we need to start from the assumption that f is hard against a non-uniform complexity class with $\Omega(\frac{1}{\varepsilon} \log |G|)$ bits of advice.
- We show that weakly black-box constructions in general cannot be realized in a low-level complexity class such as $AC^0[p]$ — the assumption that f is hard for $AC^0[p]$ is not sufficient to guarantee the existence of a hard-core set.





Acknowledgements

I am grateful to my advisor, Dr. Shi-Chun Tsai, for his guidance and encouragement. I also thank to Dr. Chi-Jen Lu for showing me how to do research. Thank to my wife, Amy, for her spiritual support.





To Amy





Table of Contents

Table of Contents	11
List of Figures	15
1 Introduction	17
1.1 Background	17
1.2 Hardness Amplification	18
1.3 Hard-core set construction	19
1.4 Black-Box Models	21
1.5 Our results and Organization of this thesis	23
1.6 Notations and Useful Facts	26
1.6.1 Tail Bounds of Binomial Distribution	27
2 Strongly Black-Box Hardness Amplification	29
2.1 Introduction	29
2.1.1 Previous Lower Bound Results	30
2.1.2 Our Results	31
2.1.3 Our Techniques	34
2.1.4 Organization of this chapter	35
2.2 Preliminaries	36
2.2.1 Strongly Black-Box Hardness Amplification and Pseudorandom Generators	36
2.2.2 Codes and Correspondence to Hardness Amplification	39
2.2.3 Noise Sensitivity	41

2.3	Impossibility of Amplification by Small-Depth Circuits	42
2.4	Impossibility of Amplification by Nondeterministic Circuits	47
2.5	Inherent Non-uniformity of Hardness Amplification	50
2.6	Impossibility Results on PRG Constructions	55
3	Weakly Black-Box Hardness Amplification	59
3.1	Introduction	59
3.1.1	Previous Results	59
3.1.2	Our Results	60
3.1.3	Organization of this chapter	62
3.2	Preliminaries	62
3.3	Impossibility of Hardness Amplification in $\text{TIME}(t)$	63
3.4	Impossibility Results in $\Sigma_k P$	66
4	Hardness Amplification in NP	73
4.1	Introduction	73
4.1.1	Organization of this chapter	74
4.2	Preliminaries	74
4.2.1	Hardness Amplification	75
4.2.2	PRGs for Branching Programs and Rectangles	77
4.3	Proof of Main Theorem	78
4.3.1	Discussion	81
5	Hardness and Pseudorandomness in NP	83
5.1	Introduction	83
5.1.1	Previous Results	84
5.1.2	Our Results	84
5.1.3	Organization of this Chapter	86
5.2	Preliminaries	86
5.2.1	Universal Hash Functions	87
5.3	Hardness from Pseudorandomness	88

<i>TABLE OF CONTENTS</i>	13
6 Hardcore Set Constructions	97
6.1 Introduction	97
6.1.1 Our Results	98
6.1.2 Bounds from Hardness Amplification	99
6.1.3 Our Techniques	99
6.1.4 Organization of this chapter	101
6.2 Preliminaries	101
6.2.1 Hardness and Hard-Core Set Lemma	102
6.2.2 Black-Box Constructions of Hard-Core Set	103
6.3 Query Complexity in Strongly Black-Box Construction	104
6.4 Advice Complexity in Weakly Black-Box Construction	107
6.5 Weakly Black-Box Construction $\notin AC^0[p]$	111
Bibliography	113





List of Figures

- 5.1 The relationship among PRG and various hardness assumptions within NP. Arrows indicate black-box transformations. **BB** and **WBB trans.** indicate black-box and weakly black-box transformations respectively. Note that the slash symbol means "the transformation cannot be done in NP". 85





Chapter 1

Introduction

1.1 Background

Understanding the power of randomness in computation is one of the central topics in theoretical computer science. A major open question is the BPP versus P question, asking whether or not all randomized polynomial-time algorithms can be converted into deterministic polynomial-time ones. A standard approach to derandomizing BPP relies on constructing the so-called pseudorandom generators (PRG), which stretch a short random seed into a long pseudorandom string that looks random to circuits of polynomial size. So far, all known constructions of PRG are based on unproven assumptions of the nature that certain functions are hard to compute. The idea of converting hardness into pseudorandomness first appeared in the work of Blum and Micali [BM82] and Yao [Yao82]. This was made more explicit by Nisan and Wigderson [NW94], who showed how to construct a PRG based on a Boolean function which is hard in an average-case sense. To get a stronger result, one would like to relax the hardness assumption, and a series of research [NW94, BFNW93, Im95] then worked on how to transform a function into a harder one. Finally, Impagliazzo and Wigderson [IW97] were able to convert a function in E that is hard in worst case into one that is hard in average case, both against circuits of exponential size. As a result, they obtained

$\text{BPP} = \text{P}$ under the assumption that some function in E cannot be computed by a circuit of sub-exponential size. Simpler proofs and better trade-offs have been obtained since then [STV01, ISW00, SU01, Uma03].

1.2 Hardness Amplification

Note that hardness amplification is the major step in derandomizing BPP in the research discussed above, as the step from an average-case hard function to a PRG is relatively simple and has low complexity. We say that a Boolean function f is β -hard (or has hardness β) against circuits of size s if any such circuit attempting to compute f must make errors on at least β fraction of the input. The error bound β is the main parameter characterizing the hardness; the size bound s also reflects the hardness, but it plays a lesser role in our study. Formally, the task of hardness amplification is to transform a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ which is β -hard against circuits of size $s(n)$ into a function $f' : \{0, 1\}^m \rightarrow \{0, 1\}$ which is β' -hard against circuits of size $s'(m)$, with $\beta < \beta'$ and $s'(m)$ close to (usually slightly smaller than) $s(n)$. Normally, one would like to have m as close to n as possible, preferably with $m = \text{poly}(n)$, so that one could have $s'(m)$ close to $s(m)$; otherwise, one would only be able to have the hardness of f' against much smaller circuits. Furthermore, one would like f' to stay in the same complexity class of f , so that one could establish the relation among hardness assumptions within the same complexity class.

In this thesis, we will consider the following issues from those works on hardness amplification.

The complexity of the amplification procedure: All previous amplification procedures going from worst-case hardness ($\beta = 2^{-n}$) to average-case hardness ($\beta' = 1/2 - 2^{-\Omega(m)}$) need exponential time [BFNW93, IW97, STV01] (or slightly better, in linear space [KM02] or $\oplus\text{ATIME}(O(1), n)$ [Vio04]). As a result, such a hardness amplification is only known for functions in high

complexity classes. Then a natural question is: can it be done for functions in lower complexity classes? For example, given a function in NP which is worst-case hard, can we transform it into another function in NP which is average-case hard? Only for some range of hardness (e.g. starting from mild hardness, with $\beta = 1/\text{poly}(n)$) is this known to be possible [Yao82, NW94, IW97, OD02, HVV04].

Non-uniformity of hardness amplification: Hardness amplification typically involves non-uniformity in the sense that hardness is usually measured against *non-uniform* circuits. In fact, one usually needs to start from a function which is hard against non-uniform circuits, even if one only wants to produce a function which is hard against uniform Turing machines. This is why most results on hardness amplification are based on non-uniform assumptions.

1.3 Hard-core set construction

One fundamental notion in complexity theory is the hardness of a function. Informally speaking, a function f is hard if any circuit of small size must fail to compute it correctly on some inputs. More precisely, we can characterize the hardness by parameters δ and s , and say that f is δ -hard (or has hardness δ) for size s if any circuit of size s must fail to compute f correctly on at least δ fraction of inputs. One may wonder if the hardness of a function basically comes from a subset of density about δ . So the question is: given any δ -hard function for size s , is there always a subset of inputs of density about δ on which f is extremely hard for circuits of size about s ? A seminal result of Impagliazzo [Im95] answers this affirmatively. He showed that any δ -hard function for size s indeed has a subset H of the inputs with density δ on which f has hardness $(1 - \varepsilon)/2$ for circuits of size $s' = O(s/(\frac{1}{\varepsilon^2} \log \frac{1}{\delta\varepsilon}))$. Such a set H is called an ε -hard-core set for size s' .

In addition to answering a very basic question in complexity theory, the

hard-core set lemma has found applications in learning theory [KS03] and cryptography [Hol05], and has become an important tool in the study of pseudo-randomness. It can be used to provide an alternative proof of Yao's celebrated XOR Lemma [Im95], or to construct pseudo-random generator directly from a mildly-hard function, bypassing the XOR lemma [STV01]. Recently, it has become a key ingredient in the study of hardness amplification for functions in NP [OD02, Tre03, HVV04, Tre05]. In spite of its importance, there are some issues of the hard-core lemma which are still not well understood and have become the bottlenecks in some applications. This calls for a more thorough study of the lemma. In this thesis, we consider the following issues concerning hard-core set constructions.

Loss of circuit size: Note that in Impagliazzo's result, the hardness on the hard-core set, although increased, is actually measured against circuits of a smaller size s' , as opposed to the initial size s . This loss of circuit size was later reduced by Klivans and Servedio [KS03] who showed the existence an ε -hard-core set of density $\delta/2$ for size $s' = O(s/(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}))$. Then a natural question is: can the size s' be further improved to, say, $\Omega(s)$?

Non-uniformity of hard-core set construction: Note that even when one only wants to have a hard-core set which is hard for uniform algorithms, one still needs to start from a function which is hard for non-uniform circuits, or algorithms supplied with advices. In fact, this becomes the bottleneck in Trevisan's work of uniform hardness amplification for functions in NP [Tre03, Tre05], in which he showed how to amplify hardness from $1 - 1/\text{poly}(n)$ to $(1 - \varepsilon)/2$ against BPP algorithms, with $\varepsilon = 1/\log^c n$ for a small constant $c < 1$. What prevents him from reaching a larger hardness, say with $\varepsilon = 1/n$, is the large number (proportional to $1/\varepsilon^2$) of advice bits needed by the hard-core set lemma. On the other hand, it is known that hardness amplification for functions in a higher complexity class, such as EXP, only requires $O(\log(1/\varepsilon))$ bits of advice [STV01]. So a natural question is: can the number of advice bits needed in the hard-core set lemma be reduced?

The complexity of decoding algorithm of hard-core set construction: The last issue is that the lemma currently does not apply to a low-level complexity class such as $AC^0[p]$. That is, one needs to start from the assumption that f is hard for a complexity class which is high enough to include the majority function. Thus, an interesting question is: for any function f which is δ -hard for $AC^0[p]$, does it always have an ε -hard-core set for $AC^0[p]$?

All these three issues seem inherent in Impagliazzo's proof and they look difficult to resolve. One may wonder that perhaps they are indeed impossible to avoid. However, proving such negative results appears to require proving circuit lower bounds, which seems to be far beyond our reach. Therefore, we would like to identify general and reasonable models for the hard-core set lemma in which such negative results can actually be proven.

1.4 Black-Box Models

Black-Box Hardness Amplification. In light of the discussion above, one would hope to show that some hardness amplification or hard-core set construction are indeed impossible. However, it is not clear what this means, especially given the possibility (in which many people believe) that average-case hard functions may indeed exist.

One important type of hardness amplification is called *strongly black-box* hardness amplification. First, the initial function f is only given as a black-box to construct the new function f' . That is, there is an oracle Turing machine AMP such that $f' = AMP^f$, so f' only uses f as an oracle and does not depend on the internal structure of f . Second, the hardness of the new function f' is proved in a black-box way. That is, there is an oracle Turing machine DEC, such that if some algorithm A computes f' correctly on β' fraction of the input, then DEC using A as an oracle can compute f correctly on β fraction of the input. Again, DEC only uses A as an oracle and does not depend on the internal structure of A . We call AMP the *encoding* procedure and DEC the *decoding* procedure. In fact, almost all previous

constructions of hardness amplification are done in a black-box way, so it is nice to establish impossibility results for such type of hardness amplification.

One relaxation is the so-called *weakly black-box* hardness amplification, in which the hardness proof is no longer required to be done in a strongly black-box way (dropping the requirement of having a decoding procedure). Precisely, its hardness proof is only to show the following statement: if there is an efficient adversary A computing \bar{f} correctly on at least $(1 - \bar{\varepsilon})$ -fraction of inputs, then there exists an efficient adversary B which computes the initial function f on at least $(1 - \varepsilon)$ fraction of inputs. Note that the analysis is arbitrary and hence is not necessarily restricted in a black-box way. In this sense, this weakly model is a natural relaxation of strongly black-box model. The difference between strongly and weakly black-box models is remarkable especially when an average-case hard function indeed exists. A hardness proof of the weakly black-box model may just to show that the resulting function \bar{f} is close to that average-case hard one. Hence this sufficiently fulfills the statement of hardness proof. However, this proof approach is not allowed for the strongly black-box model. Again, as we will see, the weakly black-box hardness amplification also has its limitation when it is unable to embed any average-case (or mildly) hard function in itself.

Black-Box Hard-Core Set Constructions. The hard-core set lemma, when stated in the contrapositive way, basically says that given any function f with no hard-core set for small circuits (on any such subset H , there is a small circuit C_H with a good correlation with f), one can find a small circuit C which is close to f . A closer look at Impagliazzo's proof of hard-core set lemma shows that the circuit C is simply the weighted majority on a small subset of those circuits C_H 's. In fact, one can replace the class of small circuits C_H 's by any class G of functions, and Impagliazzo's proof shows that given any f with no hard-core set for functions in G , one can construct a function C close to f by taking a weighted majority on a small subset of functions in G . We call this type of argument a hard-core set construction,

and note that C only uses those functions in G as an oracle (or a black box).

This motivates us to define our first model of hard-core set constructions as follows. We say that a (non-uniform) oracle algorithm $\text{DEC}^{(\cdot)}$ with a decision function $D : \{0, 1\}^q \rightarrow \{0, 1\}$ realizes a *strongly black-box* (δ, ε, k) -construction (of hard-core set) if the following holds. First, DEC will be given a family $G = \{g_1, \dots, g_k\}$ of functions as oracle together with a multi-set $I = \{i_1, \dots, i_q\}$ as advice, and for any input x , it will query the functions g_{i_1}, \dots, g_{i_q} , all at x , and then output $D(g_{i_1}(x), \dots, g_{i_q}(x))$. Moreover, it satisfies the property that for any G and for any f which has no ε -hard-core set of density $\Omega(\delta)$ for G , there exists a multi-set I of size q such that the function $\text{DEC}^{G,I}$ is δ -close to f ($\text{DEC}^{G,I}(x) \neq f(x)$ for at most δ fraction of x). We call q the query complexity of DEC , and observe that it relates to the loss of circuit size in the hard-core set lemma, with $s' = O(s/q)$. Note that the known hard-core set constructions [Im95, KS03] are in fact done in such a strongly black-box way.

Our second model of hard-core set constructions generalizes the first one by removing the constraint on how the algorithm DEC works; the algorithm DEC and its advice now are allowed to be of arbitrary form. We say that a (non-uniform) oracle algorithm $\text{DEC}^{(\cdot)}$ realizes a *weakly black-box* (δ, ε, k) -construction (of hard-core set) if the following holds. For any family G of k functions and for any function f which has no ε -hard-core set of density $\Omega(\delta)$ for G , there exists an advice string α such that $\text{DEC}^{G,\alpha}$ is δ -close to f .

1.5 Our results and Organization of this thesis

Chapter 2 - Strongly Black-Box Hardness Amplification. We show that hardness amplification from hardness $(1 - \delta)/2$ to hardness $(1 - \delta^k)/2$ cannot be carried out in some black-box way by a circuit of depth d and size $2^{o(k^{1/d})}$ or by a nondeterministic circuit of size $o(k/\log k)$ (and arbitrary depth). In particular, for $k = 2^{\Omega(n)}$, such hardness amplification cannot be

done by a strongly black-box model in $\text{ATIME}(O(1), 2^{o(n)})$. Therefore, hardness amplification in general requires a high complexity. Furthermore, we show that even without any restriction on the complexity of the amplification procedure, such a strongly black-box hardness amplification must be inherently non-uniform in the following sense. Given as an oracle any algorithm which agrees with f' on $(1 - \delta^k)/2$ fraction of the input, we still need an additional advice of length $\Omega(k \log(1/\delta))$ in order to compute f correctly on $(1 - \delta)/2$ fraction of the input. Therefore, to guarantee the hardness, even against uniform machines, of the function f' , one has to start with a function f which is hard against non-uniform circuits. Finally, we derive similar lower bounds for any strongly black-box construction of pseudorandom generators from hard functions.

Chapter 3 - Weakly Black-Box Hardness Amplification. From worst-case hardness to average-case hardness, we consider a class of hardness amplifications called weakly black-box hardness amplification, in which the initial hard function is only used as a black box to construct the harder function. First, we show that if an amplification procedure in $\text{TIME}(t)$ can amplify hardness beyond an $O(t)$ factor, then it must embed in itself a hard function computable in $\text{TIME}(t)$. As a result, it is impossible to have such a hardness amplification with hardness measured against $\text{TIME}(t)$. Next, we show that, for any $k \in \mathbb{N}$, if an amplification procedure in $\Sigma_k\text{P}$ can amplify hardness beyond a polynomial factor, then one can obtain from it a hard function in $\Sigma_k\text{P}$. A similar impossibility result can also be derived.

Chapter 4 - Hardness Amplification within NP. We study the problem of hardness amplification in NP . We prove that if there is a balanced function in NP such that any circuit of size $s(n) = 2^{\Omega(n)}$ fails to compute it on a $1/\text{poly}(n)$ fraction of inputs, then there is a function in NP such that any circuit of size $s'(n)$ fails to compute it on a $1/2 - 1/s'(n)$ fraction of inputs, with $s'(n) = 2^{\Omega(n^{2/3})}$. This improves the result of Healy et al. (STOC'04), which only achieves $s'(n) = 2^{\Omega(n^{1/2})}$ for the case with $s(n) = 2^{\Omega(n)}$.

Chapter 5 - Pseudorandomness and Hardness in NP. To build the equivalence between pseudorandomness and average-case hardness, we show how to transform a pseudorandom generator into a mildly hard function computable in NP. We give a strongly black-box construction, with both the transformation procedure and the hardness proof done in a black-box way. This improves a previous result of Nisan and Wigderson, which can only obtain a worst-case hard function from a pseudorandom generator [NW94]. Therefore, we now know that the transformations among mild hardness, average-case hardness, and pseudorandomness all can be done in the complexity class NP.

Chapter 6 - Hard-core Set Construction. We study a fundamental result of Impagliazzo (*FOCS'95*) known as the hard-core set lemma. Consider any function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ which is “mildly-hard”, in the sense that any circuit of size s must disagree with f on some δ fraction of inputs. Then the hard-core lemma says that f must have a hard-core set H of density δ on which it is “extremely hard”, in the sense that any circuit of size $s' = O(s/(\frac{1}{\varepsilon^2} \log(\frac{1}{\varepsilon\delta})))$ must disagree with f on at least $(1 - \varepsilon)/2$ fraction of inputs from H .

There are three issues of the lemma which we would like to address: the loss of circuit size, the need of non-uniformity, and its inapplicability to a low complexity class. We introduce two models of hard-core set constructions, a strongly black-box one and a weakly black-box one, and show that those issues are unavoidable in such models.

First, we show that in any strongly black-box construction, one can only prove the hardness of a hard-core set for smaller circuits of size at most $s' = O(s/(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}))$. Next, we show that any weakly black-box construction must be inherently non-uniform — to have a hard-core set for a class G of functions, we need to start from the assumption that f is hard against a non-uniform complexity class with $\Omega(\frac{1}{\varepsilon} \log |G|)$ bits of advice. Finally, we show that weakly black-box constructions in general cannot be realized in a

low-level complexity class such as $\text{AC}^0[p]$ — the assumption that f is hard for $\text{AC}^0[p]$ is not sufficient to guarantee the existence of a hard-core set.

1.6 Notations and Useful Facts

For any $n \in \mathbb{N}$, let $[n]$ denote the set $\{1, 2, \dots, n\}$ and let \mathcal{U}_n denote the uniform distribution over the set $\{0, 1\}^n$. When we sample from a finite set, the default distribution is the uniform one. For a string z , let z_i denote the i 'th bit of z . All the logarithms in this thesis will have base two. Define the binary entropy function $H(x) = -x \log x - (1 - x) \log (1 - x)$. For a finite set S , we also use S to denote the uniform distribution over S . For $q \in \mathbb{N}$, we identify the set $\{0, 1\}^q$ with $[2^q]$. For a set R , we also use R to denote its membership function. We will sometimes view a Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ as a 2^n -bit string (its truth table) and vice versa. For two strings $u, v \in \{0, 1\}^n$, let $\Delta(u, v)$ denote their relative Hamming distance $\frac{1}{n} |\{i \in [n] : u_i \neq v_i\}|$.

We need some standard complexity classes. Let $\text{ATIME}(d, t)$ denote the class of functions computed by alternating Turing machines in time t with at most d alternations, and let $\text{ATIME}(t)$ denote $\text{ATIME}(t, t)$. Let PH denote the polynomial-time hierarchy, which is $\text{ATIME}(O(1), \text{poly}(n))$. Let $\text{NTIME}(t)$ denote the class of functions computed by nondeterministic Turing machines in time t . The circuits we consider here consist of AND/OR/NOT gates, allowing unbounded fan-in for AND/OR gates. The size of a circuit is the number of non-input gates it has and the depth of circuit is the number of gates on the longest path from an input bit to the output gate. We call such circuits AC circuits.

Definition 1 Let $\text{AC}(d, s)$ ($\text{SIZE}(s)$, resp.) denote the class of functions computed by AC circuits of depth d and size s (of size s , resp.).

Note that the standard complexity class AC^0 corresponds to our class $\text{AC}(O(1), \text{poly}(n))$. We also introduce the nondeterministic version of AC

circuits. An NAC circuit C has two parts of inputs: the real input x and the witness input y . The Boolean function f computed by such a circuit C is defined as $f(x) = 1$ if and only if there exists a witness y such that $C(x, y) = 1$.

Definition 2 Let $\text{NAC}(s)$ be the class of functions computed by NAC circuits of size s .

A function with more than one output bits is said to be computed by some type of circuits (e.g. $\text{AC}(d, s)$ or $\text{NAC}(s)$) if each output bit can be computed by one such circuit. More definitions and details of complexity classes can be found in standard textbooks, such as [Pap94]. As usual in complexity theory, when we talk about a function $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$, we actually mean a sequence of functions $(f : \{0, 1\}^n \rightarrow \{0, 1\}^{m(n)})_{n \in \mathbb{N}}$, and when we make a statement about f , we usually mean that it holds for any sufficiently large $n \in \mathbb{N}$.

We say that a function $G : \{0, 1\}^\ell \rightarrow \{0, 1\}^m$ is *explicitly computable* if given $x \in \{0, 1\}^\ell$ and $i \in [m]$, the i 'th bit of $G(x)$ can be computed in time $\text{poly}(\ell, \log m)$. A function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is called *balanced* if $\Pr[f(U_n) = 1] = 1/2$. For a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, let $f^{\otimes k} : \{0, 1\}^{kn} \rightarrow \{0, 1\}^k$ be the function defined by

$$f^{\otimes k}(x_1, \dots, x_k) = (f(x_1), \dots, f(x_k)),$$

for $x_1, \dots, x_k \in \{0, 1\}^n$.

1.6.1 Tail Bounds of Binomial Distribution

We will frequently use the following simple lower bound on the tail probability of binomial distribution, a proof of which is given in the following.

Fact 1 Let Z_1, \dots, Z_t be i.i.d. binary random variables, with $\mathbb{E}[Z_i] = \mu$ for every $i \in [t]$. Suppose $\varepsilon < \frac{1}{5}$ and $t = \Omega(\frac{1}{\varepsilon^2})$. Then we have the following: (1) if $\mu \leq \frac{1+2\varepsilon}{2}$, then $\Pr[\sum_{i \in [t]} Z_i \leq \frac{1-\varepsilon}{2}t] \geq 2^{-O(\varepsilon^2 t)}$, and (2) if $\mu \geq \frac{1-2\varepsilon}{2}$, then $\Pr[\sum_{i \in [t]} Z_i \geq \frac{1+\varepsilon}{2}t] \geq 2^{-O(\varepsilon^2 t)}$.

Proof. First, consider the case that $\mu \leq \frac{1+2\varepsilon}{2}$. Note that the probability gets smaller if μ gets larger, because $\sum_{i=1}^t Z_i$ becomes more unlikely to have a small value when the probability of $Z_i = 1$ becomes higher. Thus, it suffices to show the lower bound for the case of $\mu = \frac{1+2\varepsilon}{2}$. Then,

$$\begin{aligned} \Pr \left[\sum_{i=1}^t Z_i \leq \frac{1-\varepsilon}{2}t \right] &= \sum_{0 \leq j \leq \frac{1-\varepsilon}{2}t} \binom{t}{j} \cdot \left(\frac{1+2\varepsilon}{2} \right)^j \left(\frac{1-2\varepsilon}{2} \right)^{t-j} \\ &\geq \sum_{\frac{1-2\varepsilon}{2}t \leq j \leq \frac{1-\varepsilon}{2}t} \binom{t}{j} \cdot \left(\frac{1+2\varepsilon}{2} \right)^j \left(\frac{1-2\varepsilon}{2} \right)^{t-j} \\ &\geq \frac{\varepsilon t}{2} \cdot \binom{t}{\frac{1-2\varepsilon}{2}t} \cdot \left(\frac{1+2\varepsilon}{2} \right)^{\frac{1-2\varepsilon}{2}t} \left(\frac{1-2\varepsilon}{2} \right)^{\frac{1+2\varepsilon}{2}t}. \end{aligned}$$

Using the inequality that $\binom{t}{\alpha t} \geq \frac{1}{O(\sqrt{t})} \left(\frac{1}{\alpha} \right)^{\alpha t} \left(\frac{1}{1-\alpha} \right)^{(1-\alpha)t}$ from Stirling's formula, the above becomes

$$\frac{\varepsilon t}{O(\sqrt{t})} \left(\frac{2}{1-2\varepsilon} \right)^{\frac{1-2\varepsilon}{2}t} \left(\frac{2}{1+2\varepsilon} \right)^{\frac{1+2\varepsilon}{2}t} \left(\frac{1+2\varepsilon}{2} \right)^{\frac{1-2\varepsilon}{2}t} \left(\frac{1-2\varepsilon}{2} \right)^{\frac{1+2\varepsilon}{2}t},$$

which is at least

$$\Omega(\varepsilon\sqrt{t}) \left(\frac{1-2\varepsilon}{1+2\varepsilon} \right)^{2\varepsilon t} = \Omega(\varepsilon\sqrt{t}) \left(1 - \frac{4\varepsilon}{1+2\varepsilon} \right)^{2\varepsilon t} \geq \Omega(\varepsilon\sqrt{t}) 2^{-O(\varepsilon^2 t)} \geq 2^{-O(\varepsilon^2 t)},$$

where the first inequality uses the fact that $1-x \geq 2^{-cx}$ for some constant c when $x (= \frac{4\varepsilon}{1+2\varepsilon}) \leq \frac{4}{5}$, and the last inequality follows from the condition that $t = \Omega(\frac{1}{\varepsilon^2})$.

The second case with $\mu \geq \frac{1-2\varepsilon}{2}$ follows immediately from the first case by symmetry. More precisely, define new random variables Y_1, \dots, Y_t , with $Y_i = 1 - Z_i$ for $i \in [t]$, and then we can get the desired bound by applying the bound of the first case to these new variables. \square

Chapter 2

Strongly Black-Box Hardness Amplification

2.1 Introduction

For $\delta \in (0, 1)$ and $k, n \in \mathbb{N}$, we study the task of transforming a hard function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, with which any small circuit disagrees on $(1 - \delta)/2$ fraction of the input, into a harder function f' , with which any small circuit disagrees on $(1 - \delta^k)/2$ fraction of the input. In this chapter, we show that this process cannot be realized in parallel black-box way (defined later) by a circuit of depth d and size $2^{o(k^{1/d})}$ or by a nondeterministic circuit of size $o(k/\log k)$ (and arbitrary depth). Therefore, such hardness amplification in general requires a high complexity. Furthermore, we show that even without any restriction on the complexity of the amplification procedure, such a strongly black-box hardness amplification must be inherently non-uniform in the following sense. To guarantee the hardness of the resulting function f' , even against uniform machines, one has to start with a function f which is hard against non-uniform algorithms with $\Omega(k \log(1/\delta))$ bits of advice. Finally, we derive similar lower bounds for any strongly black-box construction of a pseudorandom generator (PRG) from a hard function. To prove our results, we link the task of hardness amplifications and PRG constructions,

respectively, to some type of error-reduction codes, and then we establish lower bounds for such codes, which we hope could find interest in both coding theory and complexity theory.

2.1.1 Previous Lower Bound Results

Viola [Vio04] gave the first lower bound on the complexity required for strongly black-box hardness amplification. He showed that to transform a worst-case hard function f into a mildly hard function f' , both against circuits of size $2^{o(n)}$, the encoding function AMP cannot be realized in the complexity class $\text{ATIME}(O(1), 2^{o(n)})$. This rules out the possibility of doing such hardness amplification in PH , which explains why previous procedures all require a high computational complexity. He also showed a similar lower bound for strongly black-box construction of PRG from a worst-case hard function.

Trevisan and Vadhan [TV02] observed that a strongly black-box hardness amplification from worst-case hardness corresponds to an error-correcting code with some list-decoding property. Then results from coding theory can be used to show that for such amplification from worst-case hardness to hardness $(1 - \varepsilon)/2$, the decoding function DEC must need $\Omega(\log(1/\varepsilon))$ bits of advice in order to compute f . This explains why almost all previous hardness amplification results were done in a non-uniform setting, except [IW98, TV02] which did not work in a black-box way.

