行政院國家科學委員會專題研究計畫　期中進度報告

解隨機化之研究(2/3)

計畫類別：個別型計畫
計畫編號：NSC93-2213-E-009-035-
執行期間：93 年 08 月 01 日至 94 年 07 月 31 日
執行單位：國立交通大學資訊工程學系(所)

計畫主持人：蔡錫鈞

計畫參與人員：吳信龍，李佳蓉

報告類型：精簡報告
報告附件：出席國際會議研究心得報告及發表論文
處理方式：本計畫可公開查詢

中　華　民　國 94 年 5 月 26 日

行政院國家科學委員會補助專題研究計畫 □ 成 果 報 告
■期中進度報告

解隨機化之研究

計畫類別：■ 個別型計畫　□ 整合型計畫
計畫編號：NSC-93-2213-E-009-035
執行期間： 93 年 8 月 1 日至 94 年 7 月 31 日

計畫主持人：蔡錫鈞
共同主持人：
計畫參與人員： 吳信龍、李佳蓉

成果報告類型(依經費核定清單規定繳交)：■精簡報告 □完整報告

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、列管計畫及下列情形者外，得立即公開查詢
□涉及專利或其他智慧財產權，□一年□二年後可公開查詢

執行單位：國立交通大學資訊工程研究所
中 華 民 國 94 年 5 月 26 日

中文摘要：

一神諭過程AMP叫做半黑箱難度加大法從δ到δ’ ，假如對任何小電路均會算錯δ比例的布林函數f，任何小電路均會算AMP^f 達到δ’ 比例。在此δ < δ’且AMP只把f當成黑箱呼叫。我們証明半黑箱難度加大法從 2^{-n}到 2^{-n}不能在計算類Σ_i^P 做到。同樣的結果也適用於計算類 $SPACE(n^\varepsilon)$，在此ε<1。更進一步地，我們的方法可擴充去証明一些無條件的不可能的結果。

關鍵詞：半黑箱難度加大法、難度加大法

英文摘要：

An oracle procedure AMP is called semi-black-box hardness amplification from δ to δ’ if, for any f with which any small circuit disagrees on at least δ fraction, any small circuit must disagree with AMP^f on at least δ' fraction where δ <δ ' and AMP only uses
f as a black-box not its internal structure. We show that semi-black-box hardness amplification from 2^{-n} to 1/poly(n) cannot be done in Σ_i^P for every i unless there is a mildly hard function in Σ_i^P. The same result is also applicable to sub-linear space SPACE(n^{ε}) for constant ε <1. Moreover, our method can be extended to prove some unconditional impossibility results.

keywords: Semi-black-box hardness amplification, hardness amplification

1 Introduction

We say that a boolean function $f : \{0,1\}^n \to \{0,1\}$ is δ-hard against circuits of size $s(n)$ if any such circuit attempting to compute f fails on at least δ fraction in inputs. The parameter δ plays an important role in the hardness amplification. We often call f is worst-case hard, mildly hard and average-case hard if δ is 2^{-n}, $1/\mathrm{poly}(n)$ and $(1/2 - 2^{-\Omega(n)})$ respectively. Hardness amplification is to design a procedure which transforms a δ-hard function $f : \{0,1\}^n \to \{0,1\}$ against circuits of size $s(n)$ into δ'-hard function $f' : \{0,1\}^m \to \{0,1\}$ against circuits of size $s'(m)$ where $\delta \leq \delta'$ and $s'(m)$ is close to $s(n)$. We usually would like f' to be in the same class of f so as to establish the relation between different hardness assumptions within the same complexity class. The ultimate goal of hardness amplification is to establish the equivalence between worst-case hardness and average-case hardness in some complexity classes. Normally, to build such an equivalence is required to have exponential time complexity (e.g. E) or linear space complexity (e.g. $\mathsf{SPACE}(n)$) [10, 15, 11]. However, in some complexity classes, it is unknown whether worst-case hardness is equivalent to average-case hardness or not. For example, given a worst-case hard function in NP, is there any procedure which can convert it into an average-case hard function in NP? Only for some range of hardness (e.g. the initial function is mildly hard) is it possible to be done [18, 13, 10, 14, 8]. So the gap is from worst-case hardness to mild hardness.

