摘要

在本計劃中，我們研究如何從某些隨機來源萃取出真正的隨機元(random bits)。針對函數 f(x)與分布χ，某來源具有計算觀點最小熵 k 的意思是說，當我們依據分佈χ來取樣出 x 時，任何大小為 2^k 的電路可以猜對 f(x)的機率會低於($1/2^k$)。我們首先證明：如果不使用種子，則無法從單獨計算最小熵的來源萃取出隨機元。但如果我們可以使用具有弱隨機性的種子，此種子不必具有完美隨機性，但包含統計觀點的最小熵，則萃取器存在。甚至，當我們使用的種子雖不具有統計觀點最小熵，但具有某些計算觀點最小熵時，對應的萃取器仍然存在。由於這個結果，我們可以將傳統的萃取器推廣為考慮計算觀點最小熵的萃取器。在建造這一種萃取器時，我們也發現可以將問題轉化為計算學習理論的問題，也就是考慮在任意機率分布且具有對手雜訊的情況下，如何學習一個線性函數的問題。我們也針對這個學習問題提出了一個學習演算法。

接著考慮計算觀點的獨立符號隨機源(computational independent-symbol sources)；所謂獨立符號隨機源是在$(\{0,1\}^d)^n$ 上的分布 $X=\left(X_1,\cdots,X_n\right)$，其中$X_1,\cdots,X_n$是彼此獨立的，且 X 的 min-entropy 是 k。計算觀點的獨立符號隨機源與獨立符號隨機源類似，都包含 n 個獨立的符號 $(f_1(X_1)|X_1)$，…，$(f_n(X_n)|X_n)$，其中每個 $f_i(X_i)$都是分布在$\{0,1\}^d$上，使得當輸入 x_i是依照分布 X_i所產生時，任何一個大小為 s 的電路最多只有 2^{-k_i}的機率可以猜對 $f_i(X_i)$ 對某個數 $k_i\leq d$，且$k_1+\cdots+k_n=k$。我們將 Impagliazzo 的核心集引理[Imp95]加以推廣，並用來證明：[LLT06]中針對獨立符號隨機源的萃取器其實也可作用於計算的獨立符號隨機源，使得所萃取出的隨機性就小電路而言是非常隨機的。事實上，此針對計算的獨立符號隨機源的萃取器之結果隱含延伸的 XOR 引理。最後，我們亦提供一個在核心集的黑箱子建造法中，二元的核心集大小的上限。

關鍵詞：隨機元萃取器，計算觀點的最小熵，學習線性函數，獨立符號來源，計算觀點的獨立符號來源，延伸的核心集引理，延伸的 XOR 引理，黑箱子建造法，核心集大小。

Abstract

We study how to extract randomness from some sources. For some function f(x) and some distribution χ, we say that such a source has computational min-entropy k if any circuit of size 2^k can only predict f(x) correctly with probability at most $(1/2^k)$ given input x sampled from χ. We first show that it is impossible to have a seedless extractor to extract from one single source containing computational entropy. Then we show that it becomes possible if we are allowed a seed which is weakly random (instead of perfectly random) but contains some statistical min-entropy, or even a seed which is not random at all but contains some computational min-entropy. This can be seen as a step toward extending the study of multisource extractors from the traditional, statistical setting to a computational setting. We reduce the task of constructing such extractors to a problem in computational learning theory: learning linear functions under arbitrary distribution with adversarial noise, and we provide a learning algorithm for this problem. In fact, this problem is a well-recognized one in computational learning theory and variants of this problem have been studied intensively before. Thus, in addition to its application to extractors, our learning algorithm also has independent interest of its own.

Then, we consider computational independent-symbol sources. Just as an independent-symbol source, which is a distribution $X=(X_1,\ldots,X_n)$ over the set $(\{0,1\}^d)^n$ where these n symbols $X_1,\ldots,X_n$ are independent, and the whole min-entropy of X is k, a computational independent-symbol source consists of n mutually independent parts, $(f_1(X_1)|X_1)$ ， … ， $(f_n(X_n)|X_n)$, each $f_i(X_i)$ of length d such that for each i if given input x_i sampled from X_i, any circuit of size s can only predict $f_i(X_i)$ with probability at most 2^{-ki} for some $k_i \leq d$, and the sum of k_i's is k. We generalize Impagliazzo's well-known hardcore set lemma [Imp95] to show that the extractor for independent-symbol sources in [LLT06] still works for computational independent-symbol sources. In fact, the result of computational extractors for computational independent-symbol sources implies a generalization of the well-known XOR lemma. Finally, we provide a size upper bound on a binary hardcore set in any black-box construction of hardcore set.

Keywords: randomness extractors, computational min-entropy, learning linear functions, independent-symbol sources, computational independent-symbol sources, generalized hardcore set lemma, generalized XOR lemma, blak-box construction, size of hardcore set.

目錄

一、前言:背景與文獻探討

在資訊科學的許多領域裡，隨機性(Randomness)既是非常有用的工具，也是非常重要的計算資源。例如，就許多難以處理的問題而言[MR95, Gol98]，隨機演算法(probabilistic algorithm)往往是效率較高的解決辦法，這是因為相較於決定性演算法(deterministic algorithm)，當加入隨機性之後，使得時間以及空間的複雜度(time and space complexity)可能會更有效率。又例如，在密碼學的應用裡，我們常藉由隨機的特性來隱藏秘密，以避免在傳輸的過程中被第三者竊聽而洩露資訊。因此，是否能取得真正的隨機元(truly random bits)對於相關應用的影響非常深遠，但遺憾的是，真正的隨機元並非垂手可得。類似於我們製造亂數表的情況，在真實世界中，我們幾乎只能退而求其次地使用一個決定性的函數(deterministic function)從一些可取樣的弱隨機源(weak random sources)中萃取出真正的隨機元，其中，所謂的弱隨機源只能保證任一個字串出現的機率都不會非常高。我們說一個弱隨機源的「統計觀點最小熵(statistical min-entropy)」為 k 則表示每個字串出現的機率都不會超過 2^{-k} ，直覺上我們會認為這樣的弱來源「包含」k 個隨機元。

- 萃取器 (Extractors)

1988 年時，Chor 等人證明不存在任何決定性的函數(deterministic function)可以從單一個 statistical min-entropy < n 的弱隨機源中萃取出一個隨機元[CG88]。於是，研究者們試圖在單一個弱隨機源外，再加上一個很短(相較於弱隨機源的長度)的真正隨機元，以萃取出隨機元。我們稱這個很短的真正隨機元為種子(seed)，而這種萃取器為種子萃取器(seeded extractor)。在近幾十年來，國內外的理論學家們都致力於建造出使用最少種子，而能從各種弱隨機源中萃取出非常靠近真正隨機元的種子萃取器，如[ILL89、Zuc97、RSW00、RVW00、TSZS01、TSUZ01]等。最後終於造出了幾近完美的種子萃取器[LRVW03]。

然而在使用種子萃取器時，我們仍然需要種子，其為一些真正的隨機元。在一些應用中，我們可以解決此問題(比如在解隨機 BPP 問題中列舉所有可能的種子)，而在其他應用中，則又回歸到最初的問題: 如何獲得一些真正的隨機元呢? 這個爭議讓學者們轉而建造不需種子輔助的萃取器，即決定性萃取器(deterministic extractors or seedless extractors)。

當弱隨機源擁有某些特殊的性質時，我們確實可以從一個隨機源中萃取隨機元。一個(n, k)-固定某些位元的來源 (A (n,k)-bit fixing source) 為一個在$\{0,1\}^n$的分布 $X=(X_1,\cdots,X_n)$ ，其中 n-k 個位元是固定的，而其餘的 k 個位元則是均等的(uniform)且彼此獨立的(independent)。Kamp 等人[KZ03]提出從一個固定某些位元的來源中萃取出一些隨機元的方法，而 Gabizon 等人則在 2004 年提出改進的方法使之萃取出幾乎所有隨機源所含的隨機性[GRS04]。

此外，我們亦可從兩個甚至多個弱隨機源中不使用種子而直接萃取出很靠近均等分布(uniform distribution)的隨機元。針對兩個弱隨機源，研究者們致力於放寬對此兩個隨機源的 statistical min-entropy 的限制[CG88、DO03、DEOR04、LLTT05、Raz05]，最後，Bourgain 造出只要兩個 statistical min-entropy 均略小於 n/2 的弱隨機源即可萃取出Ω(n)個隨機元的萃取器[Bou05]。而另一方面，理論學家們也設法從越少個且含有越少 statistical min-entropy 的隨機源中萃取出隨機元[BIW04、BKS+05、Raz05]。

事實上，固定某些位元的來源以及多個隨機源的來源可以下列觀點看成是兩種極端。這兩種來源均包含多個部分而且這些部份彼此都是獨立的。固定某些位元的來源可看成是包含許多個部份，且每個部分只有單一個隨機或者為固定的位元。而多個隨機源的來源則可看成包含相對少數個部份，可是每個部份為多個位元且含足夠數量的隨機元。本計劃考慮一個介於其中的來源，稱為獨立符號的來源(independent-symbol source)，寫成 (n,D,k)-來源，其為一個在$[D]^n$（[D]={1,2,...,N}）上的分布 $X=\left(X_1,\cdots,X_n\right)$，其中 $X_1,\cdots,X_n$ 是彼此獨立的，且 X 的 min-entropy 是 k。 我們不難看出一個(n, k)-固定某些位元的來源其實就是一個(n,2,k)-來源。換句話說，固定某些位元的來源只是我們所考慮的獨立符號來源的一個特例。而對較小的 n 及較大的 D，此種來源即可涵蓋多個隨機源的來源。Kamp 等人[KRVZ06]及 Lee 等人[LLT06]均提出從一個獨立符號來源中萃取出幾乎所有隨機性的萃取器。

• 計算觀點的最小熵（Computational min-entropy）

在過去的文獻中，大多是考慮那些在統計上仍有隨機性(亦即 statistical min-entropy 不為 0)的隨機源。在本計劃中，我們將換個角度，考慮那些在統計上並不具有任何的隨機性，但在那些計算複雜度有所限制的觀察者而言仍有些許隨機性的隨機源。換句話說，我們將從傳統的統計觀點，轉換為計算觀點。由於這些隨機源在統計上本就不具任何的隨機元，因此我們也必須將萃取器的輸出限制稍作修改。相對於在原本的定義中，我們要求萃取器的輸出要在統計上(亦即不限制計算複雜度的觀察者眼中)極靠近真正的隨機元，現在我們只要求這些輸出在那些計算複雜度有限的觀察者眼中看似隨機即可。值得注意的是，這些輸出有可能在統計上是距離真正隨機元極遠的，甚或是根本不具任何隨機性的。事實上，在密碼學中，尤其是有關利用 one-way functions 建造 pseudo-random generators 的研究中，已經有一些隱含的結果存在[Yao82, GL89, HILL99]。

何謂「看似隨機」呢? 事實上，已經有一個非常好的「看似幾乎隨機」的定義[Yao82]，但關於一個隨機源「看來有些隨機」的定義以及衡量此隨機源中隨機性之測量值的定義仍不清楚。目前有一些看似合理的定義，但目前已有些證據可以證明這些定義之間有不一致之處[BSW03, HLR07]。

- 學習parity函數的演算法
  (Algorithms for learning parity functions)

在計算學習理論(computational learning theory)中，學習parity函數是一個最基本的問題。令$x=(x_1,\cdots,x_n)\in\{0,1\}^n$，則一個parity函數是一個對某個$T\subseteq\{1,2,\cdots,n\}$的函數$f(x)=\underset{i\in T}{\oplus}x_i$。在此學習模型(learning model)中，我們有一個想學習的函數$f:\{0,1\}^n\rightarrow\{0,1\}$，以及一個在$\{0,1\}^n$上的分佈W，從中我們可以取樣w並獲得訓練樣本(training example)(w,q(w))，其中q 為某個函數。如果我們得到的樣本都是正確的，亦即$q(w)=f(w),\forall w$時，則我們可以對W作多次取樣後利用所獲得的訓練樣本作高斯消去法(Gaussian elimination)則可獲得我們想要學習的函數f。但是當我們所獲得的樣本可能被雜訊影響而變得不一定正確時，如何從此種模型中學習函數f就變成一個具有挑戰性的問題了。一般廣被討論的是以下的兩種雜訊模型: 隨機雜訊(random noise)跟對手雜訊(adversarial noise)。在隨機雜訊的模型中，針對每一個訓練樣本都是獨立的並且最多有η的機率使得q(w)≠f(w)。而在對手雜訊的模型中則是當w是依照分布W所取樣時，最多有η比例的w使得q(w)≠f(w)。不難看出，隨機雜訊模型要比對手雜訊模型更容易分析。另外，對手雜訊的模型亦可看成是不可知的模型(agnostic model)[KSS94]，而此種模型確實符合我們在現實生活中所可能遭遇的現象。

針對在隨機雜訊模型中學習一個有n個變數的parity函數且$\eta\le 1/2-\Omega(1)$的問題，目前最好的結果為Blum, Kalai,和Wasserman[BKW03]於2003年提出的一個需要$2^{O\left(\frac{n}{\log n}\right)}$個樣本且時間複雜度為$2^{O\left(\frac{n}{\log n}\right)}$的演算法，更值得一提的是此演算法對任何分布W都是有效的。而在對手雜訊模型中，此問題跟條列解碼(list decode) Hadamard codes有關。然而目前已知的結果都只能作用在W 為一個均等分布的情況[GL89, FGKP06]。最近，Feldman等人[FGKP06]証明當W 為一個均等分布時，則可以利用針對隨機雜訊模型的演算法來造出針對對手雜訊模型的演算法。因此利用上述[BKW03]的演算法，則可以得到一個需要$2^{O\left(\frac{n}{\log n}\right)}$個樣本且時間複雜度為$2^{O\left(\frac{n}{\log n}\right)}$的演算法，但其中W為一個均等分布。

二、研究目的

本計劃主要目的是藉由探討 Cayley 圖以及循環矩陣(circulant matrix)的特性，進而從只擁有計算觀點最小熵 (computational min-entropy) 的隨機來源(source)中，萃取出真正的隨機元 (random bits)。