There were also impossibility results on weaker types of hardness amplification, from worst-case hardness to average-case hardness. Bogdanov and Trevisan [BT03] considered hardness amplification for functions in NP in which the black-box requirement on the *encoding* function is dropped. They showed that the decoding function cannot be computed non-adaptively in polynomial time unless PH collapses. Viola, in another recent paper [Vio05], considered hardness amplification in which the black-box requirement on the *decoding* function is dropped. He showed that if the encoding function can be computed in PH , then there exists an average-case hard function in PH

unconditionally. We will not consider such weaker types of hardness amplification in this chapter, and throughout this chapter when we refer to hardness amplification, we always mean the strongly black-box one.

2.1.2 Our Results

Previous lower bound results only address hardness in a specific range. However, whether or not one can amplify hardness beyond this range is also a natural and interesting question. For example, it is known that a strongly black-box hardness amplification from hardness $1/\text{poly}(n)$ to average-case hardness can be realized in polynomial time [Yao82, GNW95, Im95, IW97]. Can such a hardness amplification be realized in a lower complexity class, such as AC^0 ? Can it start from hardness below $1/\text{poly}(n)$ and still be realized in polynomial time? Can it be done in a uniform way (with a uniform decoding function)? In general, how does the quality of a hardness amplification (the amount of hardness increased) determine its inherent complexity or non-uniformity? All these questions will be addressed in this chapter. We generalize previous results [Vio04, TV02] and consider hardness amplification in a much broader spectrum: from hardness $(1-\delta)/2$ to hardness $(1-\delta^k)/2$, for general $\delta \in (0, 1)$ and $k \in \mathbb{N}$.

Following [Vio05], we consider a more restricted model called *parallel* black-box hardness amplification, in which oracle queries by the encoding function are done in a non-adaptive way. More precisely, we say that a circuit class \mathcal{CKT} realizes a parallel black-box hardness amplification if its encoding function AMP can be implemented in the following way. Given any input x , it first generates a circuit $T_x \in \mathcal{CKT}$ together with t query inputs $q_{x,1}, \dots, q_{x,t}$, then queries f at those t inputs, and finally computes $T_x(f(q_{x,1}), \dots, f(q_{x,t}))$ as its output. Note that here T_x and $q_{x,1}, \dots, q_{x,t}$ only depend on x but not f . Although this is a more restricted model, almost all previous constructions of hardness amplification can be done in this way, so it would be nice to know its limitation. Furthermore, through a standard simulation [FSS84, Has86], negative results in this model can in

fact be translated to those in the strongly black-box model.

Our first result addresses both the complexity issue and the non-uniformity issue in the same framework, showing how complexity constraints on the encoding function result in the inherent non-uniformity of the decoding function. Formally, we prove that if such a parallel black-box hardness amplification, from hardness $(1 - \delta)/2$ to hardness $(1 - \delta^k)/2$, is realized by circuits of depth d and size $2^{o(k^{1/d})}$, then the decoding function DEC must need an advice of length $2^{\Omega(n)}$. Translating this to the general model, we obtain the same advice lower bound when such a (general) strongly black-box hardness amplification is realized in $\text{ATIME}(O(1), k^{o(1)})$. This implies that no such hardness amplification is possible if the hardness is measured against circuits of size $2^{o(n)}$.

Our lower bound is almost tight as the well known XOR lemma [Yao82, GNW95] gives a way to realize a parallel black-box hardness amplification by circuits of depth $O(d)$ and size $2^{O(k^{1/d})}$, with DEC using an advice of length $\text{poly}(n/\delta^k)$. Note that Viola's result in [Vio04] is a special case of ours, because he only addressed explicitly the specific case with $(1 - \delta)/2 = 2^{-n}$ and $(1 - \delta^k)/2 = 1/\text{poly}(n)$ (or equivalently, $\delta = 1 - 2^{-n+1}$ and $k = 2^{\Omega(n)}$). Although it seems that his technique can be extended to show lower bounds when $(1 - \delta)/2$ is small enough, but beyond that, say with $(1 - \delta)/2 = \Omega(1)$, it fails to give a meaningful bound. We can in fact cover this case: our result implies that AC^0 circuits cannot realize a parallel black-box hardness amplification, say, from hardness $1/3$ to hardness $(1 - 2^{-\Omega(n)})/2$. On the other hand, our result when restricted to worst-case to average-case hardness amplification is incomparable to those of [BT03] and [Vio05].¹ Finally, two interesting facts follow from our result. First, it is impossible to produce in a strongly black-box way a function which is $(1 - \delta^k)/2$ -hard against a uniform low

¹In [BT03], the complexity lower bound is given on the decoding function instead, under the unproven (though widely believed) assumption that PH does not collapse. In [Vio05], a more general type of hardness amplification than ours is considered, but the possibility of such hardness amplification is not ruled out as we do; instead, it was shown that if the encoding function can be computed in PH, a hard function in PH exists unconditionally.

complexity class, say $\text{DTIME}(O(1))$, even if we start from a function which is $(1 - \delta)/2$ -hard against a uniform but arbitrarily high complexity class equipped with an advice of length $2^{o(n)}$, say $\text{DTIME}(2^{2^n})/2^{o(n)}$. On the other hand, it is easy to show that hard functions against $\text{DTIME}(O(1))$ do exist.² This demonstrates one severe weakness of strongly black-box hardness amplifications. Second, when amplifying hardness from $(1 - \delta)/2$ to $(1 - \delta^k)/2$, the complexity of such amplification is determined mainly by the parameter k ; a larger value of k results in a higher complexity requirement, for typical values of δ . Thus, to determine the complexity needed for a hardness amplification process, one should express the initial and final hardness in the forms of $(1 - \delta)/2$ and $(1 - \delta^k)/2$ respectively. This point was not clear from previous works.

Note that our first result becomes meaningless for $d = \Omega(\log k)$ as the circuit size becomes $2^{o(k^{1/d})} = O(1)$. Our second result takes care of this: we show that if a parallel black-box hardness amplification, from hardness $(1 - \delta)/2$ to hardness $(1 - \delta^k)/2$, is realized by nondeterministic circuits of size $o(k/\log k)$, even with arbitrary depth, then the decoding function DEC must need an advice of length $2^{\Omega(n)}$. For example, to amplify hardness from $\Omega(1)$ to $(1 - 2^{-\Omega(n)})/2$, our second result implies that it can not be realized by nondeterministic circuits of size $o(n/\log n)$ in a parallel black-box way.

Our third result shows that even without any complexity constraint on the encoding or decoding function, amplification between certain range of hardness is still inherently non-uniform. For the special case of amplifying hardness beyond $1/4$, the need of non-uniformity can be shown using the Plotkin bound [Pl06] from coding theory. We consider hardness amplification in a general range and obtain a quantitative bound on the amount of non-uniformity. More precisely, we show that to amplify hardness from $(1 - \delta)/2$ to $(1 - \varepsilon)/2$, the decoding function DEC must need an advice of

²For example, the parity function is $(1/2 - 2^{-\Omega(n)})$ -hard against $\text{DTIME}(O(1))$. However, according to our result, its hardness cannot be shown in such a strongly black-box way.

$\Omega(\log(\delta^2/\varepsilon))$ bits. Thus, when $\varepsilon = \delta^k$, an advice of length $\Omega(k \log(1/\delta))$ is necessary, and when $\varepsilon \leq c\delta^2$ for some constant c , such hardness amplification must be inherently non-uniform. Our result generalizes that of Trevisan and Vadhan [TV02].

Finally, we derive similar lower bounds on strongly black-box constructions of PRG from hard functions.

2.1.3 Our Techniques

Our results are obtained via a connection between strongly black-box hardness amplifications and some type of “error-reduction” codes, which generalizes the connection given by Trevisan and Vadhan [TV02] and Viola [Vio04]. A similar observation was also made by Trevisan [Tre03]. Formally, a strongly black-box amplification from hardness $(1-\delta)/2$ to hardness $(1-\varepsilon)/2$ induces a code with the following list-decoding property. Given a corrupted codeword with a fraction of less than $(1-\varepsilon)/2$ errors, we can always find a small list of candidate messages such that one of them is close to the original message, with their relative Hamming distance less than $(1-\delta)/2$. Therefore, we can focus our attention on such codes, as results on such codes immediately give results on corresponding hardness amplifications.

Our first two results are based on the following idea. A code with such a list-decoding property can only have a small number of codewords close to any codeword, so a random perturbation on an input message is unlikely to result in a close codeword. On the other hand, if such a code is computed by an algorithm which is insensitive to noise on the input, then a random perturbation on an input message is likely to result in a close codeword, and we reach a contradiction. Circuits of small size, or circuits of small depth and moderate size can be shown to be insensitive to noise on their input. Thus, they cannot be used to compute such a code and the corresponding hardness amplification. This basically follows Viola’s idea in [Vio04], but since we consider hardness amplification in a much broader spectrum, a more involved analysis is required. For example, since Viola only considered the

case with a small hardness, he only had to deal with noise of a small rate. With such a small noise rate, the output value will only be affected with a small probability, and small loss in his analysis does not matter too much. However, if a large hardness is considered, a high noise rate is needed, then the loss in his analysis will become intolerable, and his bound will become meaningless (see Remark 4 in Section 2.2.3 for details). To overcome this problem, we derive another upper bound on noise sensitivity, which works for any noise rate and thus can be used for hardness in a general range.

For the non-uniformity of hardness amplification, we show that given a corrupted codeword with a high fraction $(1 - \varepsilon)/2$ (for a small ε) of errors, one may need a long list of candidate messages in order to have one of them within a small relative distance $(1 - \delta)/2$ (for a large δ) to the original message. To show this, we would like to find a set of messages such that some ball of relative radius $(1 - \varepsilon)/2$ in the codeword space contains many of their corresponding codewords, but any ball of relative radius $(1 - \delta)/2$ in the message space contains only a small number of messages from that set. We choose these messages randomly and show that they have some chance of satisfying the condition above when $(1 - \varepsilon)/2$ is larger than $(1 - \delta)/2$ to some extent.

Finally, to prove lower bounds for strongly black-box constructions of PRG from hard functions, we discover that there is also a connection between the error-reduction codes we just considered and such PRG constructions. This new connection may have interest of its own. Then the results we obtain for such codes immediately yield results for such PRG constructions.

2.1.4 Organization of this chapter

First, some preliminaries are given in Section 2.2. Then in Section 2.3 and Section 2.4, we prove the impossibility results of hardness amplification by constant-depth circuits and non-deterministic circuits respectively. In Section 2.5, we show that hardness amplification in general is inherently non-uniform. Finally, we show the impossibility results for strongly black-box

PRG constructions from hard functions in Section 2.6.

2.2 Preliminaries

2.2.1 Strongly Black-Box Hardness Amplification and Pseudorandom Generators

Informally speaking, a function is hard if any algorithm without enough complexity must make some mistakes. Formally, we define the hardness of a function as follows.

Definition 3 *We say that a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ has hardness β against circuits of size s if for any circuit $C : \{0, 1\}^n \rightarrow \{0, 1\}$ of size s ,*

$$\Pr_{x \in U_n} [f(x) \neq C(x)] \geq \beta.$$

Note that we use the error bound β to characterize the hardness of a function, and we pay less (sometimes no) attention to the size bound s . For hardness amplification, we want to transform a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ with a smaller hardness β into a function $f' : \{0, 1\}^m \rightarrow \{0, 1\}$ with a larger hardness β' . We will focus on a special type of hardness amplification called strongly black-box hardness amplification, defined next, which consists of two oracle procedures AMP and DEC. We allow DEC to be a non-uniform oracle Turing machine, and we write $\text{DEC}^{A, \nu}$ to denote DEC taking an oracle A and an advice string ν .

Definition 4 *A strongly black-box (n, β, β', ℓ) hardness amplification consists of an oracle procedure $\text{AMP}^{(\cdot)} : \{0, 1\}^m \rightarrow \{0, 1\}$ (called encoding function) and a non-uniform oracle Turing machine $\text{DEC}^{(\cdot)} : \{0, 1\}^n \rightarrow \{0, 1\}$ (called decoding function) with the following property. For any $f : \{0, 1\}^n \rightarrow \{0, 1\}$, if a function $A : \{0, 1\}^m \rightarrow \{0, 1\}$ satisfies*

$$\Pr_{z \in U_m} [A(z) \neq \text{AMP}^f(z)] < \beta',$$

then there exists an advice string $\nu = \nu(f, A) \in \{0, 1\}^\ell$ such that

$$\Pr_{x \in U_n} [\text{DEC}^{A, \nu}(x) \neq f(x)] < \beta.$$

For a complexity class \mathcal{C} , we say that the strongly black-box hardness amplification can be realized in \mathcal{C} if for any oracle f , the procedure AMP^f can be computed in \mathcal{C}^f .

Here, the transformation of the initial function f into a harder function is done in a black-box way, as the harder function AMP^f only uses f as an oracle. Moreover, the hardness of the new function AMP^f is also guaranteed in a black-box way. Namely, any algorithm A breaking the hardness condition of AMP^f can be used as an oracle for a machine DEC to break the hardness condition of f . Note that neither of the hardness refers to circuit size, and no constraint is placed on the complexity of the procedure DEC . This freedom makes our impossibility results stronger. The parameter ℓ characterizes the amount of non-uniformity associated with this process. When $\ell \geq 1$, we say the hardness amplification is non-uniform.

Remark 1 One can also use the notion of “advantage” to characterize the hardness of a Boolean function. We say that any circuit of size s has advantage at most δ for computing f if for any such a circuit C , $\Pr_x [f(x) = C(x)] - \Pr_x [f(x) \neq C(x)] \leq \delta$. Clearly, the advantage δ is related to the hardness β in the form $\beta = \frac{1-\delta}{2}$. We will focus on the task of amplifying hardness from $\frac{1-\delta}{2}$ to $\frac{1-\delta^k}{2}$, or equivalently, reducing the advantage from δ to δ^k . We choose to present our results in terms of hardness instead of advantage for the following two reasons. First, when talking about hardness amplification, it seems more natural and less confusing to use hardness instead of advantage. Secondly, as we will see, there is some nice connection between hardness amplifications and error-correcting codes, in which hardness of functions corresponds naturally to distance in codes. However, the drawback of using hardness instead of advantage is that our notation sometimes looks more cumbersome.

Similarly, we can define the notion of strongly black-box construction of pseudo-random generators from hard functions.

Definition 5 A strongly black-box $(n, \beta, \varepsilon, \ell)$ PRG construction consists of an oracle procedure $G^{(\cdot)} : \{0, 1\}^m \rightarrow \{0, 1\}^r$ (called encoding function) and a non-uniform oracle Turing machine $\text{DEC}^{(\cdot)} : \{0, 1\}^n \rightarrow \{0, 1\}$ (called decoding function) with the following property. For any $f : \{0, 1\}^n \rightarrow \{0, 1\}$, if a function $D : \{0, 1\}^r \rightarrow \{0, 1\}$ satisfies

$$\left| \Pr_{u \in U_m} [D(G^f(u)) = 1] - \Pr_{w \in U_r} [D(w) = 1] \right| > \varepsilon,$$

then there exists an advice string $\nu = \nu(f, D) \in \{0, 1\}^\ell$ such that

$$\Pr_{x \in U_n} [\text{DEC}^{D, \nu}(x) \neq f(x)] < \beta.$$

For a complexity class \mathcal{C} , we say that the strongly black-box PRG construction can be realized in \mathcal{C} if for any oracle f , the procedure G^f can be computed in \mathcal{C}^f .

Remark 2 When talking about a strongly black-box hardness amplification or PRG construction, we usually mean a sequence of them, parameterized by the parameter $n \in \mathbb{N}$. Other parameters such as $m, \beta, \beta', \ell, r, \varepsilon, k$ are in fact allowed to be functions of n .

In this general model of strongly black-box hardness amplification or PRG construction, we do not put any restriction on how the oracle f is queried by the encoding function (AMP or G). On the other hand, we will also consider the following more restricted model, first introduced in [Vio05], in which the oracle f can only be queried in a non-adaptive way. We call such model a *parallel* black-box hardness amplification or PRG construction. More precisely, we define the following.

Definition 6 Let \mathcal{CKT} be a class of circuits, such as $\text{AC}(d, s)$ or $\text{NAC}(s)$. We say that \mathcal{CKT} realizes a parallel black-box hardness amplification, if we

have a black-box hardness amplification in which the encoding function $\text{AMP}^{(\cdot)}$ can be implemented in the following way. Given any oracle $f : \{0,1\}^n \rightarrow \{0,1\}$ and any input $x \in \{0,1\}^m$, it first generates a circuit $T_x \in \mathcal{CKT}$ together with t query inputs $q_{x,1}, \dots, q_{x,t} \in \{0,1\}^n$, then queries f at those t inputs, and finally outputs $T_x(f(q_{x,1}), \dots, f(q_{x,t}))$. The case of parallel black-box PRG construction is defined similarly.

Note that T_x and $q_{x,1}, \dots, q_{x,t}$ are produced before the oracle f is actually queried, so they depend on x but not on the oracle f . This restriction makes it easier to obtain negative (or lower bound) results in such a parallel model. Nevertheless, the following lemma provides a way to translate such results to those in the general strongly black-box model.

Lemma 1 *If a strongly black-box (n, β, β', ℓ) hardness amplification (PRG construction, resp.) can be realized in $\text{ATIME}(d, t)$, then a parallel black-box (n, β, β', ℓ) hardness amplification (PRG construction, resp.) can be realized in $\text{AC}(d + O(1), 2^{O(t)})$.*

Proof. Consider any strongly black-box hardness amplification (the case of PRG construction is similar) with the encoding function AMP such that for any oracle f , AMP^f belongs to $\text{ATIME}^f(d, t)$. It is known from [FSS84, Has86] that by adding a constant number of alternations, one can transform AMP into another procedure AMP' which only queries f once in each branch of its computation. Then by a standard simulation of alternating Turing machines by circuits [FSS84, Has86], we know that for any input x , the value of $\text{AMP}'^f(x)$ can be computed by a circuit in $\text{AC}(d + O(1), 2^{O(t)})$ with the answers to the corresponding oracle queries given as part of the input. Note that the circuit and the oracle queries depend only on the input x but not the oracle f . Thus we have a parallel black-box hardness amplification realized in $\text{AC}(d + O(1), 2^{O(t)})$. \square

2.2.2 Codes and Correspondence to Hardness Amplification

We measure the distance between two strings by their relative Hamming distance.

Definition 7 For $u, v \in \{0, 1\}^M$, define their distance $\Delta(u, v)$ as their relative Hamming distance, namely $\Delta(u, v) = \frac{1}{M} |\{i \in [M] : u_i \neq v_i\}|$.

According to this distance, we define open balls of radius β in the space $\{0, 1\}^N$.

Definition 8 For any $N \in \mathbb{N}$, $\beta \in (0, 1)$, and $x \in \{0, 1\}^N$, let $\text{BALL}_x(\beta, N) = \{x' \in \{0, 1\}^N : \Delta(x, x') < \beta\}$, which is the open ball in $\{0, 1\}^N$ of radius β centered at x . Let $\text{BALL}(\beta, N)$ denote the set consisting of all such balls.

The following simple fact gives an upper bound on the size of such a Hamming ball.

Fact 2 The size of any ball in $\text{BALL}(\beta, N)$ is at most $2^{H(\beta)N}$.

We borrow the notion of list-decodable codes, but we extend it in a way that leads to some natural correspondence with strongly black-box hardness amplifications.

Definition 9 We call $C : \{0, 1\}^N \rightarrow \{0, 1\}^M$ a (β, β', L) -list code if for any $z \in \{0, 1\}^M$, there are L balls from $\text{BALL}(\beta, N)$ such that if a codeword $C(x)$ is contained in $\text{BALL}_z(\beta', M)$, then x is contained in one of those L balls.

A (β, β', L) -list code is related to a standard list-decodable code in the way that each ball in $\text{BALL}(\beta', M)$ contains at most $L \cdot 2^{H(\beta)N}$ codewords. Next, we show how such a code arises naturally from a strongly black-box hardness amplification. Let $N = 2^n$ and $M = 2^m$. Given any oracle algorithm $\text{AMP}^{(\cdot)} : \{0, 1\}^m \rightarrow \{0, 1\}$, let us define the corresponding code $C : \{0, 1\}^N \rightarrow \{0, 1\}^M$ as $C(f) = \text{AMP}^f$. That is, seeing any function $f : \{0, 1\}^n \rightarrow \{0, 1\}$

as a vector in $\{0, 1\}^N$, $C(f)$ produces as output the function AMP^f , which is seen as a vector in $\{0, 1\}^M$. The following is a simple generalization of an observation by Viola [Vio04].

Lemma 2 *Let $\text{AMP}^{(\cdot)} : \{0, 1\}^m \rightarrow \{0, 1\}$ be the encoding function of a strongly black-box (n, β, β', ℓ) hardness amplification. Then $C : \{0, 1\}^N \rightarrow \{0, 1\}^M$, defined as $C(f) = \text{AMP}^f$, is a $(\beta, \beta', 2^\ell)$ -list code.*

Proof. Let AMP be the encoding function of a strongly black-box (n, β, β', ℓ) hardness amplification, and let DEC be the corresponding decoding function which is an oracle Turing machine with an ℓ -bit advice. Consider any $A \in \{0, 1\}^M$, seen as $A : \{0, 1\}^m \rightarrow \{0, 1\}$. For any codeword $C(f)$ with $\Delta(A, C(f)) = \Pr_z[A(z) \neq \text{AMP}^f(z)] < \beta'$, by Definition 4, there exists an $\nu \in \{0, 1\}^\ell$ such that $\Delta(\text{DEC}^{A, \nu}, f) = \Pr_x[\text{DEC}^{A, \nu}(x) \neq f(x)] < \beta$. That is, if $C(f)$ is in $\text{BALL}_A(\beta', M)$, then f is contained in one of the 2^ℓ balls of radius β centered at $\text{DEC}^{A, \nu}$ for $\nu \in \{0, 1\}^\ell$. Therefore, C is a $(\beta, \beta', 2^\ell)$ -list code. \square

Remark 3 *Note that if a circuit class CKT can realize a parallel hardness amplification, then every output bit of the corresponding code C can be computed by a circuit in CKT . This is because for any input $f \in \{0, 1\}^N$, the x -th output bit of $C(f)$ equals $\text{AMP}^f(x) = T_x(f(q_{x,1}), \dots, f(q_{x,t}))$, which is computed by some circuit $T_x \in \text{CKT}$ on some t bits of f .*

In Section 2.6, we will show that there also exists a natural correspondence between strongly black-box PRG constructions and such list-decodable codes.

2.2.3 Noise Sensitivity

Following [OD02, Vio04], we will apply Fourier analysis on Boolean functions. For any $g : \{0, 1\}^N \rightarrow \{0, 1\}$ and for any $J \subseteq [N]$, let $\hat{g}(J) = \mathbb{E}_y [(-1)^{g(y)} \cdot \prod_{i \in J} (-1)^{y_i}]$. Here is a well-known fact.

Fact 3 *For any $g : \{0, 1\}^N \rightarrow \{0, 1\}$, $\sum_{J \subseteq [N]} \hat{g}(J)^2 = 1$.*

It is known that for AC circuits of small depths, the main contribution to the above sum comes from the low-order terms.

Lemma 3 [LMN93] For any $g : \{0, 1\}^N \rightarrow \{0, 1\} \in \text{AC}(d, s)$ and for any $t \in [N]$, $\sum_{|J|>t} \hat{g}(J)^2 \leq s \cdot 2^{-\Omega(t^{1/d})}$.

This can be used to show that AC circuits of small depth are insensitive to noise on their input. We will need the following more precise relation between the noise sensitivity of a Boolean function and its Fourier coefficients.

Lemma 4 Suppose x is sampled from the uniform distribution over $\{0, 1\}^N$ and \tilde{x} is obtained by flipping each bit of x independently with probability $\frac{1-\alpha}{2}$. Then for any $g : \{0, 1\}^N \rightarrow \{0, 1\}$ and for any $t \in [N]$, $\Pr_{x, \tilde{x}}[g(x) \neq g(\tilde{x})] \leq \frac{1}{2}(1 - \alpha^t(1 - \sum_{|J|>t} \hat{g}(J)^2))$.

Proof. We know from [OD02] (Proposition 9) that $\Pr_{x, \tilde{x}}[g(x) \neq g(\tilde{x})] = \frac{1}{2}(1 - \sum_{J \subseteq [N]} \alpha^{|J|} \hat{g}(J)^2)$. Note that

$$\sum_{J \subseteq [N]} \alpha^{|J|} \hat{g}(J)^2 \geq \sum_{|J| \leq t} \alpha^{|J|} \hat{g}(J)^2 \geq \alpha^t \sum_{|J| \leq t} \hat{g}(J)^2.$$

Then the lemma follows from Fact 3. \square

Combing Lemma 3 and Lemma 4, we immediately have the following.

Corollary 1 Suppose x and \tilde{x} are sampled as in Lemma 4. Then for any $g : \{0, 1\}^N \rightarrow \{0, 1\} \in \text{AC}(d, s)$ and for any $t \in [N]$, $\Pr_{x, \tilde{x}}[g(x) \neq g(\tilde{x})] \leq \frac{1}{2}(1 - \alpha^t(1 - s \cdot 2^{-\Omega(t^{1/d})}))$.

Remark 4 In [Vio04], Viola derived a weaker bound $\Pr_{x, \tilde{x}}[g(x) \neq g(\tilde{x})] \leq O(\beta \log^d s)$, with $\beta = \frac{1-\alpha}{2}$, which becomes vacuous when β is not small enough. This prevents him from having a meaningful bound when the hardness is not small enough. The main loss in his derivation comes from his use of the inequality $\frac{1}{2}(1 - \sum_{J \subseteq [N]} \alpha^{|J|} \hat{g}(J)^2) \leq \frac{1}{2}(1 - \sum_{J \subseteq [N]} (1 - \alpha)^{|J|} \hat{g}(J)^2)$. Our Lemma 4 uses a different inequality to avoid this problem.

2.3 Impossibility of Amplification by Small-Depth Circuits

In this section, we will show that any parallel black-box $(n, \frac{1-\delta}{2}, \frac{1-\delta^k}{2}, \ell)$ hardness amplification realized in $\text{AC}(d, s)$ with small d and s must be highly non-uniform. More precisely, we will prove the following.

Theorem 1 *There exist constants c_0, c_1, c_2, c_3 such that for any $\delta \in (0, 1)$ and any $d, k \in \mathbb{N}$ with $2^{-c_0 n} \leq \delta < 1$ and $2^{-2^{c_1 n}} \leq \delta^k \leq 1 - 2^{-c_2 k^{1/d}}$, any parallel black-box $(n, \frac{1-\delta}{2}, \frac{1-\delta^k}{2}, \ell)$ hardness amplification realized in $\text{AC}(d, 2^{c_3 k^{1/d}})$ must have $\ell = 2^{\Omega(n)}$.*

Before giving the proof, let us take a closer look at the theorem itself and discuss some of its consequences. First, note that the conditions on the ranges of δ and δ^k are natural in the following sense. When $\delta \leq 2^{-\Omega(n)}$, the initial function is already hard enough, so hardness amplification is usually not needed. When $\delta^k \geq 1 - 2^{-\Omega(k^{1/d})}$, the resulting function only has a very small hardness, which is rarely what hardness amplification is used to achieve. Also, as discussed in the introduction, hardness amplifications normally have m close to n (preferably with $m = \text{poly}(n)$), therefore δ^k , which is at least 2^{-m} , would be much larger than $2^{-2^{\Omega(n)}}$.

Although Theorem 1 is on the more restricted parallel model, it in fact implies the following result on the general model of hardness amplification, according to Lemma 1.

Corollary 2 *Under the same condition in Theorem 1, no strongly black-box $(n, \frac{1-\delta}{2}, \frac{1-\delta^k}{2}, 2^{o(n)})$ hardness amplification can be realized in $\text{ATIME}(O(1), k^{o(1)})$.*

Note that Viola's result [Vio04] is a special case of ours, with initial hardness $\frac{1-\delta}{2} = 2^{-n}$ (amplifying from worst-case hardness). A closer look at his technique shows that it in fact can be extended to cases with small initial hardness. For example, with $\frac{1-\delta}{2} = n^{-\omega(1)}$, his technique can be modified to show the impossibility in PH to amplify the hardness to $\frac{1-\delta^k}{2}$ with $k =$

$n^{\omega(1)}$, which also follows from our corollary above. However, as discussed in Remark 4, when the initial hardness grows beyond a certain point, say to $\frac{1-\delta}{2} = \Omega(1)$, his technique fails to give a meaningful bound. Moreover, our lower bound almost matches the upper bound given by the well-known XOR lemma [Yao82, GNW95], while the technique in [Vio04] does not yield such a bound.

Theorem 2 *For any $\delta \in (0, 1)$ and any $k, d \in \mathbb{N}$, a parallel black-box $(n, \frac{1-\delta}{2}, \frac{1-\delta^k}{2}, \ell)$ hardness amplification can be realized in $\text{AC}(O(d), 2^{O(k^{1/d})})$ for $\ell = \text{poly}(\frac{n}{\delta^k})$.*

Proof. The encoding function is $\text{AMP}^f : (\{0, 1\}^n)^t \rightarrow \{0, 1\}$, with $t = O(k)$, defined as

$$\text{AMP}^f(x_1, \dots, x_t) = f(x_1) \oplus \dots \oplus f(x_t).$$

It is known that the parity of t bits can be computed by an $\text{AC}(d+1, 2^{O(t^{1/d})})$ circuit (c.f. [Has86]), and note that this circuit and those t query inputs do not depend on the oracle f . Furthermore, using Levin's proof for the XOR lemma given in [GNW95], one can construct a decoding function which uses an advice of length $\ell \leq \text{poly}(\frac{n}{\delta^k})$. Thus, we have the lemma. \square

Now we proceed to prove Theorem 1.