The above discussion prompts an attempt to prove that some hardness amplification is indeed impossible. We must carefully clarify what type of hardness amplification we are talking about especially when it is still possible that an mildly hard function may indeed exist. The one we concern is called *black-box* hardness amplification. First we construct a new hard function f' by using initial function f as a black-box. Precisely there is an oracle Turing machine Amp such that $\textsc{Amp}^f = f'$. Note that Amp only uses f as an oracle. Secondly the hardness of f' is proved via a black-box approach. That is, there is an oracle Turing machine Dec such that, for any algorithm A computing f' correctly on at least δ' fraction, $\textsc{Dec}^A$ computes f correctly on at least δ fraction. Again Dec only used A as an oracle. In particular, we call the amplification procedures that doesn't use the oracle decode function Dec and only use initial function as an oracle not its internal structure as *semi-black-box* hardness amplification. Almost all previous hardness amplification results are black-box-type [2, 6, 9, 10, 15, 11, 8]. Moreover the impossibility results for black-box type were also studies by Viola [16], Bogdanov and Trevisan [3], and Lu et al. [12]. The impossibility result of semi-black-box hardness amplification was first obtained by Viola [17]. Suppose that there is an Amp in PH such that, for every f which is worst-case hard against circuits of size $S(n)$, $\textsc{Amp}^f$ has constant hardness against circuits of size $S'(n)$. Viola showed that the existence of such Amp is equivalent to the existence of a function f' in PH which has constant hardness against circuits of size $S'(n)$. In short, such Amp just memories a constant-hard function if it can be realized in PH.

1.1 Our Results

In this paper, we give the negative results similar to Viola's except that we replace the complexity class PH with NP, Σ_i^{P}, or sublinear space. Also, we generalize the "fooled classes", not just circuits of certain size. More specifically, we give the definition and our main results as follows. Let $\mathcal{C}_n$ be the set $\{\mathrm{L} \cap \{0,1\}^n : \mathrm{L} \in \mathcal{C}\}$. Each element in $\mathcal{C}_n$ can be viewed as a function from $\{0,1\}^n$ to $\{0,1\}$, that is its characteristic function.

Definition 1. *Let $\mathcal{C}$ be a complexity class. We say that a function $f : \{0,1\}^n \to \{0,1\}$ has hardness δ against $\mathcal{C}$ if for any $A : \{0,1\}^n \to \{0,1\}$ in $\mathcal{C}_n$, $\Pr_{x \in U_n}[f(x) \neq A(x)] \geq \delta$. When δ*

is 2^{-n}, $1/\mathrm{poly}(n)$ *and* $1/2 - 2^{\Omega(n)}$, *f is called worst-case hard, mildly hard and average-case hard respectively.*

For the nondeterministic classes, we obtain the following.

Theorem 1. *Let $\mathcal{A}$ and $\mathcal{B}$ be two complexity classes with $|\mathcal{A}_n| = 2^{2^{o(n)}}$. Suppose there is an oracle machine* Amp *in* NP *(respectively, Σ_i^{P}) which converts every $f : \{0,1\}^n \to \{0,1\}$ of worst-case hardness against $\mathcal{A}$ into a function $\textsc{Amp}^f : \{0,1\}^m \to \{0,1\}$ of mild hardness against $\mathcal{B}$. Then there exists a function $f' : \{0,1\}^m \times \{0,1\}^{n^b} \to \{0,1\}$ in* NP *(respectively, Σ_i^{P}) such that $f'(x,t)$ has mild hardness against $\mathcal{B}$ on at least $(1 - 1/\mathrm{poly}(n))$ fraction of $t \in \{0,1\}^{n^b}$.*

In addition to nondeterministic classes, we also have the impossibility results for deterministic classes for time and space.

Theorem 2. *Let $\mathcal{A}$ and $\mathcal{B}$ be two complexity classes with $|\mathcal{A}_n| = 2^{2^{o(n)}}$. For every constant ε with $0 < \varepsilon < 1$, suppose there is an oracle machine* Amp *in* $\mathsf{SPACE}(n^\varepsilon)$ *(respectively,* $\mathsf{DTIME}(2^{n^\varepsilon})$*) which converts every $f : \{0,1\}^n \to \{0,1\}$ of worst-case hardness against $\mathcal{A}$ into a function $\textsc{Amp}^f : \{0,1\}^m \to \{0,1\}$ of mild hardness against $\mathcal{B}$. Then there exists a function $f' : \{0,1\}^m \to \{0,1\}$ in* $\mathsf{SPACE}(m^\varepsilon)$ *(respectively,* $\mathsf{DTIME}(2^{m^\varepsilon})$*) that has mild hardness against $\mathcal{B}$.*

Note that Theorem 1 and 2 are incomparable since the relation between SUBEXP and NP is not clear so far.