- 探討循環矩陣(circulant matrices)與獨立符號來源萃取器的特性與關係

在[LLT06] 中，我們知道針對獨立符號來源的萃取器即是利用在一個圖形上作隨機散步 (random walk) 的方法來萃取隨機元。我們發現在此隨機散步中每一步所對應的矩陣為一個特殊的矩陣，稱為循環矩陣(circulant matrix)[Dav79]，其為一個形式為

$$P=\begin{pmatrix} p_1 & p_2 & \cdots & p_n \\ p_n & p_1 & \cdots & p_{n-1} \\ \vdots & \vdots & & \vdots \\ p_2 & p_3 & \cdots & p_1 \end{pmatrix}$$

的矩陣，其中 $\sum_i p_i = 1$。我們希望能利用已知的關於循環矩陣的特性來簡化針對獨立符號來源萃取器的分析，並希望能獲得一個更好的結果。

- 建造出針對 computational min-entropy 的隨機源的萃取器:

由於之前針對只擁有 computational min-entropy 的隨機源的萃取器都是種子萃取器，我們希望在計算的角度更進一步地研究是否存在決定性萃取器。首先本計畫會先探討在計算架構中是否會如同在統計架構中，無法從單一個隨機源中萃取隨機元。接著我們也將探討如果將原本的種子替換成另一個只擁有些許 statistical min-entropy 的隨機源時，是否亦可從只擁有 computational min-entropy 的隨機源中萃取隨機性。更進一步地，我們也會試著從兩個或多個只擁有 computational min-entropy 的隨機源中萃取隨機性。

在本計劃中，我們考慮採用[HLR07]中的定義來計算一個隨機源在計算上的隨機量。我們在這裡所考慮的來源是一個條件的型式(f(X)|X)，其中 X 是一個在 $\{0,1\}^{n_1}$ 的分布，而 $f:\{0,1\}^{n_1}\to\{0,1\}^n$ 為某個函數。假設當輸入是依照分布 X 所產生時，任何大小為 2^k 的電路最多只有 2^{-k} 的機率可以猜對 f 的函數值時，則我們說這種條件分布(f(X)|X)擁有 computational min-entropy k。當 f 是一個決定性的函數時，亦即 f(x)的值完全由 x 所決定，則當給定 x 之後，f(x)並不具有任何在統計上的隨機性。然而，依照我們的定義，只要函數 f 是很難計算的，則(f(X)|X)還是可以擁有很大的 computational min-entropy，使得我們可以從中萃取隨機元。更準確地說，從一個分佈 f(X)中，我們希望能萃取

出一些隨機性，使得即使在給定X時，這些隨機性在某些大小的電路看來幾乎是非常隨機的。本計劃考慮放寬對隨機源的限制，試著從一個擁有statistical min-entropy以及另一個只擁有computational min-entropy的兩個隨機源中萃取出隨機元。最後希望能從兩個都只擁有computational min-entropy 的隨機源中萃取出隨機元。

- ## 考慮利用任何分布來學習一個線性函數

  在之前的研究結果中，要在對手雜訊模型中學習一個parity函數都需要分佈W為一個均等分布。因此，在學習理論中，如何在對手雜訊模型以及W為任意一個分布時學習一個parity函數是一個令研究者們很感興趣的問題。在本計劃中，我們希望能解答這個問題，甚至更進一步地，我們希望即使在W為任意一個分布時，仍能在對手雜訊模型中學習一個線性函數。

  事實上，由之前的研究我們可以得知在此種模型下的學習演算法跟從一個擁有statistical min-entropy的隨機源和另一個獨立但只擁有computational min-entropy的隨機源中萃取隨機性有很大的相關性。

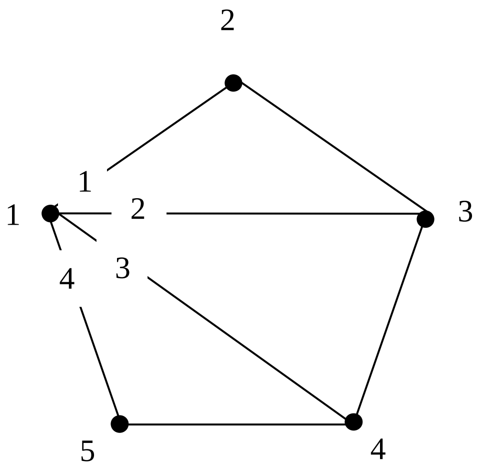
- ## 從只擁有計算觀點獨立符號的隨機來源萃取隨機元

在統計的觀點中，我們已經知道可以從單一個獨立符號來源中萃取隨機性。在本計劃中，我們考慮計算的獨立符號來源(computational independent-symbol source)。如同獨立符號來源，每個計算的(n, D, k, s)-來源(computational (n, D, k, s)-source)$\left(V\middle|X\right)=\left(V_1\middle|X_1\right)\circ\cdots\circ\left(V_n\middle|X_n\right)$，包含n個彼此獨立的部分$\left(V_1\middle|X_1\right),\cdots,\left(V_n\middle|X_n\right)$，其中每個$V_i$都是分布在[D]={1, 2, ..., D}中，而X_i則是分布在$\{0,1\}^{\ell_i}$上，且對每一個$i\in\{1,2,\cdots,n\}$，和每一個大小為s的電路C，$\Pr\left[C\left(X_i\right)=V_i\right]\le 2^{-k_i}$對某個值$k_i\le\log D$，並滿足$\sum_{i=1}^{n}k_i=k$。我們將試著證明在統計觀點中針對獨立符號來源的萃取器亦可從此種計算的獨立符號來源中萃取出隨機元。

三、研究方法

• 利用循環矩陣的特性來簡化針對獨立符號來源萃取器的證明:

在[LLT06]中我們觀察到每個 $i \in \{0,1\}^d \subset \mathbb{Z}_M$ 都會對應到一個在 $\mathbb{Z}_M$ 上排列(permutation)。例如以右圖為例，M=5，則 1 會對應到排列 $\begin{pmatrix}1 & 2 & 3 & 4 & 5\\ 2 & 3 & 4 & 5 & 1\end{pmatrix}$，而 2 會對應到排列 $\begin{pmatrix}1 & 2 & 3 & 4 & 5\\ 3 & 4 & 5 & 1 & 2\end{pmatrix}$。因此，可以對應出一個圖，其中每個點對應一個在 $\mathbb{Z}_M$ 中的元素，並且滿足如果存在一個排列使得 a 映到 b 則存在邊(a,b)。而[LLT06]的萃取器即是在此種圖上做隨機散步(只是這裡所用的並不是真正的隨機元)，並證明只要獨立符號來源的 statistical min-entropy k 夠大時，就可以很快地收斂到均等分布。在過去許多研究中已經證明在 expander 上做真正的隨機散步可以很快的收斂到均等分布[HLW06]，他們的証明方法大多是考慮此 expander 的鄰接矩陣(adjacency matrix)的第二大(取絕對值後)的特徵值(eigenvalues)。然而當 expander 是有向圖時，其所對應的特徵值不一定會是實數，因此上述的特徵值方法就不適用了。事實上，Wilmer 提出了一些其他在無法使用特徵值法時，可以證明收斂速度的方法[Wil99]。在[LLT06]中，其隨機散步所對應的矩陣之特徵值並不保證一定是實數，因此亦無法用上述特徵值的方法証明，而另採其他的方法。我們希望能將[LLT06]中的証明方法用來證明某些 expander 的收斂性，比如說 Cayley 圖(Cayley graph)，而提供一種新的不同於[Wil99]的証明收斂速度的方法。



經過編號的ℤ5

• 建造出針對擁有 computational min-entropy 的隨機源的萃取器:

我們首先會考慮是否可以從單一個只擁有 computational min-entropy 的隨機源中萃取出隨機元。接下來，由於已經知道如何從兩個擁有 statistical min-entropy 的隨機源的萃取出隨機元，接著，考慮放寬對隨機源的限制，試著從一個擁有 statistical min-entropy 以及另一個只擁有 computational min-entropy 的兩個隨機源中萃取出隨機元。跟著我們進一

步放寬對隨機源的限制，希望能從兩個都只擁有 computational min-entropy 的隨機源中萃取出隨機元。我們利用隨機方法證明無法從單一個長度為 n，computational min-entropy 為 n-2 的隨機源中萃取出隨機元，即使我們僅希望萃取出一個隨機元。

考慮[LLTT05]中針對兩個弱隨機源的萃取器 $Ext:\{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$，其中 m|n。在[LLTT05]中，我們把任何一個 $v \in \{0,1\}^n$，看成一個 ℓ-dimensional vector $(v_1,\ldots,v_l)$，其中 ℓ=n/m，而定義 $Ext(x,y) \equiv \langle x,y \rangle_m = \sum_i x_i \cdot y_i (\bmod m)$。我們希望能證明對任何一個只擁有 computational min-entropy 的隨機源(V|X)與另一個擁有 statistical min-entropy 的隨機源 W，任何的小電路都無法分辨 $(X,W,\langle V,W \rangle_m)$ 以及 (X,W,U)。我們利用反證法，假設存在一個小電路可以分辨 $(X,W,\langle V,W \rangle_m)$ 以及 (X,W,U)。則可以推得會存在一台推測器（predictor）Q 使得對許多的 (x,v)，給定 x,w 之後，Q 有足夠的機率可以猜對 $\langle v,w \rangle_m$ 的值。給定一對此種 (x,v)，我們希望能有足夠的機率可以從 x 猜對 v。而這個問題可以轉化成由一些有誤差的訓練樣本，即(w,q(w))其中 q(w)=Q(x,w)，中學習一個線性函數 $\langle v,\cdot \rangle_m$ 的問題。對此，我們建造了一個可以輸出所有可能的線性函數之學習演算法。最後我們從這些可能的線性函數中隨機選一個當我們的輸出，我們可以證明此輸出所包含的函數並不會太多，因此有足夠的機率會猜中正確的 v，而這就會跟(V|X)有足夠 computational min-entropy 的假設相矛盾。由以上的方法，我們可以證明[LLTT05]中的萃取器可以從一個擁有 computational min-entropy 為 $k_1 = n - k + O(k/\log k)$ 的隨機源與另一個擁有 statistical min-entropy 為 k 的隨機源中萃取出隨機元。

最後我們考慮由兩個只擁有 computational min-entropy 的隨機源萃取出隨機元。我們發現如果一個隨機源（W|Y）擁有 computational min-entropy k，則 W 所擁有的 statistical min-entropy 至少為 k。經由以上的觀察，我們就可以利用上述的証明方式證明[LLTT05]的萃取器亦可從一個擁有 computational min-entropy 為 $k_1 = n - k + O(k/\log k)$ 的隨機源與另一個擁有 computational min-entropy 為 k 的隨機源中萃取出隨機元。

• 利用任何分布來學習一個線性函數

在建造這一種萃取器時，我們也發現可以將問題轉化為計算學習理論的問題，也就是考慮在任意機率分布且具有對手雜訊的情況下，如何學習一個線性函數的問題。我們也針對這個學習問題提出了一個學習演算法。具體而言，在之前的研究結果中，要在對手雜訊模型中學習一個 parity 函數都需要分佈 W 為一個均等分布。因此，在學習理論中，如何在對手雜訊模型以及 W 為任意一個分布時學習一個 parity 函數是一個令研究者們很感興趣的問題。在本計劃中，我們希望能解答這個問題，甚至更進一步地，我們希望即使在

W 為任意一個分布時，仍能在對手雜訊模型中學習一個線性函數。

事實上，由之前的研究我們可以得知在此種模型下的學習演算法跟從一個擁有 statistical min-entropy 的隨機源和另一個獨立但只擁有 computational min-entropy 的隨機源中萃取隨機性有很大的相關性。

• 將核心集引理 (hardcore set lemma)加以推廣

Impagliazzo 的核心集引理 [Imp95] 告訴我們，如果一個布林函數 $f:\{0,1\}^{\ell}\rightarrow\{0,1\}$，其滿足對每個大小為 s 的電路 h，$\Pr_{x\in\{0,1\}^{\ell}}\left[h(x)\neq f(x)\right]>\delta$，則對每個ε>0，存在一個大小至少為$\delta 2^{\ell}$的 hardcore set $H\subseteq\{0,1\}^{\ell}$ 使得對任何大小為 $\Omega\left(\varepsilon^2\delta^2 s\right)$ 的電路 C，$\Pr_{x\in H}\left[C(x)\neq f(x)\right]>\frac{1}{2}-\varepsilon$。我們延伸此結果考慮一個函數 $f:\{0,1\}^{\ell}\rightarrow[D]$，其滿足對每個大小為 s 的電路 h，$\Pr_{x\in\{0,1\}^{\ell}}\left[h(x)\neq f(x)\right]>\delta$，並證明此時依然存在一群總大小至少為$(\delta/2)2^{\ell}$的二元核心集(binary hardcore sets) $T_1,\cdots,T_r$，亦即對每一個$i\in[r]$，存在一個大小為 2 的子集合$I_i\subseteq[D]$使得 $T_i\subseteq f^{-1}(I_i)$，且對每個大小為 $\Omega\left(\varepsilon^2\delta^2 s/D^6\right)$ 的電路 C，$\Pr_{x\in T_i}\left[C(x)\neq f(x)\right]>\frac{1}{2}-\varepsilon$。

我們發現上述針對計算獨立符號來源的萃取器的結果其實可以推得延伸的 XOR 引理。Yao 的 XOR 引理 [Yao82] 告訴我們，如果一個布林函數 $f:\{0,1\}^{\ell}\rightarrow\{0,1\}$，其滿足對每個大小為 s 的電路 h，$\Pr_{x\in\{0,1\}^{\ell}}\left[h(x)\neq f(x)\right]>\delta$，則對$\varepsilon>2(1-\delta)^t$，任何大小為$\Omega\left(\varepsilon^2 s/\ell\right)$的電路 C，

$$\Pr_{x_1,\cdots,x_t\in\{0,1\}^{\ell}}\left[C(x_1,\cdots,x_t)=\sum_i f(x_i)\right]<\frac{1}{2}+\varepsilon \text{。}$$