Proof.(of Theorem 1) Consider any parallel black-box $(n, \frac{1-\delta}{2}, \frac{1-\delta^k}{2}, \ell)$ hardness amplification realized in $\text{AC}(d, s)$, with $s = 2^{c_3 k^{1/d}}$ for a small enough positive constant c_3 . Let $N = 2^n$ and $M = 2^m$. Recall from Lemma 2 that such a hardness amplification induces a $(\frac{1-\delta}{2}, \frac{1-\delta^k}{2}, 2^\ell)$ -list code $C : \{0, 1\}^N \rightarrow \{0, 1\}^M$. Then from Remark 3, it suffices to show that any such a code C computed by an $\text{AC}(d, s)$ circuit must have $\ell = 2^{\Omega(n)}$.

The basic idea behind the proof is the following. Suppose C has only a small number of codewords close to any codeword. Then a random perturbation on an input message is unlikely to result in a close codeword. On the other hand, if C is computed by an $\text{AC}(d, s)$ circuit with small d and s , which is insensitive to noise on the input, then a random perturbation on an input message is likely to result in a close codeword, and we reach a contradiction.

2.3. IMPOSSIBILITY OF AMPLIFICATION BY SMALL-DEPTH CIRCUITS 45

Now we give the details. Let x be sampled from the uniform distribution over $\{0, 1\}^N$ and let \tilde{x} be the random variable obtained by flipping each bit of x independently with some probability $\frac{1-\alpha}{2}$. We set $\alpha = \delta^{1.1}$ so that $\frac{1-\alpha}{2}$ is only slightly larger than $\frac{1-\delta}{2}$.³ We call any two codewords *close* if their (relative) distance is less than $\frac{1-\delta^k}{2}$. The next lemma gives a lower bound on the probability that $C(\tilde{x})$ is close to $C(x)$, which relies on the fact that such an AC circuit is insensitive to noise on the input.

Lemma 5 *There exist constants c_2, c_3, c_4 such that for any $\delta \in (0, 1)$ and any $k, d \in \mathbb{N}$ with $\delta^k \leq 1 - 2^{-c_2 k^{1/d}}$, if $C \in \text{AC}(d, 2^{c_3 k^{1/d}})$, then*

$$\Pr_{x, \tilde{x}}[C(x) \text{ is close to } C(\tilde{x})] \geq \delta^{c_4 k}.$$

Proof. Suppose $C \in \text{AC}(d, 2^{c_3 k^{1/d}})$ for a small enough constant c_3 . Suppose $\delta^k \leq 1 - 2^{-c_2 k^{1/d}}$ for some constant c_2 such that $\delta^{0.5k} \leq 1 - 2^{-c_3 k^{1/d}}$. Then using Corollary 1 with $t = k/3$, we have that for each $i \in [M]$,

$$\begin{aligned} \Pr_{x, \tilde{x}}[C(x)_i \neq C(\tilde{x})_i] &\leq \frac{1}{2} \left(1 - \alpha^t \left(1 - 2^{-c_3 k^{1/d}} \cdot 2^{-\Omega(t^{1/d})} \right) \right) \\ &\leq \frac{1}{2} \left(1 - \delta^{0.4k} \left(1 - 2^{-c_3 k^{1/d}} \right) \right) \\ &\leq \frac{1}{2} (1 - \delta^{0.9k}). \end{aligned}$$

Therefore, $\mathbb{E}_{x, \tilde{x}}[\Delta(C(x), C(\tilde{x}))] \leq \frac{1}{2}(1 - \delta^{0.9k})$, which implies that

$$\Pr_{x, \tilde{x}}[C(x) \text{ is not close to } C(\tilde{x})] \leq \frac{1 - \delta^{0.9k}}{1 - \delta^k}$$

by Markov inequality. Thus

$$\Pr_{x, \tilde{x}}[C(x) \text{ is close to } C(\tilde{x})] \geq 1 - \frac{1 - \delta^{0.9k}}{1 - \delta^k} \geq \frac{\delta^{0.9k} - \delta^k}{1 - \delta^k} \geq \delta^{0.9k} - \delta^k \geq \delta^{c_4 k},$$

for some constant c_4 . \square

Next, we give an upper bound on the probability that $C(\tilde{x})$ is close to $C(x)$, which relies on the fact that each codeword is only close to a small number of other codewords. This requires a more careful analysis than that in [Vio04], in order to get the tighter bound we need.

³We do not attempt to optimize parameters here, and in fact it suffices to set $\alpha = \delta(1 - o(1))$.

Lemma 6 For any $(\frac{1-\delta}{2}, \frac{1-\delta^k}{2}, 2^\ell)$ -list code C , $\Pr_{x, \tilde{x}}[C(x) \text{ is close to } C(\tilde{x})] \leq 2^\ell \cdot 2^{-\Omega(\delta^2 N)}$.

Proof. Consider any fixed $x \in \{0, 1\}^N$. Since C is a $(\frac{1-\delta}{2}, \frac{1-\delta^k}{2}, 2^\ell)$ -list code, there are at most $2^{\ell+H(\frac{1-\delta}{2})N}$ different y 's such that $C(y)$ is close to $C(x)$. The lemma would follow easily if each such y had a very small probability to occur. However, this may not be the case in general. We will show that although some y 's may occur with higher probability, there are not too many of them, so their overall contribution is still tolerable.

For any $y \in \{0, 1\}^N$, $\Pr_{\tilde{x}}[\tilde{x} = y] = \left(\frac{1-\alpha}{2}\right)^{\Delta(x,y)N} \left(\frac{1+\alpha}{2}\right)^{(1-\Delta(x,y))N}$, which decreases as $\Delta(x, y)$ increases. Let $\beta = \alpha^{0.91} = \delta^{1.001}$.⁴ Call $y \in \{0, 1\}^N$ *good* for x if $\Delta(x, y) \geq \frac{1-\beta}{2}$ and call y *bad* for x otherwise. Note that for any y which is good for x ,

$$\begin{aligned} \Pr_{\tilde{x}}[\tilde{x} = y] &\leq \left(\frac{1-\alpha}{2}\right)^{\frac{1-\beta}{2}N} \left(\frac{1+\alpha}{2}\right)^{\frac{1+\beta}{2}N} \\ &= 2^{\left(\frac{1-\beta}{2} \log \frac{1-\alpha}{2} + \frac{1+\beta}{2} \log \frac{1+\alpha}{2}\right)N} \\ &\leq 2^{-H\left(\frac{1-\beta}{2}\right)N}. \end{aligned}$$

On the other hand, \tilde{x} is only bad for x with a small probability. This is because \tilde{x} is obtained by flipping each bit of x independently with probability $\frac{1-\alpha}{2}$, so $\mathbb{E}_{\tilde{x}}[\Delta(x, \tilde{x})] = \frac{1-\alpha}{2}$, and by Chernoff bound,

$$\Pr_{\tilde{x}}[\tilde{x} \text{ is bad for } x] = \Pr_{\tilde{x}}\left[\Delta(x, \tilde{x}) < \frac{1-\beta}{2}\right] \leq 2^{-\Omega(\beta^2 N)}.$$

Thus, $\Pr_{\tilde{x}}[C(\tilde{x}) \text{ is close to } C(x)]$ is at most

$$\begin{aligned} &\Pr_{\tilde{x}}[C(\tilde{x}) \text{ is close to } C(x) \wedge \tilde{x} \text{ is good for } x] + \Pr_{\tilde{x}}[\tilde{x} \text{ is bad for } x] \\ &\leq 2^{\ell+H\left(\frac{1-\delta}{2}\right)N} \cdot 2^{-H\left(\frac{1-\beta}{2}\right)N} + 2^{-\Omega(\beta^2 N)} \\ &= 2^\ell \cdot 2^{H\left(\frac{1-\delta}{2}\right)N - H\left(\frac{1-\beta}{2}\right)N} + 2^{-\Omega(\beta^2 N)} \\ &\leq 2^\ell \cdot 2^{-\Omega(\delta^2 N)} + 2^{-\Omega(\beta^2 N)} \\ &\leq 2^\ell \cdot 2^{-\Omega(\delta^2 N)}. \end{aligned}$$

⁴Again, we make no attempt on optimizing the parameter here. In fact it suffices to set $\beta = \alpha(1 + o(1))$ while still maintaining $\beta = \delta(1 - o(1))$.

Since this holds for every x , the lemma follows. \square

Suppose $2^{-c_0 n} \leq \delta < 1$ and $2^{-2^{c_1 n}} \leq \delta^k \leq 1 - 2^{-c_2 k^{1/d}}$ for suitable constants c_0, c_1, c_2 . Then from Lemma 5 and Lemma 6, we get

$$\delta^{c_4 k} \leq \Pr_{x, \tilde{x}} [C(x) \text{ is close to } C(\tilde{x})] \leq 2^\ell \cdot 2^{-\Omega(\delta^2 N)},$$

which implies that

$$2^\ell \geq \delta^{c_4 k} \cdot 2^{\Omega(\delta^2 N)} \geq 2^{2^{\Omega(n)}}.$$

Thus, we have the following.

Lemma 7 *There exist constant c_0, c_1, c_2, c_3 such that for any $\delta \in (0, 1)$ and any $d, k \in \mathbb{N}$ with $2^{-c_0 n} \leq \delta < 1$ and $2^{-2^{c_1 n}} \leq \delta^k \leq 1 - 2^{-c_2 k^{1/d}}$, if $C : \{0, 1\}^{2^n} \rightarrow \{0, 1\}^M$ is a $(\frac{1-\delta}{2}, \frac{1-\delta^k}{2}, 2^\ell)$ -list code computable by an $\text{AC}(d, 2^{c_3 k^{1/d}})$ circuit, then $2^\ell = 2^{2^{\Omega(n)}}$.*

Combining this lemma with Lemma 2, we obtain Theorem 1. \square

2.4 Impossibility of Amplification by Non-deterministic Circuits

Note that the result in the previous section becomes meaningless for $d = \Omega(\log k)$, as it only rules out circuits in $\text{AC}(d, s)$ with $s = 2^{O(k^{1/d})} = O(1)$. In this section, we show that even without any restriction on the circuit depth, a meaningful lower bound on the circuit size can still be derived. Formally, we have the following theorem.

Theorem 3 *There exist constants c_0, c_1, c_2, c_3 such that for any $\delta \in (0, 1)$ and any $k \in \mathbb{N}$ with $2^{-c_0 n} \leq \delta < 1$ and $2^{-2^{c_1 n}} \leq \delta^k \leq 1 - k^{-c_2}$, any parallel black-box $(n, \frac{1-\delta}{2}, \frac{1-\delta^k}{2}, \ell)$ hardness amplification realized in $\text{NAC}(\frac{k}{c_3 \log k})$ must have $\ell = 2^{\Omega(n)}$.*

From Lemma 1, this implies the following impossibility result on general black-box hardness amplification.

Corollary 3 *Under the same condition as in Theorem 3, it is impossible to realize black-box $(n, \frac{1-\delta}{2}, \frac{1-\delta^k}{2}, 2^{o(n)})$ hardness amplification in $\text{ATIME}(c \log k)$, for some constant $c > 0$.*

Now we prove the theorem.

Proof.(of Theorem 3) The basic proof idea is similar to that for Theorem 1. The only difference is to replace Lemma 5 by an analogous one for NAC circuits. Here we use the method of random restriction. A restriction on a set of variables $V = \{x_i : i \in [N]\}$ is a mapping $\rho : V \rightarrow \{0, 1, \star\}$, which either fixes the value of a variable x_i with $\rho(x_i) \in \{0, 1\}$ or leaves x_i free with $\rho(x_i) = \star$. For $p \in (0, 1)$, let \mathbf{R}_p denote the distribution on such restrictions such that each variable x_i is mapped independently with $\Pr_{\rho \in \mathbf{R}_p}[\rho(x_i) = \star] = p$ and $\Pr_{\rho \in \mathbf{R}_p}[\rho(x_i) = 0] = \Pr_{\rho \in \mathbf{R}_p}[\rho(x_i) = 1] = (1 - p)/2$. For a Boolean function g and a restriction ρ , let g_ρ denote the function obtained from g by applying the restriction ρ to its variables. That is, $g_\rho(x_1, \dots, x_N) = g(y_1, \dots, y_N)$ with $y_i = x_i$ if $\rho(x_i) = \star$ and $y_i = \rho(x_i)$ otherwise.

Define the degree of a function g as $\text{deg}(g) = \max_J\{|J| : \hat{g}(J) \neq 0\}$. It is not hard to verify that a constant function has degree 0 and a function depending on only t input bits has degree at most t . We need the following lemma which bounds the contribution of higher-order Fourier coefficients.

Lemma 8 [LMN93] *Let $p \in (0, 1)$ and $t \in \mathbb{N}$ with $pt > 8$. Then for any Boolean function g , $\sum_{|J|>t} \hat{g}(J)^2 \leq 2 \cdot \Pr_{\rho \in \mathbf{R}_p}[\text{deg}(g_\rho) \geq pt/2]$.*

The following is the key lemma in this section, which gives a concrete bound on the sum above for NAC circuits.

Lemma 9 *For any $g : \{0, 1\}^N \rightarrow \{0, 1\} \in \text{NAC}(s)$, $\sum_{|J|>t} \hat{g}(J)^2 \leq s \cdot 2^{-\Omega(t/s)}$, when $9 \leq t \leq N$.*

Proof. Suppose g is computed by an NAC circuit of size s , which divides its input into the real input part and the witness part. Let \mathcal{B} be the set of gates which receive some real input variables directly. Consider applying a random restriction $\rho \in \mathbf{R}_p$ on the real input variables. We say a gate in \mathcal{B} is

killed if it is an AND gate and receives a real input variable which is fixed to 0 by ρ , or if it is an OR gate and receives a real input variable which is fixed to 1 by ρ . For a gate $A \in \mathcal{B}$, let $\#(A)$ denote the number of real input variables it receives. For a restriction ρ , let $\#(A_\rho)$ denote the the number of remaining real input variables it receives if A is not killed by ρ , and let $\#(A_\rho) = 0$ otherwise. Set p to be any constant in $(0, 1)$ so that $pt > 8$. Then

$$\begin{aligned} \Pr_{\rho \in \mathcal{R}_p} [\deg(g_\rho) \geq pt/2] &\leq \Pr_{\rho \in \mathcal{R}_p} [\exists A \in \mathcal{B} : \#(A_\rho) \geq pt/(2s)] \\ &\leq s \cdot \max_{A \in \mathcal{B}} \Pr_{\rho \in \mathcal{R}_p} [\#(A_\rho) \geq pt/(2s)]. \end{aligned}$$

Any $A \in \mathcal{B}$ with $\#(A) < pt/(2s)$ clearly has $\Pr_{\rho \in \mathcal{R}_p} [\#(A_\rho) \geq pt/(2s)] = 0$. On the other hand, any $A \in \mathcal{B}$ with $\#(A) \geq pt/(2s)$ is likely to be killed, so that $\Pr_{\rho \in \mathcal{R}_p} [\#(A_\rho) \geq pt/(2s)] \leq \Pr_{\rho \in \mathcal{R}_p} [A \text{ is not killed by } \rho] \leq (1 - (1-p)/2)^{pt/(2s)} = 2^{-\Omega(t/s)}$. From Lemma 8, we have $\sum_{|J|>t} \hat{g}(J)^2 \leq 2s \cdot 2^{-\Omega(t/s)} = s \cdot 2^{-\Omega(t/s)}$. \square

Then analogously to Lemma 5 (in the previous section), we have the following.

Lemma 10 *There exist constants c_2, c_3, c_4 such that for any $\delta \in (0, 1)$ and any $k \in \mathbb{N}$ with $\delta^k \leq 1 - k^{-c_2}$, if $C \in \text{NAC}(\frac{k}{c_3 \log k})$, then*

$$\Pr_{x, \tilde{x}} [C(x) \text{ is close to } C(\tilde{x})] \geq \delta^{c_4 k}.$$

Proof. Suppose $C : \{0, 1\}^N \rightarrow \{0, 1\}^M \in \text{NAC}(\frac{k}{c_3 \log k})$, for some large enough constant c_3 . Using Lemma 4 and Lemma 9 with $t = k/3$, we have that for each $i \in [M]$,

$$\begin{aligned} \Pr_{x, \tilde{x}} [C(x)_i \neq C(\tilde{x})_i] &\leq \frac{1}{2} \left(1 - \alpha^t \left(1 - \frac{k}{c_3 \log k} \cdot 2^{-\Omega(c_3 \log k)} \right) \right) \\ &\leq \frac{1}{2} (1 - \delta^{0.4k} (1 - k^{-\Omega(1)})) \\ &\leq \frac{1}{2} (1 - \delta^{0.9k}), \end{aligned}$$

when $\delta^k \leq 1 - k^{-c_2}$ for some suitable constant c_2 . Then the rest is the same as that for Lemma 5, and we can have $\Pr_{x, \tilde{x}} [C(x) \text{ is close to } C(\tilde{x})] \geq \delta^{c_4 k}$ for some constant c_4 . \square

Suppose $2^{-c_0 n} \leq \delta < 1$ and $2^{-2^{c_1 n}} \leq \delta^k \leq 1 - k^{-c_2}$, for suitable constants c_0, c_1, c_2 . By combining Lemma 10 with Lemma 6, we get $2^\ell \geq \delta^{c_4 k} \cdot 2^{\Omega(\delta^{2N})} \geq 2^{2^{\Omega(n)}}$, which gives the following.

Lemma 11 *There exist constant c_0, c_1, c_2, c_3 such that for any $\delta \in (0, 1)$ and any $k \in \mathbb{N}$ with $2^{-c_0 n} \leq \delta < 1$ and $2^{-2^{c_1 n}} \leq \delta^k \leq 1 - k^{-c_2}$, if $C : \{0, 1\}^{2^n} \rightarrow \{0, 1\}^M$ is a $(\frac{1-\delta}{2}, \frac{1-\delta^k}{2}, 2^\ell)$ -list code computable by $\text{NAC}(\frac{k}{c_3 \log k})$, then $2^\ell = 2^{2^{\Omega(n)}}$.*

Combining this with Lemma 2, we obtain Theorem 3. \square

2.5 Inherent Non-uniformity of Hardness Amplification

In the previous two sections, we have proven that any strongly black-box hardness amplification must be very non-uniform when the computational complexity of the amplification procedure AMP is bounded in certain ways. In this section, we prove that even without any such complexity bound, there still exists some inherent non-uniformity.

First, we state the following simple result which seems to be a folklore. For completeness we include its proof.

Theorem 4 *For some constant c and for any $\gamma \in (0, 1)$, no oracle algorithm $\text{AMP}^{(\cdot)} : \{0, 1\}^m \rightarrow \{0, 1\}$ can realize a strongly black-box $(n, \frac{1-\gamma}{4}, \frac{1}{4}, 0)$ hardness amplification with $c\gamma 2^{n/2} > m + 1$.*

Proof. From Lemma 2, this reduces to the following coding-theoretical question: for which values of α and β do we have a $(\alpha, \beta, 1)$ -list code?

We call $C : \{0, 1\}^N \rightarrow \{0, 1\}^M$ an $[N, M, \alpha]$ code if the (relative Hamming) distance of any two codewords is at least α . We need the following good code, which can be constructed using, say, the concatenation of Reed-Solomon code with Hadamard code.

Fact 4 $[N, O((\frac{N}{\gamma})^2), \frac{1-\gamma}{2}]$ codes exist for any $\gamma \in (0, 1)$.

This says that unique decoding is possible if the fraction of error is slightly smaller than $\frac{1}{4}$. On the other hand, according to the following Plotkin bound, unique decoding is basically impossible if the fraction of error grows beyond $\frac{1}{4}$.

Fact 5 (Plotkin Bound [Plo60]) An $[N, M, \alpha]$ code with $\alpha \geq \frac{1}{2}$ must have $N \leq \log(2M)$.

Combining these two facts, we have the following.

Lemma 12 For some constant c and for any $\gamma \in (0, 1)$, any $(\frac{1-\gamma}{4}, \frac{1}{4}, L)$ -list code $C : \{0, 1\}^N \rightarrow \{0, 1\}^M$ with $c\gamma\sqrt{N} > \log(2M)$ must have $L \geq 2$.

Proof. From Fact 4, there exists a $[K, N, \frac{1-\gamma}{2}]$ code C' with $K \geq c\gamma\sqrt{N}$ for some constant c . Suppose that C is a $(\frac{1-\gamma}{4}, \frac{1}{4}, L)$ -list code with $c\gamma\sqrt{N} > \log(2M)$. If $L = 1$, then $C \circ C' : \{0, 1\}^K \rightarrow \{0, 1\}^M$ is a $[K, M, \frac{1}{2}]$ code with $K > \log(2M)$, which is impossible according to Fact 5. \square

Then from Lemma 2, we obtain Theorem 4. \square

As discussed in the introduction, hardness amplifications normally have $m = \text{poly}(n)$. Thus, the theorem basically says that amplifying hardness beyond $\frac{1}{4}$ must introduce non-uniformity in general. However, the theorem does not provide a quantitative bound on the non-uniformity. This is addressed by our next theorem.

Theorem 5 Suppose $\varepsilon < \frac{1}{c}$ for some suitable constant c , and suppose $2^n = \omega(\frac{1}{\delta^2} \log \frac{1}{\varepsilon})$. Then any strongly black-box $(n, \frac{1-\delta}{2}, \frac{1-\varepsilon}{2}, \ell)$ hardness amplification must have $\ell = \Omega(\log \frac{\delta^2}{\varepsilon})$.

Thus, any such hardness amplification, even without any complexity constraint, must be inherently non-uniform, with $\ell \geq 1$ when $\varepsilon \leq c'\delta^2$ for some constant c' , or with $\ell = \Omega(k \log \frac{1}{\delta})$ when $\varepsilon = \delta^k$. Note that our lower bound generalizes that of Trevisan and Vadhan [TV02]: they only considered the

case with $\delta = 1 - 2^{-n+1}$ (or equivalently $\frac{1-\delta}{2} = 2^{-n}$) and obtained the lower bound $\ell = \Omega(\log \frac{1}{\varepsilon})$, while we consider general δ and obtain the lower bound $\ell = \Omega(\log \frac{\delta^2}{\varepsilon})$.

Now we proceed to the proof of Theorem 5.

Proof.(of Theorem 5) Consider an arbitrary code $C : \{0, 1\}^N \rightarrow \{0, 1\}^M$. We would like to show that for some constant c , one can find a string $z \in \{0, 1\}^M$ and a set $S \subseteq \{0, 1\}^N$ such that the following two conditions hold:

- For every $x \in S$, $C(x)$ is contained in the ball $\text{BALL}_z(\frac{1-\varepsilon/c}{2}, M)$.
- S needs $\Omega(\frac{\delta^2}{\varepsilon})$ balls in $\text{BALL}(\frac{1-\delta}{2}, N)$ to cover with.

For this, we first choose x^1, \dots, x^t uniformly and independently from $\{0, 1\}^N$ to form the set R , for some $t = \Theta(\frac{1}{\varepsilon^2})$. Call the set R δ -good if $|R| = t$ (i.e. $x^i \neq x^j$ for any $i \neq j$) and any ball in $\text{BALL}(\frac{1-\delta}{2}, N)$ contains $O(\frac{1}{\delta^2})$ elements of R . Later, we will derive the set S from a δ -good R .

Lemma 13 *When $N = \omega(\frac{1}{\delta^2} \log \frac{1}{\varepsilon})$, R is δ -good with probability $1 - 2^{-\Omega(N)}$.*

Proof. First, the probability that $x^i = x^j$ for some $i \neq j$ is at most $\binom{t}{2} \cdot 2^{-N} \leq 2^{2 \log t - N}$. Next, the probability that some ball in $\text{BALL}(\frac{1-\delta}{2}, N)$ contains r elements of R is at most $2^N \cdot \binom{t}{r} \cdot 2^{(H(\frac{1-\delta}{2})-1)Nr} \leq 2^{N+r \log t - \Omega(\delta^2)rN}$. For some $r = O(\frac{1}{\delta^2})$, both probabilities above are $2^{-\Omega(N)}$ when $N = \omega(\frac{1}{\delta^2} \log t)$. This proves the lemma. \square

We want to choose a string $z \in \{0, 1\}^M$ such that the ball $\text{BALL}_z(\frac{1-\varepsilon}{2}, M)$ contains a lot of codewords coming from a δ -good R . We will fix some of z 's bits first.

Definition 10 *For each $y \in [M]$, let b_y be the bit such that*

$$\Pr_{x \in \{0,1\}^N} [C(x)_y \neq b_y] \leq \frac{1}{2}.$$

Call R (δ, ε) -good for y if R is δ -good and $\Pr_{x \in R} [C(x)_y \neq b_y] \leq \frac{1-\varepsilon}{2}$.

Lemma 14 *Suppose $N = \omega(\frac{1}{\delta^2} \log \frac{1}{\varepsilon})$. Then for any $y \in [M]$, R is (δ, ε) -good for y with probability $\Omega(1)$.*

Proof. From Lemma 13, R is not δ -good with probability $2^{-\Omega(N)}$. Now fix any $y \in [M]$. Let I_i , for $i \in [t]$, be the indicator random variable such that $I_i = 1$ if $C(x^i)_y \neq b_y$ and $I_i = 0$ otherwise. Then, by letting $T(R) = 1$ if and only if $\Pr_{x \in R} [C(x)_y \neq b_y] \leq \frac{1-\varepsilon}{2}$,

$$\begin{aligned} \Pr_R [T(R) = 1] &= \Pr_{x^1, \dots, x^t} \left[\frac{1}{t} |\{i \in [t] : C(x^i)_y \neq b_y\}| \leq \frac{1-\varepsilon}{2} \right] \\ &= \Pr_{x^1, \dots, x^t} \left[\frac{1}{t} \sum_{i \in [t]} I_i \leq \frac{1-\varepsilon}{2} \right]. \end{aligned}$$

Note that I_1, \dots, I_t form a sequence of i.i.d., with $\mathbb{E}[I_i] \leq \frac{1}{2}$ for each i . Let J_1, \dots, J_t be the sequence of i.i.d. binary random variables with $\mathbb{E}[J_i] = \frac{1}{2}$ for each i . Then,

$$\Pr \left[\frac{1}{t} \sum_{i \in [t]} I_i \leq \frac{1-\varepsilon}{2} \right] \geq \Pr \left[\frac{1}{t} \sum_{i \in [t]} J_i \leq \frac{1-\varepsilon}{2} \right].$$

Therefore, by Fact 1, we have

$$\Pr_R [T(R) = 1] \geq \Pr \left[\frac{1}{t} \sum_{i \in [t]} J_i \leq \frac{1-\varepsilon}{2} \right] = \Omega(1),$$

as $t = \Theta(\frac{1}{\varepsilon^2})$. Then R is (δ, ε) -good for y with probability at least $\Omega(1) - 2^{-\Omega(N)} = \Omega(1)$. \square

An averaging argument immediately gives the following.

Corollary 4 *Suppose $N = \omega(\frac{1}{\delta^2} \log \frac{1}{\varepsilon})$. Then there exist a set $R \subseteq \{0, 1\}^N$ with $|R| = \Omega(\frac{1}{\varepsilon^2})$ and a set $A \subseteq [M]$ with $|A| = \Omega(M)$ such that for any $y \in A$, R is (δ, ε) -good for y .*

Let us fix the sets R and A guaranteed by the corollary above. Next, we want to show that many x 's from R satisfy the property that the codeword $C(x)$ has enough agreement with the vector b (with each bit b_y defined in Definition 10) on those dimensions in A .

Lemma 15 *There exists $R' \subseteq R$ with $|R'| = \Omega(\frac{1}{\varepsilon})$ such that for any $x \in R'$, $\Pr_{y \in A} [C(x)_y \neq b_y] < \frac{1-\varepsilon/2}{2}$.*

Proof. For any $y \in A$, R is (δ, ε) -good for y , so

$$\mathbb{E}_{x \in R} \left[\Pr_{y \in A} [C(x)_y \neq b_y] \right] = \mathbb{E}_{y \in A} \left[\Pr_{x \in R} [C(x)_y \neq b_y] \right] \leq \frac{1 - \varepsilon}{2}.$$

Let $P(x) = 1$ if and only if $\Pr_{y \in A} [C(x)_y \neq b_y] \geq \frac{1 - \varepsilon/2}{2}$. By Markov's inequality,

$$\Pr_{x \in R} [P(x) = 1] \leq \frac{\frac{1 - \varepsilon}{2}}{\frac{1 - \varepsilon/2}{2}} \leq 1 - \frac{\varepsilon}{2}.$$

Thus, there exists $R' \subseteq R$ of size $\frac{\varepsilon}{2}|R| = \Omega(\frac{1}{\varepsilon})$ such that for any $x \in R'$, $\Pr_{y \in A} [C(x)_y \neq b_y] < \frac{1 - \varepsilon/2}{2}$. \square

We let the vector z inherit from the vector b those bits indexed by A , and it remains to set the values for the remaining bits. It is easy to show that there exist $v \in \{0, 1\}^M$ (in fact, v can be chosen from $\{0^M, 1^M\}$) and $S \subseteq R'$ with $|S| \geq \frac{1}{2}|R'|$ such that for any $x \in S$, $\Pr_{y \notin A} [C(x)_y \neq v_y] \leq \frac{1}{2}$. So we just define $z \in \{0, 1\}^M$ as $z_y = b_y$ if $y \in A$ and $z_y = v_y$ otherwise. Then, for any $x \in S$,

$$\begin{aligned} & \Delta(C(x), z) \\ &= \Pr_{y \in [M]} [y \in A] \cdot \Pr_{y \in A} [C(x)_y \neq b_y] + \Pr_{y \in [M]} [y \notin A] \cdot \Pr_{y \notin A} [C(x)_y \neq v_y] \\ &< \frac{|A|}{M} \cdot \frac{1 - \varepsilon/2}{2} + \frac{M - |A|}{M} \cdot \frac{1}{2} \\ &= \frac{1}{2} \left(1 - \frac{|A|(\varepsilon/2)}{M} \right) \\ &\leq \frac{1 - \varepsilon/c}{2}, \end{aligned}$$

for some constant c .