Our argument in the proof of the main theorems implies some impossibility results. For example, let $\mathcal{A} = \mathsf{BPP}$ and $\mathcal{B} = \mathsf{SUBEXP}$. We immediately obtain that there is a function in SUBEXP which is hard against SUBEXP. That is a contradiction. Hence no semi-black-box hardness amplification that can convert any function which is worst-case hard against BPP into another one which is δ-hard against SUBEXP can be realized in SUBEXP.

There is another interesting fact from our results. Consider the following question: **Given a class $\mathcal{C}$ and a value $\delta < 1$, how complex is it to build an operator G such that, for all $f \notin \mathcal{C}$, G^f is δ-hard against $\mathcal{C}$?** If $\delta = 2^{-n}$, then it is trivial to make G the identity function. So the complexity is low. Our results show that for some complexity classes, such as P, if δ is somewhat non-negligible, then such G cannot be constructed in P. We will give more discussion in Section 5.

1.2 Organization of this paper

First, some preliminaries are given in Section 2. Then in Section 3 and Section 4, we prove the impossibility results of semi-black-box hardness amplification in NP (Σ_i^{P}) and sublinear space respectively. In Section 5, we use the technique developed in this paper to show the unconditional impossibility results for semi-black-box construction.

2 Preliminaries

Let $\mathbf{F} : \{0,1\}^n \to \{0,1\}$ be a uniform random function. N denotes 2^n, unless mentioned otherwise. U_n is the uniform distribution on $\{0,1\}^n$ for each integer n. We will use the method of random restriction. A restriction on a set of variables $V = \{x_i : i \in \{0,1\}^n\}$ is a mapping $\rho : V \to \{0,1,*\}$, which either fixes the value of a variable x_i with $\rho(x_i) \in \{0,1\}$ or leaves x_i free with $\rho(x_i) = *$.

A boolean function $f : \{0,1\}^n \to \{0,1\}$ is usually viewed as a N-bit string, i.e. its truth table. Given a restriction $\rho : \{0,1\}^n \to \{0,1,*\}$ and a function $f : \{0,1\}^n \to \{0,1\}$, f_ρ is the N-bit truth table obtained from ρ by substituing the *'s with the corresponding bits of f. For any function $B : \{0,1\}^N \to \{0,1\}$, let $B_\rho(f) \stackrel{def}{=} B(f|_\rho)$ for each $f : \{0,1\}^n \to \{0,1\}$ and let $\widetilde{\text{BIAS}}[B] = \max\{\Pr[B(U_N) = 0], \Pr[B(U_N) = 1]\}$. Using this definition we can easily get the following lemma.

Lemma 1. *Let U'_N be a distribution which is independently and identically distributed to U_N. Suppose that $\Pr_{U_N,U'_N}[B(U_N) \neq B(U'_N)] \leq n^{-c}$. Then $\widetilde{\text{BIAS}}[B] \geq 1 - n^{-c}$ for any constant c.*

Proof. Let η be $\Pr_{U_N}[B(U_N) = 1]$. Without loss of generality, suppose $\eta \geq 1/2$. So $(1-\eta) \leq 2\eta(1-\eta)) = \Pr_{U_N,U'_N}[B(U_N) \neq B(U'_N)] \leq n^{-c}$. □

The circuits we consider consist of AND/OR/NOT gates, allowing unbounded fan-in for AND/OR gates. The size of a circuit is the number of its gates 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 2. *Let $\mathsf{AC}(s)$ be the class of boolean functions computed by AC circuits of size s. Let $\mathsf{AC}(d,s)$ denote the class of boolean functions computed by AC circuits of depth d and size s. If the fan-in is bounded, let $\mathsf{SIZE}(s)$ be the class of boolean functions computed by bounded fanin circuits of size s.*

Note that the standard complexity class AC^0 corresponds to our class $\mathsf{AC}(O(1), \text{poly}(n))$. We also introduce the non-deterministic circuits. A non-deterministic circuit C has two kind 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 y such that $C(x,y) = 1$.

Definition 3. *Let $\mathsf{NSIZE}(s)$ be the class of functions computed by non-deterministic circuits of size s. In particular, we denote $\mathsf{NSIZE}(\text{poly}) = \bigcup_{s:s \text{ is a polynomial}} \mathsf{NSIZE}(s)$.*

3 Proof of Theorem 1

In this section we carefully analyze Viola's proof [17] to obtain a more elaborate result. First we need a lemma proved by Viola.

Lemma 2. *[17] For every constant $c > 0$, there is a distribution $\tilde{R}_c$ on restrictions $\rho : \{0,1\}^n \to \{0,1,*\}$ such that*

(1) Each ρ in support of $\tilde{R}_c$ can be generated by a polynomial time algorithm which outputs $\rho_t(x)$ with input $x \in \{0,1\}^n$ and $\text{poly}(n)$-bit random string t.