我們延伸此結果考慮 n 個函數 $f_1,\cdots,f_n$，其中每個函數 $f_i:\{0,1\}^{\ell_i}\rightarrow[D]$，其滿足對每個大小為 s 的電路 h，$\Pr_{x_i\in\{0,1\}^{\ell_i}}\left[h(x_i)\neq f_i(x_i)\right]>\delta_i$，並證明如果 $\delta=\sum_{i=1}^{n}\delta_i\geq\Omega\left(M^2\log D\right)$，則對每個大小為$\Omega\left(s\left(\log n/nD\delta\right)^2\right)$的電路 C，

$$\Pr_{x_1\in\{0,1\}^{\ell_1},\cdots,x_n\in\{0,1\}^{\ell_n}}\left[C(x_1,\cdots,x_n)=\sum_{i=1}^{n} f_i(x_i)\right]\geq\frac{1}{M}+\frac{M^2\log n}{\delta} \text{。}$$

- ## 建造針對計算獨立符號來源的萃取器

我們首先利用[STV01]的方法以及延伸的核心集引理證明對每一個來源$(V_i|X_i)$都存在一個來源Y_i，使得任何小電路都無法分辨(X_i,V_i)以及(X_i,Y_i)，且來源$(Y_i|X_i)$的 statistical min-entropy 與$(V_i|X_i)$的 computational min-entropy 有關。所以給定一個計算的(n,D,k,s)-來源$(V|X)=(V_1|X_1)\circ\cdots\circ(V_n|X_n)$，其中$k\ge\Omega\left(2^{2m}\log^2 D\right)$，對任何大小為$\Omega\left(s\left(\log n/nkD\right)^2\right)$的電路 C，

$$\left|\Pr\left[C\left(X_1,\cdots,X_n,V_1,\cdots,V_n\right)=1\right]-\Pr\left[C\left(X_1,\cdots,X_n,Y_1,\cdots,Y_n\right)=1\right]\right|\le\frac{2^{2m}\log n}{k}\text{。}$$

進一步地，我們可以證明對任何大小為$\Omega\left(s\left(\log n/nkD\right)^2\right)$的電路 C，

$$\left|\Pr\left[C\left(X_1,\cdots,X_n,\sum_{i=1}^{n}V_i\right)=1\right]-\Pr\left[C\left(X_1,\cdots,X_n,\sum_{i=1}^{n}Y_i\right)=1\right]\right|\le\frac{2^{2m}\log n}{k}\text{。}$$

另一方面，有$1-e^{-\Omega(k/\log D)}$的機率，即使在知道$X_1,\cdots,X_n$後，來源$Y=Y_1\circ\cdots\circ Y_n$的 min-entropy 至少也有$\Omega(k/\log D)$。再加上$(V_1|X_1),\cdots,(V_n|X_n)$是彼此獨立的，我們可推得$Y_1,\cdots,Y_n$也是彼此獨立的。因此，我們可以利用[LLT06]中，

$$Ext\left(V_1,\cdots,V_n\right)=\sum\nolimits_{i=1}^{n}V_i$$

是針對獨立符號來源的萃取器的結果，證明

$$\Delta\left(\left(X_1,\cdots,X_n,\sum\nolimits_{i=1}^{n}Y_i\right),\left(X_1,\cdots,X_n,U_m\right)\right)\le e^{-\Omega\left(k/2^{2m}\log D\right)}\text{。}$$

結合以上的結果，我們推得對任何大小為$\Omega\left(s\left(\log n/nkD\right)^2\right)$的電路 C，

$$\left|\Pr\left[C\left(X_1,\cdots,X_n,\sum\nolimits_{i=1}^{n}V_i\right)=1\right]-\Pr\left[C\left(X_1,\cdots,X_n,U_m\right)=1\right]\right|\le O\left(\frac{2^{2m}\log n}{k}\right),$$

亦即$Ext\left(V_1,\cdots,V_n\right)=\sum\nolimits_{i=1}^{n}V_i$為針對計算獨立符號來源的萃取器。

- ## 黑箱子建造法中二元核心集大小的上限

在延伸的核心集引理中，我們證明存在若干個二元核心集，其大小總和至少為$(\delta/2)2^{\ell}$。大家可能會猜想是否存在單一個夠大的核心集，比如說大小為$(\delta/D)2^{\ell}$。我們最後證明一個對任何黑箱子建造法的二元核心集大小的上限。假如一個演算法 Dec$^{(\cdot)}$滿足對任意的函數$f:\{0,1\}^{\ell}\to[D]$，以及任何的函數集合$G=\left\{g_I\,\middle|\,I\subseteq[D],|I|=2\right\}$，其中任何在$G$裡面的函數$g_I$，以及任何大小

為 s 的子集$H \subseteq f^{-1}(I)$，$\Pr_{x \in H}\left[g_I(x) \neq f(x)\right] \leq (1-\varepsilon)/2$，可推得

$$\Pr_{x \in \{0,1\}^{\ell}}\left[Dec^{G}(x) \neq f(x)\right] \leq \delta ，$$

則我們說演算法 Dec$^{(\cdot)}$是一個核心集的黑箱子 (δ, ε, D)-建造法 (black-box (δ, ε, D)-construction of a hardcore set)，而其中 s 為黑箱子建造法的大小複雜度(size complexity)。

我們最後一個結果即利用機率方法證明當δ ≥ Ω(1/D^{c}) 對某個常數 c ，D ≥ 4，以及ε < 1/5 時，任何核心集的黑箱子(δ, ε, D)-建造法的大小複雜度為$O\left(\delta 2^{\ell}/D^{2}\right)$。

四、結果與討論

• 獨立符號源萃取器的建造與簡化版本的證明
（此部份之結果發表於畢業論文 [Lee 10]）

由於在[LLT06]中的隨機散步其每一步所對應的矩陣即為一個循環矩陣(circulant matrix)，且此種矩陣的特徵向量（eigenvectors）是互相垂直的，我們即可利用此矩陣的特徵值（eigenvalues）來幫助我們證明每走一步後所得到的分布都會比原先的分布更接近均等分布，且得到一個比[LLT06]更好的結果。更精確地說，我們考慮在 $\mathbb{Z}_M$ 上做隨機散步，目前的分布為 $P=(P_1,\ldots, P_M)$, $U=(1/M,\ldots,1/M)$為均等分布，而利用一個 min-entropy 為 k 的來源 X_i 走一步隨機散步之後所得到的分布為 P’。在[LLT06]中，我們可以證明

$$\|P'-U\|_2^2 \le \|P-U\|_2^2 \cdot \left(1-\frac{k}{4M^2\log D}\right) \le \|P-U\|_2^2 \cdot e^{-\frac{k}{4M^2\log D}}.$$

然而，利用循環矩陣的特性，我們可以證明

$$\|P'-U\|_2^2 \le \|P-U\|_2^2 \cdot e^{-2\frac{2^{2k}-1}{M^2-1}}.$$

我們針對獨立符號源建造與證明萃取器時，雖然結果與 [KRVZ06] 類似，但所採取的方法卻相當不同。我們也發現傳統上為了增加最小熵都會同時使用加法與乘法，但這個證明卻顯示其實只要使用加法即可增加最小熵，雖然缺點是最小熵增加的速度相對之下比較慢。

• 關於計算觀點 min-entropy 的不可能結果
（此部份之結果發表於 [LLT11a]，原文請參看附錄）

[定理] 對於任何自然數 n 與 n_1 ，且對於任何函數 $EXT:\{0,1\}^n \to \{0,1\}$ ，只要 $n_1 > 3n$ 成立，則必存在決定性的函數 $f:\{0,1\}^{n_1} \to \{0,1\}^n$ ，使得對於所有 $\chi = U_{n_1}$ ，有 $H_c(f(\chi)|\chi) = n-2$ ，且 $EXT(f(x))$ 的值在任何 x 都會一樣。因此，我們可以很容易區別隨機元與 $EXT(f(x))$ 。

這個結果顯示：在沒有種子的前提下，即使 computational min-entropy 高達 $n-2$ ，我們也無法從弱隨機源萃取出真正的隨機元。另外，這個結果可以和已知的統計式萃取器[CG88]相互比較。

- ## 混合型與計算觀點型式的萃取器

(此部份之結果發表於 [LLT11a]，原文請參看附錄)

我們先定義一個函數$EXT:\{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$如下：

$$EXT(v,w) = \langle v,w \rangle$$

則針對這種內積型式的函數，有下列結果：

[定理] 對於所有任何符合下列條件的參數(k,m,ε,s)：$k \geq \Omega(\log^2 n)$，$m|n$且$m \leq O(\sqrt{k/\log k})$，$\varepsilon \geq 2^{-O(\sqrt{k/\log k})}$，以及$s \geq 2^{n-k+O(k/\log k)}$，存在適合的$k_1$(其估計值為$k_1 = n-k+O(k/\log k)$)，使得上述所定義的$EXT:\{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$同時兼具「$(k_1,k,\varepsilon,s)$-混合型萃取器」，以及「$(k_1,k,\varepsilon,s)$-計算觀點型式萃取器」的特徵。

[定理] 對於所有任何符合下列條件的參數(k,m,δ)：$k \geq \Omega(\log^2 n)$，$m|n$且$m \leq O(\sqrt{k/\log k})$，$\delta \geq 2^{-O(\sqrt{k/\log k})}$，存在一個具備下列性質的學習演算法 A：針對佈於$\{0,1\}^n = F^l$的給定來源$W$，若$H_\infty(W) \geq k$且對於任意函數$q:F^l \to F$，演算法 A 根據分佈$(W,q(W))$取出$2^{O(k/\log k)}$個訓練範例，並且執行時間在$2^{n-k+O(k/\log k)}$以內，可輸出一串長度為$2^{n-k+O(k/\log k)}$的結果，而且該串結果有$1-o(1)$的機率會包含所有滿足下列條件式的$v \in F$：

$$\Pr_{w \in W}[q(w) = \langle v,w \rangle] \geq \frac{1}{2^m} + \delta \text{ 。}$$

- ## 學習含有雜訊的線性函數

(此部份之結果發表於 [LLT11a]，原文請參看附錄)

給定任何一個 statistical min-entropy 為 k 的隨機源 W，我們提出一個需要 K=$2^{O(k/\log k)}$個訓練樣本(w, q(w))，時間複雜度為$2^{n-k+O(k/\log k)}$，且有很大機率會輸出一個包含所有滿足$\Pr_{w \in W}[q(w) \neq v(w)] \leq 1-2^{-O(\sqrt{k/\log k})}$的線性函數 v 的學習演算法。我們注意到，把 v 看成(v_1, …, v_ℓ)，則每一個訓練樣本(w, q(w))，可以給我們一個線性方程式 w_1v_1+…+$w_\ell v_\ell$=q(w)，所以我們的問題可以轉化成是去解一個有$2^{O(k/\log k)}$個方程式的線性方程組[$W^{(0)}$|$q^{(0)}$]，其中 $W^{(0)}$是一個 K× ℓ 的矩陣，而 $q^{(0)}$是一個 K-dimensional vector，且滿足對每一個訓練樣本(w, q(w))，$W^{(0)}$有一列為 w，而 $q^{(0)}$有一個元素為 q(w)。我們的學習演算法(如下一頁所示)包含兩個階段: Forward phas 和 Backward Phase，其中 d=k/mT，而 T 及 L 為適當的參數。

最後我們證明任何一個滿足$\Pr_{w \in W}[q(w) \neq v(w)] \leq 1-2^{-O(\sqrt{k/\log k})}$的 v 都有足夠的機率會被包含在 $V^{(0)}$中，而我們可以經由重複以上過程多次且輸出所有 $V^{(0)}$之聯集的方法來提高此成功機率。

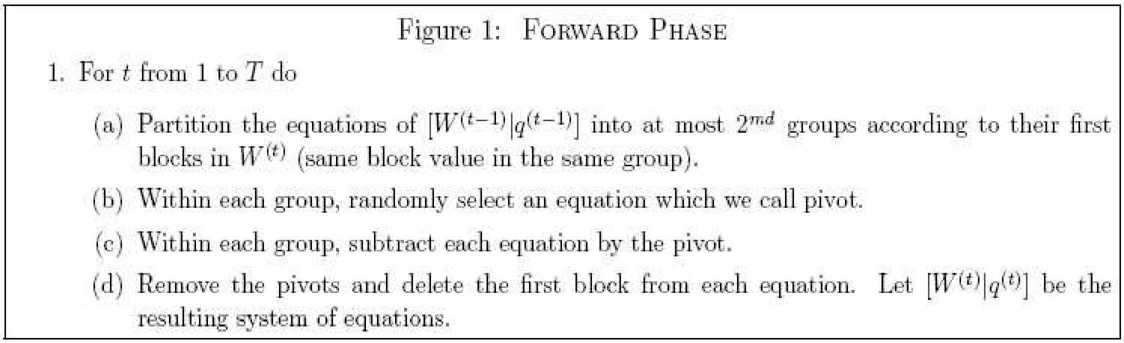


Figure 1: FORWARD PHASE

```
1. For $t$ from 1 to $T$ do
    (a) Partition the equations of $[W^{(t-1)}|q^{(t-1)}]$ into at most $2^{md}$ groups according to their first
        blocks in $W^{(t)}$ (same block value in the same group).
    (b) Within each group, randomly select an equation which we call pivot.
    (c) Within each group, subtract each equation by the pivot.
    (d) Remove the pivots and delete the first block from each equation. Let $[W^{(t)}|q^{(t)}]$ be the
        resulting system of equations.
```