Furthermore, as $S \subseteq R$ and R is δ -good, any ball in $\text{BALL}(\frac{1 - \delta}{2}, N)$ contains $O(\frac{1}{\delta^2})$ elements of S , and hence S must need $\frac{|S|}{O(1/\delta^2)} = \Omega(\frac{\delta^2}{\varepsilon})$ such balls to cover with. This shows that any $(\frac{1 - \delta}{2}, \frac{1 - \varepsilon/c}{2}, 2^\ell)$ -list code must have $2^\ell = \Omega(\frac{\delta^2}{\varepsilon})$. Replacing the parameter ε/c by ε , we have the following.

Lemma 16 *Suppose $\varepsilon < \frac{1}{c}$ for some suitable constant c , and suppose $N = \omega(\frac{1}{\delta^2} \log \frac{1}{\varepsilon})$. Then any $(\frac{1 - \delta}{2}, \frac{1 - \varepsilon}{2}, 2^\ell)$ -list code must have $2^\ell = \Omega(\frac{\delta^2}{\varepsilon})$.*

This, combined with Lemma 2, proves the theorem. \square

Remark 5 *Recently Guruswami and Vadhan [GV05] use a more involved argument to prove that any $(2^{-n}, \frac{1-\varepsilon}{2}, 2^\ell)$ -list code must have $L = \Omega(\frac{1}{\varepsilon^2})$. Their proof can be extended to show that any $(\frac{1-\delta}{2}, \frac{1-\varepsilon}{2}, 2^\ell)$ -list code must have $2^\ell = \Omega(\frac{\delta^2}{\varepsilon^2})$. Therefore any such strongly black-box hardness amplification with $\delta \geq c_0\varepsilon$, for some constant c_0 , must be inherently non-uniform.*

2.6 Impossibility Results on PRG Constructions

In this section, we prove lower bound (impossibility) results for strongly black-box PRG constructions from hard functions. For this, we establish a connection between strongly black-box PRG constructions and codes. Then using those lower bound results for codes in previous sections, we obtain lower bound results for strongly black-box PRG constructions.

Consider any strongly black-box PRG construction with an encoding function $G^{(\cdot)} : \{0, 1\}^m \rightarrow \{0, 1\}^r$. We call the ratio $\frac{r}{m}$ as the stretch factor of the PRG construction. Let $N = 2^n$ and $M = r2^m$, and define the corresponding code $C : \{0, 1\}^N \rightarrow \{0, 1\}^M$ as $C(f) = G^f$. That is, seeing any function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ as a vector in $\{0, 1\}^N$, $C(f)$ produces as output the function G^f , which is seen as a vector in $(\{0, 1\}^r)^{2^m} = \{0, 1\}^M$ (the concatenation of $G^f(u)$'s over $u \in \{0, 1\}^m$). Analogously to Lemma 2, we have the following connection between PRG constructions and codes.

Lemma 17 *Suppose $G^{(\cdot)} : \{0, 1\}^m \rightarrow \{0, 1\}^r$ is the encoding function of a strongly black-box $(n, \beta, \frac{\varepsilon}{2}, \ell)$ PRG construction with a stretch factor $\frac{r}{m} = \omega(\frac{1}{\varepsilon^2})$. Then $C : \{0, 1\}^N \rightarrow \{0, 1\}^M$, defined as $C(f) = G^f$, is a $(\beta, \frac{1-\varepsilon}{2}, 2^\ell)$ -list code.*

Proof. Suppose G is the encoding function of a strongly black-box $(n, \beta, \frac{\varepsilon}{2}, \ell)$ PRG construction, and DEC is the decoding function, which is an oracle

Turing machine with an ℓ -bit advice. Consider any string $A \in \{0, 1\}^M$, which can be seen as a function $A : \{0, 1\}^m \rightarrow \{0, 1\}^r$. We want to show that not many codewords are close to A . For this, we show that there exists a distinguisher D_A such that if any $C(f)$ is close to A , then D_A can distinguish G^f from random.

Define the distinguisher $D_A : \{0, 1\}^r \rightarrow \{0, 1\}$ as

$$D_A(w) = 1 \quad \text{if and only if} \quad \exists u \in \{0, 1\}^m : \Delta(w, A(u)) \leq \frac{1 - \varepsilon/4}{2}.$$

Suppose $r = \omega(\frac{m}{\varepsilon^2})$, and assume without loss of generality that $2^{-\omega(m)} \leq \frac{\varepsilon}{4}$.⁵ Then,

$$\begin{aligned} \Pr_{w \in \mathcal{U}_r} [D_A(w) = 1] &\leq \sum_{u \in \{0, 1\}^m} \Pr_{w \in \mathcal{U}_r} \left[\Delta(w, A(u)) \leq \frac{1 - \varepsilon/4}{2} \right] \\ &\leq 2^m \cdot 2^{-\Omega(\varepsilon^2 r)} \\ &\leq 2^{-\omega(m)} \\ &\leq \frac{\varepsilon}{4}. \end{aligned}$$

Consider any codeword $C(f)$ with $\Delta(A, C(f)) < \frac{1 - \varepsilon}{2}$. Now as

$$\mathbb{E}_{u \in \mathcal{U}_m} [\Delta(A(u), G^f(u))] = \Delta(A, C(f)),$$

by Markov inequality we have

$$\Pr_{u \in \mathcal{U}_m} [\Delta(A(u), G^f(u)) > \frac{1 - \varepsilon/4}{2}] < \frac{1 - \varepsilon}{1 - \varepsilon/4} \leq 1 - \frac{3\varepsilon}{4}.$$

Thus,

$$\Pr_{u \in \mathcal{U}_m} [D_A(G^f(u)) = 1] \geq \Pr_{u \in \mathcal{U}_m} \left[\Delta(G^f(u), A(u)) \leq \frac{1 - \varepsilon/4}{2} \right] > \frac{3\varepsilon}{4}.$$

Therefore, we have

$$\left| \Pr_{u \in \mathcal{U}_m} [D_A(G^f(u)) = 1] - \Pr_{w \in \mathcal{U}_r} [D_A(w) = 1] \right| > \frac{3\varepsilon}{4} - \frac{\varepsilon}{4} = \frac{\varepsilon}{2}.$$

⁵For a PRG $G : \{0, 1\}^m \rightarrow \{0, 1\}^r$, one can only expect $\varepsilon \geq 2^{-m} - 2^{-r}$, because this can be achieved by a simple distinguisher T defined as $T(z) = 1$ if and only if $z = G(0^r)$. Since G is a PRG, $r \geq m + 1$, $\varepsilon \geq 2^{-m} - 2^{-(m+1)} = 2^{-(m+1)}$ and we have $2^{-\omega(m)} \leq \frac{\varepsilon}{4}$.

From Definition 19, this implies that there exists an $\nu \in \{0, 1\}^\ell$ such that $\Delta(\text{DEC}^{D_A, \nu}, f) = \Pr_x[\text{DEC}^{D_A, \nu}(x) \neq f(x)] < \beta$.

We have shown that if $C(f)$ is in $\text{BALL}_A(\frac{1-\varepsilon}{2}, M)$, then f is contained in one of the 2^ℓ balls of radius β centered at $\text{DEC}^{D_A, \nu}$ for $\nu \in \{0, 1\}^\ell$. This implies that C is a $(\beta, \frac{1-\varepsilon}{2}, 2^\ell)$ -list code. \square

With the help of this lemma, lower bound results on codes in previous sections now immediately yield results on strongly black-box constructions of PRG.

First, observe that if the PRG construction has a parallel realization in a circuit class, then every output bit of C can be computed by a circuit in the class. Then by combining Lemma 17 with Lemma 7, we have the following theorem on parallel black-box PRG constructions realized by small-depth AC circuits.

Theorem 6 *There exist constants c_0, c_1, c_2, c_3 such that for any $\delta \in (0, 1)$ and any $d, k \in \mathbb{N}$ with $2^{-c_0 n} \leq \delta < 1$ and $2^{-2^{c_1 n}} \leq \delta^k \leq 1 - 2^{-c_2 k^{1/d}}$, any parallel black-box $(n, \frac{1-\delta}{2}, \frac{\delta^k}{2}, \ell)$ realized in $\text{AC}(d, 2^{c_3 k^{1/d}})$ with a stretch factor $\omega(\frac{1}{\delta^{2k}})$ must have $\ell = 2^{\Omega(n)}$.*

Next, by combining Lemma 17 with Lemma 11, we immediately have the following theorem on parallel black-box PRG constructions realized by NAC circuits.

Theorem 7 *There exist constants c_0, c_1, c_2, c_3 such that for any $\delta \in (0, 1)$ and any $k \in \mathbb{N}$ with $2^{-c_0 n} \leq \delta < 1$ and $2^{-2^{c_1 n}} \leq \delta^k \leq 1 - k^{-c_2}$, any parallel black-box $(n, \frac{1-\delta}{2}, \frac{\delta^k}{2}, \ell)$ PRG construction realized in $\text{NAC}(\frac{k}{c_3 \log k})$ with a stretch factor $\omega(\frac{1}{\delta^{2k}})$ must have $\ell = 2^{\Omega(n)}$.*

Similar to those in Sections 2.3 & 2.4, the two theorems above on the parallel model immediately imply impossibility results on general strongly black-box PRG constructions, via Lemma 1.

Finally, by combining Lemma 17 with Lemma 16, we have the following theorem on the inherent non-uniformity of strongly black-box PRG constructions.

Theorem 8 *Suppose $\varepsilon < \frac{1}{c}$ for some suitable constant c , and suppose $2^n = \omega(\frac{1}{\delta^2} \log \frac{1}{\varepsilon})$. Then any strongly black-box $(n, \frac{1-\delta}{2}, \varepsilon, \ell)$ PRG construction with a stretch factor $\omega(\frac{1}{\varepsilon^2})$ must have $\ell = \Omega(\log \frac{\delta^2}{\varepsilon})$.*



Chapter 3

Weakly Black-Box Hardness Amplification

3.1 Introduction

In this chapter, we study the problem of transforming a hard function into a harder one via a procedure called weakly black-box hardness amplification, in which the initial hard function is only used as a black box to construct the harder function. First, we show that if a weakly black-box hardness amplification procedure in $\text{TIME}(t)$ can amplify hardness beyond an $O(t)$ factor, then it must embed in itself a hard function computable in $\text{TIME}(t)$. As a result, it is impossible to have such a hardness amplification with hardness measured against $\text{TIME}(t)$. Next, we show that, for any $k \in \mathbb{N}$, if a weakly black-box hardness amplification procedure in $\Sigma_k\text{P}$ can amplify hardness beyond a polynomial factor, then one can obtain from it a hard function in $\Sigma_k\text{P}$. A similar impossibility result can also be derived.

3.1.1 Previous Results

For the case of strongly black-box hardness amplification, Viola [Vio04] proved that no amplification procedures from worst-case hardness to mild hardness is computable in PH . Lu et al. [LTW05] proved a more general

result, showing the impossibility of amplifying hardness from $(1 - \delta)/2$ to $(1 - \delta^k)/2$ in PH for any super-polynomial k . Furthermore, they showed that such a hardness amplification must be highly non-uniform in nature, in the sense that one must start from a function f which is hard against a very non-uniform complexity class even if one only wants to obtain a function \bar{f} which is hard against a uniform complexity class [LTW05] (presented in Chapter 2).

Since the strongly black-box approach has its limitation, one may look for a weaker type of hardness amplification. Bogdanov and Trevisan [BT03] showed that even if one drops the constraint on the encoding procedure, one still cannot amplify from worst-case hardness to mild hardness for functions in NP unless PH collapses, when the decoding procedure is required to be computable non-adaptively in P .

The other possibility is to consider weakly black-box hardness amplification, in which the constraint on the decoding procedure is dropped, while the encoding procedure is still required to be done in a black-box way. Viola [Vio05] proved that if a weakly black-box procedure amplifying from worst-case hardness to mild hardness can be realized in PH , then one can obtain from it a mildly hard function computable in PH . Although this can be seen as a negative result, it does not rule out the possibility of such a weakly black-box hardness amplification. In fact, it appears difficult to establish impossibility results for such a hardness amplification. This is because if an average-case hard function indeed exists, an amplification procedure may simply ignore the initial hard function and compute the average-case hard function from scratch. This raises the question: can one prove any meaningful impossibility result for weakly black-box hardness amplification?

3.1.2 Our Results

We derive two negative results for weakly black-box hardness amplification. First, we prove that if a weakly black-box hardness amplification realized in $\text{TIME}(t)$ can amplify hardness by an $\omega(t)$ factor, from $o(\varepsilon/t)$ to ε , then it

must embed in it a function computable in $\text{TIME}(t)$ with hardness about ε . Note that a function in $\text{TIME}(t)$ cannot be hard against a class containing $\text{TIME}(t)$. Therefore, we obtain an unconditional impossibility result: it is impossible to use a procedure in $\text{TIME}(t)$ to transform a function which is $o(\varepsilon/t)$ -hard against the class $\mathcal{C} = \text{SIZE}(2^{n/3})$ into a function which is ε -hard against a class $\bar{\mathcal{C}} \supseteq \text{TIME}(t)$. This rules out the possibility of using a low-complexity procedure to do such a hardness amplification for high-complexity functions.¹ Note that when $t = 2^{o(n)}$, this gives an impossibility result for amplifying from worst-case hardness to mild hardness in sub-exponential time. We also extend this impossibility result to the case with \mathcal{C} being any uniform complexity class equipped with an advice of length at most $2^{n/3}$. This says that such a weakly hardness amplification, just as in the strongly black-box case [LTW05], must also be highly non-uniform in nature: it is impossible to have such a weakly hardness amplification if one start from an initial function which is hard against any complexity class with only $2^{n/3}$ bits of non-uniformity (even of arbitrarily high uniform complexity). Second, we prove that if a weakly black-box hardness amplification realized in NP ($\Sigma_k\text{P}$, respectively) can amplify hardness beyond a polynomial factor, from $\varepsilon^2/n^{\omega(1)}$ to ε , then one can obtain from it a function computable in NP ($\Sigma_k\text{P}$, respectively) with hardness about ε . This improves the result in [Vio05], as the hard function obtained there seems to need at least the complexity of $\Sigma_{k+1}\text{P}$, one level higher than ours in PH . Again, this enables us to derive an unconditional impossibility result: it is impossible to use a procedure in NP ($\Sigma_k\text{P}$, respectively) for such a hardness amplification, if the new function's hardness is measured against a class containing NP/poly ($\Sigma_k\text{P}$, respectively),

¹It is possible to use a low complexity (oracle) procedure to amplify hardness within certain range for functions in high complexity classes. For example, the derandomized XOR lemma [IW97] (the XOR lemma [Yao82, NW94], respectively) allows us to use a polynomial-time (oracle) procedure to amplify from mild hardness to average-case hardness (hardness close to average-case hardness, respectively) for functions in high complexity classes, such as E . Our result says that this becomes impossible if one wants to amplify hardness beyond certain factor.

when the initial function is hard against a uniform complexity class equipped an advice of length $2^{n/3}$. Note that this excludes the possibility of having such a hardness amplification from worst-case hardness to mild hardness for functions in NP. Following our result, we widen the gap between worst-case and mild hardness within NP.

3.1.3 Organization of this chapter

First, some preliminaries are given in Section 3.2. Then in Section 3.3 and in Section 3.4, we show the results of weakly black-box hardness amplification sub-exponential time and $\Sigma_k P$ respectively.

3.2 Preliminaries

First, we generalize the hardness from the circuit model to arbitrary ones.

Definition 11 *We say that a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is $(\varepsilon, \mathcal{C})$ -hard, for a complexity class \mathcal{C} , if for any $C \in \mathcal{C}$, $\Pr_{x \in \mathcal{U}_n} [C(x) \neq f(x)] > \varepsilon$. We will call f ε -hard when the complexity class \mathcal{C} is clear.*

The parameter ε in the definition above is allowed to be a function of n , so a better notation should be $\varepsilon(n)$, but for simplicity we drop the parameter n . In previous works, people usually consider hardness against circuits, i.e., with $\mathcal{C} = \text{SIZE}(s)$ for some s . Since we will consider hardness against other complexity classes, we introduce this slightly more general definition. Next, we define the notions of weakly black-box hardness amplification [RTV04].

Definition 12 *Let \mathcal{C} and $\bar{\mathcal{C}}$ be complexity classes. We say that an oracle algorithm $\text{AMP}^{(\cdot)} : \{0, 1\}^{\bar{n}} \rightarrow \{0, 1\}$ realizes a weakly black-box $(n, \varepsilon, \bar{\varepsilon}, \mathcal{C}, \bar{\mathcal{C}})$ hardness amplification, if given any $(\varepsilon, \mathcal{C})$ -hard function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, the function $\text{AMP}^f : \{0, 1\}^{\bar{n}} \rightarrow \{0, 1\}$ is $(\bar{\varepsilon}, \bar{\mathcal{C}})$ -hard.*

Here, the reduction from the initial function f to the harder function is done in a black-box way, as the harder function AMP^f only uses f as an oracle.

One can also define the notions of strongly black-box hardness amplification. For their definitions, we refer the readers to Chapter 2 in this thesis.

3.3 Impossibility of Hardness Amplification in TIME(t)

In this section, we show that if a weakly black-box hardness amplification realized in TIME(t), can amplify hardness beyond an $O(t)$ factor, then it must basically embed a hard function in it.

Theorem 9 *Suppose a weakly black-box $(n, \varepsilon, \bar{\varepsilon}, \mathcal{C}, \bar{\mathcal{C}})$ hardness amplification can be realized in TIME(t) with $2^{-n/2} \leq \varepsilon \leq o(\bar{\varepsilon}/t)$, $\mathcal{C} = \text{SIZE}(2^{n/3})$, and $\bar{\mathcal{C}}$ being any complexity class. Then one can obtain from it an $(\bar{\varepsilon}/2, \bar{\mathcal{C}})$ -hard function $\bar{A} : \{0, 1\}^{\text{poly}(n)} \rightarrow \{0, 1\}$ computable in TIME(t).*

Using $t = 2^{o(n)}$, this implies that if such a hardness amplification from worst-case hardness to mild hardness can be realized in sub-exponential time (or sub-linear space), then it must basically embed a mildly hard function in it. Furthermore, since a function in TIME(t) cannot be hard against TIME(t), we have the following unconditional impossibility result on weakly black-box hardness amplification.

Corollary 5 *It is impossible to realize a weakly black-box $(n, \varepsilon, \bar{\varepsilon}, \mathcal{C}, \bar{\mathcal{C}})$ hardness amplification in TIME(t), with $2^{-n/2} \leq \varepsilon \leq o(\bar{\varepsilon}/t)$, $\mathcal{C} = \text{SIZE}(2^{n/3})$, and $\bar{\mathcal{C}} \supseteq \text{TIME}(t)$.*

Now we prove Theorem 9.

Proof.(of Theorem 9) Assume that such a weakly hardness amplification can be realized by $\text{AMP} \in \text{TIME}(t)$. We will show that the function $\text{AMP}^{\vec{0}}$ is $(\bar{\varepsilon}/2, \bar{\mathcal{C}})$ -hard, where $\vec{0}$ is the constant zero function which always outputs zero for every input. The idea is to choose a certain kind of random function f such that f is likely to be hard and AMP is unlikely to tell it apart from

the function $\vec{0}$. A natural candidate is the following, which is obtained by adding random noise of certain rate to the function $\vec{0}$.²

Definition 13 Let \mathbb{F}^δ denote the distribution of functions $f : \{0, 1\}^n \rightarrow \{0, 1\}$ such that for any $x \in \{0, 1\}^n$, $f(x) = 0$ with probability $1 - \delta$, and $f(x)$ is given a random bit with probability δ .

Let $\delta = 4\varepsilon$. It remains to show that such a random function can do the work.

Lemma 18 $\Pr_{f \in \mathbb{F}^\delta}[\text{AMP}^f \text{ is } (\bar{\varepsilon}, \bar{\mathcal{C}})\text{-hard}] \geq 1 - 2^{-\Omega(n)}$.

Proof. Consider any $D \in \mathcal{C} = \text{SIZE}(2^{n/3})$. Note that for any $x \in \{0, 1\}^n$, $\Pr_f[D(x) \neq f(x)] \geq \delta/2 = 2\varepsilon$. Define $T(f) = 1$ if and only if

$$\Pr_x[D(x) \neq f(x)] < \varepsilon.$$

Let $\tilde{T}(f) = \Pr_x[D(x) \neq f(x)]$. So we have $\mathbb{E}_f[\tilde{T}(f)] \geq 2\varepsilon$ and $\Pr_f[T(f) = 1] \leq 2^{-\Omega(\varepsilon 2^n)}$ by a Chernoff bound. Now it is clear that

$$\begin{aligned} & \Pr_f[f \text{ is not } (\varepsilon, \mathcal{C})\text{-hard}] \\ &= \Pr_f[\exists D \in \text{SIZE}(2^{n/3}) : T(f) = 1] \\ &\leq 2^{O(2^{n/3} \cdot n/3)} \cdot 2^{-\Omega(\varepsilon 2^n)} \\ &\leq 2^{-\Omega(n)}. \end{aligned}$$

As a result, $\Pr_f[\text{AMP}^f \text{ is not } (\bar{\varepsilon}, \bar{\mathcal{C}})\text{-hard}] \leq \Pr_f[f \text{ is not } (\varepsilon, \mathcal{C})\text{-hard}] \leq 2^{-\Omega(n)}$.

□

Lemma 19 $\Pr_{f \in \mathbb{F}^\delta}[\Delta(\text{AMP}^f, \text{AMP}^{\vec{0}}) \leq \bar{\varepsilon}/2] \geq 1 - o(1)$.

Proof. For any input \bar{x} , $\text{AMP}^f(\bar{x}) \neq \text{AMP}^{\vec{0}}(\bar{x})$ only when $\text{AMP}^f(\bar{x})$ ever makes an oracle query x to f with $f(x) \neq 0$. AMP runs in time t and

²A similar idea also appeared in [LTW05a] for the problem of amplifying hardness of one-way permutations.

3.3. IMPOSSIBILITY OF HARDNESS AMPLIFICATION IN TIME(T)⁶⁵

can make at most t queries to the oracle, so for every \bar{x} , $\Pr_f[\text{AMP}^f(\bar{x}) \neq \text{AMP}^{\bar{0}}(\bar{x})] \leq t \cdot \delta = o(\bar{\varepsilon})$. Define $\text{Bad}(f) = 1$ if and only if $\Pr_{\bar{x}}[\text{AMP}^f(\bar{x}) \neq \text{AMP}^{\bar{0}}(\bar{x})] \geq \bar{\varepsilon}/2$. Then $\Pr_{f,\bar{x}}[\text{AMP}^f(\bar{x}) \neq \text{AMP}^{\bar{0}}(\bar{x})] = o(\bar{\varepsilon})$, and by Markov's inequality, $\Pr_f[\text{Bad}(f) = 1] = o(1)$. \square

From the two lemmas above, there exists a function f such that AMP^f is $(\bar{\varepsilon}, \bar{\mathcal{C}})$ -hard and $\Delta(\text{AMP}^f, \text{AMP}^{\bar{0}}) \leq \bar{\varepsilon}/2$. This implies that the function $\text{AMP}^{\bar{0}}$ is $(\bar{\varepsilon}/2, \bar{\mathcal{C}})$ -hard. Since $\text{AMP}^{\bar{0}}$ is clearly computable in $\text{TIME}(t)$, we have Theorem 9. \square

In fact, we can have essentially the same impossibility result even if we replace the class $\text{SIZE}(2^{n/3})$ by any uniform complexity class \mathcal{B} equipped with an advice of length $2^{n/3}$, denoted as $\mathcal{B}/2^{n/3}$. This means that such a weakly black-box hardness amplification must be highly non-uniform.

Theorem 10 *It is impossible to realize a weakly black-box $(n, \varepsilon, \bar{\varepsilon}, \mathcal{C}, \bar{\mathcal{C}})$ hardness amplification in $\text{TIME}(t)$, with $2^{-n/2} \leq \varepsilon \leq o(\bar{\varepsilon}/(t \cdot n^2))$, $\mathcal{C} = \mathcal{B}/2^{n/3}$ for any uniform complexity class \mathcal{B} , and $\bar{\mathcal{C}} \supseteq \text{TIME}(t)$.*

Proof. The proof can be slightly modified from that for Theorem 9. We will use the following lemma, known as the Borel-Cantelli Lemma (see e.g. [Bil95]).

Lemma 20 *Let E_1, E_2, \dots be a sequence of probability events on the same probability space. Suppose that $\sum_{n=1}^{\infty} \Pr[E_n] < \infty$. Then $\Pr[\bigwedge_{k=1}^{\infty} \bigvee_{n \geq k} E_n] = 0$.*

The proof is largely based on that for Theorem 9, but here we need to treat things in a more careful way. As now we often need to talk about a sequence of functions, one on each input length, we change the notation slightly by adding a subscript n to a function of input length n . That is, now we write $f_n : \{0, 1\}^n \rightarrow \{0, 1\}$, and write f for the sequence of functions $(f_n)_{n \in \mathbb{N}}$. Similarly, we write \mathbb{F}_n^δ for the distribution of functions $f_n : \{0, 1\}^n \rightarrow \{0, 1\}$ in the proof of Theorem 9, and write \mathbb{F}^δ for the sequence of distributions $(\mathbb{F}_n^\delta)_{n \in \mathbb{N}}$.

Lemma 21 *With measure one over $f \in \mathbb{F}^\delta$, AMP^f is $(\bar{\varepsilon}, \bar{\mathcal{C}})$ -hard.*

Proof. Consider any Turing machine M in the uniform complexity class \mathcal{B} . Consider any input length n and let E_n denote the event that there exists an advice $\nu \in \{0, 1\}^{2^{n/3}}$ such that $\Pr_{x \in \mathcal{U}_n} [M_n^\nu(x) \neq f_n(x)] < \varepsilon$. Then as in Lemma 18, one can show that $\Pr_{f_n \in \mathbb{F}_n^\delta} [E_n] \leq 2^{O(2^{n/3} \cdot n/3)} \cdot 2^{-\Omega(\varepsilon 2^n)} < 1/n^2$. Since $\sum_{n=1}^\infty 1/n^2 < \infty$, the Borel-Cantelli Lemma implies that E_n happens for infinitely many n with measure zero over $f \in \mathbb{F}^\delta$. Since there are only a countable many Turing machines M 's, we conclude that f is not $(\varepsilon, \mathcal{C})$ -hard with measure zero over $f \in \mathbb{F}^\delta$. As AMP^f is not $(\bar{\varepsilon}, \bar{\mathcal{C}})$ -hard only when f is not $(\varepsilon, \mathcal{C})$ -hard, we have the lemma. \square

Lemma 22 *With measure one over $f \in \mathbb{F}^\delta$, $\Delta(\text{AMP}^{f_n}, \text{AMP}^{\bar{0}_n}) \geq \bar{\varepsilon}/2$ for only finitely many n .*

Proof. As in the proof of Lemma 19, now with $\varepsilon \leq o(\bar{\varepsilon}/(t \cdot n^2))$, one can show that for any n , $\Pr_{f_n, \bar{x}} [\text{AMP}^{f_n}(\bar{x}) \neq \text{AMP}^{\bar{0}_n}(\bar{x})] = o(\bar{\varepsilon}/n^2)$, and $\Pr_{f_n} [\Delta(\text{AMP}^{f_n}, \text{AMP}^{\bar{0}_n}) \geq \bar{\varepsilon}/2] < 1/n^2$. Since $\sum_{n=1}^\infty 1/n^2 < \infty$, the lemma immediately follows from the Borel-Cantelli Lemma. \square

Then as in Theorem 9, the two lemmas above imply that the function $\text{AMP}^{\bar{0}}$ is $(\bar{\varepsilon}/2, \bar{\mathcal{C}})$ -hard. Since $\text{AMP}^{\bar{0}}$ is computable in $\text{TIME}(t)$, it cannot be hard against any $\bar{\mathcal{C}} \supseteq \text{TIME}(t)$, and we have Theorem 10. \square

3.4 Impossibility Results in $\Sigma_k\text{P}$

In this subsection, we consider weakly black-box hardness amplification realized in $\Sigma_k\text{P}$ (or PH). We will show that if it can amplify hardness beyond a certain factor, then it must basically embed a hard function in it.

Theorem 11 *Suppose a weakly black-box $(n, \varepsilon, \bar{\varepsilon}, \mathcal{C}, \bar{\mathcal{C}})$ hardness amplification can be realized in NP ($\Sigma_k\text{P}$, respectively), with $2^{-n/2} \leq \varepsilon \leq \bar{\varepsilon}^2/n^{\omega(1)}$, $\mathcal{C} = \text{SIZE}(2^{n/3})$, and $\bar{\mathcal{C}}$ satisfying $\bar{\mathcal{C}}/\text{poly} = \bar{\mathcal{C}}$. Then one can obtain from it an $(\bar{\varepsilon}/3, \bar{\mathcal{C}})$ -hard function $\bar{A} : \{0, 1\}^{\text{poly}(n)} \rightarrow \{0, 1\}$ computable in NP ($\Sigma_k\text{P}$, respectively).*

Since a function in NP ($\Sigma_k\text{P}$, resp.) cannot be hard against NP/poly ($\Sigma_k\text{P}/\text{poly}$, resp.), we have the following unconditional impossibility result on weakly black-box hardness amplification.