(2) Let $\mathbf{F}' : \{0,1\}^n \to \{0,1\}$ be a random function which is independent and identically distributed to $\mathbf{F}$. For every N-bit circuit B in $\mathsf{AC}(c, 2^{n^c})$,

$$\Pr_{\rho_t \in \tilde{R}_c}\left[\Pr_{\mathbf{F},\mathbf{F}'}\left[B(\mathbf{F}|_{\rho_t}) \neq B(\mathbf{F}'|_{\rho_t})\right] \leq \frac{1}{n^{c/2}}\right] \geq (1 - O(\frac{1}{n^{c/2}})).$$

(3) With probability $(1 - O(\frac{1}{n^{c/2}}))$ over $\rho_t \in \tilde{R}_c$, ρ_t has at least $2^n/3n^{c^2}$ $*$'s.

We use the notion of promise set [1] to rephrase a lemma proved by Nisan and Wigderson [13].

Lemma 3. *[13] Suppose there is a promise set (A, B) such that there is an oracle machine M in NP (respectively, Σ_i^{P}) with $\Pr_{\mathbf{F}}\left[\ \forall x \in A \cup B,\ x \in A \Leftrightarrow M^{\mathbf{F}}(x) = 1\right] \geq \frac{2}{3}$. Then there is a generator $G : \{0,1\}^{\mathrm{poly}(n)} \to \{0,1\}^{2^n}$ such that*

(1) given r and index $i \leq 2^n$, the i-th bit of $G(r)$ can be computed in $\mathsf{DTIME}(\mathrm{poly}(n))$ and

(2) $\Pr_r\left[\ \forall x \in A \cup B,\ x \in A \Leftrightarrow M^{G(r)}(x) = 1\right] \geq 1 - 2^{-n}$.

Note that, for any $r \in \{0,1\}^{\mathrm{poly}(n)}$, $M^{G(r)}$ is in NP (respectively, Σ_i^{P}) while M is in NP (respectively, Σ_i^{P}). Now we use these two lemmas to prove our first main theorem.

Proof. (of Theorem 1) For each input x, $\textsc{Amp}^f(x)$ can be simulated in $\mathsf{AC}(k, 2^{O(n^k)})$ with input f where n^k is the running time of $\textsc{Amp}$ [5, 7]. By Lemma 2 (2), for each input x,

$$\Pr_{\rho_t \in \tilde{R}_k}\left[\Pr_{\mathbf{F},\mathbf{F}'}\left[\textsc{Amp}^{\mathbf{F}|_{\rho_t}}(x) \neq \textsc{Amp}^{\mathbf{F}'|_{\rho_t}}(x)\right] \leq \frac{1}{n^{k/2}}\right] \geq (1 - O(\frac{1}{n^{k/2}})).$$

By Lemma 1, this implies that, for each x,

$$\Pr_{\rho_t \in \tilde{R}_k}\left[\widetilde{\textsc{Bias}}\left[\textsc{Amp}^{\mathbf{F}|_{\rho_t}}(x)\right] \geq (1 - n^{-k/2})\right] \geq (1 - O(\frac{1}{n^{k/2}})).$$

We call ρ_t is good for x if $\widetilde{\textsc{Bias}}\left[\textsc{Amp}^{\mathbf{F}|_{\rho_t}}(x)\right] \geq (1 - n^{-k/2})$ and bad otherwise. So we have

$$\Pr_{x,\rho_t}[\rho_t \text{ is good for } x] \geq (1 - O(\frac{1}{n^{k/2}})).$$

By Markov argument,

$$\Pr_{\rho_t}\left[\Pr_x[\rho_t \text{ is good for } x] \geq 1 - O(n^{-k/4})\right] \geq 1 - O(n^{-k/4}).$$

Call ρ_t **good** if $\Pr_x[\rho_t \text{ is good for } x] \geq 1 - O(n^{-k/4})$. With this definition, $\Pr[\rho_t \text{ is } \mathbf{bad}] \leq O(n^{-k/4})$. For each **good** ρ_t, we define a promise set (A_{ρ_t}, B_{ρ_t}) such that, for any $x \in A_{\rho_t} \cup B_{\rho_t}$,

- $x \in A_{\rho_t}$ if $\Pr_{\mathbf{F}}\left[\textsc{Amp}^{\mathbf{F}|_{\rho_t}}(x) = 1\right] \geq (1 - n^{-k/2})$ and
- $x \notin B_{\rho_t}$ if $\Pr_{\mathbf{F}}\left[\textsc{Amp}^{\mathbf{F}|_{\rho_t}}(x) = 0\right] \geq (1 - n^{-k/2})$.