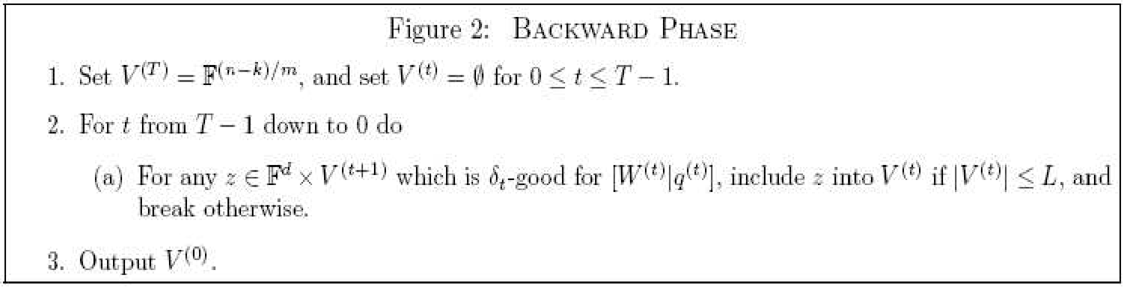


Figure 2: BACKWARD PHASE

```
1. Set $V^{(T)} = \mathbb{F}^{(n-k)/m}$, and set $V^{(t)} = \emptyset$ for $0 \le t \le T-1$.
2. For $t$ from $T-1$ down to 0 do
    (a) For any $z \in \mathbb{F}^d \times V^{(t+1)}$ which is $\delta_t$-good for $[W^{(t)}|q^{(t)}]$, include $z$ into $V^{(t)}$ if $|V^{(t)}| \le L$, and
        break otherwise.
3. Output $V^{(0)}$.
```

- **核心集的推廣**
  （此部份之結果發表於 [LLT11a]，原文請參看附錄）

我們將 Impagliazzo 著名的 hardcore lemma 加以推廣，得到下列結果：

［Lemma］令 $f:X\to[V]$ 為(δ, s)-hard 的函數，其中 $\delta\ge 1-\frac{1}{L}(1-\gamma)$ ， $L\in[V-1]$ 且 $\gamma\in(0,1)$ 。則對於任何 $\varepsilon>0$ ，都存在 $s'=s/poly(V,1/\varepsilon,\log(1/\gamma))$ 與 $I\in\binom{V}{L+1}$ 使得 f 具有密度為 $|H_1|/|X|\ge\gamma\Big/\binom{V}{L+1}$ 且難度為 (I,ε,s')-hard 的核心集 H_I 。

我們可以進一步將上述的想法推廣到一般的函數 $f:X\to[V]$，其中 $V\ge 3$ 。

［Lemma］假設對於某個 $I\subseteq[V]$， $f:X\to[V]$ 在 X 上不具有難度為 (I,ε,s')-hard 的核心集。則存在輸入的一個子集合 $T_I\subseteq f^{-1}(I)$ ，該子集在 X 中的密度小於 ρ ，也存在尺寸大小為 $|A_I|\le O((1/\varepsilon^2)\log(1/\rho))$ 的電路集合 $A_I\subseteq SIZE(s')$ 使得對於所有 $x\in f^{-1}(I)$ 下列成立:

$$\Pr_{A\in A_I}[A(x)=f(x)]>\frac{1}{|I|} 。$$

- ## 核心集的密度

（此部份之結果發表於 [LLT11b]，原文請參看附錄）

很自然的一個問題是：能否保證有更大的核心集？基本上，下列結果顯示無法有這樣的保證：

［定理］ 對任何$\delta = 1 - \frac{1}{L}(1-\gamma)$，其中$\gamma \in (0, 1/_{2(L+1)})$且$L \le V - 1$，都存在一個難度為$(\delta, s)$-hard 的函數$f:\{0,1\}^n \to [V]$，使得針對某個$s \ge poly(\gamma, 1/_L, 2^n)$，下列性質成立：

對於所有$I \in \binom{[V]}{L+1}$與$\varepsilon < \frac{1}{2L}$，存在某個$s' \le poly(n)$，使函數$f$在$\{0,1\}^n$上不具有密度為${}^{4(L+1)\gamma}/_{\binom{V}{L+1}}$的$(I, \varepsilon, s')$-hard 核心集。

這個結果說明即便我們有一個函數，且此函數相對於尺寸大如指數等級的電路而言是難以計算的，都只能夠保證該函數相對於多項式等級大小的電路可以具有低密度的核心集。

另一方面，這個定理和前一小節「核心集的推廣」裡所提到的結果相比較，兩者之間有$4(L+1)$的倍數差異。

- ## 計算觀點隨機元的萃取

（此部份之結果發表於 [LLT11b]，原文請參看附錄）

［Lemma］ 令$f: X \to [V]$是一個難度為(δ, s)-hard 的函數，其中$\delta \ge 1 - 1/_{2^k}$，而k是某個固定的正實數。令χ是X上的均等分布。則對於任何$\varepsilon \in (0,1)$，存在$s' \ge s / poly(V, 1/_\varepsilon)$與分佈$\nu$（與$\chi$相關）會使下列兩條件成立：
(1)針對分布$(\chi, f(\chi))$與分布(χ, ν)不存在(ε, s')-distinguisher(識別器)。
(2)$\Pr_{x \in \chi}\left[H_\infty(\nu \mid \chi = x) < \lceil k/_3 \rceil\right] \le 1/_{2^{(k/3)}}$。

［定理］ 令$f: X \to [V]$是一個難度為(δ, s)-hard 的函數，其中$\delta \ge 1 - 1/_{2^k}$，而k是某個固定的正實數。令χ是X上的均等分布。則對於任何可以被大小為$SIZE(s_0)$的電路計算，且具有種子的$(k/3, \varepsilon)$-萃取器$EXT: \{0,1\}^l \times \{0,1\}^d \to \{0,1\}^m$而言，會有某一組$\bar{\varepsilon} \le 2\varepsilon + 2^{-k/3}$與$\bar{s} \ge s / poly(2^l, 1/\varepsilon)$，使得分布$(\chi, EXT(f(\chi), U_d))$與$(\chi, U_m)$不存在$(\bar{\varepsilon}, \bar{s})$-distinguisher(識別器)。

［定理］ 令$i \in [t]$，且$f^{(i)}: X^{(i)} \to \{0,1\}^l$是一個難度為$(\delta^{(i)}, s)$-hard 的函數，其中

$\delta^{(i)} \ge 1-2^{-k^{(i)}}$ 且 $k=\sum_{i\in[t]} k^{(i)}$ 。令 $\chi=(\chi^{(1)},\ldots,\chi^{(t)})$ ，其中每一個 $\chi^{(i)}$ 都是在 $X^{(i)}$ 上的獨立均等分布。令 $f(\chi)=(f^{(1)}(\chi^{(1)}),\ldots,f^{(t)}(\chi^{(t)}))$ 。對於任何不具有種子的 t-來源 $(k/7,\varepsilon)$-萃取器 $EXT:(\{0,1\}^l)^t \to \{0,1\}^m$ ，若此萃取器可被大小為 $SIZE(s_0)$ 的電路計算，則存在某一組 $\bar{\varepsilon} \le (t+1)\varepsilon + 2^{-\Omega(k^2/tl^2)}$ 與 $\bar{s} \ge s/poly(2^l,1/\varepsilon)-s_0$ ，使得分布 $(\chi, EXT(f(\chi),U_d))$ 與 (χ,U_m) 不存在 $(\bar{\varepsilon},\bar{s})$ -distinguisher（識別器）。

- 電路大小的差距

  （此部份之結果發表於 [LLT11b]，原文請參看附錄）

［定理］假設 $V \ge \omega(1)$ ，$0<\delta\le 1-(4\log V)/V$ ，$0<\varepsilon\le 1/3$，$0<\rho<1$，且 $S \ge \Omega((V^{k+1}k^3/\varepsilon^2)\log(1/\rho))$ 。考慮任何配備有 oracle 的演算法，該演算法使用長度 $\tau \le o(\delta|X|)$ 的 advice（建議），並針對 $F_{X,V}$ 中的函數實現一個核心集的 $(\delta,k,\varepsilon,S,\rho)$-黑箱子證明。則此演算法詢問 oracle 的次數至少為 $\Omega((Vk/\varepsilon^2)\log(1/\delta))$ 。

參考文獻

[BHK08] B. Barak, M. Hardt, S. Kale, The Uniform Hardcore Lemma via Approximate Bregman Projectionss. In: SODA 2008, pp. 1193–1200, 2008.

[BIW04] B. Barak, R. Impagliazzo, and A. Wigderson. Extracting Ranodmness from Few Independent Sources. FOCS 2004.

[BKS+05] B.Barak, G. Kindler, R. Shaltiel, B. Sudakov, A. Wigderson. Simulating Independence: New Constructions of Condensers, Ramsey Graphs, Dispersers and Extractors. In Proc. 37th STOC. ACM, 2005.

[BKW03] A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. J. ACM, 50(4):506-519,2003.

[BSW03] B. Barak, R. Shaltiel, and A. Wigderson. Computational Analogues of Entropy. In RANDOM-APPROX 2003, 200-215.

[Bou05] J. Bourgain. More on the sum-product phenomenon in prime fields and its applications. International Journal of Number Theory, 1:1-32, 2005.

[CG88] B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. SIAM Journal on Computing. 17(2):230-261, 1988.

[Dav79] P. J. Davis. Circulant matrices. John Wiley, 1979.

[DO03] Y. Dodis, R. Oliveira. On Extracting Private Randomness over a Public Channel. RANDOM-APPROX 2003, 252-263.

[DORS08] Y. Dodis, R. Ostrovsky, L.Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM J. Comput., 38(1):97–139, 2008.

[DEOR04] Y. Dodis, A. Elbaz, R. Oliveira, R. Raz. Improved Randomness Extraction from Two Independent Sources. RANDOM-APPROX 2004.

[FGKP09] V. Feldman, P. Gopalan, S. Khot, and A. Ponnuswami, On agnostic learning of parities, monomials, and halfspaces, SIAM J. Comput., vol. 39, no. 2, pp. 606–645, 2009.

[FGKP06] V. Feldman, P. Gopalan, S. Khot, and A. K. Ponnuswami. New results for learning noisy parities and halfspaces. In Proc. 47th IEEE Symposium on Foundations of Computer Science (FOCS’06), 2006.

[GL89] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In Proc. 21st Annual ACM Symposium on Theory of Computing (STOC'89),1989.

[Gol98] O. Goldreich. Modern Cryptography, Probabilistic Proofs and Pseudorandomness. Springer-Verlag, Algorithms and Combinatorics, 1998.

[GRS04] A. Gabizon, R. Raz and R. Shaltiel. Deterministic extractors for bit-fixing sources by obtaining an independent seed. In Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, 2004.

[HILL99] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. SIAM J. Comput., 28(4):1364--1396, 1999.

[HLR07] C. Y. Hsiao, C. J. Lu, and L. Reyzin. Conditional computational entropy, or toward separating pseudoentropy form compressibility. In Proc. Advances in Cryptology-EUROCRYPT07, 2007.

[IJKW08] R. Impagliazzo, R. Jaiswal, V. Kabanets, and A. Wigderson. Uniform direct product theorems: simplified, optimized, and derandomized. In Proc.40th Annual ACM Symposium on Theory of Computing (STOC’08), pages 579–588, 2008.

[ILL89] R. Impagliazzo, L. A. Levin, and M. Luby. Pseudorandom generation from one-way functions. In Proceedings of the 21st ACM Symposium on Theory of Computing, 1989.

[Imp95] R. Impagliazzo, Hard-core distributions for somewhat hard problems. In: FOCS 1995, pp. 538–545 (1995)

[KMV08] A. Kalai, Y. Mansour, and E. Verbin, “On agnostic boosting and parity learning,” in Proc. 40th Annu. ACMSymp. Theory Comput. (STOC’08), pp. 629–638.

[KRVZ06] J. Kamp, A. Rao, S. Vadhan, and D. Zuckerman. Deterministic Extractors for Small-Space Sources. In Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC `06), pages 691-700, May 2006.

[KSS94] M. Kearns, R. Schapire, and L.Sellie. Toward efficient agnostic learning. Machine Learning, 17(2/3): 115-142, 1994.

[KZ03] J. Kamp and D. Zuckerman, Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. In Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, 2003.

[KZ07] J. Kamp and D. Zuckerman, Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. SIAM Journal on Computing, 36(5):1231–1247, 2007.

[Lee10] C. J. Lee, Seedless Eztractors: Constructions and Analysis, Doctor dissertation, National Chiao Tung University, Hsinchu, Taiwan, 2010.

[LLTT05] C. J. Lee, C. J. Lu, S. C. Tsai, and W. G. Tzeng. Extracting randomness from n independent weak random sources. IEEE Transactions on Information Theory (SCI), 51(6) 2224-2227, 2005.

[LLT06] C. J. Lee, C. J. Lu, and S. C. Tsai, Deterministic extractors for independent-symbol sources. In Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP), pages 84-95, 2006.

[LLT11a] C. J. Lee, C. J. Lu, and S. C. Tsai, Extracting Computational Entropy and Learning Noisy Linear Functions, IEEE Transactions on Information Theory, Vol. 57(8) 5485-5496, 2011.

[LLT11b] C. J. Lee, C. J. Lu, and S. C. Tsai, Computational Randomness from Generalized Hardcore Sets, 18th International Symposium on Fundamentals of Computation Theory, Oslo, NORWAY, August 22-25, 2011.

[LRVW03] C. J. Lu, O. Reingold, S. Vadhan, and A. Wigderson. Extractors: Optimal up to Constant Factors. In Proceedings of the 35th ACM Symposium on Theory of Computing, pages 601-611, 2003.

[MR95] R. Motwani and P. Raghavan. Randomized algorithms. Cambridge University press, 1995.

[Rao09] A. Rao, Extractors for a constant number of polynomially small minentropy independent sources, SIAM J. Comput., vol. 39, no. 1, pp. 168–194, 2009.

[Raz05] R. Raz. Extractors with Weak Random Seeds. Proceeding of the 37th STOC, 2005, pp. 11-20.

[RSW00] O. Reingold, R. Shaltiel, and A. Wigderson. Extracting randomness via repeated condensing. In Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science, 2000.

[RVW00] O. Reigold, S. Vadhan, and A. Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. In Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science, 2000.

[TSUZ01] A. Ta-Shma, C. Umans, and D. Zuckerman. Loss-less condensers, unbalanced expanders, and extractors. In Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, 2001.

[TSZS01] A. Ta-Shma, D. Zucherman and S. Safra. Extractors from Reed-Muller codes. In Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science, 2001.

[Yao82] A. Yao. Theory and applications of trapdoor functions. In Proc. 23rd Annual Symposium on Foundations of Computer Science (FOCS'82), 1982.

[Zuc97] D. Zuckerman. Randomness-optimal oblivious sampling. Random Structures and Algorithms, 11:345-367, 1997.