Corollary 6 *It is impossible to realize a weakly black-box $(n, \varepsilon, \bar{\varepsilon}, \mathcal{C}, \bar{\mathcal{C}})$ hardness amplification in NP ($\Sigma_k\text{P}$, resp.), with $2^{-n/2} \leq \varepsilon \leq \bar{\varepsilon}^2/n^{\omega(1)}$, $\mathcal{C} = \text{SIZE}(2^{n/3})$, and any $\bar{\mathcal{C}} \supseteq \text{NP}/\text{poly}$ ($\Sigma_k\text{P}/\text{poly}$, resp.) satisfying $\bar{\mathcal{C}}/\text{poly} = \bar{\mathcal{C}}$.*

We will need the notion of random restriction [FSS84, Has86]. A restriction ρ on N variables is an element of $\{0, 1, \star\}^N$, or seen as a function $\rho : [N] \rightarrow \{0, 1, \star\}$. A variable is fixed by ρ if it receives a value in $\{0, 1\}$ while a variable remains free if it receives the symbol \star . For a string $y \in \{0, 1\}^N$ and a restriction $\rho \in \{0, 1, \star\}^N$, let $y|_\rho \in \{0, 1\}^N$ be the restriction of y with respect to ρ : for $i \in [N]$, the i 'th bit of $y|_\rho$ is y_i if $\rho_i = \star$ and is ρ_i if $\rho_i \in \{0, 1\}$.

Suppose there exists such a weakly black-box hardness amplification, with AMP realized in NP ($\Sigma_k\text{P}$, resp.). Then AMP can be computed by an $\text{AC}(c, 2^{nc})$ circuit, for some constant c , with the truth table of the oracle function given as part of the input (c.f. [FSS84]). We will show how to derive a hard function from it.

The idea, which basically follows that of Viola's [Vio05], is the following. We know that a random function f is likely to be hard, and so is the function AMP^f , but we do not know which f gives a hard function. One attempt is to include f as part of the input in the new function, but the description of f is too long. The idea is that by choosing a suitable random restriction $\bar{\rho}$, the function $f|_{\bar{\rho}}$ is still likely to be hard, and so is the function $\text{AMP}^{f|_{\bar{\rho}}}$. On the other hand, a random restriction is likely to kill off the effect of a random function f on $\text{AMP}^{f|_{\bar{\rho}}}$, so it becomes possible to replace the random function by a pseudo-random one \bar{f} , which has a short description. Therefore, if we have a random restriction which has a short description and satisfies the properties above, we can define the new function which includes $\bar{\rho}$ and \bar{f} as part of the input and computes the function $\text{AMP}^{\bar{f}|_{\bar{\rho}}}$. The existence of such

a random restriction is guaranteed by the following lemma of Viola's [Vio05]. For our purpose here, we state it in a slightly more general form.

Lemma 23 [Vio05] *For any $n \in \mathbb{N}$, any constant c , and any $\varepsilon, \bar{\varepsilon} \in (0, 1)$ such that $2^{-n} \leq \varepsilon \leq \bar{\varepsilon}^2/n^{\omega(1)}$, there is a distribution $\bar{\mathbb{R}}$ on restrictions $\bar{\rho} : \{0, 1\}^n \rightarrow \{0, 1, \star\}$ such that the following holds.*

- *Every $\bar{\rho} \in \bar{\mathbb{R}}$ can be described by $\text{poly}(n)$ bits, and given such a description and $x \in \{0, 1\}^n$, one can compute $\bar{\rho}(x)$ in time $\text{poly}(n)$.*
- $\Pr_{\bar{\rho} \in \bar{\mathbb{R}}} [|\{x : \bar{\rho}(x) = \star\}| < 3\varepsilon 2^n] = o(\bar{\varepsilon})$.
- *For any $C : \{0, 1\}^{2^n} \rightarrow \{0, 1\} \in \text{AC}(c, 2^{n^c})$, $\Pr_{\bar{\rho} \in \bar{\mathbb{R}}; y, y' \in \mathcal{U}_{2^n}} [C(y|_{\bar{\rho}}) \neq C(y'|_{\bar{\rho}})] = o(\bar{\varepsilon}^2)$.*

Let \mathbb{F} denote the set of all functions $f : \{0, 1\}^n \rightarrow \{0, 1\}$. It remains to show that the random restriction given in Lemma 23 can accomplish the task we discussed above. First, by using the second item of Lemma 23, we show that the function $\text{AMP}^{f|_{\bar{\rho}}}$ is hard with high probability over $\bar{\rho} \in \bar{\mathbb{R}}$ and $f \in \mathbb{F}$.

Lemma 24 $\Pr_{\bar{\rho} \in \bar{\mathbb{R}}, f \in \mathbb{F}} [\text{AMP}^{f|_{\bar{\rho}}} \text{ is not } (\bar{\varepsilon}, \bar{\mathcal{C}})\text{-hard}] = o(\bar{\varepsilon})$.

Proof. Call a restriction $\bar{\rho} \in \bar{\mathbb{R}}$ *good* if $|\{x : \bar{\rho}(x) = \star\}| \geq 3\varepsilon 2^n$. Consider any good $\bar{\rho}$ and any $D \in \mathcal{C} = \text{SIZE}(2^{n/3})$. Note that for any x such that $\bar{\rho}(x) = \star$, $\Pr_f [D(x) \neq f|_{\bar{\rho}}(x)] = 1/2$. Define $\text{Bad}(f) = 1$ if and only if $\Pr_x [D(x) \neq f|_{\bar{\rho}}(x)] < \varepsilon$ and define $\tilde{B}(f) = \Pr_x [D(x) \neq f|_{\bar{\rho}}(x)]$. Thus, $\mathbb{E}_f [\tilde{B}(f)] \geq 3\varepsilon/2$, and $\Pr_f [\text{Bad}(f) = 1] \leq 2^{-\Omega(\varepsilon 2^n)}$, by a Chernoff bound. As a result, for any good $\bar{\rho}$, $\Pr_f [f|_{\bar{\rho}} \text{ is not } (\varepsilon, \mathcal{C})\text{-hard}]$ is

$$\Pr_f [\exists D \in \text{SIZE}(2^{n/3}) : \text{Bad}(f) = 1] \leq 2^{O(2^{n/3} \cdot n/3)} \cdot 2^{-\Omega(\varepsilon 2^n)} = o(\bar{\varepsilon}).$$

From Lemma 23, $\Pr_{\bar{\rho} \in \bar{\mathbb{R}}} [\bar{\rho} \text{ is not good}] = o(\bar{\varepsilon})$, and by definition, $\text{AMP}^{f|_{\bar{\rho}}}$ is $(\bar{\varepsilon}, \bar{\mathcal{C}})$ -hard whenever $f|_{\bar{\rho}}$ is $(\varepsilon, \mathcal{C})$ -hard. Therefore,

$$\begin{aligned} & \Pr_{\bar{\rho}, f} [\text{AMP}^{f|_{\bar{\rho}}} \text{ is not } (\bar{\varepsilon}, \bar{\mathcal{C}})\text{-hard}] \\ & \leq \Pr_{\bar{\rho}} [\bar{\rho} \text{ is not good}] + \Pr_{\bar{\rho}, f} [f|_{\bar{\rho}} \text{ is not } (\varepsilon, \mathcal{C})\text{-hard} \mid \bar{\rho} \text{ is good}] \\ & = o(\bar{\varepsilon}). \quad \square \end{aligned}$$

From this lemma, we know that the function $\text{AMP}^{f \upharpoonright_{\bar{\rho}}}$ is hard for most $\bar{\rho} \in \bar{\mathbf{R}}$ and $f \in \mathbb{F}$, but we do not know which $\bar{\rho}$ and f give a hard function. While $\bar{\rho}$ has a short description, f does not, so we cannot just include both $\bar{\rho}$ and f as part of the input. Viola's approach in [Vio05] is to remove the dependence of f altogether, by considering the function A' which on input $(\bar{x}, \bar{\rho})$ outputs the majority value of $\text{AMP}^{f \upharpoonright_{\bar{\rho}}}(\bar{x})$ over $f \in \mathbb{F}$. The hardness of A' is guaranteed by the third item in Lemma 23, because for most $\bar{\rho}$ and for most f , the function $\text{AMP}^{f \upharpoonright_{\bar{\rho}}}$ is hard and $A'(\bar{\rho}, \cdot)$ is close to it. However, to compute such a majority value over $f \in \mathbb{F}$ costs one additional level in the polynomial hierarchy in [Vio05], and with $\text{AMP} \in \mathbf{NP}$ ($\Sigma_k\mathbf{P}$, respectively), Viola needs at least $\Sigma_2\mathbf{P}$ (or $\Sigma_{k+1}\mathbf{P}$) to compute the function A' . Our idea is to replace the random function by a pseudorandom one.

Definition 14 Let $\text{NIS} : \{0, 1\}^{r_1} \rightarrow \{0, 1\}^{2^n}$ be Nisan's $o(\bar{\epsilon}^2)$ -PRG for $\text{AC}(c+2, 2^{n^c+2})$, with $r_1 = \text{poly}(n)$ [Nis91]. Let $\bar{\mathcal{F}}$ be the class of functions $\bar{f}_{z_1} : \{0, 1\}^n \rightarrow \{0, 1\}$, with $z_1 \in \{0, 1\}^{r_1}$, defined as $\bar{f}_{z_1}(x) = \text{NIS}(z_1)_x$, the x 'th bit in $\text{NIS}(z_1)$.

There seems to be an obstacle in front of us. Unlike a random function, such a pseudo-random \bar{f} is not hard at all. Then how do we guarantee the hardness of the function $\text{AMP}^{\bar{f} \upharpoonright_{\bar{\rho}}}$? We resolve this by showing that the function $\text{AMP}^{\bar{f} \upharpoonright_{\bar{\rho}}}$ is likely to be close to a hard function $\text{AMP}^{f \upharpoonright_{\bar{\rho}}}$. For this, we first show the following.

Lemma 25 For any $\bar{x} \in \{0, 1\}^{\bar{n}}$, $\Pr_{\bar{\rho} \in \bar{\mathbf{R}}, f \in \mathbb{F}, \bar{f} \in \bar{\mathcal{F}}}[\text{AMP}^{f \upharpoonright_{\bar{\rho}}}(\bar{x}) \neq \text{AMP}^{\bar{f} \upharpoonright_{\bar{\rho}}}(\bar{x})] = o(\bar{\epsilon}^2)$.

Proof. Let $N = 2^n$. Fix any $\bar{x} \in \{0, 1\}^{\bar{n}}$, and let $C : \{0, 1\}^N \rightarrow \{0, 1\}$ be the function which takes a function $g : \{0, 1\}^n \rightarrow \{0, 1\}$, seen as $g \in \{0, 1\}^N$, as the input and outputs the value $\text{AMP}^g(\bar{x})$. Clearly, $C \in \text{AC}(c, 2^{n^c})$. Now for $\bar{\rho} \in \bar{\mathbf{R}}$, let $\bar{C}_{\bar{\rho}} : \{0, 1\}^N \times \{0, 1\}^N \rightarrow \{0, 1\}$ be the function such that $\bar{C}_{\bar{\rho}}(f, f') = 1$ if and only if $C(f \upharpoonright_{\bar{\rho}}) \neq C(f' \upharpoonright_{\bar{\rho}})$, which is computable by an

$\text{AC}(c + 2, 2^{n^c+2})$ circuit. Since NIS is an $o(\bar{\varepsilon}^2)$ -PRG for such circuits,

$$\begin{aligned} & \left| \Pr_{\bar{\rho}, f, \bar{f}} [\bar{C}_{\bar{\rho}}(f, \bar{f}) = 1] - \Pr_{\bar{\rho}, f, f'} [\bar{C}_{\bar{\rho}}(f, f') = 1] \right| \\ & \leq \mathbb{E}_{\bar{\rho}, f} \left[\left| \Pr_{z_1} [\bar{C}_{\bar{\rho}}(f, \text{NIS}(z_1)) = 1] - \Pr_{f'} [\bar{C}_{\bar{\rho}}(f, f') = 1] \right| \right] = o(\bar{\varepsilon}^2). \end{aligned}$$

From Lemma 23, $\Pr_{\bar{\rho}, f, f'} [\bar{C}_{\bar{\rho}}(f, f') = 1] = o(\bar{\varepsilon}^2)$, and we have

$$\Pr_{\bar{\rho}, f, \bar{f}} [\bar{C}_{\bar{\rho}}(f, \bar{f}) = 1] \leq \Pr_{\bar{\rho}, f, f'} [\bar{C}_{\bar{\rho}}(f, f') = 1] + o(\bar{\varepsilon}^2) = o(\bar{\varepsilon}^2).$$

□

From this, one can show that the function $\text{AMP}^{f \uparrow \bar{\rho}}$ is hard for most $\bar{\rho} \in \bar{\mathcal{R}}$ and $\bar{f} \in \bar{\mathcal{F}}$.

Lemma 26 $\Pr_{\bar{\rho} \in \bar{\mathcal{R}}, \bar{f} \in \bar{\mathcal{F}}} [\text{AMP}^{f \uparrow \bar{\rho}} \text{ is not } (\bar{\varepsilon}/2, \bar{\mathcal{C}})\text{-hard}] = o(\bar{\varepsilon})$.

Proof. From Lemma 24, we know that $\Pr_{\bar{\rho}, f} [\text{AMP}^{f \uparrow \bar{\rho}} \text{ is not } (\bar{\varepsilon}, \bar{\mathcal{C}})\text{-hard}] = o(\bar{\varepsilon})$. From Lemma 25, we know that $\Pr_{\bar{\rho}, f, \bar{f}, \bar{x}} [\text{AMP}^{f \uparrow \bar{\rho}}(\bar{x}) \neq \text{AMP}^{\bar{f} \uparrow \bar{\rho}}(\bar{x})] = o(\bar{\varepsilon}^2)$, and by Markov's inequality, we have that $\Pr_{\bar{\rho}, f, \bar{f}} [\Delta(\text{AMP}^{f \uparrow \bar{\rho}}, \text{AMP}^{\bar{f} \uparrow \bar{\rho}}) > \bar{\varepsilon}/2] = o(\bar{\varepsilon})$. Note that $\text{AMP}^{f \uparrow \bar{\rho}}$ is $(\bar{\varepsilon}/2, \bar{\mathcal{C}})$ -hard when $\text{AMP}^{f \uparrow \bar{\rho}}$ is $(\bar{\varepsilon}, \bar{\mathcal{C}})$ -hard and $\Delta(\text{AMP}^{f \uparrow \bar{\rho}}, \text{AMP}^{\bar{f} \uparrow \bar{\rho}}) \leq \bar{\varepsilon}/2$. So, $\Pr_{\bar{\rho}, \bar{f}} [\text{AMP}^{\bar{f} \uparrow \bar{\rho}} \text{ is not } (\bar{\varepsilon}/2, \bar{\mathcal{C}})\text{-hard}]$ is at most $\Pr_{\bar{\rho}, f} [\text{AMP}^{f \uparrow \bar{\rho}} \text{ is not } (\bar{\varepsilon}, \bar{\mathcal{C}})\text{-hard}] + \Pr_{\bar{\rho}, f, \bar{f}} [\Delta(\text{AMP}^{f \uparrow \bar{\rho}}, \text{AMP}^{\bar{f} \uparrow \bar{\rho}}) > \bar{\varepsilon}/2] = o(\bar{\varepsilon})$. □

From the lemma above, we know that $\text{AMP}^{f \uparrow \bar{\rho}}$ is hard for most $\bar{\rho}$ and \bar{f} . We do not know which $\bar{\rho}$ and \bar{f} give a hard function, but since they have short description, we can include them as part of the input. Define the function $\bar{A} : \{0, 1\}^{\bar{n}} \times \bar{\mathcal{R}} \times \bar{\mathcal{F}} \rightarrow \{0, 1\}$ as $\bar{A}(\bar{x}, \bar{\rho}, \bar{f}) = \text{AMP}^{\bar{f} \uparrow \bar{\rho}}(\bar{x})$. Note that the input length of \bar{A} is at most $\text{poly}(n)$ as $\bar{\rho}$ and \bar{f} can be described by $\text{poly}(n)$ bits.

Lemma 27 *The function \bar{A} is $(\bar{\varepsilon}/3, \bar{\mathcal{C}})$ -hard.*

Proof. Consider any $\bar{D} : \{0, 1\}^{\bar{n}} \times \bar{\mathcal{R}} \times \bar{\mathcal{F}} \rightarrow \{0, 1\} \in \bar{\mathcal{C}}$. Note that for any $\bar{\rho} \in \bar{\mathcal{R}}$ and $\bar{f} \in \bar{\mathcal{F}}$ such that $\text{AMP}^{\bar{f} \uparrow \bar{\rho}}$ is $(\bar{\varepsilon}/2, \bar{\mathcal{C}})$ -hard, $\Pr_{\bar{x}} [\bar{D}(\bar{x}, \bar{\rho}, \bar{f}) =$

$\bar{A}(\bar{x}, \bar{\rho}, \bar{f})] = \Pr_{\bar{x}}[\bar{D}(\bar{x}, \bar{\rho}, \bar{f}) = \text{AMP}^{\bar{f}|_{\bar{\rho}}}(\bar{x})] < 1 - \bar{\varepsilon}/2$. Therefore,

$$\begin{aligned} & \Pr_{\bar{x}, \bar{\rho}, \bar{f}} [\bar{D}(\bar{x}, \bar{\rho}, \bar{f}) = \bar{A}(\bar{x}, \bar{\rho}, \bar{f})] \\ & < \Pr_{\bar{\rho}, \bar{f}} [\text{AMP}^{\bar{f}|_{\bar{\rho}}} \text{ is not } (\bar{\varepsilon}/2, \bar{\mathcal{C}})\text{-hard}] + 1 - \bar{\varepsilon}/2 \\ & \leq 1 - \bar{\varepsilon}/3. \end{aligned}$$

□

Now we prove Theorem 11.

Proof.(of Theorem 11) By Lemma 27, \bar{A} is $(\bar{\varepsilon}/3, \bar{\mathcal{C}})$ -hard. Note that given $\bar{f} \in \bar{\mathcal{F}}$, $\bar{\rho} \in \bar{\mathcal{R}}$, and any $x \in \{0, 1\}^n$, one can compute $\bar{f}|_{\bar{\rho}}(x)$ in time $\text{poly}(n)$. Therefore, the function \bar{A} belongs to NP, the same class as AMP. □

Similar to Theorem 10, we have the following.

Theorem 12 *It is impossible to realize a weakly black-box $(n, \varepsilon, \bar{\varepsilon}, \mathcal{C}, \bar{\mathcal{C}})$ hardness amplification in NP ($\Sigma_k\text{P}$, resp.) with $2^{-n/2} \leq \varepsilon \leq \bar{\varepsilon}^2/n^{\omega(1)}$, $\mathcal{C} = \mathcal{B}/2^{n/3}$ for any uniform complexity class \mathcal{B} , and any $\bar{\mathcal{C}} \supseteq \text{NP}/\text{poly}$ ($\Sigma_k\text{P}/\text{poly}$, resp.) satisfying $\bar{\mathcal{C}}/\text{poly} = \bar{\mathcal{C}}$.*

Proof. Similar to how we modify the proof of Theorem 9 to prove Theorem 10, we can also modify the proof of Theorem 11 to prove Theorem 12. Again, we will use the the Borel-Cantelli Lemma. We will also change the notation slightly by adding a subscript n to a function of input length n . That is, now we write $\bar{f}_n \in \bar{\mathcal{F}}_n$ and $\bar{\rho}_n \in \bar{\mathcal{R}}_n$ for functions and restrictions on inputs of length n , respectively, and we write $\bar{f}, \bar{\rho}, \bar{\mathcal{F}}, \bar{\mathcal{R}}$ for the sequences $(\bar{f}_n)_{n \in \mathbb{N}}, (\bar{\rho}_n)_{n \in \mathbb{N}}, (\bar{\mathcal{F}}_n)_{n \in \mathbb{N}}, (\bar{\mathcal{R}}_n)_{n \in \mathbb{N}}$, respectively. Then it is easy to check that one can modify the proof in Theorem 11 to show the following lemma.

Lemma 28 *With measure one over $\bar{\rho} \in \bar{\mathcal{R}}$ and $\bar{f} \in \bar{\mathcal{F}}$, $\text{AMP}^{\bar{f}|_{\bar{\rho}}}$ is $(\bar{\varepsilon}/2, \bar{\mathcal{C}})$ -hard.*

Unlike in Theorem 11, we now cannot show show that the function \bar{A} is hard. Instead, from the lemma above, we know that for any large enough n , there exists $\bar{\rho}_n \in \bar{\mathcal{R}}_n$ and $\bar{f}_n \in \bar{\mathcal{F}}_n$ such that the function $\bar{A}(\bar{\rho}_n, \bar{f}_n, \cdot) =$

$\text{AMP}^{\bar{f}_n \upharpoonright_{\bar{\rho}_n}}(\cdot)$ is hard. We can see such $\bar{\rho}_n$'s and \bar{f}_n 's as advice strings, which are of length $\text{poly}(n)$, and as a result we have a hard function which is computable in NP/poly ($\Sigma_k\text{P}/\text{poly}$, resp.). Since such a function cannot be hard against any $\bar{\mathcal{C}} \supseteq \text{NP}/\text{poly}$ ($\Sigma_k\text{P}/\text{poly}$, resp.), we have Theorem 12. \square



Chapter 4

Hardness Amplification in NP

4.1 Introduction

In this chapter, we focus on the task of transforming mild hardness to average-case hardness for the complexity class NP. One attempt is to use Yao's XOR lemma [Yao82, GNW95], which transforms a given function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ into a function $f' : (\{0, 1\}^n)^k \rightarrow \{0, 1\}$ defined by $f'(x_1, \dots, x_k) = f(x_1) \oplus \dots \oplus f(x_k)$. However, we do not know if this works here, since we do not know if NP is closed under the XOR operation. O'Donnell [OD02] gave the first result along this line, showing how to convert any balanced function $f \in \text{NP}$ which is mildly hard for polynomial-size circuits into another $f' \in \text{NP}$ which is $(1/2 - 1/n^{1/2-\alpha})$ -hard for polynomial-size circuits, for any constant $\alpha > 0$. He considered transformations of the form: $f'(x_1, \dots, x_k) = C(f(x_1), \dots, f(x_k))$, where C is a polynomial-time computable *monotone* function. Then he used the "tribes" function and the "recursive majority" function, and took their composition as the function C . Recently, Healy et al. [HVV04] were able to amplify hardness beyond $1/2 - 1/\text{poly}(n)$, showing how to convert any balanced function in NP which is mildly hard for circuits of size $s(n)$ into one in NP which is $(1/2 - 1/s'(n))$ -hard for circuits of size $s'(n)$, with $s'(n) = s(n^{1/2})^{\Omega(1)}$. In particular, $s'(n) = n^{\omega(1)}$ when $s(n) = n^{\omega(1)}$, $s'(n) = 2^{n^{\Omega(1)}}$ when $s(n) = 2^{n^{\Omega(1)}}$,

and $s'(n) = 2^{\Omega(n^{1/2})}$ when $s(n) = 2^{\Omega(n)}$. A key source of their improvement came from derandomizing O'Donnell's proof (the other source being the use of nondeterminism in computing the new function). They observed that the function C used by O'Donnell can be computed by a small-size read-once branching program and thus can be fooled by the pseudorandom generator of Nisan [Nis92]. Unfortunately, this generator becomes the bottleneck of their approach when $s(n) = 2^{\Omega(n)}$, which prevents them from achieving the goal of having $s'(n) = 2^{\Omega(n)}$.

In this chapter, we make a further progress towards this goal, at the high end of the spectrum:

Theorem 13 *Suppose there is a balanced function in NP which is mildly hard for circuits of size $s(n) = 2^{\Omega(n)}$. Then there is a function in NP which is $(1/2 - 1/s'(n))$ -hard for circuits of size $s'(n)$, with $s'(n) = 2^{\Omega(n^{2/3})}$.*

Our improvement comes from a closer look into the structure of the function C used by Healy et al., which enables us to construct a better pseudorandom generator to fool C . More precisely, we observe that the function C , which is the composition of the tribes function (a DNF) with the recursive majority function, can be seen as some kind of combinatorial rectangle, though the range in each dimension is large. This suggests that we fool each dimension by a separate copy of Nisan's generator and provide their seeds using the output of Lu's pseudorandom generator for rectangles [Lu02]. Our generator then is the composition of these two generators.

4.1.1 Organization of this chapter

First, some preliminaries are given in Section 4.2. Then in Section 4.3, we give the proof of our main theorem in this chapter.

4.2 Preliminaries

First of all, we recall the definition of a hard function. For $0 \leq \delta \leq 1/2$ and $s(n) \leq 2^{O(n)}$, we say that a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is δ -hard for size $s(n)$ if every circuit of size $s(n)$ fails to compute f on at least a δ fraction of inputs. As shown by Impagliazzo [Im95], one can basically see a δ -hard function as a δ -random function defined below, as they cannot be distinguished by circuits of size slightly smaller than $s(n)$.

Definition 15 *A probabilistic function $g : \{0, 1\}^n \rightarrow \{0, 1\}$ is called δ -random if it is balanced and there is a subset $H \subset \{0, 1\}^n$ with $|H| = 2\delta 2^n$ such that $g(x)$ is an independent random bit for $x \in H$ and $g(x)$ is deterministic for $x \notin H$.*

Note that any probabilistic function g can also be seen as a deterministic function with respect to a random string y , and we will use g_y to denote this deterministic function.

4.2.1 Hardness Amplification

Given a hard function f , one would like to transform it into a harder function f' . One typical way is to apply a function $C : \{0, 1\}^k \rightarrow \{0, 1\}$ to the function $f^{\otimes k}$ to get the function $f' = C \circ f^{\otimes k} : \{0, 1\}^{nk} \rightarrow \{0, 1\}$, defined as $(C \circ f^{\otimes k})(x_1, \dots, x_k) = C(f(x_1), \dots, f(x_k))$. For hardness amplification within NP, to ensure that $C \circ f^{\otimes k} \in \text{NP}$ whenever $f \in \text{NP}$, O'Donnell [OD02] choose C to be a polynomial-time computable *monotone* function. In particular, he considered the functions TRIBES and RMAJ, defined as follows.

Definition 16 *Define the function $\text{TRIBES}_t : \{0, 1\}^t \rightarrow \{0, 1\}$ as*

$$\text{TRIBES}_t(x_1, \dots, x_t) = (x_1 \wedge \dots \wedge x_b) \vee (x_{b+1} \wedge \dots \wedge x_{2b}) \vee \dots \vee (x_{\lceil t/b \rceil b - b + 1} \wedge \dots \wedge x_t),$$

where b is the largest integer such that $(1 - 2^{-b})^{t/b} \geq 1/2$. Note that this makes $b = O(\log t)$. Also, let MAJ be the majority function, and define the

function $\text{RMAJ}_r : \{0, 1\}^{3^r} \rightarrow \{0, 1\}$ recursively as

$$\text{RMAJ}_1(x_1, x_2, x_3) = \text{MAJ}(x_1, x_2, x_3)$$

and

$$\begin{aligned} & \text{RMAJ}_r(x_1, \dots, x_{3^r}) \\ &= \text{RMAJ}_{r-1}(\text{MAJ}(x_1, x_2, x_3), \dots, \text{MAJ}(x_{3^{r-2}}, x_{3^{r-1}}, x_{3^r})). \end{aligned}$$

Given any $\delta \geq 1/\text{poly}(n)$, to amplify from a δ -hard function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, O'Donnell used the composition

$$\text{AMP}_k^\delta = \text{TRIBES}_t \circ \text{RMAJ}_r^{\otimes t}$$

as the function for C , with $k = t3^r$, $t \in \mathbb{N}$, and $r = O(\log(1/\delta))$. He showed that the resulting function $f' = C \circ f^{\otimes k} : \{0, 1\}^{n'} \rightarrow \{0, 1\}$ has hardness $1/2 - 1/k^c$ for some constant c . Note that the new function f' now has an input length $n' = kn$, so its hardness when expressed in terms of the input length n' is only $1/2 - 1/\text{poly}(n')$. That is, even if a super-polynomial k is used, the resulting hardness does not go beyond $1/2 - 1/\text{poly}(n')$.

To overcome this bottleneck, Healy et al. [HVV04] showed that one can take a super-polynomial k while keeping the input size of f' small (polynomial in n) if the k inputs to $f^{\otimes k}$ are generated in some pseudorandom way, in the following sense.¹ To simplify our presentation, we state things in a slightly different way from [HVV04].

Definition 17 [HVV04] For a probabilistic function $h : \{0, 1\}^n \rightarrow \{0, 1\}$, define its expected collision probability as $\text{EXPCP}[h] = \mathbb{E}_x[2 \cdot \Pr_{y, y'}[h_y(x) = h_{y'}(x)] - 1]$.


Definition 18 We say that a generator $G : \{0, 1\}^\ell \rightarrow (\{0, 1\}^n)^k$ ε -fools the δ -EXPCP of a function $C : \{0, 1\}^k \rightarrow \{0, 1\}$ if for any δ -random function $g : \{0, 1\}^n \rightarrow \{0, 1\}$,

$$|\text{EXPCP}[(C \circ g^{\otimes k})] - \text{EXPCP}[(C \circ g^{\otimes k}) \circ G]| \leq \varepsilon.$$

¹Another issue when using a super-polynomial k is that the function AMP_k^δ is no longer computable in time $\text{poly}(n)$. As shown in [HVV04], this can be handled using non-determinism.