Note that $|A_{\rho_t} \cup B_{\rho_t}| \geq 2^n \cdot (1 - O(n^{-k/4}))$. So if ρ_t is **good**, then we have

$$\Pr_{\mathbf{F}}\left[\forall x \in A_{\rho_t} \cup B_{\rho_t}\ , x \in A_{\rho_t} \Leftrightarrow \textsc{Amp}^{\mathbf{F}|_{\rho_t}}(x) = 1\right] = 1 - o(1).$$

By Lemma 3, there exists a generator $G : \{0,1\}^{\mathrm{poly}(n)} \to \{0,1\}^N$ such that

$$\Pr_r\left[\forall x \in A_{\rho_t} \cup B_{\rho_t}\ , x \in A_{\rho_t} \Leftrightarrow \textsc{Amp}^{G(r)|_{\rho_t}}(x) = 1\right] \geq 1 - \frac{1}{2^n}.$$

Now we define

$$f'(x,t,r) \stackrel{def}{=} \textsc{Amp}^{G(r)|_{\rho_t}}(x).$$

Amp is in NP (respectively, Σ_i^{P}), so is f'.

Next we analyze the hardness of f'. We bound the following probability:

$$\begin{aligned}
& \Pr_{x,t,r,\mathbf{F}}\left[f'(x,t,r) \neq \textsc{Amp}^{\mathbf{F}|_{\rho_t}}(x)\right] \\
\leq\ & \Pr\left[\rho_t \text{ is } \mathbf{bad}\right] + \Pr\left[\textsc{Amp}^{G(r)|_{\rho_t}}(x) \neq \textsc{Amp}^{\mathbf{F}|_{\rho_t}}(x) \mid \rho_t \text{ is } \mathbf{good}\right] \\
\leq\ & O(n^{-k/4}) + 2^{-n} + O(n^{-k/4}) \\
=\ & O(n^{-k/4}).
\end{aligned}$$

So

$$\Pr_{t,r,F}\left[\Pr_x\left[f'(x,t,r) \neq \textsc{Amp}^{\mathbf{F}|_{\rho_t}}(x)\right] \geq n^{-k/8}\right] \leq O(n^{-k/8}).$$

By Lemma 2 (3), we know that there is a constant c_1 such that with probability $(1 - O(\frac{1}{n^{c/2}}))$ over $\rho_t \in \tilde{R}_c$, by a counting argument

$$\Pr_{\mathbf{F}}\left[\mathbf{F}|_{\rho_t} \text{ is worst-case hard against } \mathcal{A}\right] \geq (1 - 2^{-2^{c_1 n}}).$$

So, by the assumption of Theorem 1, there is some constant d,

$$\Pr_{\mathbf{F}}\left[\textsc{Amp}^{\mathbf{F}|_{\rho_t}} \text{ has hardness } n^{-d} \text{ against } \mathcal{B}\right] \geq (1 - 2^{-2^{c_1 n}}).$$

Let C be any algorithm with input (x,t,r). Suppose that $C(x,t,r)$ is in $\mathcal{B}_n$ for any t and r. Then there is a constant e such that

$$\begin{aligned}
& \Pr_{x,t,r}\left[f'(x,t,r) = C(x,t,r)\right] \\
\leq\ & \Pr\left[\rho_t \text{ is } \mathbf{bad}\right] + \Pr\left[\textsc{Amp}^{G(r)|_{\rho_t}}(x) = C(x,t,r) \mid \rho_t \text{ is } \mathbf{good}\right] \\
\leq\ & O(n^{-k/4}) + \Pr\left[\textsc{Amp}^{\mathbf{F}|_{\rho_t}}(x) = C(x,t,r) \mid \rho_t \text{ is } \mathbf{good}\right] + 2^{-n} \\
\leq\ & \Pr\left[\textsc{Amp}^{\mathbf{F}|_{\rho_t}} \text{ doesn't have hardness } n^{-d}\right] + (1 - n^{-d}) + O(n^{-k/4}) \\
\leq\ & 1 - n^{-e}.
\end{aligned}$$

By Markov argument,

$$\Pr_{t,r}\left[\Pr_x\left[f'(x,t,r) = C(x,t,r)\right] \leq 1 - n^{-e}/2\right] \leq 1 - n^{-e}/2.$$

Therefore $f'(x,t,r)$ is mildly hard against $\mathcal{B}$ on at least $(1 - n^{-e}/2)$ fraction (t,r).