計畫成果自評

一、我們利用循環矩陣的特性來簡化針對獨立符號來源萃取器的證明並獲得一個更好的結果。

二、由計算的角度，我們發現無法從一個只擁有 computational min-entropy 的隨機源中萃取出一個隨機元。

三、我們能證明[LLTT05]所提出的針對兩個弱隨機源的萃取器也是一個針對兩個只擁有 computational min-entropy 的隨機源的萃取器。

四、我們提出一個可利用任何分布來學習線性函數的學習演算法。

五、我們延伸核心集引理從原本只考慮函數 f 為布林函數的情況到考慮 $f:\{0,1\}^{\ell}\to[D]$。

六、我們利用延伸的核心集引理證明[LLT06]中針對獨立符號來源的萃取器亦可從計算的獨立符號來源中萃取出隨機元，其在小電路的眼中看起來是非常隨機的。

七、我們亦利用證明針對計算獨立符號來源之萃取器的方法證明延伸的 XOR 引理，其將原本只考慮函數 f 為布林函數的情況延伸到考慮 $f:\{0,1\}^{\ell}\to[D]$。

八、最後我們證明一個核心集的黑箱子建造法中二元核心集大小的上限。

九、前述結果已分別整理並發表於期刊：IEEE Transactions on Information Theory, Vol.57(8) 5485-5496, 2011 以及國際會議 18th International Symposium on Fundamentals of Computation Theory, Oslo, NORWAY.

Extracting Computational Entropy and Learning Noisy Linear Functions

Chia-Jung Lee, Chi-Jen Lu, and Shi-Chun Tsai, *Member, IEEE*

***Abstract*—We study the task of deterministically extracting randomness from sources containing computational entropy. The sources we consider have the form of a conditional distribution $(f(\mathcal{X})|\mathcal{X})$, for some function f and some distribution $\mathcal{X}$, and we say that such a source has computational min-entropy k if any circuit of size 2^k can only predict $f(x)$ correctly with probability at most 2^{-k} given input x sampled from $\mathcal{X}$. We first show that it is impossible to have a seedless extractor to extract from one single source of this kind. Then we show that it becomes possible if we are allowed a seed which is weakly random (instead of perfectly random) but contains some statistical min-entropy, or even a seed which is not random at all but contains some computational min-entropy. This can be seen as a step toward extending the study of multisource extractors from the traditional, statistical setting to a computational setting. We reduce the task of constructing such extractors to a problem in computational learning theory: learning linear functions under arbitrary distribution with adversarial noise, and we provide a learning algorithm for this problem. In fact, this problem is a well-recognized one in computational learning theory and variants of this problem have been studied intensively before. Thus, in addition to its application to extractors, our learning algorithm also has independent interest of its own, and it can be considered as the main technical contribution of this paper.**

***Index Terms*—Computational min-entropy, randomness extractors, learning linear functions, computational complexity.**

I. Introduction

RANDOMNESS has become a useful tool in computer science, as the most efficient algorithms known for many important problems are randomized. However, when analyzing the performance of a randomized algorithm, we usually assume that the algorithm has access to a perfectly random source. In reality, the random sources we have access to are usually not perfect but may contain some amount of randomness. The amount of randomness in a source is usually measured by its min-entropy, where a source has min-entropy at least k if every element occurs with probability at most 2^{-k}. From a source with some min-entropy, we would like to have a procedure, called an *extractor* [30], [22], to extract almost perfect randomness, which can then be used for randomized algorithms.

Most works on extractors focused on *seeded* extractors, which can utilize an additional seed to aid the extraction. There has been a long and fruitful line of results on constructing seeded extractors (see [25] for a nice survey), which culminated in [21] and [13] with an optimal construction (up to constant factors). However, there is an issue with using seeded extractors. Namely, we need a seed which is perfectly random and independent of the source we extract from. How do we get such a seed? For some applications, this can be taken care of (e.g., by enumerating through all possible seed values), but for others, this seems to go back to the problem which we try to solve using extractors. Can we get rid of the need for a seed and have *seedless* extractors? For general sources, the answer has been known to be negative [7]. On the other hand, when the sources are restricted and have special structure, it becomes possible to have seedless extractors. Examples of such sources include samplable sources [28], bit-fixing sources [8], [18], [10], independent-symbol sources [17], [19], and multiple independent sources [7], [2], [3], [24], [6], [23].

In this paper, we would like to look for a more general class of sources from which seedless extraction is still possible. In particular, we will consider sources which may contain no randomness at all in a statistical sense, but *look* slightly random to computational-bounded observers, such as small circuits. That is, we will go from a traditional, statistical setting to a computational one. It is conceivable that in many situations when we consider a source random, it may in fact only appear so to us, while its actual statistical min-entropy may be much smaller (or even zero) especially if we take into account some correlated information which we can observe. Another application of this notion is in cryptography, and in fact the idea of extracting computational randomness has appeared implicitly long ago since [29], [11], [14], for the task of constructing pseudorandom generators from one-way functions. The idea is that given a one-way function g, it is hard to invert $g(y)$ to get y, and this means that given the (correlated) information $g(y)$, y still looks somewhat random, from which one can extract some bits that look almost random. However, while there is a natural and well-accepted definition for what it means that a distribution looks almost random [29], it seems less clear how to define that a distribution looks slightly random and how to measure the amount of randomness in it. In fact, there are several alternatives which all seem reasonable,

Manuscript received October 05, 2009; revised November 10, 2010; accepted December 08, 2010. Date of current version July 29, 2011. The work of C.-J. Lu was supported in part by the National Science Council of Taiwan under Contract NSC-97-2221-E-001-012-MY3. The work of S.-C. Tsai was supported in part by the National Science Council of Taiwan under Contracts NSC-97-2221-E-009-064-MY3 and NSC-98-2221-E-009-078-MY3. The material in this paper was presented in part at the 15th International Computing and Combinatorics Conference (COCOON), Niagara Falls, NY, 2009.

C.-J. Lee was with the Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan. She is now with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan (e-mail: leecj@iis.sinica.edu.tw).

C.-J. Lu is with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan (e-mail: cjlu@iis.sinica.edu.tw).

S.-C. Tsai is with the Department of Computer Science, National Chiao Tung University, Hsinchu 300, Taiwan (e-mail: sctsai@csie.nctu.edu.tw).

Communicated by K. M. Martin, Associate Editor for Complexity and Cryptography.

Digital Object Identifier 10.1109/TIT.2011.2158897

but there are provable discrepancies among them [4], [15]. To extract randomness from a source with so-called HILL-entropy [4], the strongest among them, one can simply use any statistical extractor, but we would like to extract randomness from a broader class of sources. Here we consider a weaker (more general) notion of computational randomness, which appears in [15], and we call it *computational min-entropy*. A comparison with other notions of computational randomness can be found in [15].

A. Computational Min-Entropy

To model the more general situation that one may observe some correlated information about the source, we consider the setting with a pair of jointly distributed random variables $\mathcal{V}$ and $\mathcal{X}$, where $\mathcal{V}$ is the source from which we want to extract and $\mathcal{X}$ (could be empty) is some information which one can observe. To stress that we want to measure the randomness of $\mathcal{V}$ conditioned on $\mathcal{X}$ and to extract randomness from $\mathcal{V}$ given the information $\mathcal{X}$, we use the notation $(\mathcal{V}|\mathcal{X})$ to denote such a joint distribution. The correlation between $\mathcal{V}$ and $\mathcal{X}$ is modeled by $\mathcal{V} = f(\mathcal{X})$ for some function f. In the example of one-way permutation, f is the inverse function g^{-1}, which is hard to compute, and $\mathcal{X}$ is the distribution of $g(y)$ over a random y. Here in our definition, we allow f to be probabilistic and we even do not require it to have an efficient (or even computable) algorithm, and furthermore, we do not require $\mathcal{X}$ to be efficiently samplable either. We say that such a source $(f(\mathcal{X})|\mathcal{X})$ has computational min-entropy k if given input x sampled from $\mathcal{X}$, any circuit of size 2^k can only predict $f(x)$ correctly with probability at most 2^{-k}.[1] From the distribution $f(\mathcal{X})$, we would like to extract randomness which when given $\mathcal{X}$ still looks random to circuits of a certain size. Note that a source $\mathcal{Y}$ with statistical min-entropy k can be seen as such a source $(f(\mathcal{X})|\mathcal{X})$ with computational min-entropy k, where we can simply have no $\mathcal{X}$ or just have $\mathcal{X}$ taking a fixed value, and let f be a probabilistic function with $\mathcal{Y}$ as its output distribution. This means that extractors for sources with computational min-entropy can immediately work for sources with statistical min-entropy, and thus results in the computational setting can be seen as a generalization of those in the traditional, statistical setting. On the other hand, for a deterministic function f, $f(x)$ has no statistical min-entropy at all when given x. Still, according to our definition, as long as f is hard to compute, $(f(\mathcal{X})|\mathcal{X})$ in fact can have high computational min-entropy.

Extractors for such sources were implicitly proposed before [11], [14], and they are seeded ones. That is, they need an additional seed which must be perfectly random and independent of the source. In fact, it is known that any seeded statistical extractor with some additional *reconstruction* property (in the sense of [27]) gives a seeded extractor for such sources [4], [26], [15]. However, just as in the statistical setting, several natural questions arise in the computational setting too. To extract from such sources, do we really need a seed? Can we use a weaker seed which is only slightly random, instead of perfectly random, in a statistical sense, or an even weaker seed which only looks slightly random in a computational sense but may contain no randomness in a statistical sense? Seeing the seed as an additional independent source, a general question is: Can we have seedless extractors for multiple independent sources in which each source contains some computational min-entropy? We will try to answer these questions in this paper. One can see this as a step toward extending the study of multisource extractors from the traditional, statistical setting to a new, computational setting. One can also see this as providing a finer map for the landscape of statistical extractors, according to the degree of their reconstruction property.

B. Our Results

First, we show that it is impossible to have seedless extractors for one single source, even if the source of length n can have a computational min-entropy as high as $n-2$ and even if we only want to extract one bit.

Next, we show that with the help of a weak seed, it becomes possible to extract randomness from such sources. We use a two-source extractor of Lee *et al.* [20], denoted as EXT, which takes two input strings $v, w \in \{0,1\}^n$, sees them as vectors from $\mathbb{F}^\ell$, where $\mathbb{F} = GF(2^m)$ for some m with $n = m\ell$, and outputs their inner product, denoted as $\langle v, w\rangle$, over $\mathbb{F}$. As shown in [20], it works for any two independent sources both containing some statistical min-entropy. Moreover, it is also known to work when one source contains some computational min-entropy and the other, the seed, is perfectly random (in a statistical sense) [12]. Our second result shows that it even works when the seed only contains some statistical min-entropy. More precisely, we show that given any source $(f(\mathcal{X})|\mathcal{X})$ with computational min-entropy $k_1 = n - k + O(k/\log k)$ and another independent source $\mathcal{W}$ with statistical min-entropy k, the output $\text{EXT}(f(\mathcal{X}), \mathcal{W})$ given $\mathcal{X}$ cannot be distinguished from random with advantage $\varepsilon = 2^{-O(\sqrt{k/\log k})}$ by circuits of size $s = 2^{n-k+O(k/\log k)}$. That is, for any such Boolean circuit D, $|\Pr[D(\mathcal{X}, \text{EXT}(f(\mathcal{X}))) = 1] - \Pr[D(\mathcal{X}, \mathcal{U}) = 1]| \le \varepsilon$, where $\mathcal{U}$ denotes the uniform distribution. Then we proceed to show that the extractor even works when the seed only contains computational min-entropy. More precisely, when we replace the source $\mathcal{W}$ by a source $(g(\mathcal{Y})|\mathcal{Y})$ with computational min-entropy k, $\text{EXT}(f(\mathcal{X}), g(\mathcal{Y}))$ given $(\mathcal{X}, \mathcal{Y})$ still cannot be distinguished with advantage ε by circuits of size about s. This can be seen as a seedless extractor for two independent sources, both with computational min-entropy.

We do not know if the statistical extractors of [2], [3], [24], [6], and [23] for multiple independent sources can also work in the computational setting, since to work in this setting, we need them to have some reconstruction property. For the extractors from [11] and [12], this property can be translated to a task in learning theory, and the proofs there can be recast as providing an algorithm for learning linear functions under *uniform* distribution with adversarial noise. Our second result can be seen as a generalization of [11] and [12], but we are facing a more challenging learning problem: learning linear functions under *arbitrary* distribution with adversarial noise. Our third result provides an algorithm for this problem, which, in addition to being used to prove our second result, may have interest of its own.

In the learning problem, there is some unknown linear function $v : \mathbb{F}^\ell \to \mathbb{F}$, defined as $v(w) = \langle v, w\rangle$, which we want

[1] A more general definition is to have the circuit size as a separate parameter, but our extractor construction does not seem to work for this more general definition.

to learn, and there is a distribution $\mathcal{W}$ over $\mathbb{F}^\ell = \{0,1\}^n$ from which we can sample w to obtain a training example $(w, q(w))$, for some function $q : \mathbb{F}^\ell \to \mathbb{F}$. The function q can be seen as a noisy version of v with some noise rate α, and there are two noise models. In the adversarial-noise model, q is a deterministic function such that $\Pr_{w \in \mathcal{W}}[q(w) \neq v(w)] \leq \alpha$. In the random-noise model, q is a probabilistic function such that independently for any w, $\Pr[q(w) \neq v(w)] \leq \alpha$. We consider the more difficult adversarial-noise model, and our algorithm works for an arbitrary distribution $\mathcal{W}$, while its complexity depends on the min-entropy k of $\mathcal{W}$. More precisely, our algorithm samples $2^{O(k/\log k)}$ training examples, runs in time $2^{n-k+O(k/\log k)}$, and with high probability outputs a list containing every linear function v satisfying $\Pr_{w \in \mathcal{W}}[q(w) \neq v(w)] \leq \alpha$, for $\alpha = (1 - 1/|\mathbb{F}|) - 2^{-O(\sqrt{k/\log k})}$. The factor 2^{n-k} in our running time is in fact unavoidable because one can easily find a distribution $\mathcal{W}$ (e.g., the first k bits perfectly random and the rest fixed) for which the number of such v's, and thus the running time, is in fact at least 2^{n-k}. Note that when $\mathcal{W}$ is the uniform distribution (with $k = n$), our algorithm runs in time $2^{O(n/\log n)}$ and takes $2^{O(n/\log n)}$ samples.