Given such a generator G for $C = \text{AMP}_k^\delta$, the amplified function f' is defined as $f' = C \circ f^{\otimes k} \circ G$. The seed length of the generator G now becomes the input length of the new function f' . The following lemma states that the task of hardness amplification can be reduced to that of constructing such a generator.

Lemma 29 [HVV04] *Suppose for any $\delta \geq 1/\text{poly}(n)$ and any $k = t3^r \leq 2^{O(n)}$, with $t \in \mathbb{N}$ and $r = O(\log(1/\delta))$, there exists an explicitly computable generator $G : \{0, 1\}^{\ell(n)} \rightarrow (\{0, 1\}^n)^k$ which $2^{-\Omega(n)}$ -fools the δ -EXPCP of the function AMP_k^δ . Then for any $\delta \geq 1/\text{poly}(n)$ and any $s(n) \geq 2^{\Omega(n)}$, if there exists a balanced function in NP which is δ -hard for size $s(n)$, one can convert it into a function in NP which is $(1/2 - 1/s'(n))$ -hard for size $s'(n)$, for some $s'(n) \geq 2^{\Omega(\ell^{-1}(n))}$.*



Remark 6 *Lemma 29 does not appear explicitly in [HVV04] but can be derived from arguments therein. In fact, Healy et al. [HVV04] used two kinds of generators: one for preserving indistinguishability and one for fooling the δ -EXPCP of the function AMP_k^δ . In the case with $s(n) \geq 2^{\Omega(n)}$, the bottleneck lies on that for AMP_k^δ . They observed that the function AMP_k^δ can be computed by a read-once branching program of small size, and they showed that a pseudorandom generator fooling such branching programs (see Definition 19 and 20) can fool the δ -EXPCP of the function AMP_k^δ . Therefore, they reduced the task of hardness amplification to that of finding a pseudorandom generator to fool such branching programs. (See for example Theorem 6.2 in the journal version of [HVV04].) However, using currently available generators for branching programs, they were only able to obtain a generator of seed length $\Omega(n^2)$ for fooling the δ -EXPCP of AMP_k^δ . Instead of fooling branching programs, we will show that it suffices to fool a simpler class of tests: combinatorial rectangles, for which a better generator can indeed be found.*

4.2.2 PRGs for Branching Programs and Rectangles

Our generator will be based on pseudorandom generators that fool read-once branching programs and combinatorial rectangles, respectively.

Definition 19 A function $G : \{0, 1\}^\ell \rightarrow \{0, 1\}^t$ is called an ε -PRG for a class of functions from $\{0, 1\}^t$ to $\{0, 1\}$ if for any T in this class,

$$\left| \Pr [T(U_t)] - \Pr [T(G(U_\ell))] \right| \leq \varepsilon.$$

Definition 20 A read-once branching program of size s with block-length n is a finite state machine of s states, with each edge labelled by a subset of $\{0, 1\}^n$. The computation proceeds as follows. The input is read sequentially in one pass, one block of n bits at a time. When the machine reads a block $\beta \in \{0, 1\}^n$, it goes from the current state to the state reached by the edge labelled with β . Let $\text{BP}(s, n)$ denote the class of functions computed by such read-once branching programs.

Definition 21 For $m, d \in \mathbb{N}$, let $\mathbf{R}(m, d)$ denote the collection of rectangles $R = R_1 \times \cdots \times R_d \subseteq [m]^d$, with $R_i \subseteq [m]$ for all $i \in [d]$.

To fool these two classes of functions, we will use the PRGs of Nisan [Nis92] and Lu [Lu02], respectively.

Lemma 30 [Nis92] For any $n \in \mathbb{N}$ and any $s \leq 2^n$, there exists an explicitly computable $2^{-\Omega(n)}$ -PRG $G_N : \{0, 1\}^\ell \rightarrow \{0, 1\}^{sn}$ for $\text{BP}(s, n)$ with $\ell = O(n \log s)$.

Lemma 31 [Lu02] For any $m, d \in \mathbb{N}$ and any $\varepsilon \in (0, 1)$, there exists an explicitly computable ε -PRG $G_L : \{0, 1\}^\ell \rightarrow [m]^d$ for $\mathbf{R}(m, d)$ with $\ell = O(\log m + \log d + \log^{3/2}(1/\varepsilon))$.

4.3 Proof of Main Theorem

Now we prove Theorem 13. Consider any $\delta \geq 1/\text{poly}(n)$ and any $k = t3^r \leq 2^{O(n)}$, with $t \in \mathbb{N}$ and $r = O(\log(1/\delta))$. Given Lemma 29, our task is to construct a generator with a short seed which fools the δ -EXPCP of the function AMP_k^δ . Let OR_d denote the OR function on d bits and let AND_b denote the AND function on b bits. Consider an arbitrary δ -random function $g : \{0, 1\}^n \rightarrow \{0, 1\}$, and let $A : \{0, 1\}^{kn} \rightarrow \{0, 1\}$ be the function

$$A = \text{AMP}_k^\delta \circ g^{\otimes k} = \text{OR}_d \circ (\text{AND}_b \circ \text{RMAJ}_r^{\otimes b} \circ g^{\otimes b3^r})^{\otimes d},$$

where $k = db3^r = 2^{O(n)}$, $b = \text{poly}(n)$, $d = 2^{O(n)}$, and $r = O(\log(1/\delta)) = O(\log n)$. Note that A is a probabilistic function (because of g), and let A_y denote the function A taking the random string y . Observe that each $A_y^{-1}(0)$ can be seen as a rectangle in $\mathbb{R}(2^{b3^r n}, d)$. As we will see, to fool the δ -EXPCP of AMP_k^δ , it suffices to have a good PRG for rectangles in $\mathbb{R}(2^{b3^r n}, d)$. However, the range in each dimension of such rectangles is too large for us to apply Lemma 31 effectively. To resolve this, we use d copies of Nisan's PRGs to fool the d functions in the d dimensions respectively, with the d seeds coming from the output of Lu's PRG. Formally, we use the following two generators, with $\varepsilon = 2^{-\Omega(n)}$:

- Let $G_N : \{0, 1\}^q \rightarrow \{0, 1\}^{b3^r n}$ be Nisan's (ε/d) -PRG for $\text{BP}(n^c, n)$, for some large enough constant c . From Lemma 30, one can have $q = O(n \log n)$.
- Let $G_L : \{0, 1\}^\ell \rightarrow \{0, 1\}^{dq}$ be Lu's ε -PRG for $\mathbb{R}(2^q, d)$. From Lemma 31, one can have $\ell = O(n \log n) + O(n^{3/2}) = O(n^{3/2})$.

Then define our generator $G : \{0, 1\}^\ell \rightarrow \{0, 1\}^{db3^r n}$ as

$$G(u) = (G_N^{\otimes d} \circ G_L)(u) = G_N^{\otimes d}(G_L(u)).$$

It is easy to see that G is explicitly computable since both G_N and G_L are. To show that G is a good generator, we shall bound the value $|\text{EXPCP}[A] -$

$\text{EXPCP}[A \circ G]$. Let $C_{y,y'}$ be the function defined as $C_{y,y'}(x) = 1$ if $A_y(x) \neq A_{y'}(x)$ and $C_{y,y'}(x) = 0$ otherwise. Then

$$\begin{aligned} & |\text{EXPCP}[A] - \text{EXPCP}[A \circ G]| \\ &= 2 \left| \mathbb{E}_x \left[\Pr_{y,y'} [C_{y,y'}(x) = 0] \right] - \mathbb{E}_u \left[\Pr_{y,y'} [C_{y,y'}(G(u)) = 0] \right] \right| \\ &\leq 2 \mathbb{E}_{y,y'} \left[\left| \Pr_x [C_{y,y'}(x) = 0] - \Pr_u [C_{y,y'}(G(u)) = 0] \right| \right]. \end{aligned}$$

It remains to show that for any y and y' , G fools the function $C_{y,y'}$. However, neither $C_{y,y'}^{-1}(0)$ nor $C_{y,y'}^{-1}(1)$ appears to be a rectangle, so some twist is needed.

In fact, $C_{y,y'}^{-1}(1)$ is the symmetric difference of the two rectangles $R^y = A_y^{-1}(0)$ and $R^{y'} = A_{y'}^{-1}(0)$ in $\mathbb{R}(2^{b3^r n}, d)$, denoted as $R^y \ominus R^{y'}$. That is,

$$C_{y,y'}^{-1}(1) = R^y \ominus R^{y'} = (R^y \setminus R^{y'}) \cup (R^{y'} \setminus R^y).$$

Then $\Pr_x[x \in R^y \ominus R^{y'}] = \Pr_x[x \in R^y] + \Pr_x[x \in R^{y'}] - 2\Pr_x[x \in R^y \cap R^{y'}]$ and similarly for $\Pr_u[G(u) \in R^y \ominus R^{y'}]$. Thus

$$\begin{aligned} & \left| \Pr_x [C_{y,y'}(x) = 1] - \Pr_u [C_{y,y'}(G(u)) = 1] \right| \\ &= \left| \Pr_x [x \in R^y \ominus R^{y'}] - \Pr_u [G(u) \in R^y \ominus R^{y'}] \right| \\ &\leq \left| \Pr_x [x \in R^y] - \Pr_u [G(u) \in R^y] \right| + \left| \Pr_x [x \in R^{y'}] - \Pr_u [G(u) \in R^{y'}] \right| + \\ &\quad 2 \left| \Pr_x [x \in R^y \cap R^{y'}] - \Pr_u [G(u) \in R^y \cap R^{y'}] \right|. \end{aligned}$$

Note that R^y , $R^{y'}$, and $R^y \cap R^{y'}$ are all rectangles in $\mathbb{R}(2^{b3^r n}, d)$. Furthermore, they all satisfy the property that the membership function of the set in each dimension can be computed in $\text{BP}(n^c, n)$ for some constant c , according to [HVV04].² Therefore, it remains to show the following.

²Recall that $A = \text{AMP}_k^\delta \circ g^{\otimes k} = \text{OR}_d \circ (\text{AND}_b \circ \text{RMAJ}_r^{\otimes b} \circ g^{\otimes b3^r})^{\otimes d}$. From [HVV04], the function $\text{AND}_b \circ \text{RMAJ}_r^{\otimes b}$ is in $\text{BP}(\text{poly}(n), 1)$, and the probabilistic function $\text{AND}_b \circ \text{RMAJ}_r^{\otimes b} \circ g^{\otimes b3^r}$ can be computed by a probabilistic $\text{BP}(\text{poly}(n), n)$. Thus by fixing the random string of A to any string y , the set $R^y = A_y^{-1}(0)$ seen as a rectangle in $\mathbb{R}(2^{b3^r n}, d)$ has the property that each of its d dimensions has its membership function in $\text{BP}(\text{poly}(n), n)$. Next, we argue that for any y and y' , each dimension of the rectangle $R^y \cap R^{y'}$ also has its

Lemma 32 For any $R = R_1 \times \cdots \times R_d \in \mathcal{R}(2^{b^{3^r}n}, d)$ such that $R_i \in \text{BP}(n^c, n)$ for any $i \in [d]$, $|\Pr_x[x \in R] - \Pr_u[G(u) \in R]| \leq 2\varepsilon$.

Proof. Observe that $|\Pr_x[x \in R] - \Pr_u[G(u) \in R]|$ is at most

$$\left| \Pr_x[x \in R] - \Pr_v[G_N^{\otimes d}(v) \in R] \right| + \left| \Pr_v[G_N^{\otimes d}(v) \in R] - \Pr_u[G_N^{\otimes d}(G_L(u)) \in R] \right|, \quad (4.1)$$

where v is sampled from U_{dq} with $dq = O(d \cdot n \log n)$. It remains to bound the two terms above.

First, note that G_N is an (ε/d) -PRG for $\text{BP}(n^c, n)$ and can fool each R_i . Using a standard hybrid argument (see e.g. [Gol01]), one can show that $G_N^{\otimes d}$ is an $(d \cdot \varepsilon/d)$ -PRG for such a rectangle R . Thus, the first term in (4.1) above is at most ε .

To bound the second term, consider the rectangle $R' = \{v : G_N^{\otimes d}(v) \in R\} \in \mathcal{R}(2^q, d)$ with $q = O(n \log n)$. That is, $R' = R'_1 \times \cdots \times R'_d$, with $R'_i = G_N^{-1}(R_i) \subseteq \{0, 1\}^q$ for $i \in [d]$. Since G_L is an ε -PRG for $\mathcal{R}(2^q, d)$, we have $|\Pr_v[v \in R'] - \Pr_u[G_L(u) \in R']| \leq \varepsilon$. Thus, the second term in (4.1) above is also at most ε . Therefore we have the lemma. \square

Combining the lemma and the discussion above, we have

$$|\text{EXPCP}[A] - \text{EXPCP}[A \circ G]| \leq 2(2\varepsilon + 2\varepsilon + 4\varepsilon) = 16\varepsilon = 2^{-\Omega(n)}.$$

That is, our generator G , which uses a seed of length $\ell(n) = O(n^{3/2})$, is able to $2^{-\Omega(n)}$ -fool the δ -EXPCP of the function AMP_k^δ . Then by Lemma 29, we have our main theorem.

membership function in $\text{BP}(\text{poly}(n), n)$. To see this, first observe that the i 'th dimension of $R^y \cap R^{y'}$ is exactly $R_i^y \cap R_i^{y'}$, where R_i^y and $R_i^{y'}$ are the i 'th dimension of R^y and $R^{y'}$, respectively. Suppose R_i^y and $R_i^{y'}$ are recognized by read-once branching programs B and B' with state spaces S and S' , respectively. Then the set $R_i^y \cap R_i^{y'}$ is recognized by the read-once branching program B'' with state space $S \times S'$, which goes from state (s, s') to state (t, t') when reading an input block x if and only if B goes from s to t and B' goes from s' to t' , respectively, when reading x . The initial state of B'' is the state (s_0, s'_0) where s_0 and s'_0 are the initial states of B and B' , respectively, while the accepting states of B'' are exactly those states (t, t') where t and t' are the accepting states of B and B' , respectively. Thus, if B and B' are both in $\text{BP}(\text{poly}(n), n)$, so is B'' .

4.3.1 Discussion

Our generator uses a seed of length $O(n^{3/2})$, and as a result, we can only amplify hardness to $1/2 - 1/s'(n)$ against size $s'(n)$ with $s'(n) = 2^{\Omega(n^{2/3})}$. The main bottleneck is the generator for rectangles. However, to achieve the goal of having $s'(n) = 2^{\Omega(n)}$ using our approach, we need to improve both the generator for branching programs and the generator for rectangles. Without improving Nisan's PRG, even if we could have an optimal ε -PRG for $R(m, d)$, with seed length $\Theta(\log m + \log d + \log(1/\varepsilon))$, the resulting generator would still need a seed of length $\Theta(n \log n) + O(n) = \Omega(n \log n)$ (see the definition of the generator G and the calculation of its seed length in the previous page), and we would only be able to achieve $s'(n) = 2^{\Omega(n/\log n)}$.



Chapter 5

Hardness and Pseudorandomness in NP

5.1 Introduction

In this chapter, we study the problem of transforming a pseudorandom generator into a hard function and the problem of transforming a hard function into a harder one. It is known that in a high complexity class such as exponential time, one can convert from worst-case hardness to average-case hardness, from average-case hardness to pseudorandomness, and from pseudorandomness back to worst-case hardness. However, in lower complexity classes, such as NP, some of the relationships remains unclear. We establish the equivalence between pseudorandomness and average-case hardness, and widen the gap between worst-case hardness and average-case hardness within NP.

By the result of Impagliazzo and Levin ([IL90] in *FOCS' 90*), one can build the equivalence between pseudorandomness and average-case hardness within NP. For completeness of this thesis, we give a proof which shows how to transform a pseudorandom generator into a mildly hard function computable in NP. We give a strongly black-box construction, with both the transformation procedure and the hardness proof done in a black-box way.

This improves a previous result of Nisan and Wigderson, which can only obtain a worst-case hard function from a pseudorandom generator [NW94]. Therefore, we now know that the transformations among mild hardness, average-case hardness, and pseudorandomness all can be done in the complexity class NP.

5.1.1 Previous Results

The reduction from average-case hardness to pseudorandomness within NP is done in [NW94]. However, from pseudorandomness to average-case hardness, the method in [NW94] can only transform a PRG back to a worst-case hard function [NW94] within NP. Since the hardness amplification from worst-case hardness to average-case hardness is believed to be impossible (see Chapter 2 and Chapter 3), it is not clear how to obtain the reduction from PRG to average-case hardness within NP. One can use the method developed in [IL90] to achieve this reduction.

Figure 5.1.1 summarizes these known relationships between various hardness assumptions and pseudorandomness. Note that the above transformation can be done in a black-box way, in which the decoding procedure is realized in P while the encoding procedure needs the complexity of NP. This raises the following two questions. First, can the complexity of the encoding procedure be reduced? Next, since the transformation from worst-case hardness to average-case hardness seems to require high complexity, can we transform a PRG directly into an average-case hard function, using a low-complexity procedure, say in NP (or even in P)?

5.1.2 Our Results

In this chapter, we provide strongly black-box constructions of average-case hard functions from PRGs although it also can be done via method developed in [IL90]. As a result, we build the equivalence between PRG and average-case hardness within NP as shown in Figure 5.1.1. Combining with

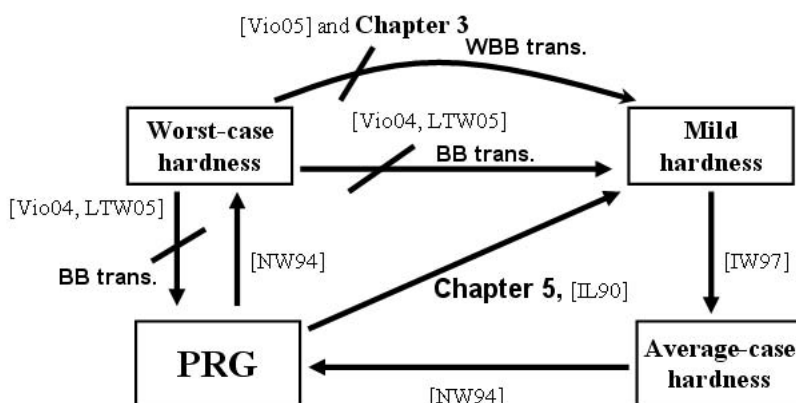


Figure 5.1: The relationship among PRG and various hardness assumptions within NP. Arrows indicate black-box transformations. **BB** and **WBB trans.** indicate black-box and weakly black-box transformations respectively. Note that the slash symbol means "the transformation cannot be done in NP".

results in the previous chapter, we are able to look closer the relationship between pseudorandomness and hardness within NP.

Our main result gives strongly black-box constructions of average-case hard functions from PRGs. The first construction has the encoding procedure realized in NP and the decoding procedure realized in P/poly (or randomized polynomial time). This improves the result of [NW94] which, using an encoding procedure in NP as well, obtains only a worst-case hard function. A natural question then is: can we further reduce the complexity of the encoding procedure, or can we prove a complexity lower bound? We give a partial answer to this by providing another strongly black-box construction with the encoding procedure realized in P but at the expense of increasing the complexity of the decoding procedure to NP, which rules out the possibility of proving a complexity lower bound for the encoding procedure without restricting the complexity of the decoding procedure. This still leaves open the question of whether or not one can have both the encoding

and decoding procedures realized in P . Our positive results also imply some impossibility results. By combining with the impossibility results of strongly black-box hardness amplification in [Vio04, LTW05] and the previous impossibility results of weakly black-box hardness amplification in Chapter 3, respectively, we can obtain corresponding impossibility results of strongly and weakly black-box PRG constructions from hard functions.

5.1.3 Organization of this Chapter

First, some preliminaries are given in Section 5.2. Then in Section 5.3, we give strongly black-box constructions of hard function from PRG.

5.2 Preliminaries

We say that a distribution is close to random if no distinguisher can tell it apart from the uniform distribution.

Definition 22 *We say that a distribution Z over $\{0, 1\}^n$ is δ -random if for any $D : \{0, 1\}^n \rightarrow \{0, 1\}$, $|\Pr_{z \in Z}[D(z) = 1] - \Pr_{u \in \mathcal{U}_n}[D(u) = 1]| \leq \delta$.*

We say that a distribution is pseudorandom if it is not random but a distinguisher with a bounded complexity cannot tell it apart from the random one. This is captured by the notion of pseudo-random generators, which are functions that stretch a short random seed to a long random-looking string.

Definition 23 *We say that a function $G : \{0, 1\}^n \rightarrow \{0, 1\}^m$ is a (δ, \mathcal{C}) -PRG, for a complexity class \mathcal{C} , if for any test $T : \{0, 1\}^m \rightarrow \{0, 1\} \in \mathcal{C}$, $|\Pr_{u \in \mathcal{U}_n}[T(G(u)) = 1] - \Pr_{w \in \mathcal{U}_m}[T(w) = 1]| < \delta$. We will call g a δ -PRG when the complexity class \mathcal{C} is clear or irrelevant in the context.*

We will need the following notion of (strongly) black-box construction of hard functions from PRGs.

Definition 24 We say that an oracle algorithm $\text{ENC} : \{0, 1\}^{\bar{n}} \rightarrow \{0, 1\}$ realizes a (strongly) black-box construction of an ε -hard function from any δ -PRG $G : \{0, 1\}^n \rightarrow \{0, 1\}^m$ if there exists a non-uniform oracle Turing machine DEC that satisfies the following. For any $G : \{0, 1\}^n \rightarrow \{0, 1\}^m$ and any $A : \{0, 1\}^{\bar{n}} \rightarrow \{0, 1\}$, there exists an advice string $\nu = \nu(G, A) \in \{0, 1\}^\ell$, for some $\ell \in \mathbb{N}$, such that if $\Pr_{\bar{x} \in \mathcal{U}_{\bar{n}}} [A(\bar{x}) \neq \text{ENC}^f(\bar{x})] < \varepsilon$, then $|\Pr_{x \in \mathcal{U}_n} [\text{DEC}^{A, \nu}(G(x)) = 1] - \Pr_{z \in \mathcal{U}_m} [\text{DEC}^{A, \nu}(z) = 1]| > \delta$.

Note that we do not include complexity classes in this definition. In fact, this definition implies the definition of constructing an $(\varepsilon, \bar{\mathcal{C}})$ -hard function from a $(\delta, \mathcal{A}^{\bar{\mathcal{C}}})$ -PRG, where \mathcal{A} is the complexity class of DEC .

5.2.1 Universal Hash Functions

We need the notion of universal hash functions and their efficient constructions.

Lemma 33 [CW79] For any $n, m \in \mathbb{N}$ with $n \geq m$, there is a family \mathcal{H}_m^n of hash functions $h : \{0, 1\}^n \rightarrow \{0, 1\}^m$ satisfying the following two properties.

- Each $h \in \mathcal{H}_m^n$ can be described by $O(\log n)$ bits, and given such a description and any $x \in \{0, 1\}^n$, one can compute $h(x)$ in $\text{poly}(n)$ time.
- $\Pr_{h \in \mathcal{H}_m^n} [h(\mathcal{U}_n) \neq \mathcal{U}_m] = 2^{-\Omega(n)}$.
- For any distinct $x, y \in \{0, 1\}^n$, $\Pr_{h \in \mathcal{H}_m^n} [h(x) = h(y)] = 2^{-m}$.

Universal hash functions are known to be good extractors, in the sense that they can extract almost perfect randomness from a slightly random source. We need the following, which is known as the leftover hash lemma.

Lemma 34 [IZ89, HILL99] Let $\mathcal{H} = \mathcal{H}_m^n$. Then for any $X \subseteq \{0, 1\}^n$ with $|X| \geq 2^k$, the distribution of $(h(x), h)$, with (x, h) sampled from $X \times \mathcal{H}$, is $2^{-(k-m)/2}$ -random.

A simple corollary is the following, which says that a random $h \in \mathcal{H}$ is likely to be a good extractor. We omit the proof since it can be easily derived using a Markov inequality.

Corollary 7 *Let $\mathcal{H} = \mathcal{H}_m^n$. Then for any $X \subseteq \{0, 1\}^n$ with $|X| \geq 2^k$,*

$$\Pr_{h \in \mathcal{H}} [h(X) \text{ is not } \delta\text{-random}] \leq 2^{-(k-m)/2} / \delta.$$

5.3 Hardness from Pseudorandomness

In this section, we consider constructions of hard functions from pseudorandom generators. Our first result shows that one can construct a mildly-hard function from a PRG.

Theorem 14 *There exists a black-box construction of $\Omega(1/n^4)$ -hard function from $1/(3n)$ -PRG such that the encoding procedure can be realized in NP while the decoding procedure can be realized in P/poly.*

Note that one can transform a mildly hard function into an average-case hard one by a polynomial-time procedure [IW97]. Therefore, one can have a black-box construction of average-case hard function from PRG realized in NP. Furthermore, we can combine Theorem 14 with the impossibility results of black-box hardness amplification in [Vio04, LTW05] to obtain corresponding impossibility results for black-box PRG construction from hard function. We can also combine Theorem 14 with our results for weakly black-box hardness amplification to obtain corresponding results for weakly black-box PRG construction from hard function.

One unsatisfying aspect of Theorem 14 is that the encoding procedure needs the complexity of NP. A natural question is whether or not its complexity can be reduced. Theorem 15 shows that this is in fact possible, but at the expense of increasing the complexity of the decoding procedure to NP.

Theorem 15 *There exists a black-box construction of $(1 - \delta)/2$ -hard function from a $\delta/2$ -PRG $G : \{0, 1\}^n \rightarrow \{0, 1\}^m$, with $m = \omega(n/\delta^2)$, such that*

encoding procedure can be realized in \mathbf{P} while the decoding procedure can be realized in \mathbf{NP} .

Proof. Suppose $G : \{0, 1\}^n \rightarrow \{0, 1\}^m$, with $m = \omega(n/\delta^2)$, is an $\delta/2$ -PRG. Define the hard function $f : \{0, 1\}^n \times [m] \rightarrow \{0, 1\}$ by $f(x, i) = G(x)_i$. We will show that f is $(1 - \delta)/2$ -hard.

Suppose there is a function $A : \{0, 1\}^n \times [m] \rightarrow \{0, 1\}$ such that

$$\Pr_{u,i} [f(u, i) \neq A(u, i)] < \frac{1 - \delta}{2}.$$

We define the distinguisher $D^A : \{0, 1\}^m \rightarrow \{0, 1\}$ by

$$D^A(w) = 1 \quad \text{if and only if} \quad \exists u \in \{0, 1\}^n : \Pr_i [w_i \neq A(u, i)] \leq \frac{1 - \delta/4}{2}.$$

We will show that D^A is an $\delta/2$ -distinguisher for G .

Define $I(w) = 1$ if and only if $\Pr_i [w_i \neq A(u, i)] \leq \frac{1 - \delta/4}{2}$.

First, we bound the probability $\Pr_w [D^A(w) = 1]$, which is at most

$$\sum_u \Pr_w [I(w) = 1] \leq 2^n \cdot 2^{-\Omega(\delta^2 m)} \leq 2^{-\omega(n)} \leq \frac{\delta}{4}.$$

Next, we bound the probability $\Pr_u [D^A(G(u)) = 1]$. Since

$$\Pr_{x,i} [f(x, i) \neq A(x, i)] < (1 - \delta)/2,$$

Markov's inequality gives $\Pr_u [I(G(u)) = 0] < \frac{1 - \delta}{1 - \delta/4} \leq 1 - \frac{3\delta}{4}$. Thus,

$$\Pr_u [D^A(G(u)) = 1] = \Pr_u [I(G(u)) = 1] > \frac{3\delta}{4}.$$

Therefore $\Pr_u [D^A(G(u)) = 1] - \Pr_w [D^A(w) = 1] > \frac{\delta}{2}$, which contradicts to the assumption that G is a $\delta/2$ -PRG.

Note that the distinguisher D^A is computable in \mathbf{NP}^A . Therefore, we have a black-box proof for the hardness of f , in which the decoding procedure can be realized in \mathbf{NP} . Finally, note that the function f can be easily computed in polynomial time given G as an oracle, so the encoding procedure can be realized in \mathbf{P} . This proves Theorem 15. \square

Now we proceed to prove Theorem 14.

Proof. Suppose $G : \{0, 1\}^n \rightarrow \{0, 1\}^m$, with $n < m$, is a $1/(3n)$ -PRG. This means that it is hard to tell the image of G from a random string. Therefore, Nisan and Wigderson [NW94] considered the function $f^G : \{0, 1\}^m \rightarrow \{0, 1\}$ defined by $f^G(y) = 1$ if and only if $y \in \text{image}(G)$. This function is clearly worst-case hard, because otherwise it can serve as a distinguisher for G . However, there are two issues which prevent us from proving a large hardness for such a function in general. First, $\text{image}(G)$ may only be a relatively small subset of $\{0, 1\}^m$; in this case, one can approximate f^G well simply by outputting 0 for every input. The second issue is that G may be highly non-injective so that elements of $\text{image}(G)$ may have large pre-image sizes; in this case, different elements of $\{0, 1\}^m$ may carry very different weights from G , so even if one can approximate f^G well, one may be still unable to distinguish G well enough. In fact, when G is injective and $m = n + 1$, with both issues gone, one can indeed show that f^G has constant hardness. Then a natural question is: can we transform any PRG G into another PRG which has a relatively large image and is almost injective?