□

Observing the proof of Theorem 1, we can extend the condition that $|\mathcal{A}_n| = 2^{2^{dn}}$ for some constant $d < 1$. The theorem still holds. Let $\mathcal{A} = \mathsf{SIZE}(2^{cn})$ and $\mathcal{B} = \mathsf{SIZE}(2^{dn})$. Clearly We obtain the following corollary.

Corollary 1. *Suppose there is an oracle machine* AMP *in* NP *(respectively,* Σ_i^{P}*) which converts every* $f : \{0,1\}^n \to \{0,1\}$ *of worst-case hardness against* $\mathsf{SIZE}(2^{cn})$ *into a function* $\text{AMP}^f : \{0,1\}^m \to \{0,1\}$ *of mild hardness against* $\mathsf{SIZE}(2^{dm})$*. Then there exists a function* $f' : \{0,1\}^m \times \{0,1\}^{n^b} \to \{0,1\}$ *in* NP *(respectively,* Σ_i^{P}*) such that* $f'(x,t)$ *has mild hardness against* $\mathsf{SIZE}(2^{dm})$ *for some constant* c *and* d*.*

4 Proof of Theorem 2

In this section, we prove the second main result.

Proof. (of Theorem 2) Unlike the method used in the previous section, we want to claim that if AMP satisfies the condition of Theorem 2, then $\text{AMP}^{\vec{0}}$ is a constant-hard function where $\vec{0}$ is the zero boolean function. We use a probabilistic argument to prove it. Suppose our AMP can be realized in $\mathsf{SPACE}(\log T)$ (respectively, $\mathsf{DTIME}(T)$) where $T = O(2^{n^\varepsilon})$. Note that every algorithm in $\mathsf{SPACE}(\log T)$ can be computed by a algorithm in $\mathsf{DTIME}(T)$. So we only consider $\mathsf{DTIME}(T)$. Let $\delta = T^{-2}$. Define the following random function $\mathbf{g}$: for any $x \in \{0,1\}^n$,

$$\mathbf{g}(x) = \begin{cases} U_1 & \text{with probability } \delta \\ 0 & \text{with probability } 1-\delta\ . \end{cases}$$

We claim that such a random function $\mathbf{g}$ satisfies the following properties.

Claim 1. *W.h.p. over* $\mathbf{g}$*,* $\mathbf{g}$ *is worst-case hard against* $\mathcal{A}$*. Therefore* $\text{AMP}^{\mathbf{g}}$ *is mildly hard against* $\mathcal{B}$*.*

Proof. Let $\#\mathbf{g}$ be the number of inputs whose outputs are random bits. On average, $\#\mathbf{g}$ is about $\delta \cdot 2^n$. In fact, by Chernoff bound,

$$\Pr\left[\#\mathbf{g} < \delta \cdot 2^{n-1}\right] \le \Pr\left[|\#\mathbf{g} - \delta \cdot 2^n| > \delta \cdot 2^{n-1}\right] \le 2e^{-\frac{\delta 2^n}{8}} = 2^{-2^{\Omega(n)}}.$$

Therefore, with probability $(1 - 2^{-2^{\Omega(n)}})$, $\#\mathbf{g} \ge \delta \cdot 2^{n-1} = 2^{\Omega(n)}$. Again, by a counting argument, w.h.p. $\mathbf{g}$ is worst-case hard against $\mathcal{A}$. Hence w.h.p. $\text{AMP}^{\mathbf{g}}$ is mildly hard against $\mathcal{B}$. □

Claim 2. *For any* $x \in \{0,1\}^n$*, w.h.p. over* $\mathbf{g}$*,* $\text{AMP}^{\mathbf{g}}(x) = \text{AMP}^{\vec{0}}(x)$*.*

Proof. For every input x, the running time of AMP is at most T. So it queries at most T times. Therefore, for any input x,

$$\Pr_{\mathbf{g}}\left[\text{AMP}^{\mathbf{g}}(x) = \text{AMP}^{\vec{0}}(x)\right] \ge (1-\delta)^T \ge 1 - \delta \cdot T = 1 - T^{-1}.$$

□

By Claim 2, we know that

$$\Pr_{\mathbf{g},x}\left[\text{AMP}^{\mathbf{g}}(x) \ne \text{AMP}^{\vec{0}}(x)\right] \le T^{-1}.$$

By Markov argument,

$$\Pr_{\mathbf{g}}\left[\Pr_x\left[\text{AMP}^{\mathbf{g}}(x) \ne \text{AMP}^{\vec{0}}(x)\right] \ge T^{-1/2}\right] \le T^{-1/2}.$$

So with probability $1 - T^{-1/2}$ over $\mathbf{g}$, we have

$$\Pr_x \left[\text{AMP}^{\mathbf{g}}(x) \neq \text{AMP}^{\vec{0}}(x) \right] \leq T^{-1/2}.$$