Previously, the algorithm of Blum, Kalai, and Wasserman [5] can learn under arbitrary distribution but in the random-noise model, while that of Feldman *et al.* [9] can learn in the adversarial-noise model but under the uniform distribution. Both algorithms learn the parity functions on n variables, tolerate a noise rate $\alpha \leq 1/2 - \Omega(1)$, run in time $2^{O(n/\log n)}$, and take $2^{O(n/\log n)}$ samples. Very recently, Kalai, Mansour, and Verbin [16] gave an algorithm which can learn the parity functions under arbitrary distribution in the adversarial-noise model, but the hypothesis they produce is not in the linear form, so it cannot be used for our extractors. Furthermore, their algorithm only produces one hypothesis instead of all the legitimate ones, and their technique does not seem to generalize from the parity functions to the linear functions over larger fields. Thus, to the best of our knowledge, the task our learning algorithm achieves has not been accomplished before. Finally, just as the result of [11] can yield a list-decoding algorithm for Hadamard codes, so can ours, while that of [16] cannot. In fact, our list-decoding algorithm can work even when all but 2^k symbols from the codeword are erased and an α fraction of the remaining symbols are corrupted. It can also be seen as list-decoding a *punctured* Hadamard code, where a punctured code is obtained from a code by deleting all but a small number of symbols from the codeword.

C. Our Techniques

For our impossibility result, we show that for any function $\textsc{Ext} : \{0,1\}^n \to \{0,1\}$, there exists a function $f : \{0,1\}^{3n} \to \{0,1\}^n$ such that $(f(\mathcal{X})|\mathcal{X})$ has computational min-entropy $n - 2$, but $\textsc{Ext}(f(x))$ takes an identical value for all x. We show the existence of such a function f by a standard probabilistic argument: in fact, a random function from $\{0,1\}^{3n}$ to $\textsc{Ext}^{-1}(b)$ is likely to work, for the $b \in \{0,1\}$ with the larger $\textsc{Ext}^{-1}(b)$.

To show that our extractor works in the computational setting, we follow the approach of [11] and reduce it to the task of learning linear functions as we just discussed. More precisely, for the case when the source $(f(\mathcal{X})|\mathcal{X})$ has computational min-entropy and the seed $\mathcal{W}$ has statistical min-entropy, the reduction works as follows. Assume our extractor $\textsc{Ext}$ does not work, and thus some efficient distinguisher can tell the distribution of $\textsc{Ext}(f(x), \mathcal{W}) = \langle f(x), \mathcal{W}\rangle$ from random given x, for a large fraction of x from $\mathcal{X}$. For any such x, we can then predict the value $\langle f(x), \mathcal{W}\rangle$ with a good probability, given the ability to sample from $\mathcal{W}$, which can then be used by the learning algorithm to learn $f(x)$. This would give us an efficient algorithm for predicting $f(x)$ for those x's, if we could in fact sample $\mathcal{W}$ efficiently. However, this may not be the case in general as $\mathcal{W}$ could be any arbitrary distribution. Still, by an average argument, there must exist a small set of samples from $\mathcal{W}$ which preserve this predicting probability, so we can hard-wire them in to get a circuit which predicts f well. If the function f is hard, this is impossible, so the assumed distinguisher cannot exit, and $\textsc{Ext}$ indeed works. For the case that the seed comes from a distribution $(g(\mathcal{Y})|\mathcal{Y})$ with computational min-entropy, observe that $g(\mathcal{Y})$ alone (without conditioning on $\mathcal{Y}$) must have some statistical min-entropy, because otherwise it becomes easy to predict. Then a very similar argument as above can be used.

Note that our results on extractors still depend on the existence of a good learning algorithm, and our main technical contribution can be seen as providing such an algorithm. Our algorithm can be seen as extending that of [5] from the random-noise model to the adversarial-noise model. Note that in the random-noise model, it is possible to predict the value of $v(w)$ with confidence for an input w by taking the majority vote on several independent predictions, while in the adversarial-noise model, this does not seem so and the learning task becomes much harder.

Our learning algorithm works as follows. We start by sampling some number K of training examples $(w, q(w))$ from $(\mathcal{W}, q(\mathcal{W}))$. Note that each example $(w, q(w))$ gives us a linear equation $\langle v, w\rangle = q(w)$ for the unknown v, so the K examples gives us a system of K linear equations, some of which may be wrong. We reduce the original problem of learning the unknown v to the problem of solving such a noisy system of linear equations. To solve the system, we proceed in two phases. In the forward phase, we start from the system, and use several iterations to produce smaller and smaller systems with fewer and fewer variables, until we have a small enough system which we can afford to solve using brute force. Then we enter the backward phase, and starting from the last system produced by the forward phase, we work backward on larger and larger systems produced in the forward phase to obtain solutions for more and more variables. Since the possible solutions may not be unique, we keep them all in a list in each iteration, and the list in the final iteration of the backward phase is our output, which we hope contains the correct v.

The forward phase is similar in spirit to an approach in [5]. The key is to guarantee that after each iteration, the solution v is still good for the new system in the sense that the new system still contains a good fraction of correct equations with respect to v, so that v will not be lost when solving this new system. Using an argument similar to that in [5], we can show that this does hold with a significant probability. On the other hand, it is not clear whether or not some iteration in the forward phase would turn many originally bad solutions into good ones for the new system (satisfying a good fraction of its equations). That is, not only is v a good solution for the system, there are in fact too many good solutions for it. If this happens, then in the backward phase when we try to solve this system, we cannot afford to keep all such solutions, and we have the risk of losing the actual solution v. This tricky situation does not arise in the random-noise model considered in [5], so a much simpler algorithm works there. However, in the adversarial-noise model, this seems unavoidable. Fortunately, we can show that with high probability, the systems we produce indeed do not have too many good solutions. This turns out to rely on the fact that our extractor is also a good *statistical* extractor, together with the property, which we will show, that each system is likely to have a distribution which is close to some good distribution with high statistical min-entropy.

II. Preliminaries

For any $m \in \mathbb{N}$, let $\mathcal{U}_m$ denote the uniform distribution over $\{0,1\}^m$. Let $\mathsf{SIZE}(s)$ be the class of functions computable by Boolean circuits of size s. We say that a function $D : \{0,1\}^n \to \{0,1\}$ is an ε-distinguisher for two distributions $\mathcal{X}$ and $\mathcal{Y}$ over $\{0,1\}^n$ if

$$|\Pr[D(\mathcal{X})=1] - \Pr[D(\mathcal{Y})=1]| \geq \varepsilon.$$

All logarithms in this paper will have base two.

We consider two types of min-entropy: *statistical* min-entropy and *computational* min-entropy. The notion of statistical min-entropy is a standard one, usually just called min-entropy.

Definition 1: We say that a distribution $\mathcal{X}$ has *statistical min-entropy* at least k, denoted by $\mathrm{H}_\infty(\mathcal{X}) \geq k$, if for any x, $\Pr[\mathcal{X} = x] \leq 2^{-k}$.

Next, we define the notion of computational min-entropy. Here, we consider the more general setting of measuring the randomness of a distribution $\mathcal{V}$ given a correlated distribution $\mathcal{X}$, and we use $(\mathcal{V}|\mathcal{X})$ to denote such a joint distribution. The correlation between $\mathcal{V}$ and $\mathcal{X}$ is modeled by $\mathcal{V} = f(\mathcal{X})$ for some function f, which could be either probabilistic or deterministic.

Definition 2: We say that a distribution $(\mathcal{V}|\mathcal{X})$ has *computational min-entropy* k, denoted by $\mathrm{H}_c(\mathcal{V}|\mathcal{X}) = k$, if for any $C \in \mathsf{SIZE}(2^k)$, $\Pr[C(\mathcal{X}) = \mathcal{V}] \leq 2^{-k}$.

We consider three kinds of extractors: *statistical* extractors, *hybrid* extractors and *computational* extractors. The notion of statistical extractors is a standard one for 2-source extractors, usually just called 2-source extractors, while we introduce the notions of hybrid extractors and computational extractors.

Definition 3: A function $\mathrm{EXT} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ is called a

- (k_1, k_2, ε)*-statistical-extractor* if for any source $\mathcal{V}$ with $\mathrm{H}_\infty(\mathcal{V}) \geq k_1$ and any source $\mathcal{W}$, independent of $\mathcal{V}$, with $\mathrm{H}_\infty(\mathcal{W}) \geq k_2$, there is no ε-distinguisher (without any complexity bound) for the distributions $(\mathcal{W}, \mathrm{EXT}(\mathcal{V},\mathcal{W}))$ and $(\mathcal{W}, \mathcal{U}_m)$.
- $(k_1, k_2, \varepsilon, s)$*-hybrid-extractor* if for any source $(\mathcal{V}|\mathcal{X})$ with $\mathrm{H}_c(\mathcal{V}|\mathcal{X}) \geq k_1$ and any source $\mathcal{W}$, independent of $(\mathcal{V}|\mathcal{X})$, with $\mathrm{H}_\infty(\mathcal{W}) \geq k_2$, there is no ε-distinguisher in $\mathsf{SIZE}(s)$ for the distributions $(\mathcal{X}, \mathcal{W}, \mathrm{EXT}(\mathcal{V},\mathcal{W}))$ and $(\mathcal{X}, \mathcal{W}, \mathcal{U}_m)$.
- $(k_1, k_2, \varepsilon, s)$*-computational-extractor* if for any source $(\mathcal{V}|\mathcal{X})$ with $\mathrm{H}_c(\mathcal{V}|\mathcal{X}) \geq k_1$ and any source $(\mathcal{W}|\mathcal{Y})$, independent of $(\mathcal{V}|\mathcal{X})$, with $\mathrm{H}_c(\mathcal{W}|\mathcal{Y}) \geq k_2$, there is no ε-distinguisher in $\mathsf{SIZE}(s)$ for the distributions $(\mathcal{X}, \mathcal{Y}, \mathcal{W}, \mathrm{EXT}(\mathcal{V},\mathcal{W}))$ and $(\mathcal{X}, \mathcal{Y}, \mathcal{W}, \mathcal{U}_m)$.

Remark 1: Note that the definition above corresponds to the notion of strong extractors in the setting of seeded statistical extractors, which guarantees that even given the seed (the second source), the output still looks random.

We will need the following statistical extractor from [20], which generalizes the construction from [7]. For any $m \in \mathbb{N}$ with $m|n$, let $\ell = n/m$, and see any $x \in \{0,1\}^n$ as an ℓ-dimensional vector $x = (x_1, x_2, \ldots, x_\ell)$ over $\mathbb{F} = GF(2^m)$. Then for any $x, y \in \mathbb{F}^\ell$, let $\langle x, y\rangle$ be their inner product over $\mathbb{F}$ defined as

$$\langle x, y\rangle = \sum_{i=1}^{\ell} x_i \cdot y_i.$$

Theorem 1: [20] The function $\mathrm{EXT} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ defined as $\mathrm{EXT}(u,v) = \langle u, v\rangle$ is a (k_1, k_2, ε)-statistical-extractor when $k_1 + k_2 \geq n + m + 2\log(1/\varepsilon) - 2$.

We will need the following fact about statistical extractors.

Lemma 1: Let $\mathrm{EXT} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ be any (k_1, k_2, ε)-statistical-extractor. Then for any source $\mathcal{W}$ over $\{0,1\}^n$ with $\mathrm{H}_\infty(\mathcal{W}) = k_2$ and any function $q : \{0,1\}^n \to \{0,1\}^m$, there are at most 2^{k_1} different v's satisfying

$$\Pr_{w \in \mathcal{W}}[q(w) = \mathrm{EXT}(v,w)] \geq 1/2^m + \varepsilon.$$

Proof: Let V be the set consisting of such v's and let $\mathcal{V}$ be the uniform distribution over V. Consider the distinguisher D defined as $D(w,u) = 1$ if $q(w) = u$ and $D(w,u) = 0$ otherwise. Then, the difference

$$\Pr_{v \in \mathcal{V}, w \in \mathcal{W}}[D(w, \mathrm{EXT}(v,w)) = 1] - \Pr_{w \in \mathcal{W}, u \in \mathcal{U}_m}[D(w,u) = 1]$$

is equal to

$$\Pr_{v \in \mathcal{V}, w \in \mathcal{W}}[q(w) = \mathrm{EXT}(v,w)] - \Pr_{w \in \mathcal{W}, u \in \mathcal{U}_m}[q(w) = u]$$

which is at least

$$1/2^m + \varepsilon - 1/2^m = \varepsilon.$$

This implies that $\log|V| = \mathrm{H}_\infty(\mathcal{V}) \leq k_1$, because otherwise it would contradict the fact that EXT is a good statistical extractor. ■

Finally, we will need the following lemma about obtaining predictors from distinguishers. The Boolean case $(m = 1)$ is well known, and a proof for general m can be found in [12].

Lemma 2: For any source $\mathcal{Z}$ over $\{0,1\}^n$ and any function $b : \{0,1\}^n \to \{0,1\}^m$, if there is an ε-distinguisher D for the distributions $(\mathcal{Z}, b(\mathcal{Z}))$ and $(\mathcal{Z}, \mathcal{U}_m)$, then there is a predictor P with D as oracle which calls D once and runs in time $O(m)$ such that

$$\Pr_{z \in \mathcal{Z}}[P^D(z) = b(z)] \geq (1+\varepsilon)/2^m.$$

III. An Impossibility Result

Just as in the statistical setting [7], we show that seedless extractors do not exist either in the computational setting. In fact, we show the impossibility result even for sources with a computational min-entropy as high as $n-2$.

Theorem 2: For any $n_1, n \in \mathbb{N}$ with $n_1 \geq 3n$ and for any function $\textsc{Ext} : \{0,1\}^n \to \{0,1\}$, there exists a deterministic function $f : \{0,1\}^{n_1} \to \{0,1\}^n$ such that $\mathrm{H}_c(f(\mathcal{X})|\mathcal{X}) = n-2$ for $\mathcal{X} = \mathcal{U}_{n_1}$ but $\textsc{Ext}(f(x))$ takes the same value for all x (so can be easily distinguished from random).