To handle the first issue, we would like to choose a hash function h to map the space $\{0, 1\}^m$ down to a smaller one, the smaller the better, without two elements of $\text{image}(G)$ being hashed to the same value. To handle the second issue, we would like to add to the output more information $g(x)$ extracted from the seed x , the more the better, without compromising the security. For both purposes, we would like to know the pre-image size of $G(x)$ for any given seed x . We define

$$i_x = \lfloor \log |G^{-1}(G(x))| \rfloor.$$

For a seed x , if we know the value i_x , then we would like to choose the hash function h with output length about $n - i_x$ and the function g with output length about i_x . We will use the well-known construction of universal hash functions, given in Lemma 33 in the Appendix 5.2.1. Let \mathcal{H}_m^n denote such a family of hash functions $h : \{0, 1\}^n \rightarrow \{0, 1\}^m$. Formally, we consider the following family of generators.

Definition 25 Given $\delta \in (0, 1)$, let $r = \log(4n)$. For $i \in [n]$, $m_i = n - i + 2r$, $\ell_i = i - r$, $h \in \mathcal{H}_{m_i}^m$, and $g \in \mathcal{H}_{\ell_i}^n$, define the function $G_{h,g}^i : \{0, 1\}^n \rightarrow \{0, 1\}^{m_i} \times \{0, 1\}^{\ell_i}$ by

$$G_{h,g}^i(x) = (h(G(x)), g(x)).$$

The problem is that G in general may not be regular, i.e. the values of i_x may not be the same for every x , and the value of i_x may not be easy to compute given x . Instead, we will show that for some specific value of i , for most h and g , determining the image of $G_{h,g}^i$ is already a hard function. Let $f_{h,g}^i$ be the function such that

$$f_{h,g}^i(y, z) = 1 \quad \text{if and only if} \quad (y, z) \in \text{image}(G_{h,g}^{i*}).$$

For $i \in [n]$, define the set $B_i = \{G(x) : x \in \{0, 1\}^n \wedge i_x = i\}$. Clearly, these sets B_1, \dots, B_n form a partition of $\text{image}(G)$. Note that for any $i \in [n]$, $|B_i| \leq 2^{n-i}$, since each $y \in B_i$ has $|G^{-1}(y)| \geq 2^i$. Furthermore, since $\Pr_x[G(x) \in \cup_{i \in [n]} B_i] = 1$, there must exist some $i^* \in [n]$ such that $\Pr_x[G(x) \in B_{i^*}] \geq 1/n$ and $i^* \geq 2r$. From now on, we will focus on this i^* . Let $B = B_{i^*}$, $\mathcal{H} = \mathcal{H}_{m_{i^*}}^m$, and $\mathcal{K} = \mathcal{H}_{\ell_{i^*}}^n$.

Call $(h, g) \in \mathcal{H} \times \mathcal{K}$ *good* if both h and g are good by satisfying the following:

- $h(\mathcal{U}_m)$ is perfectly random, i.e., $h(\mathcal{U}_m) = \mathcal{U}_{m_{i^*}}$.
- $\Pr_x[G(x) \in L \mid G(x) \in B] \leq 1/8$, for $L = \{w \in B : \exists w' \in B \text{ with } w' \neq w \text{ and } h(w) = h(w')\}$.
- For any $y \in B$, the distribution of $g(x)$, over $x \in G^{-1}(y)$, is $1/8$ -random.

The following shows that a random (h, g) is likely to be good.

Lemma 35 $\Pr_{h \in \mathcal{H}, g \in \mathcal{K}}[(h, g) \text{ is not good}] = o(1)$.

Proof. Recall that $\mathcal{H} = \mathcal{H}_{m_{i^*}}^m$ and $\mathcal{K} = \mathcal{H}_{\ell_{i^*}}^n$. First, from Lemma 33, we have

$$\Pr_{h \in \mathcal{H}} [h(\mathcal{U}_m) \neq \mathcal{U}_{m_{i^*}}] = 2^{-\Omega(m)}.$$

Next, for any x , $\Pr_{h \in \mathcal{H}}[G(x) \in L \mid G(x) \in B] = \Pr_{h \in \mathcal{H}}[\exists w \in B \setminus \{G(x)\} : h(G(x)) = h(w)]$, which by the definition of \mathcal{H} is at most

$$|B| \cdot 2^{-m_{i^*}} \leq 2^{n-i^*} \cdot 2^{-(n-i^*+2r)} = 2^{-2r}.$$

Thus, $\mathbb{E}_{h \in \mathcal{H}}[\Pr_{x \in \mathcal{U}_n}[G(x) \in L \mid G(x) \in B]] = \Pr_{x \in \mathcal{U}_n, h \in \mathcal{H}}[G(x) \in L \mid G(x) \in B] \leq 2^{-2r}$. Define $B(h) = 1$ if and only if $\Pr_{x \in \mathcal{U}_n}[G(x) \in L \mid G(x) \in B] > 1/8$. Then by Markov's inequality, we have

$$\Pr_{h \in \mathcal{H}}[B(h) = 1] < 2^{-2r+3}.$$

Finally, consider any $y \in B$, and note that $|G^{-1}(y)| \geq 2^{i^*}$. Let X denote the uniform distribution over $G^{-1}(y)$. Then from Corollary 7,

$$\Pr_{g \in \mathcal{K}}[g(X) \text{ is not } 1/8\text{-random}] \leq 2^{-(i^* - \ell_{i^*})/2} / (1/8) \leq 2^{-r/2+3}.$$

Therefore, $\Pr_{h \in \mathcal{H}, g \in \mathcal{K}}[(h, g) \text{ is not good}] \leq 2^{-\Omega(m)} + 2^{-2r+3} + 2^{-r/2+3} = o(1)$.
□

Next, we show that a good (h, g) gives a hard function. Fix a good (h, g) , and let $f = f_{h,g}^{i^*}$. Suppose that there exists a function C such that $\Pr_{y,z}[C(y, z) \neq f(y, z)] = o(1/n^3)$. For any z , define the distinguisher $D_z : \{0, 1\}^m \rightarrow \{0, 1\}$ for G by $D_z(w) = 1$ if and only if $C(h(w), z) = 1$. Then we have the following two claims.

Claim 1 $\Pr_{w,z}[D_z(w) = 1] \leq 1/(3n)$.

Proof. Recall that $\Pr_{w,z}[D_z(w) = 1] = \Pr_{w,z}[C(h(w), z) = 1]$. The idea is to show that this probability is close to $\Pr_{y,z}[f(y, z) = 1] = \Pr_{y,z}[(y, z) \in \text{image}(f)]$, which is small because $\text{image}(f)$ is relatively small.

First, since h is good, the distribution of $h(w)$ is perfectly random, which implies that

$$\Pr_{w,z}[C(h(w), z) = 1] = \Pr_{y,z}[C(y, z) = 1].$$

Next, by the assumption that C approximates f well, we have

$$\begin{aligned} \Pr_{y,z}[C(y, z) = 1] &\leq \Pr_{y,z}[f(y, z) = 1] + \Pr_{y,z}[C(y, z) \neq f(y, z)] \\ &\leq \Pr_{y,z}[f(y, z) = 1] + o(1/n^3). \end{aligned}$$

Finally, since $\text{image}(G)$, of size 2^n , is a small subset of $\{0, 1\}^{n+r}$, we have

$$\Pr_{y,z} [f(y, z) = 1] = \Pr_{y,z} [(y, z) \in \text{image}(f)] = 2^n / 2^{n+r} = 2^{-r} = 1/(4n).$$

As a result, we have $\Pr_{w,z} [D_z(w) = 1] \leq 1/(4n) + o(1/n^3) \leq 1/(3n)$. \square

Claim 2 $\Pr_{x,z} [D_z(G(x)) = 1] \geq 2/(3n)$.

Proof. Recall that $\Pr_{x,z} [D_z(G(x)) = 1] = \Pr_{x,z} [C(h(G(x)), z) = 1]$, which is at least

$$\Pr_x [G(x) \in B] \cdot \Pr_{x,z} [C(h(G(x)), z) = 1 \mid G(x) \in B], \quad (5.1)$$

where $B = B_{i^*}$. The first factor above is at least $1/n$, by the choice of i^* . For the second factor, we will show that it is close to $\Pr_x [f(h(G(x)), g(x)) = 1 \mid G(x) \in B]$, which is 1 by definition.

Define $T(x, z) = 1$ if and only if $C(h(G(x)), z) \neq f(h(G(x)), z)$. Note that the second factor is at least

$$\Pr_{x,z} [f(h(G(x)), z) = 1 \mid G(x) \in B] - \Pr_{x,z} [T(x, z) = 1 \mid G(x) \in B]. \quad (5.2)$$

It remains to show that the first term is large while the second term is small. Since g is good, the distribution of $g(x)$ is $1/8$ -random, which implies that the first term in (5.2) is at least

$$\Pr_{x,z} [f(h(G(x)), g(x)) = 1 \mid G(x) \in B] - 1/8 = 7/8.$$

Next, we show that the second term in (5.2) is not far from

$$\Pr_{y,z} [C(y, z) \neq f(y, z)],$$

which is small. Observe that the difference in these two probabilities is that the first argument of C (and f) comes from two different distributions: one is $h(G(x))$ for a random $x \in G^{-1}(B)$ and the other is a random y from $\mathcal{U}_{m_{i^*}}$. It suffices to show that for most $y \in \text{image}(h \circ G)$, the probability assigned to y by the first distribution, which is $\Pr_x [h(G(x)) = y \mid G(x) \in B]$, is within a small factor of that by the second distribution, which is 2^{-m_i} .

Recall that L is the set of $w \in \text{image}(G)$ which has a different $w' \in B$ with $h(w) = h(w')$. Note that the second term in (5.2) is at most

$$\Pr_x [G(x) \in L \mid G(x) \in B] + \Pr_{x,z} [T(x,z) = 1 \wedge G(x) \notin L \mid G(x) \in B]. \quad (5.3)$$

Since h is good, the first term in (5.3) is at most $1/8$. It remains to bound the second term in (5.3). Consider any $y \notin G(L)$, which has at most one $w \in B$ such that $h(w) = y$. As any $w \in B$ has at most 2^{i^*+1} different x 's such that $G(x) = w$, $\Pr_x [h(G(x)) = y \mid G(x) \in B]$ is

$$\frac{\Pr_x [h(G(x)) = y \wedge G(x) \in B]}{\Pr_x [G(x) \in B]} \leq \frac{2^{i^*+1-n}}{(1/n)} = 2n \cdot 2^{-m_{i^*}+2r} = 32n^3 \cdot 2^{-m_{i^*}}.$$

Thus, the second term in (5.3) is at most $32n^3 \cdot \Pr_{y,z} [C(y,z) \neq f(y,z)] = o(1)$.

Combining the bounds for (5.1), (5.2), and (5.3), we conclude that

$$\Pr_{x,z} [D_z(G(x)) = 1] \geq (1/n)(7/8 - 1/8 - o(1)) \geq 2/(3n).$$

□

From the two claims above, we have $\mathbb{E}_z [\Pr_x [D_z(G(x)) = 1] - \Pr_w [D_z(w) = 1]] \geq 1/(3n)$, which implies that for some z , D_z can distinguish G with advantage at least $1/(3n)$. Note that z can be seen as an advice, and D_z uses C in a black-box way. That is, we have given a black-box proof that $f_{h,g}^{i^*}$ is $\Omega(1/n^3)$ -hard, for any good (h,g) , when G is a $1/(3n)$ -PRG.

The remaining problem is that we do not know what i^* is and which (h,g) is good. Our solution is, as in Section ??, to include i, h, g in the input. Therefore, define our hard function \hat{f} by $\hat{f}(y, i, h, g) = f_{h,g}^i(y)$. That is,

$$\hat{f}(y, i, h, g) = 1 \quad \text{if and only if} \quad y \in \text{image}(G_{h,g}^i).$$

Next, we prove the hardness of \hat{f} . Suppose there exists a function C such that

$$\Pr_{y,i,h,g} [C(y, i, h, g) \neq \hat{f}(y, i, h, g)] = o(1/n^4).$$

Then $\Pr_{y,h,g} [C(y, i^*, h, g) \neq \hat{f}(y, i^*, h, g)] = o(1/n^3)$, and

$$\begin{aligned} \Pr_{y,h,g} [C(y, i^*, h, g) \neq f_{h,g}^{i^*}(y) \mid (h,g) \text{ is good}] &= o(1/n^3) / \Pr_{h,g} [(h,g) \text{ is good}] \\ &= o(1/n^3). \end{aligned}$$

This implies that for some good (h, g) , the function $f_{h,g}^{i^*}$ is not $\Omega(1/n^3)$ -hard, which then implies that G is not a $1/(3n)$ -PRG. Again, we can see i^* together with a good (h, g) as the advice string. Therefore, we have shown a black-box proof that \hat{f} is $\Omega(1/n^4)$ -hard when G is a $1/(3n)$ -PRG.

Finally, note that \hat{f} can be computed in **NP** with G given as an oracle, so we have proved Theorem 14. \square





Chapter 6

Hardcore Set Constructions

6.1 Introduction

In this chapter, we study a fundamental result of Impagliazzo (*FOCS'95*) known as the hard-core set lemma. Consider any function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ which is “mildly-hard”, in the sense that any circuit of size s must disagree with f on some δ fraction of inputs. Then the hard-core lemma says that f must have a hard-core set H of density δ on which it is “extremely hard”, in the sense that any circuit of size $s' = O(s/(\frac{1}{\varepsilon^2} \log(\frac{1}{\varepsilon\delta})))$ must disagree with f on at least $(1 - \varepsilon)/2$ fraction of inputs from H .

There are three issues of the lemma which we would like to address: the loss of circuit size, the need of non-uniformity, and its inapplicability to a low complexity class. We introduce two models of hard-core set constructions, a strongly black-box one and a weakly black-box one, and show that those issues are unavoidable in such models.

First, we show that in any strongly black-box construction, one can only prove the hardness of a hard-core set for smaller circuits of size at most $s' = O(s/(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}))$. Next, we show that any weakly black-box construction must be inherently non-uniform — to have a hard-core set for a class G of functions, we need to start from the assumption that f is hard against a non-uniform complexity class with $\Omega(\frac{1}{\varepsilon} \log |G|)$ bits of advice. Finally, we

show that weakly black-box constructions in general cannot be realized in a low-level complexity class such as $\text{AC}^0[p]$ — the assumption that f is hard for $\text{AC}^0[p]$ is not sufficient to guarantee the existence of a hard-core set.

6.1.1 Our Results

We have three results, which give negative answers to the three questions we raised before, with respect to our models of black-box constructions. Note that our lower bounds for our second model (weakly black-box one) also hold for our first model as the first model is a special case of the first one.

Our first result shows that any *strongly* black-box (δ, ε, k) -construction must require a query complexity of $q = \Omega(\frac{1}{\varepsilon^2} \log \frac{1}{\delta})$. Our lower bound explains why it is very difficult to have a smaller loss of circuit size in the hard-core set lemma; in fact, any strongly black-box construction must suffer a loss of such a large factor q . Note that our lower bound is tight as it is matched (up to a constant factor) by the upper bound from the construction of Klivans and Servedio [KS03].

Our second result shows that any *weakly* black-box (δ, ε, k) -construction must require an advice of length $\Omega(\frac{1}{\varepsilon} \log k)$. This explains why it is difficult to have a uniform version of the hard-core set lemma; in fact, any weakly black-box construction is inherently non-uniform. Moreover, one cannot hope to improve Trevisan's uniform hardness amplification results [Tre03, Tre05] by reducing the number of advice bits needed in the hard-core set construction, unless one can come up with a non-black-box approach. Note that from the query upper bound of [KS03], one can immediately have an advice upper bound of $O(\frac{1}{\varepsilon^2} (\log \frac{1}{\delta}) \log k)$, which has a gap from our lower bound. It is not clear which bound can be further improved, but our feeling is that this upper bound may likely be improved.

Our third result shows that no *weakly* black-box (δ, ε, k) -construction can be realized in a low-level complexity class such as $\text{AC}^0[p]$ for a prime p , when $\delta \geq 1/20$ and $\varepsilon \leq 1/n$. More precisely, we show that the function realizing such a black-box construction can be used to approximate

the majority function, but on the other hand, the majority function cannot be approximated by an $\text{AC}^0[p]$ circuit. Therefore one cannot have a hard-core set lemma for $\text{AC}^0[p]$, unless one can prove it in a non-black-box way.

6.1.2 Bounds from Hardness Amplification

There is no previous result working directly on the lower bounds of hard-core set constructions. However, one can obtain such bounds from lower bounds for the task of hardness amplification [Vio06, LTW05]. This is because the hard-core set lemma can be used for hardness amplification, as shown in [Im95], and a closer inspection shows that a black-box construction of hard-core set in fact yields hardness amplification in a similar black-box model.

In particular, one can have the following. First, using a recent result of Viola [Vio06], we can derive a lower bound of $\min(\frac{1}{10\epsilon}, \frac{n}{5 \log n})$ on the query complexity of any strongly black-box construction of hard-core set. Note that this bound is always smaller than our bound. Second, we can use the result in [LTW05] to derive an advice lower bound of $\Omega(\log \frac{(1-\delta)^2}{\epsilon})$ for any weakly black-box construction of hard-core set. Note that this bound is exponentially worse than ours. Finally, we can use another result of Viola [Vio06] to show that for weakly black-box construction of hard-core set, if the function DEC satisfies the additional condition that it only needs a short (logarithmic in the circuit size of DEC) advice, then it cannot belong to $\text{AC}^0[p]$. Note that this additional condition is not required in our result and our proof is much simpler.¹

6.1.3 Our Techniques

To have our query lower bound, we show that if a strongly black-box construction does not make enough number of queries, then there exist a family

¹On the other hand, under this additional condition, Viola achieved something stronger: such DEC can be used as oracle gates by an AC^0 circuit to compute the majority function exactly (instead of approximately). We can also achieve this, but we omit it here as our proof is similar to Viola's.

$G = \{g_1, \dots, g_k\}$ of functions and a function f violating the guarantee of the construction. We prove the existence of G and f by a probabilistic argument. We choose f randomly and then choose g_1, \dots, g_k independently as k noisy versions of f , with each $g_i(x)$ being $f(x)$ added by a noise of rate $(1 - 2\varepsilon)/2$. We can show that f is unlikely to have an ε -hard-core set for G , because it is very unlikely to have a subset on which every g_i has a large deviation from f , when k is large enough. On the other hand, we can show that if the function DEC does not make enough number of queries to functions in G , there is a good chance that it is not close to f . This implies the existence of G and f for which DEC fails to work. Thus, we conclude that the query complexity must be high.

To have our advice lower bound, we show the existence of a family $G = \{g_1, \dots, g_k\}$ of functions such that one can find a large collection Γ of functions with the property that every function in Γ has no hard-core set but no two functions in Γ are close. The candidates for Γ are functions G_I , with $I = \{i_1, \dots, i_t\}$, defined as $G_I(x) = \text{MAJ}(g_{i_1}(x), \dots, g_{i_t}(x))$, where MAJ denotes the majority function. We will let $t = \lfloor 1/\varepsilon \rfloor$, so that every G_I has a good correlation with some g_i for $i \in I$, which implies that G_I has no ε -hard-core set for G . On the other hand, for any G_I and G_J with small $I \cap J$, they are likely to be far away because for any input x , $\sum_{i \in I \cap J} g_i(x)$ is likely to be small, so there is a good chance that the values of $G_I(x)$ and $G_J(x)$ are dominated by $\sum_{i \in I \setminus J} g_i(x)$ and $\sum_{j \in J \setminus I} g_j(x)$, respectively, and hence there is a good chance that $G_I(x) \neq G_J(x)$. This implies that with high probability, each G_I is far away from many other G_J 's, and by the well-known Turán's theorem, there must be many G_I 's which are far away from each other, and they form the set Γ . This gives an advice lower bound of $\log |\Gamma|$.

To show that a weakly black-box construction can be used to approximate the majority function, we again use the observation that for any G , the function G_I , with $|I| = t \leq 1/\varepsilon$, has no ε -hard-core set for G . When $t \leq n$, we can define the functions g_i , for $1 \leq i \leq t$, as $g_i(x) = x_i$ (the i -th bit of x). Then for some advice α , $\text{DEC}^{G, \alpha}(x) = G_I(x) = \text{MAJ}(x_1, \dots, x_t)$ for at

least δ fraction of x , and by an average argument there must exist some fixed $\bar{x}_{t+1}, \dots, \bar{x}_n$ such that $\text{DEC}^{G,\alpha}(x_1, \dots, x_t, \bar{x}_{t+1}, \dots, \bar{x}_n) = \text{MAJ}(x_1, \dots, x_t)$ for at least δ fraction of x_1, \dots, x_t . By hard-wiring α and $\bar{x}_{t+1}, \dots, \bar{x}_n$ into the circuit for DEC, we get a circuit which is δ -close to the majority function on t bits.

6.1.4 Organization of this chapter

First, in Section 6.2 we give some preliminaries and define our two models for black-box constructions of hard-core set. In Section 6.3, we prove a query lower bound for such a strongly black-box construction. Then we show a lower bound on the advice length needed in such a weakly black-box construction in Section 6.4. Finally, in Section 6.5 we show that no such weakly black-box construction can be realized in $\text{AC}^0[p]$.

6.2 Preliminaries

Let F_n denote the set of all Boolean functions $f : \{0, 1\}^n \rightarrow \{0, 1\}$. Let $\text{AC}^0[p](s)$ denote the class of Boolean functions computed by constant-depth circuits of size s equipped with mod_p gates (which output 0 exactly when the input bits sum to 0 modulo p), and let $\text{AC}^0[p] = \text{AC}^0[p](\text{poly}(n))$. Given a multi-set (or simply a set) S , we let $|S|$ denote the number of elements in it, counting multiplicity. Given a set $G = \{g_1, \dots, g_k\} \subseteq F_n$, together with a multi-set $I = \{i_1, \dots, i_q\} \subseteq [k]$ of indices, let g_I denote the function such that $g_I(x) = (g_{i_1}(x), \dots, g_{i_q}(x))$ for $x \in \{0, 1\}^n$. We say that two functions f and g in F_n are δ -close if $\Pr_{x \in \mathcal{U}_n}[f(x) \neq g(x)] \leq \delta$.

We will also need the following result, known as Turán's Theorem, which can be found in standard textbooks (e.g. [AS00]).

Fact 6 (*Turán's Theorem*) *Given a graph $G = (V, E)$, let d_v denote the degree of a vertex v and $\alpha(G)$ the size of the maximum independent set. Then $\alpha(G) \geq \sum_{v \in V} \frac{1}{d_v + 1}$.*

6.2.1 Hardness and Hard-Core Set Lemma

We recall that a function is hard if no small circuit is close to it. Formally, we say that a function $f \in F_n$ is δ -hard (or has hardness δ) for size s , if for any $C \in \text{SIZE}(s)$, $\Pr_{x \in \mathcal{U}_n} [C(x) \neq f(x)] > \delta$. Impagliazzo introduced the following notion of hard-core set of a hard function.

Definition 26 [Im95] *We say that a function $f \in F_n$ has an ε -hard-core set $H \subseteq \{0, 1\}^n$ for size s , if for any $C \in \text{SIZE}(s)$, $\Pr_{x \in H} [C(x) \neq f(x)] > \frac{1-\varepsilon}{2}$.*

Now we can state Impagliazzo's hard-core set lemma [Im95], which is the focus of this chapter.

Lemma 36 [Im95] *Any function $f \in F_n$ which is δ -hard for size s must have an ε -hard-core set H for size s' , with $|H| \geq \delta 2^n$ and $s' = O(s/(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}))$.*

Note that in this lemma, the hardness on the set H is measured against a smaller circuit size s' , as compared to the original circuit size s . This was later improved by Klivans and Servedio [KS03] to $s' = O(s/(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}))$ but at the expense of having a slightly smaller hard-core set of size $\delta 2^{n-1}$. A closer look at their proofs shows that they work for the more general setting with hardness measured against any class of functions instead of just circuits. For this, let us first formalize the notion that a function has no hard-core set with respect to a class $G \subseteq F_n$.

Definition 27 *Given a set $G = \{g_1, \dots, g_k\} \subseteq F_n$, we say that a function $f \in F_n$ is (δ, ε, G) -easy if for any $H \subseteq \{0, 1\}^n$ of size $\delta 2^n$, there is a function $g \in G$ such that $\Pr_{x \in H} [g(x) \neq f(x)] \leq \frac{1-\varepsilon}{2}$.*

Then from [Im95] and its improvement in [KS03], one actually has the following.

Lemma 37 *For some $q = O(\frac{1}{\varepsilon^2} \log \frac{1}{\delta})$, there exists a function $D : \{0, 1\}^q \rightarrow \{0, 1\} \in \text{SIZE}(\text{poly}(q))$ such that for some constant c the following holds. For any $G = \{g_1, \dots, g_k\} \subseteq F_n$, if a function $f \in F_n$ is $(c\delta, \varepsilon, G)$ -easy, then there is a multi-set I with $|I| = q$ such that $\Pr_x [D(g_I(x)) \neq f(x)] \leq \delta$.*

In [Im95], $c = 1$ and D is the majority function (and $q = O(\frac{1}{\varepsilon^2} \log \frac{1}{\delta\varepsilon})$), while in [KS03], $c = 1/2$ and D is the majority of majority functions.

6.2.2 Black-Box Constructions of Hard-Core Set

Now we introduce our two models for black-box construction of hard-core set. The first one is stronger than the second.

Definition 28 *We say that a (non-uniform) oracle algorithm $\text{DEC}^{(\cdot)}$ realizes a strongly black-box (δ, ε, k) -construction (of hard-core set) if for some $q \in \mathbb{N}$ it has a decision function $D : \{0, 1\}^q \rightarrow \{0, 1\}$ such that for some constant c the following holds. For any $G = \{g_1, \dots, g_k\} \subseteq F_n$, if a function $f \in F_n$ is $(c\delta, \varepsilon, G)$ -easy, then there is a multi-set I with $|I| = q$ such that $\text{DEC}^{G, I}(x) = D(g_I(x))$ and $\Pr_x[\text{DEC}^{G, I}(x) \neq f(x)] \leq \delta$. We call q the query complexity of DEC .*

Note that we do not place any requirement on the computational complexity of DEC , for either computing D or finding I , which will make our lower bound stronger. In this model, I can be seen as an advice, so the advice is of the form of a multi-set $I = \{i_1, \dots, i_q\}$, and the algorithm DEC is restricted to be of the following form: on input x , it queries the functions g_{i_1}, \dots, g_{i_q} all on the input x , applies the function D on the q answer bits, and outputs $D(g_{i_1}(x), \dots, g_{i_q}(x))$. Note that the known hard-core set constructions are in fact done in our first model [Im95, KS03]

Our second model generalizes the first one by removing the format constraint on the algorithm DEC and its advice. That is, the algorithm DEC and its advice now are allowed to be of arbitrary form, and as in the previous model, we do not place any requirement on its computational complexity.

Definition 29 *We say that a (non-uniform) oracle algorithm $\text{DEC}^{(\cdot)}$ realizes a weakly black-box (δ, ε, k) -construction (of hard-core set) if for some constant c the following holds. For any $G = \{g_1, \dots, g_k\} \subseteq F_n$, if a function $f \in F_n$ is $(c\delta, \varepsilon, G)$ -easy, then there is an advice string α such that $\Pr_x[\text{DEC}^{G, \alpha}(x) \neq f(x)] \leq \delta$.*

We will consider implementing the oracle algorithm DEC by a circuit. In this case, we will allow the functions g_1, \dots, g_k to be used as oracle gates for the circuit.

6.3 Query Complexity in Strongly Black-Box Construction

In this section, we give a lower bound on the query complexity of any strongly black-box construction of hard-core set. Formally, we have the following.

Theorem 16 *Suppose $2^{-c_1 n} \leq \varepsilon, \delta < c_2$, and $\omega(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}) \leq k \leq 2^{2^{c_3 n}}$, for small enough constants $c_1, c_2, c_3 > 0$. Then any strongly black-box (δ, ε, k) -construction must have a query complexity of $q = \Omega(\frac{1}{\varepsilon^2} \log \frac{1}{\delta})$.*

Our lower bound is optimal since it is matched (up to a constant factor) by the upper bound from the construction of Klivans and Servedio (Lemma 37). Note that the assumption $k \geq \omega(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}) \geq 2^{\Omega(n)}$ is reasonable, since in standard setting of hard-core set lemma G typically consists of circuits of polynomial (or larger) size, which gives $k = |G| \geq 2^{\text{poly}(n)}$.

The roadmap for the proof is the following. Consider any DEC which realizes such a strongly black-box construction. We would like to show the existence of a function f and a family $G = \{g_1, \dots, g_k\}$ of functions such that f is (δ, ε, G) -easy but the algorithm DEC without making enough queries cannot approximate f well. We will prove their existence by a probabilistic argument.

Now we proceed to the proof of the theorem. Suppose the parameters ε, δ, k satisfy the condition stated in the theorem. Suppose we have such a black-box construction realized by an oracle algorithm DEC with decision function D . Consider k independent random functions b_1, \dots, b_k from F_n , which will serve as noise vectors, such that for any i and x , $\Pr[b_i(x) = 0] = \frac{1+2\varepsilon}{2}$.

6.3. QUERY COMPLEXITY IN STRONGLY BLACK-BOX CONSTRUCTION 105

Now let f be a perfectly random function from F_n , so that $\Pr[f(x) = 1] = \frac{1}{2}$ for any x , and let g_1, \dots, g_k be k independent noisy versions of f defined as $g_i(x) = f(x) \oplus b_i(x)$, for any i and x . Let $B = \{b_1, \dots, b_k\}$ and $G = \{g_1, \dots, g_k\}$. First, we show that f is likely to be $(c\delta, \varepsilon, G)$ -easy.