Therefore, by Claim 1, we can fix a particular g such that AMP^g has mild hardness against $\mathcal{B}$ and $\Pr_x \left[\text{AMP}^g(x) \neq \text{AMP}^{\vec{0}}(x) \right] \leq T^{-1/2}$. It follows that $\text{AMP}^{\vec{0}}$ has hardness $T^{-1/2}$ against $\mathcal{B}$. It is easy to get $\text{AMP}^{\vec{0}}$ is in $\mathsf{SPACE}(n^\varepsilon)$ (respectively, $\mathsf{DTIME}(2^{n^\varepsilon})$). This proves Theorem 2. □

Under the same condition of Corollary 1, we can obtain the corollary similar to Corollary 1.

Corollary 2. *For every constant ε with $0 < \varepsilon < 1$, suppose there is an oracle machine* AMP *in* $\mathsf{SPACE}(n^\varepsilon)$ *(respectively,* $\mathsf{DTIME}(2^{n^\varepsilon})$*) which converts every $f : \{0,1\}^n \to \{0,1\}$ of worst-case hardness against* $\mathsf{SIZE}(2^{cn})$ *into a function* $\text{AMP}^f : \{0,1\}^m \to \{0,1\}$ *of mild hardness against* $\mathsf{SIZE}(2^{dm})$*. Then there exists a function $f' : \{0,1\}^m \to \{0,1\}$ in* $\mathsf{SPACE}(m^\varepsilon)$ *(respectively,* $\mathsf{DTIME}(2^{m^\varepsilon})$*) that has mild hardness against* $\mathsf{SIZE}(2^{dm})$*.*

5 Some Impossibility Results for Semi-Black-Box Hardness Amplification

From the observation of Theorem 1 and 2, we can generalize the semi-black-box hardness amplification problem as follows: for complexity classes $\mathcal{A}$,$\mathcal{B}$ and $\mathcal{C}$, is there an AMP constructed in $\mathcal{C}$ with the following property:

$$\text{for any } f \text{ which is } \delta\text{-hard against } \mathcal{B} \overset{\text{AMP}}{\Rightarrow} \text{AMP}^f \text{ has hardness } \delta' \text{ against } \mathcal{C}?$$

For $\mathcal{A} = \Sigma_i^{\mathsf{P}}$ (respectively $\mathsf{SPACE}(n^\varepsilon)$), $\mathcal{B} = \mathsf{SIZE}(2^{O(n)})$, $\mathcal{C} = \mathsf{SIZE}(2^{O(n)})$, $\delta = 2^{-n}$ and $\delta' = 1/\text{poly}(n)$, we already gave some results that the above construction is impossible to be realized unless there is a mildly hard function in class $\mathcal{C}$. These are conditional results. In fact, we can achieve some unconditional results. For example, under the same above setting except $\mathcal{A} = BPP$, such semi-black-box hardness amplification is impossible since we can obtain a contradiction that there is a function in BPP, indeed $f'(x,t,r) = \text{AMP}^G(r)|_{\rho_t}(x)$, which is hard against circuits of size $2^{\Omega(n)}$ but $\mathsf{BPP} \subset \mathsf{SIZE}(2^O(n))$ [4]. So we have the following result of Theorem 1.

Corollary 3. *Under the same setting of Corollary 1, such semi-black-box hardness amplification cannot be done in* $\mathsf{SIZE}(2^{cn})$ *for some constant c.*

Indeed, we rule out the possibility that the hardness amplification machine can simulate a hard function in some class. We would like to give a general result to illustrate these impossibility results. First of all, we need to estimate the size of class PSPACE.

Lemma 4. $|\mathsf{PSPACE}_n| = 2^{2^{o(n)}}$.

Proof. Each language L in PSPACE can be determined by nondeterministic polynomial-time Turing machine B and a polynomial p. That is, for every x of length n,

$$x \in \text{L} \Leftrightarrow (Q_1 y_1, |y_1| \leq p(n))(Q_2 y_2, |y_2| \leq p(n)) \cdots (Q_m y_m, |y_m| \leq p(n))(x, y_1, \cdots, y_m) \in B,$$

where each Q_i is either $\exists$ or $\forall$, and $m \leq p(n)$. For the proof, we refer the reader to Du and Ko [4]. The lemma follows that there are at most $2^{2^{o(n)}}$ polynomial-time Turing machines. □