Proof: Consider any function $\textsc{Ext} : \{0,1\}^n \to \{0,1\}$. Assume without loss of generality that $|\textsc{Ext}^{-1}(1)| \geq 2^{n-1}$. Then we will show the existence of a function f such that $\mathrm{H}_c(f(\mathcal{X})|\mathcal{X}) = n-2$ but $\textsc{Ext}(f(x)) = 1$ for all x. In fact, a standard argument can show that a random function is likely to work, as we will describe next.

Consider a random function $f : \{0,1\}^{n_1} \to \textsc{Ext}^{-1}(1)$. Fix any $C : \{0,1\}^{n_1} \to \{0,1\}^n \in \mathsf{SIZE}(2^{n-2})$, and for each $x \in \{0,1\}^{n_1}$, define a binary random variable C_x such that $C_x = 1$ if and only if $C(x) = f(x)$. Observe that $\sum_x C_x$ is the number of x satisfying $C(x) = f(x)$. Note that

$$\begin{aligned}\mathop{\mathrm{E}}_f\left[\sum_x C_x\right] &= \sum_x \mathop{\mathrm{E}}_f[C_x] \\ &= \sum_x \Pr_f[C(x) = f(x)] \\ &\leq 2^{n_1-(n-1)},\end{aligned}$$

and let $\mu = 2^{n_1-(n-1)}$. Then by a Chernoff bound (see e.g., [1]), we have

$$\Pr_f\left[\sum_x C_x \geq 2\mu\right] \leq 2^{-\Omega(\mu)} = 2^{-\Omega(2^{n_1-n})}.$$

Since $|\mathsf{SIZE}(2^{n-2})| \leq 2^{O(n2^n)}$ and $n_1 \geq 3n$, a union bound gives

$$\begin{aligned}&\Pr_f\left[\exists C \in \mathsf{SIZE}(2^{n-2}) \text{ s.t. } \sum_x C_x \geq 2\mu\right] \\ &\leq 2^{O(n2^n)} \cdot 2^{-\Omega(2^{n_1-n})} \\ &< 1.\end{aligned}$$

Hence, there exists some f, such that $\Pr_x[C(x) = f(x)] < 2\mu \cdot 2^{-n_1} = 2^{-(n-2)}$ for any $C \in \mathsf{SIZE}(2^{n-2})$, but $\textsc{Ext}(f(x)) = 1$ for any x. This completes the proof.

IV. Hybrid and Computational Extractors

In this section, we show that the function $\textsc{Ext} : \mathbb{F}^\ell \times \mathbb{F}^\ell \to \mathbb{F}$ defined in Theorem 1 as

$$\textsc{Ext}(v, w) = \langle v, w \rangle,$$

which is known to be a good statistical extractor, is also a good hybrid extractor and a good computational extractor.

Theorem 3: For any $k \geq \Omega(\log^2 n)$, any $m \leq O(\sqrt{k/\log k})$ dividing n, any $\varepsilon \geq 2^{-O(\sqrt{k/\log k})}$, any $s \leq 2^{n-k+O(k/\log k)}$, and for some $k_1 = n - k + O(k/\log k)$, the function $\textsc{Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ defined above is both a (k_1, k, ε, s)-hybrid-extractor and a (k_1, k, ε, s)-computational-extractor.

The proof for Theorem 3 relies on the following result, which gives an algorithm for the problem of learning linear functions under arbitrary distribution with adversarial noise.

Theorem 4: For any $k \geq \Omega(\log^2 n)$, any $m \leq O(k/\log k)$ dividing n, and any $\delta \geq 2^{-O(\sqrt{k/\log k})}$, there exists a learning algorithm A with the following property. Given any source $\mathcal{W}$ over $\{0,1\}^n = \mathbb{F}^\ell$ with $\mathrm{H}_\infty(\mathcal{W}) \geq k$ and any function $q : \mathbb{F}^\ell \to \mathbb{F}$, the algorithm A samples $2^{O(k/\log k)}$ training examples from the distribution $(\mathcal{W}, q(\mathcal{W}))$ and then runs in time $2^{n-k+O(k/\log k)}$ to output a list of size $2^{n-k+O(k/\log k)}$ which with probability $1 - o(1)$ contains every $v \in \mathbb{F}^\ell$ satisfying

$$\Pr_{w \in \mathcal{W}}[q(w) = \langle v, w \rangle] \geq 1/2^m + \delta.$$

Note that as in a standard learning-theoretical setting, we do not count the complexity of sampling the training examples (or just count each sampling as unit cost) in Theorem 4. We will prove the theorem in the next section, and now let us see how it is used to show Theorem 3.

Proof: (of Theorem 3)

First, we prove that the function Ext is a good hybrid extractor. Consider any source $(\mathcal{V}|\mathcal{X})$ with $\mathrm{H}_c(\mathcal{V}|\mathcal{X}) \geq k_1$ and any source $\mathcal{W}$, which is independent of $(\mathcal{V}|\mathcal{X})$, with $\mathrm{H}_\infty(\mathcal{W}) \geq k$. Assume for the sake of contradiction that there exists an ε-distinguisher $D \in \mathsf{SIZE}(s)$ for the distributions $(\mathcal{X}, \mathcal{W}, \langle \mathcal{V}, \mathcal{W} \rangle)$ and $(\mathcal{X}, \mathcal{W}, \mathcal{U}_m)$. By Lemma 2, this implies the existence of a predictor $Q \in \mathsf{SIZE}(s + O(m))$ with

$$\Pr_{x \in \mathcal{X}, v \in \mathcal{V}, w \in \mathcal{W}}[Q(x, w) = \langle v, w \rangle] \geq (1 + \varepsilon)/2^m.$$

Let $\delta = \varepsilon/2^{m+1} \geq 2^{-O(\sqrt{k/\log k})}$, and call any (x, v) heavy if

$$\Pr_{w \in \mathcal{W}}[Q(x, w) = \langle v, w \rangle] \geq 1/2^m + \delta.$$

Then a Markov inequality shows that

$$\Pr_{x \in \mathcal{X}, v \in \mathcal{V}}[(x, v) \text{is heavy}] \geq \delta.$$

Given any heavy (x, v), we want to predict v from x with a good probability. This can be reduced to the task of learning the linear function $\langle v, \cdot \rangle$, through noisy training examples $(w, q(w))$, with $q(w) = Q(x, w)$, under the distribution $w \in \mathcal{W}$. Consider the algorithm C which on input x calls the algorithm A in Theorem 4 using the function $q(\cdot) = Q(x, \cdot)$, and outputs a random element in the list produced by A. It samples $2^{O(k/\log k)}$ independent elements, denoted as W, from $\mathcal{W}$, makes $2^{O(k/\log k)}$ calls to Q, and for any heavy (x, v) it outputs v with probability $(1 - o(1)) \cdot 2^{-(n-k+O(k/\log k))}$. Then $\Pr_{x,v,W}[C(x) = v]$ is at least

$$\Pr_{x,v}[(x, v) \text{ is heavy}] \cdot \Pr_{x,v,W}[C(x) = v \mid (x, v) \text{ is heavy}]$$

which is at least

$$\delta \cdot (1 - o(1)) \cdot 2^{-(n-k+O(k/\log k))} \geq 2^{-(n-k+O(k/\log k))}.$$

That is, we have

$$\Pr_{x,v,W}[C(x)=v] \geq 2^{-(n-k+O(k/\log k))}.$$

We are almost done except that we still cannot bound the complexity of the algorithm C because it needs a way to sample elements from the source $\mathcal{W}$ which may not have an efficient sampling algorithm, unlike in the learning setting where one does not count the complexity of sampling. Fortunately, by an average argument, the bound above still holds for some fixed W, and we can simply hard-wire it into C. Similarly, we can do this for other random choices of C, and it is not hard to show that one can have a resulting circuit of size

$$|W|^{O(1)} + 2^{O(k/\log k)} \cdot (s + O(m)) + 2^{n-k+O(k/\log k)}$$

which is at most

$$2^{n-k+O(k/\log k)}.$$

Thus, for some large enough $k_1 = n-k+O(k/\log k)$, we have a circuit of size smaller than 2^{k_1} which can predict v correctly with probability at least

$$2^{-(n-k+O(k/\log k))} > 2^{-k_1}.$$

This contradicts the assumption that $\mathrm{H}_c(\mathcal{V}|\mathcal{X}) \geq k_1$, which means that the distinguisher D assumed at the beginning cannot exist, so EXT is a good hybrid extractor as claimed.

Next, we prove that EXT is also a good computational extractor, and the proof is almost identical. Consider two independent sources $(\mathcal{V}|\mathcal{X})$ and $(\mathcal{W}|\mathcal{Y})$, with $\mathrm{H}_c(\mathcal{V}|\mathcal{X}) \geq k_1$ and $\mathrm{H}_c(\mathcal{W}|\mathcal{Y}) \geq k$. Observe that the distribution of $\mathcal{W}$ must have statistical min-entropy at least k, because otherwise the predictor which always outputs the value with the largest measure can predict $\mathcal{W}$ correctly with probability larger than 2^{-k}, a violation of the assumption that $\mathrm{H}_c(\mathcal{W}|\mathcal{Y}) \geq k$. Then we can follow the proof above: assuming the existence of a distinguisher for EXT, we can obtain a predictor of size smaller than 2^{k_1}, with some $2^{O(k/\log k)}$ elements from $(\mathcal{W},\mathcal{Y})$ hard-wired in it, which can predict $\mathcal{V}$ correctly with probability larger than 2^{-k_1}. This contradicts the fact that $\mathrm{H}_c(\mathcal{V}|\mathcal{X}) \geq k_1$, so EXT is a good computational extractor. ■

V. LEARNING NOISY LINEAR FUNCTIONS

In this section, we prove Theorem 4. Recall that given any source $\mathcal{W}$ over $\{0,1\}^n = \mathbb{F}^\ell$ with $\mathrm{H}_\infty(\mathcal{W}) \geq k$, any $\delta \geq 2^{-O(\sqrt{k/\log k})}$, and any function $q : \mathbb{F}^\ell \to \mathbb{F}$, we would like to learn some unknown $v \in \mathbb{F}^\ell$ such that

$$\Pr_{w \in \mathcal{W}}[q(w) = \langle v, w\rangle] \geq 1/2^m + \delta. \tag{1}$$

Since such v may not be unique, we will list them all. Let us first imagine one such fixed v.

We start by randomly choosing $K = 2^{c(k/\log k)}$ independent training examples (with replacement) from the distribution $(\mathcal{W}, q(\mathcal{W}))$, for some large enough constant c (depending on δ). Let $W^{(0)}$ denote the $K \times \ell$ matrix and $q^{(0)}$ the K-dimensional vector, both over $\mathbb{F}$, such that for each training example $(w, q(w))$, $W^{(0)}$ has $w \in \mathbb{F}^\ell$ as a row and $q^{(0)}$ has $q(w) \in \mathbb{F}$ as an entry. Note that each training example $(w, q(w))$, with $w = (w_1, w_2, \ldots, w_\ell)$, gives us a linear equation

$$w_1v_1 + w_2v_2 + \cdots + w_\ell v_\ell = q(w)$$

for $v = (v_1, v_2, \ldots, v_\ell) \in \mathbb{F}^\ell$. Thus from these K training examples, we obtain a system of K linear equations, denoted as $[W^{(0)}|q^{(0)}]$, and we would like to reduce the task of learning v to that of solving this system of linear equations. However, this system is highly noisy as about $1 - 1/2^m$ fraction of the equations are likely to be wrong, according to (1). We will roughly follow the approach of Gaussian elimination (which works for noiseless systems of linear equations), but will make substantial changes in order to deal with our noisy case.

```
1) For $t$ from 1 to $T$ do
   a) Partition the equations of $[W^{(t-1)}|q^{(t-1)}]$
      into at most $2^{md}$ groups (recall that
      $|\mathbb{F}| = 2^m$) according to their first blocks
      in $W^{(t)}$ (same block value in the same
      group).
   b) Within each group, randomly select an
      equation which we call pivot.
   c) Within each group, subtract the pivot
      from each equation.
   d) Remove the pivots and delete the first
      block from each equation. Let $[W^{(t)}|q^{(t)}]$
      be the resulting system of equations.
```

Fig. 1. FORWARD PHASE.

```
1) Set $V^{(T)} = \mathbb{F}^{(n-k)/m}$, and set $V^{(t)} = \emptyset$ for
   $0 \leq t \leq T-1$.
2) For $t$ from $T-1$ down to 0 do
   (a) For any $z \in \mathbb{F}^d \times V^{(t+1)}$ which is $\delta_t$-good
       for $[W^{(t)}|q^{(t)}]$:
           if $|V^{(t)}| \leq L$
               then include $z$ into $V^{(t)}$,
               else report "error" and halt.
3) Output $V^{(0)}$.
```

Fig. 2. BACKWARD PHASE.

Our algorithm consists of two phases: the forward phase, shown in Fig. 1, and the backward phase, shown in Fig. 2. The forward phase works as follows, which is similar to an approach of Blum *et al.* [5]. Starting from the system $[W^{(0)}|q^{(0)}]$ of linear equations, we use several iterations to produce smaller and smaller systems with fewer and fewer variables, until we have a small enough system which we can afford to solve using brute force. More precisely, we choose the parameters

$$T = \log\sqrt{k/\log k} \text{ and } d = k/(mT),$$

divide each row of $W^{(0)}$ into ℓ/d blocks, with each block containing d elements in $\mathbb{F}$, and proceed in T iterations, as shown in Fig. 1. Note that after iteration t, we have the system $[W^{(t)}|q^{(t)}]$ which has $\ell - dt$ variables and $K^{(t)}$ equations, with

$$\begin{aligned} K^{(t)} &\geq K - t2^{md} \\ &= 2^{c(k/\log k)} - t2^{k/T} \\ &\geq 2^{c(k/\log k)}/2 \\ &= K/2, \end{aligned}$$

for a large enough constant c. The key is to guarantee that the system still contains a good fraction of correct equations. Let

$$\delta_0 = \delta/2 \text{ and } \delta_t = (\delta_{t-1}/2)^2 \text{ for } t \geq 1.$$

A simple induction shows that for $t < T$,

$$\delta_t = \delta^{2^t}/2^{3\cdot 2^t - 2} \geq (\delta/8)^{2^t} \geq 2^{-0.1c(k/\log k)} = K^{-0.1},$$

for a large enough constant c. We say that any $z \in \mathbb{F}^{\ell - dt}$ is *δ_t-good* for the system $[W^{(t)}|q^{(t)}]$ if it satisfies at least $1/2^m + \delta_t$ fraction of equations in the system. Let $v^{(t)} \in \mathbb{F}^{\ell-dt}$ denote v without its first t blocks, and we call the forward phase *good* if for every t, $v^{(t)}$ is δ_t-good for $[W^{(t)}|q^{(t)}]$. Lemma 3 below, which will be proved in Section V-A, guarantees that the forward phase is good with a significant probability.