Lemma 38 *If $k = \omega(\frac{1}{\varepsilon^2} \log \frac{1}{\delta})$, then $\Pr_{f,G} [f \text{ is not } (c\delta, \varepsilon, G)\text{-easy}] = o(1)$.*

Proof. Consider any $H \subseteq \{0, 1\}^n$ of size $c\delta 2^n$. We call f hard on H if $\Pr_{x \in H} [g_i(x) \neq f(x)] > \frac{1-\varepsilon}{2}$ for every i . Note that for any i , the random variables $b_i(x)$'s, for $x \in H$, are i.i.d. with $\mathbb{E}[b_i(x)] = \frac{1-2\varepsilon}{2}$, and $b_i(x) = 1$ exactly when $g_i(x) \neq f(x)$. Then the probability that f is hard on H equals

$$\begin{aligned} \Pr_B \left[\forall i \in [k] : \sum_{x \in H} b_i(x) > \frac{1-\varepsilon}{2} |H| \right] &= \prod_{i \in [k]} \Pr_{b_i} \left[\sum_{x \in H} b_i(x) > \frac{1-\varepsilon}{2} |H| \right] \\ &\leq \left(2^{-\Omega(\varepsilon^2 \delta 2^n)} \right)^k, \end{aligned}$$

where the equality is due to the fact that each b_i is independent from others and the inequality uses the Chernoff bound.

Recall that f is not $(c\delta, \varepsilon, G)$ -easy exactly when f is hard on some H of size $c\delta 2^n$. Then by a union bound, we conclude that it happens with probability at most

$$\binom{2^n}{c\delta 2^n} \cdot 2^{-\Omega(\varepsilon^2 \delta 2^n k)} \leq 2^{O(\delta 2^n \log \frac{1}{\delta})} \cdot 2^{-\Omega(\varepsilon^2 \delta 2^n k)},$$

which is $o(1)$ when $k = \omega(\frac{1}{\varepsilon^2} \log \frac{1}{\delta})$. \square

Next, we show that with a small q , DEC is unlikely to approximate f well. Recall that for a multi-set $I = \{i_1, \dots, i_q\} \subseteq [k]$, $g_I(x) = (g_{i_1}(x), \dots, g_{i_q}(x))$. We say that DEC can δ -approximate f if there is a multi-set $I \subseteq [k]$ with $|I| = q$ such that $D \circ g_I$ is δ -close to f (i.e., $\Pr_x [D(g_I(x)) \neq f(x)] \leq \delta$).

Lemma 39 *If $q = o(\frac{1}{\varepsilon^2} \log \frac{1}{\delta})$, then $\Pr_{f,G} [\text{DEC can } \delta\text{-approximate } f] = o(1)$.*

Proof. Consider any multi-set $I \subseteq [k]$ with $|I| = q$. First we show the following.

Claim 3 For any $x \in \{0, 1\}^n$, $\Pr_{f,G}[D(g_I(x)) \neq f(x)] \leq 2\delta$.

Proof. Let \tilde{I} denote the set of elements from I , removing multiplicity, and \tilde{D} the function such that $\tilde{D}(g_{\tilde{I}}(x)) = D(g_I(x))$. For example, for $I = \{1, 1, 2\}$, we have $\tilde{I} = \{1, 2\}$ and $\tilde{D}(g_1(x), g_2(x)) = D(g_1(x), g_1(x), g_2(x))$. Then

$$\Pr_{f,G}[D(g_I(x)) \neq f(x)] = \Pr_{f,G}[\tilde{D}(g_{\tilde{I}}(x)) \neq f(x)] = \frac{1}{2}p(0) + \frac{1}{2}p(1),$$

where $p(c) = \Pr_{f,G}[\tilde{D}(g_{\tilde{I}}(x)) = c \mid f(x) = 1 - c]$, so it suffices to give a lower bound for either $p(0)$ or $p(1)$. Let $\tilde{I} = \{i_1, \dots, i_{\tilde{q}}\}$, where \tilde{q} is clearly at most q . Assume without loss of generality that $|\tilde{D}^{-1}(1)| \geq 2^{\tilde{q}-1}$, and we will give a lower bound for $p(1)$ (otherwise, we bound $p(0)$ in a similar way). Let $Z = (Z_1, \dots, Z_{\tilde{q}})$ denote the sequence of random variables $(b_{i_1}(x), \dots, b_{i_{\tilde{q}}}(x))$, which are i.i.d. with $\mathbb{E}[Z_i] = \frac{1-2\varepsilon}{2}$. Note that $g_i(x) = b_i(x)$ when $f(x) = 0$, so we have

$$p(1) = \Pr_B \left[\tilde{D}(b_{i_1}(x), \dots, b_{i_{\tilde{q}}}(x)) = 1 \right] = \sum_{y \in \tilde{D}^{-1}(1)} \Pr[Z = y].$$

The above is the sum of $|\tilde{D}^{-1}(1)| \geq 2^{\tilde{q}-1}$ values from the $2^{\tilde{q}}$ values: $\Pr[Z = y]$ for $y \in \{0, 1\}^{\tilde{q}}$, so it is clearly no less than the sum of the $2^{\tilde{q}-1}$ smallest values from them. Observe that $\Pr[Z = y] = \left(\frac{1-2\varepsilon}{2}\right)^{\#_1(y)} \left(\frac{1+2\varepsilon}{2}\right)^{\tilde{q}-\#_1(y)}$, where $\#_1(y)$ denotes the number of 1's in the string y , so $\Pr[Z = y] \leq \Pr[Z = y']$ whenever $\#_1(y) \geq \#_1(y')$. As a result, $p(1)$ is at least

$$\sum_{y: \#_1(y) > \frac{1}{2}\tilde{q}} \Pr[Z = y] = \Pr \left[\sum_{i \in [\tilde{q}]} Z_i > \frac{1}{2}\tilde{q} \right] \geq \Pr \left[\sum_{i \in [\tilde{q}]} Z_i > \frac{1-\varepsilon}{2}\tilde{q} \right] \geq 2^{-O(\varepsilon^2\tilde{q})},$$

by Fact 1 (2). So when $\tilde{q} \leq q = o\left(\frac{1}{\varepsilon^2} \log \frac{1}{\delta}\right)$, we have $\Pr_{f,G}[D(g_I(x)) \neq f(x)] \geq \frac{1}{2}p(1) \geq 2\delta$. \square

Now for any multi-set I with $|I| = q$, let Y_x , for $x \in \{0, 1\}^n$, denote the binary random variable such that $Y_x = 1$ if and only if $D(g_I(x)) \neq f(x)$. Clearly, they are i.i.d. random variables, and we know from above that $\mathbb{E}[Y_x] \geq 2\delta$ for any x . So by Chernoff bound, $\Pr_{f,G}[D \circ g_I \text{ is } \delta\text{-close to } f] =$

$\Pr[\sum_x Y_x \leq \delta 2^n] \leq 2^{-\Omega(\delta^2 2^n)}$. Then a union bound gives

$$\Pr_{f,G}[\exists I \text{ with } |I| = q : D \circ g_I \text{ is } \delta\text{-close to } f] \leq k^q \cdot 2^{-\Omega(\delta^2 2^n)},$$

which is $o(1)$ as we assume $\varepsilon, \delta \geq 2^{-c_1 n}$ and $k \leq 2^{2^{c_3 n}}$, for small enough constants $c_1, c_3 > 0$. \square

From Lemma 38 and 39, we conclude that there exist $f \in F_n$ and $G = \{g_1, \dots, g_k\} \subseteq F_n$ which satisfy the following:

- f is (δ, ε, G) -easy, and
- for every multi-set $I \subseteq [k]$ of size $q = o(\frac{1}{\varepsilon^2} \log \frac{1}{\delta})$, $\Pr_x [D(g_I(x)) \neq f(x)] > \delta$.

Therefore, any algorithm DEC which realizes a strongly black-box (δ, ε, k) -construction must have $q = \Omega(\frac{1}{\varepsilon^2} \log \frac{1}{\delta})$, when $k = \omega(\frac{1}{\varepsilon^2} \log \frac{1}{\delta})$. This completes the proof of Theorem 16.

6.4 Advice Complexity in Weakly Black-Box Construction

In this section, we show a length lower bound on the advice needed in any weakly black-box construction of hard-core set. This explains why a uniform version of the hard-core set lemma is hard to come by and any black-box construction is inherently non-uniform. Formally, we have the following.

Theorem 17 *Suppose $2^{-c_1 n} \leq \varepsilon, \delta < c_2$, and $\frac{1}{\varepsilon^3} \leq k \leq 2^{2^{c_3 n}}$, for small enough constants $c_1, c_2, c_3 > 0$. Then any weakly black-box (δ, ε, k) -construction must need an advice of length $\Omega(\frac{1}{\varepsilon} \log k)$.*

As a comparison, the construction of Klivans and Servedio (Lemma 37) provides an upper bound of $O(\frac{1}{\varepsilon^2} \log \frac{1}{\delta} \log k)$ on the advice length, so there is a gap of a factor $O(\frac{1}{\varepsilon} \log \frac{1}{\delta})$ between our lower bound and their upper bound.

As in Theorem 16, one can also argue that the range assumption on the parameters is reasonable.

The roadmap for the proof is the following. Consider any DEC realizing such a weakly black-box construction. We will show the existence of a family $G = \{g_1, \dots, g_k\} \subseteq F_n$ with respect to which we can find a large collection Γ of functions satisfying the following two properties: (1) any function in Γ is $(c\delta, \varepsilon, G)$ -easy, and (2) any two functions in Γ are not 2δ -close. This then implies a lower bound of $\log |\Gamma|$ on the advice length. Again, we will show the existence of G by a probabilistic argument.

Now we proceed to the proof of the theorem. First, we independently sample k perfectly random functions $g_1, \dots, g_k \in F_n$ (for any i and x , $g_i(x) = 1$ with probability exactly $\frac{1}{2}$), and let $G = \{g_1, \dots, g_k\}$. Now for any set $I = \{i_1, \dots, i_t\} \subseteq [k]$, let G_I be the function such that

$$G_I(x) = \text{MAJ}(g_{i_1}(x), \dots, g_{i_t}(x)),$$

where MAJ denotes the majority function. Then we have the following, which follows from the known result that any majority gate has a good correlation with one of its input bits [HMP⁺87, GHR92]. For completeness we give its proof.

Lemma 40 *Let $G = \{g_1, \dots, g_k\}$ be any set of functions from F_n . Then for any $I \subseteq [k]$, the function G_I is $(c\delta, \frac{1}{|I|}, G)$ -easy.*

Proof. Let I be a multi-set of size t . For any $H \subseteq \{0, 1\}^n$, consider the $|H| \times t$ matrix M such that for $x \in H$ and $j \in I$, $M_{x,j} = 1$ if $g_j(x) = G_I(x)$ and 0 otherwise. Clearly, each row of M has more 1's than 0's, so the fraction of 1's must be at least $\frac{1}{2}(1 + \frac{1}{t})$ (otherwise, the number of 1's minus the number of 0's is less than $t \cdot \frac{1}{t} = 1$, a contradiction). Then by an averaging argument, some column must have at least this fraction of 1's. That is, for any $H \subseteq \{0, 1\}^n$ (including those of size $c\delta 2^n$), there exists a function $g_j \in G$ such that

$$\Pr_{x \in H} [g_j(x) = G_I(x)] \geq \frac{1}{2} \left(1 + \frac{1}{t} \right).$$

Therefore, G_I is a $(c\delta, \frac{1}{t}, G)$ -easy function. \square

Let $t = \lfloor \frac{1}{\varepsilon} \rfloor$, let $V^t = \{I \subseteq [k] : |I| = t\}$, and consider the class $\{G_I : I \in V^t\}$ of functions. From the previous lemma, we know that each function in the class is $(c\delta, \varepsilon, G)$ -easy. Our next step is to find a large collection of functions from this class such that any two of them are not close. Note that whether or not two functions G_I, G_J are close really depends on the choice of G . We will show that if I and J have a small intersection, then G_I and G_J are unlikely to be close for a random G .

Lemma 41 *For any $I, J \in V^t$ with $|I \cap J| < \frac{t}{2}$, $\Pr_G [G_I \text{ is } 2\delta\text{-close to } G_J] \leq 2^{-\Omega(2^n)}$.*

Proof. Consider any such I and J . First, we prove the following.

Claim 4 *For any $x \in \{0, 1\}^n$, $\Pr_G [G_I(x) \neq G_J(x)] = \Omega(1)$.*

Proof. Note that for any x , $g_1(x), \dots, g_k(x)$ can be seen as a sequence of i.i.d. binary random variables Z_1, \dots, Z_k , with $E[Z_i] = \frac{1}{2}$ for each i . Let Z_I denote the subsequence of random variables Z_i for $i \in I$, and similarly for Z_J . Thus our goal is to show that $\Pr[\text{MAJ}(Z_I) \neq \text{MAJ}(Z_J)] = \Omega(1)$.

Let $K = I \cap J$, $I_1 = I \setminus K$, and $J_1 = J \setminus K$, and note that $|K| < |I_1|, |J_1|$. Consider the following three events.

- A_1 : $\left| \sum_{i \in K} Z_i - \frac{|K|}{2} \right| \leq \frac{1}{2} \sqrt{|K|}$. By Chernoff bound, $\Pr[\neg A_1] < \alpha$ for a constant $\alpha < 1$, so $\Pr[A_1] = \Omega(1)$.
- A_2 : $\sum_{i \in I_1} Z_i \leq \frac{1}{2}(|I_1| - \sqrt{|I_1|})$. By Fact 1 (1) with $\mu = \frac{1}{2}$, $\Pr[A_2] = \Omega(1)$.
- A_3 : $\sum_{i \in J_1} Z_i \geq \frac{1}{2}(|J_1| + \sqrt{|J_1|})$. By Fact 1 (2) with $\mu = \frac{1}{2}$, $\Pr[A_3] = \Omega(1)$.

Now observe that if $A_1 \wedge A_2$, then

$$\sum_{i \in I} Z_i \leq \frac{1}{2}(|K| + |I_1| + \sqrt{|K|} - \sqrt{|I_1|}) < \frac{|K| + |I_1|}{2} = \frac{|I|}{2},$$

which implies that $\text{MAJ}(Z_I) = 0$. Similarly, if $A_1 \wedge A_3$, then

$$\sum_{i \in J} Z_i \geq \frac{1}{2}(|K| + |J_1| - \sqrt{|K|} + \sqrt{|J_1|}) > \frac{|K| + |J_1|}{2} = \frac{|J|}{2},$$

which implies that $\text{MAJ}(Z_J) = 1$. That is, if $A_1 \wedge A_2 \wedge A_3$, then $\text{MAJ}(Z_I) = 0 \wedge \text{MAJ}(Z_J) = 1$, so $\text{MAJ}(Z_I) \neq \text{MAJ}(Z_J)$. Since the events A_1, A_2, A_3 are independent from each other (as each depends on a separate set of random variables), we have

$$\Pr[\text{MAJ}(Z_I) \neq \text{MAJ}(Z_J)] \geq \Pr[A_1 \wedge A_2 \wedge A_3] = \Pr[A_1] \cdot \Pr[A_2] \cdot \Pr[A_3] \geq \Omega(1).$$

□

From this, we next show that G_I and G_J are unlikely to agree on many x . For any $x \in \{0, 1\}^n$, consider the binary random variable Y_x such that $Y_x = 1$ if and only if $G_I(x) \neq G_J(x)$. From the above claim, we know that $\mathbb{E}_G[Y_x] \geq c_0$ for some constant c_0 . So by Chernoff bound, we have

$$\Pr_G[G_I \text{ is } 2\delta\text{-close to } G_J] = \Pr\left[\sum_x Y_x \leq 2\delta 2^n\right] \leq 2^{-\Omega((c_0 - 2\delta)^2 2^n)} \leq 2^{-\Omega(2^n)},$$

as we assume that $\delta < c_2$ for a small enough constant c_2 . □

Call G nice if for any $I, J \in V^t$ with $|I \cap J| < \frac{t}{2}$, G_I is not 2δ -close to G_J . By a union bound,

$$\Pr_G[G \text{ is not nice}] \leq \binom{k}{t}^2 \cdot 2^{-\Omega(2^n)} \leq 2^{2t \log k} \cdot 2^{-\Omega(2^n)} < 1,$$

as we assume that $t \leq \frac{1}{\epsilon} \leq 2^{c_1 n}$ and $k \leq 2^{2c_3 n}$, for small enough constants $c_1, c_3 > 0$. This guarantees the existence of a nice G , and from now on, we will fix on one such G .

Consider the undirected graph $\mathcal{G} = (V, E)$ where $V = \{G_I : I \in V^t\}$ and E consists of those pairs of G_I, G_J which are 2δ -close to each other. Then we have the following.

Lemma 42 \mathcal{G} has an independent set of size at least $k^{\Omega(t)}$.

Proof. Since G is nice, there cannot be an edge between vertices G_I and G_J if $|I \cap J| < \frac{t}{2}$. Thus, the degree of any vertex G_I is at most the number of G_J with $|I \cap J| \geq \frac{t}{2}$, which is at most

$$\sum_{\frac{t}{2} \leq i < t} \binom{t}{i} \binom{k-t}{t-i} \leq \sum_{\frac{t}{2} \leq i < t} \binom{t}{i} \binom{k}{\frac{t}{2}} \leq 2^t \binom{k}{\frac{t}{2}} \leq \left(\frac{8ek}{t}\right)^{t/2} \leq k^{t/2},$$

where the first and last inequalities, respectively, hold under the conditions that $k \geq \frac{1}{\varepsilon^3} \geq t^3$ and $t = \lfloor \frac{1}{\varepsilon} \rfloor$ is at least a large enough constant, while the third inequality uses the fact that $\binom{n}{m} \leq \left(\frac{en}{m}\right)^m$. Then by Turán's theorem (Fact 6), \mathcal{G} has an independent set of size

$$\binom{k}{t} \frac{1}{k^{t/2} + 1} \geq \left(\frac{k}{t}\right)^t \frac{1}{k^{t/2} + 1} \geq k^{2t/3} \frac{1}{k^{t/2} + 1} \geq k^{\Omega(t)},$$

where the second inequality follows from the assumption that $k \geq t^3$. \square

Now we are ready to finish the proof of the theorem. From Lemma 42, we know that \mathcal{G} has an independent set Γ of size $k^{\Omega(t)}$. Note that any two $G_I, G_J \in \Gamma$ are not 2δ -close. Furthermore, we know from Lemma 40 that every $G_I \in \Gamma$ is $(c\delta, \varepsilon, G)$ -easy, since $|I| = t \leq \frac{1}{\varepsilon}$. Therefore, an advice of length $\log |\Gamma| = \Omega(t \log k) = \Omega(\frac{1}{\varepsilon} \log k)$ is required, because for each advice α , $\text{DEC}^{G, \alpha}$ is only δ -close to at most one $G_I \in \Gamma$. This proves Theorem 17.

6.5 Weakly Black-Box Construction $\notin \text{AC}^0[p]$

In this section, we show that no weakly black-box construction of hard-core set can be realized in $\text{AC}^0[p]$. More precisely we have the following.

Theorem 18 *Suppose $1/20 \leq \delta < 1$, $0 < \varepsilon < 1$, $k \geq n$, and p a prime. Let $t = \min(\lfloor 1/\varepsilon \rfloor, n)$. Then no weakly black-box (δ, ε, k) -construction can be realized in $\text{AC}^0[p](2^{\text{poly}(\log t)})$.*

The idea is the following. Suppose we have a function DEC realizing such a black-box construction. Let $I = [t]$ and note that $1/t \geq \varepsilon$. From the previous section, we know that for any G , the function G_I is $(c\delta, \varepsilon, G)$ -easy,

so there must exist some advice α such that $\text{DEC}^{G,\alpha}$ is δ -close to the function G_I , which is the majority function over g_1, \dots, g_t . As will be shown later, by defining G properly, we can use $\text{DEC}^{G,\alpha}$ to approximate the majority function on t input variables. Then we need to show a lower bound on the majority function. To obtain it, we need two following theorems.

Theorem 19 [Smo87] *For any $C : \{0, 1\}^t \rightarrow \{0, 1\}$ in $\text{AC}^0[p](2^{\text{poly}(\log t)})$, there is a polynomial Q over $GF(p)$ of degree $\text{poly}(\log t)$ such that $\Pr_x[C(x) \neq Q(x)] \leq 2^{-\text{poly}(\log t)}$.*

Theorem 20 [Sze89, Tar91] *For any polynomial Q of degree $\text{poly}(\log t)$ and for a large enough t , $\Pr_x[Q(x) \neq \text{MAJ}(x)] \geq 1/10$.*

Now we are able to give a lower bound on the majority function.

Lemma 43 *For any $C : \{0, 1\}^t \rightarrow \{0, 1\}$ in $\text{AC}^0[p](2^{\text{poly}(\log t)})$ and for a large enough t , we have $\Pr_x[C(x) \neq \text{MAJ}(x)] \geq 1/20$.*

Proof. This lemma can be easily obtained from the above two theorems. In fact, by Theorem 19 and Theorem 20, we have $\Pr_x[C(x) \neq \text{MAJ}(x)] \geq 1/10 - 2^{-\text{poly}(\log n)} \geq 1/20$ for any $C : \{0, 1\}^t \rightarrow \{0, 1\}$ in $\text{AC}^0[p](2^{\text{poly}(\log t)})$ when t is large enough. \square

We define the function g_i , for $i \in I$, as $g_i(x) = x_i$, and define g_j , for $j \notin I$, as $g_j(x) = 0$, for $x \in \{0, 1\}^n$. Let $G = \{g_1, \dots, g_k\}$. Then $G_I(x) = \text{MAJ}(x_1, \dots, x_t)$ for any $x \in \{0, 1\}^n$, so there must be some advice α such that $\Pr_x[\text{DEC}^{G,\alpha}(x) \neq \text{MAJ}(x_1, \dots, x_t)] \leq \delta$, and by an average argument there must be some fixed $\bar{x}_{t+1}, \dots, \bar{x}_n$ such that

$$\Pr_{x_1, \dots, x_t} [\text{DEC}^{G,\alpha}(x_1, \dots, x_t, \bar{x}_{t+1}, \dots, \bar{x}_n) \neq \text{MAJ}(x_1, \dots, x_t)] \leq \delta.$$

Such α and $\bar{x}_{t+1}, \dots, \bar{x}_n$ can be hard-wired into the circuit for DEC , and observe that each oracle query of DEC can be simulated easily. So if DEC is computable by an $\text{AC}^0[p](2^{\text{poly}(\log t)})$ circuit, we can get another $\text{AC}^0[p](2^{\text{poly}(\log t)})$ circuit which is δ -close to the majority function on t bits and contradicts Lemma 43. This proves Theorem 18.

Bibliography

- [AS00] Noga Alon and Joel Spencer. *The probabilistic method*. Wiley-Interscience, New York, second edition, 2000.
- [Bil95] Patrick Billingsley. *Probability and measure, 3rd Edition*. Wiley & Sons, 1995.
- [BFNW93] László Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3(4), pages 307–318, 1993.
- [BM82] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo random bits. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 112–117, 1982.
- [BT03] Andrej Bogdanov and Luca Trevisan. On worst-case to average-case reductions for NP problems. In *44th Annual Symposium on Foundations of Computer Science*, Cambridge, Massachusetts, pages 11–14, 2003.
- [CW79] Larry Carter and Mark N. Wegman. Universal Classes of Hash Functions. *Journal of Computer and System Sciences*, 18(2), pages 143–154, 1979.

- [CG89] Benny Chor and Oded Goldreich. On the power of two-point based sampling. *Journal of Complexity*, 5(1): pages 96-106, March 1989.
- [DK00] Ding-Zhu Du and Ker-I Ko. *Theory of Computational Complexity*. John Wiley & Sons, Inc. New York, 2000.
- [FSS84] Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1), pages 13–27, 1984.
- [Gol01] Oded Goldreich. *Foundations of Cryptography: Volume 1, Basic Tools*. Cambridge University Press, Cambridge, 2001.
- [GHR92] Mikael Goldmann, Johan Håstad, and Alexander A. Razborov. Majority gates vs. general weighted threshold gates. *Computational Complexity*, 2, pages 277–300, 1992.
- [GNW95] Oded Goldreich, Noam Nisan, and Avi Wigderson. On Yao’s XOR lemma. Technical Report TR95–050, Electronic Colloquium on Computational Complexity, 1995.
- [GV05] Venkatesan Guruswami and Salil Vadhan. A lower bound on list size for list decoding. In *Proceedings of the 8th International Workshop on Randomization and Computation (RANDOM ‘05)*, pages 318–329, 2005.
- [Has86] Johan Håstad. *Computational limitations for small depth circuits*. PhD thesis, MIT Press, 1986.
- [HILL99] Johan Håstad, Russel Impagliazzo, Leonid Levin and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4), pages 1364–1396, 1999.
- [HMP+87] Andras Hajnal, Wolfgang Maass, Pavel Pudlák, Máriaó Szegedy, and György Turán. Threshold circuits of bounded depth, In

Proceedings of the 28th Annual IEEE Symposium on Foundations of Computer Science, pages 99–110, 1987.

- [HVV04] Alexander Healy, Salil P. Vadhan, and Emanuele Viola. Using nondeterminism to amplify hardness. In *Proceedings of the 36th ACM Symposium on Theory of Computing*, pages 192–201, 2004.
- [Hol05] Thomas Holenstein. Key agreement from weak bit agreement. In *Proceedings of the 37th ACM Symposium on Theory of Computing*, pages 664–673, 2005.
- [Im95] Russel 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.
- [IL90] Russel Impagliazzo and Leonid Levin. No better ways to generate hard NP instances than picking uniformly at random. In *Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science*, pages 812–821, 1990.
- [ISW00] Russell Impagliazzo, Ronen Shaltiel, and Avi Wigderson. Extractors and pseudo-random generators with optimal seed length. In *Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 1–10, 2000.
- [IW97] Russel Impagliazzo and Avi Wigderson. P=BPP if E requires exponential circuits: derandomizing the XOR lemma. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 220–229, 1997.
- [IW98] Russel Impagliazzo and Avi Wigderson. Randomness vs. time: de-randomization under a uniform assumption. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, pages 734–743, 1998.

- [IZ89] Russel Impagliazzo and David Zuckerman. How to recycle random bits. In *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, pages 248–253, 1989.
- [KM02] Adam Klivans and Dieter van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM Journal on Computing*, 31(5), pages 1501–1526, 2002.
- [KS03] Adam Klivans and Rocco A. Servedio. Boosting and hard-core sets. *Machine Learning*, 51(3): pages 217–238, 2003.
- [LMN93] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, Fourier transform, and learnability. *Journal of the ACM*, 40(3), pages 607–620, 1993.
- [Lu02] Chi-Jen Lu. Improved pseudorandom generators for combinatorial rectangles. *Combinatorica*, 22(3), pages 417–434, 2002.
- [LTW05] Chi-Jen Lu, Shi-Chun Tsai, and Hsin-Lung Wu. On the complexity of hardness amplification. In *Proceedings of the 20th Annual IEEE Conference on Computational Complexity*, pages 170–182, 2005.
- [LTW05a] Henry Lin, Luca Trevisan, and Hoeteck Wee. On hardness amplification of one-way functions. In *Proceedings of the 2nd Theory of Cryptography Conference*, pages 34–49, 2005.
- [Nis91] Noam Nisan. Pseudorandom bits for constant depth circuits. *Combinatorica*, 11(1), pages 63–70, 1991.
- [Nis92] Noam Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4), pages 449–461, 1992.

- [NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of Computing System Science*, 49(2), pages 149–167, 1994.
- [OD02] Ryan O’Donnell. Hardness amplification within NP. In *Proceedings of the 34th ACM Symposium on Theory of Computing*, pages 751–760, 2002.
- [Pap94] Christos Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
- [Plo60] M. Plotkin. Binary codes with specified minimum distance. *IEEE Transactions on Information Theory*, 6, pages 445–450, 1960.
- [RTV04] Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Notions of Reducibility between Cryptographic Primitives. In *Proceedings of the 1st Theory of Cryptography Conference*, pages 1–20, 2004.
- [Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proceedings of the 19th ACM Symposium on Theory of Computing*, pages 77–82, 1987.
- [Sze89] Mario Mária Szegedy. *Algebraic methods in lower bounds for computational models with limited communication*. Ph.D. thesis, University of Chicago, 1989.
- [STV01] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62(2), pages 236–266, 2001.
- [SU01] Ronen Shaltiel and Christopher Umans. Simple extractors for all min-entropies and a new pseudo-random generator. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 648–657, 2001.

- [Tar91] Jun Tarui. Degree complexity of boolean functions and its applications to relativized separations. In *Proceedings of the 6th Annual IEEE Conference on Structure in Complexity Theory*, pages 382–390, 1991.
- [Tre03] Luca Trevisan. List decoding using the XOR lemma. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 126–135, 2003.
- [Tre05] Luca Trevisan. On uniform amplification of hardness in NP. In *Proceedings of the 37th ACM Symposium on Theory of Computing*, pages 31–38, 2005.
- [TV02] Luca Trevisan and Salil Vadhan. Pseudorandomness and average-case complexity via uniform reductions. In *Proceedings of the 17th Computational Complexity Conference*, pages 129–138, IEEE, 2002.
- [Uma03] Christopher Umans. Pseudo-random generators for all hardnesses. *Journal of Computer and System Sciences*. 67(2), pages 419–440, 2003.
- [Vio04] Emanuele Viola. The Complexity of Constructing Pseudorandom Generators from Hard Functions. In *Computational Complexity*, 13(3-4), pages 147–188, 2004.
- [Vio05] Emanuele Viola. On Constructing Parallel Pseudorandom Generators from One-Way Functions. In *Proceedings of the 20th Computational Complexity Conference*, pages 183–197, IEEE, 2005.
- [Vio06] Emanuele Viola. *The Complexity of Hardness Amplification and Derandomization*. Ph.D. thesis, Harvard University, 2006.

- [Yao82] Andrew Chi-Chih Yao. Theory and applications of trapdoor functions. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 80–91, 1982.