We restate the question we mentioned in Introduction. **Given a class $\mathcal{C}$ and a value $\delta < 1$, how complex is it to build an operator G such that, for all $f \notin \mathcal{C}$, G^f is δ-hard against $\mathcal{C}$?** If $\delta = 2^{-n}$, i.e. worst-case hard, then we just trivially make G the identity function. Clearly the complexity is low. From the argument of Theorem 2, we can answer some impossibility results for this problem. Back to the condition of Theorem 1 and 2, we have $\mathcal{A}_n \leq 2^{2^{o(n)}}$ for every subclass $\mathcal{A} \subset \mathsf{PSPACE}$. Consider the class $\mathsf{DTIME}(n^k)$ where k is any constant. Since $\mathsf{DTIME}(n^k) \subset \mathsf{PSPACE}$, it follows from the proof of Theorem 2 that no Amp which can convert any function $f \notin \mathsf{DTIME}(n^k)$ into Amp^f slightly hard against $\mathsf{DTIME}(n^k)$ can be done in $\mathsf{DTIME}(n^k)$. If not, then we can get a function $\text{Amp}^{\vec{0}}$ which is clearly in $\mathsf{DTIME}(n^k)$ is slightly hard against $\mathsf{DTIME}(n^k)$. This is a contradiction. It is indeed interesting. One can easily obtain an operator which maps any function not in $\mathsf{DTIME}(n^k)$ into another one not in $\mathsf{DTIME}(n^k)$, that is, let the operator be identity function. However, if we require that such an operator must map them into those function which is slightly hard against $\mathsf{DTIME}(n^k)$, then this is impossible to be done in $\mathsf{DTIME}(n^k)$. There is a complexity gap between these two requirements. The same argument can apply to P, SUBEXP and $\mathsf{SPACE}(n^\varepsilon)$ where ε is a constant less than 1. Formally we have the following theorem.

Theorem 3. *Given any complexity class $\mathcal{A} \subset \mathsf{DTIME}(2^{n^\varepsilon})$ for some constant $\varepsilon < 1$, no operator G that maps any function f against $\mathcal{A}$ into another $2^{-n^\varepsilon/2}$-hard G^f against $\mathcal{A}$ can be done in $\mathcal{A}$.*

Above argument is based on the low deterministic computational complexity. Is there a similar result for non-deterministic complexity? The answer is unknown since our proof in Theorem 1 may lose some parameter. However, we can obtain a non-uniform results as follows.

Theorem 4. *No operator G that maps any function f against $\mathsf{NSIZE}(\text{poly})$ inot another mildly hard G^f against $\mathsf{NSIZE}(\text{poly})$ can be realized in $\mathsf{NSIZE}(\text{poly})$.*

Proof. First of all, observe that the cardinality of $\mathsf{NSIZE}(\text{poly})_n$ is at most $2^{2^{o(n)}}$. Then applying the same argument of Theorem 1, we can obtain the theorem. □

References

[1] H. Buhrman and L. Fortnow. One-sided versus two-sided randomness. In *Proceedings of the 16th Symposium on Theoretical Aspects of Computer Science*, vol. 1563 of Lecture Notes in Computer Science, pages 100-109. Springer, Berlin, 1999.

[2] László Babai, Lance Fortnow, Noam Nisan, Avi Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3(4), pages 307–318, 1993.

[3] 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, October 2003.

[4] Ding-Zhu Du and Ker-I Ko. *Theory of Computational Complexity.* John Wiley & Sons, Inc. New York, 2000.

[5] 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.

[6] Oded Goldreich, Noam Nisan, and Avi Wigderson. On Yao's XOR lemma. Technical Report TR95–050, Electronic Colloquium on Computational Complexity, 1995.

[7] Johan Håstad. *Computational limitations for small depth circuits*. PhD thesis, MIT Press, 1986.

[8] 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.

[9] 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.

[10] 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.

[11] Adam Klivans and Dieter van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 659-667. ACM, New York, 1999.

[12] Chi-Jen Lu, Shi-Chun Tsai, and Hsin-Lung Wu. On the Complexity of Hardness Amplification. Submitted.

[13] Noam Nisan and Avi Wigderson. Hardness vs Randomness. *Journal of Computing System Science*,49(2):149-167, October 1994.

[14] Ryan O'Donnell. Hardness amplification within NP. In *Proceedings of the 34th ACM Symposium on Theory of Computing*, pages 751-760, 2002.

[15] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62(2), pp. 236-266, March 2001.

[16] Emanuele Viola. The Complexity of Constructing Pseudorandom Generators from Hard Functions. To appear in *Computational Complexity*.

[17] Emanuele Viola. On parallel pseudorandom generators. Technical Report TR04-074, Electronic Colloquium on Computational Complexity, 2004.

[18] 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.