Lemma 3: The forward phase is good with probability at least $2^{-O(k/\log k)}$.

For the backward phase, we start from the last system $[W^{(T)}|q^{(T)}]$ produced by the forward phase, and work backward on larger and larger systems produced in the forward phase to obtain solutions for more and more variables. More precisely, we go from $t = T - 1$ down to $t = 0$, and while in iteration t, we try to find all possible solutions which extend solutions from iteration $t + 1$ and are δ_t-good for $[W^{(t)}|q^{(t)}]$, as shown in Fig. 2. However, in order to bound the running time, we will stop including the solutions once their number grows beyond the threshold

$$L = 2^{n-k+m+T+2\log(1/\delta_T)} = 2^{n-k+O(k/\log k)}.$$

If this happens, we may fail to include the actual solution v in our final list. Call the backward phase *good* if for every t, the number of such δ_t-good solutions for $[W^{(t)}|q^{(t)}]$ is at most L, or equivalently, it never reports "error." Lemma 4 below, which will be proved in Section V-B, guarantees that the backward phase is indeed good with a high probability.

Lemma 4: The backward phase is not good with probability at most $2^{-\Omega(k)}$.

From Lemma 3 and Lemma 4, the probability that both the forward and backward phases are good is at least

$$2^{-O(k/\log k)} - 2^{-\Omega(k)} = 2^{-O(k/\log k)}.$$

Assuming that both phases are good, a simple induction shows that $v^{(t)} \in V^{(t)}$ for any t and hence $v \in V^{(0)}$. Thus, we have shown that any fixed v satisfying the bound in (1) is contained in the list $V^{(0)}$ of size at most L with probability $2^{-O(k/\log k)}$. We can further reduce the probability of missing this v to $2^{-\omega(n)}$ by repeating the process $2^{O(k/\log k)}$ times, and take the union of the produced lists. Then a union bound shows that some v satisfying (1) is not included in the final output with probability only $o(1)$.

Finally, let us measure the complexity of our algorithm. First, $K \leq 2^{O(k/\log k)}$ training examples are sampled from the distribution $(\mathcal{W}, q(\mathcal{W}))$. Next, each iteration of the forward phase works on a system of at most K equations with at most n variables and runs in time $\mathrm{poly}(K, n)$, and hence the whole forward phase runs in time

$$\begin{aligned} T \cdot \mathrm{poly}(K, n) &= O(\log(k/\log k)) \cdot \mathrm{poly}(K) \\ &\leq 2^{O(k/\log k)}, \end{aligned}$$

since $k \geq \Omega(\log^2 n)$. Then, each iteration of the backward phase runs in time

$$\begin{aligned} &O(2^{md} \cdot L \cdot K) \\ &\leq 2^{O(k/\log k)} \cdot 2^{n-k+O(k/\log k)} \cdot 2^{O(k/\log k)} \\ &\leq 2^{n-k+O(k/\log k)}, \end{aligned}$$

so the whole backward phase runs in time

$$O(\log(k/\log k)) \cdot 2^{n-k+O(k/\log k)} \leq 2^{n-k+O(k/\log k)}.$$

Finally, the process is repeated for $2^{O(k/\log k)}$ times, and thus the total running time is

$$\begin{aligned} &2^{O(k/\log k)} \cdot \left(2^{O(k/\log k)} + 2^{n-k+O(k/\log k)}\right) \\ &\leq 2^{n-k+O(k/\log k)}. \end{aligned}$$

As a result, we have Theorem 4. To complete the proof, it remains to prove Lemma 3 and Lemma 4, which we do next.

A. *Proof of Lemma 3*

First, by a Chernoff bound, we know that $v = v^{(0)}$ satisfies less than $1/2^m + \delta_0$ fraction of equations in $[W^{(0)}|q^{(0)}]$ with probability at most $2^{-\Omega(\delta_0^2 K)} = o(1)$. That is, $v^{(0)}$ is δ_0-good for $[W^{(0)}|q^{(0)}]$ with probability $1 - o(1)$. Next, we need the following lemma.

Lemma 5: In the forward phase, if $v^{(t-1)}$ is δ_{t-1}-good for $[W^{(t-1)}|q^{(t-1)}]$, then $v^{(t)}$ is δ_t-good for $[W^{(t)}|q^{(t)}]$ with probability at least δ_t.

Proof: Let $\tau = \delta_{t-1}$. Assume that $v^{(t-1)}$ is τ-good, so it satisfies at least $\frac{1}{2^m} + \tau$ fraction of equations in the system $[W^{(t-1)}|q^{(t-1)}]$. Partition equations in the system $[W^{(t-1)}|q^{(t-1)}]$ into groups according to their first blocks, as in Step 1(a) of the forward phase. Suppose group i contains p_i fraction of equations in $[W^{(t-1)}|q^{(t-1)}]$ and $v^{(t-1)}$ satisfies $\frac{1}{2^m} + \tau_i$ fraction of equations in the group, for some $\tau_i \in [-\frac{1}{2^m}, 1 - \frac{1}{2^m}]$. Then we have

$$\sum_i p_i \cdot \left(\frac{1}{2^m} + \tau_i\right) \geq \frac{1}{2^m} + \tau. \quad (2)$$

We would like to count the expected fraction of new equations satisfied by $v^{(t-1)}$, where we count equations in their multiplicity. Before doing that, let us first count the fraction with respect to the system obtained before Step 1(d) (before removing pivots). Let us denote a generic equation of the system $[W^{(t-1)}|q^{(t-1)}]$ by $(w^{(t-1)}|q^{(t-1)})$. Consider any group i. For

$u \in \mathbb{F}$, let α_u denote the fraction of equations $(w^{(t-1)}|q^{(t-1)})$ in the group which are off by a value u in the sense that

$$q^{(t-1)} = \langle v^{(t-1)}, w^{(t-1)} \rangle + u.$$

Note that for $v^{(t-1)}$ to satisfy a new equation, which is the difference between two equations, these two involved equations must be off by the same value. Therefore, the expected fraction of new satisfied equations in this group is $\sum_u \alpha_u^2$, which under the constraint $\alpha_0 = \frac{1}{2^m} + \tau_i$ achieves its minimum when $\alpha_u = \frac{1}{2^m} - \frac{\tau_i}{2^m-1}$ for all other $u \neq 0$. Hence, after one iteration, the expected fraction of new equations in group i (before removing pivots) satisfied by $v^{(t-1)}$ is at least

$$\begin{aligned}&\left(\frac{1}{2^m} + \tau_i\right)^2 + (2^m - 1) \cdot \left(\frac{1}{2^m} - \frac{\tau_i}{2^m - 1}\right)^2 \\ &= 2^m \cdot \left(\frac{1}{2^m}\right)^2 + \frac{2-2}{2^m} \cdot \tau_i + \frac{2^m - 1 + 1}{2^m - 1} \cdot \tau_i^2 \\ &\geq \frac{1}{2^m} + \tau_i^2.\end{aligned}$$

Combing all groups together, the expected fraction of satisfied equations overall (before removing the pivots) is at least

$$\begin{aligned}\sum_i p_i \left(\frac{1}{2^m} + \tau_i^2\right) &= \frac{1}{2^m} + \sum_i p_i \tau_i^2 \\ &\geq \frac{1}{2^m} + \left(\sum_i p_i \tau_i\right)^2 \\ &\geq \frac{1}{2^m} + \tau^2,\end{aligned}$$

where the first inequality is due to Jensen inequality, and the second inequality uses the bound $\sum_i p_i \tau_i \geq \tau$ implied by that in (2).

To get the expected fraction of satisfied equations in the final system $[W^{(t)}|q^{(t)}]$, after performing Step 1(d), observe that we only need to discard at most $2^{md} = 2^{O(k/\log k)}$ equations, each with measure $\frac{1}{K^{(t)}} \leq \frac{2}{K}$, so the total discarded measure, denoted as μ, is at most

$$2^{md} \cdot \frac{2}{K} \leq 2^{O(k/\log k)} \cdot 2 \cdot 2^{-c(k/\log k)} \leq \frac{\tau^2}{2},$$

for a large enough constant c. As a result, the expected fraction of equations in $[W^{(t)}|q^{(t)}]$ satisfied by $v^{(t)}$ is at least

$$\begin{aligned}\frac{1}{1-\mu} \cdot \left(\frac{1}{2^m} + \tau^2 - \mu\right) &\geq \frac{1}{2^m} + \tau^2 - \mu \\ &\geq \frac{1}{2^m} + \frac{\tau^2}{2} \\ &= \frac{1}{2^m} + 2\delta_t,\end{aligned}$$

by recalling that $\tau = \delta_{t-1}$ and $\delta_t = (\delta_{t-1}/2)^2$. Finally, by a Markov inequality, we have the lemma. ■

Then by Lemma 5 and an induction, the forward phase is good with probability at least

$$\begin{aligned}(1 - o(1)) \prod_{t=1}^{T} \delta_t &\geq (1 - o(1)) \prod_{t=1}^{T} (\delta/8)^{2^t} \\ &\geq (1 - o(1))(\delta/8)^{2^{T+1}} \\ &\geq 2^{-O(k/\log k)}.\end{aligned}$$

This proves Lemma 3.

B. Proof of Lemma 4

Recall that a solution is δ_t-good for the system $[W^{(t)}|q^{(t)}]$ if it satisfies at least $1/2^m + \delta_t$ fraction of the equations. For any t such that $0 \leq t \leq T-1$, consider the following event:

- $B^{(t)}$: the number of δ_t-good solutions for $[W^{(t)}|q^{(t)}]$ exceeds L.

Thus, our goal is to show that

$$\Pr\left[\bigvee_{t=0}^{T-1} B^{(t)}\right] \leq 2^{-\Omega(k)}.$$

We will prove this by a union bound, so our goal is reduced to bounding each $\Pr[B^{(t)}]$ for $0 \leq t \leq T-1$.

To get a quick idea, let us first consider how to bound $\Pr[B^{(0)}]$. Note that since EXT is a good *statistical* extractor and $\mathcal{W}$ has a high min-entropy, Lemma 1 guarantees that the number of z satisfying the probability bound $\Pr_{w \in \mathcal{W}}[q(w) = \langle z, w \rangle] \geq 1/2^m + \delta_0/2$ is at most L. Any other z is very unlike to be δ_0-good for $[W^{(0)}|q^{(0)}]$ by a Chernoff bound because each row of $W^{(0)}$ is sampled independently from $\mathcal{W}$. Since $B^{(0)}$ happens only when any such z (not satisfying that probability bound) is δ_0-good, a union bound shows that $\Pr[B^{(0)}]$ is indeed small.

Now for $t \geq 1$, to follow this idea to bound $\Pr[B^{(t)}]$, we would also like the distribution of $W^{(t)}$, denoted as $\mathcal{S}^{(t)}$, to have the nice property that each of its rows comes independently from a high min-entropy source. Unfortunately, this is not true in general,[2] and a much more involved analysis is needed. Our approach is to consider the distribution $\mathcal{S}^{(t)}$ conditioned on the choice of pivots in the first t iterations. We call a particular choice of the pivots a *restriction* of the pivots, which includes fixing the indices and the values of some rows as pivots while leaving other rows free. We will show that the distribution $\mathcal{S}^{(t)}$ conditioned on most restrictions is close to a distribution with the nice property. For our purpose here, instead of using the standard definition of "closeness" (which would be measured according to the statistical distance), we consider the following one.

Definition 4: We say that two distributions are γ-close if the probabilities of any event according to the two distributions are within a multiplicative factor of γ from each other.

Observe that one can generate the matrix $W^{(t)}$ in an alternative way by first choosing the pivots in t iterations and then generating the matrices $W^{(1)}, \ldots, W^{(t)}$ consistent with the pivots. Formally, the distribution $\mathcal{S}^{(t)}$ (the distribution of the matrix $W^{(t)}$) can be generated in two passes as follows. In the first pass, we select a restriction of pivots in the first t iterations, denoted as $R^{(1)}, \ldots, R^{(t)}$, by running the forward phase on the matrix $W^{(0)}$ sampled from $\mathcal{W}$ and collecting the pivots, which include the indices and the values of rows as pivots, in each iteration. In

[2] This is true in the simple case considered by [5] that one has $\mathcal{W} = \mathcal{U}_n$ to start with. In this case, for each t, one can easily show that each row of $W^{(t)}$ does come independently from the uniform distribution $\mathcal{U}_{n-tmd}$.

the second pass, we again sample a matrix $W^{(0)}$ from $\mathcal{W}$ and then run the forward phase accordingly for t iterations to derive the matrix $W^{(t)}$, under the condition, denoted as $I_{R^{(1)},\ldots,R^{(t)}}$, that the pivots selected in the t iterations match $R^{(1)},\ldots,R^{(t)}$. Let $\tilde{\mathcal{D}}^{(t)} = (\mathcal{S}^{(t)}|I_{R^{(1)},\ldots,R^{(t)}})$ denote such a conditional distribution of $W^{(t)}$ with respect to the restriction $R^{(1)},\ldots,R^{(t)}$. Now consider the following event about $\tilde{\mathcal{D}}^{(t)}$, over the distribution of $R^{(1)},\ldots,R^{(t)}$ selected in the first pass.

- $E^{(t)}$: the distribution $\tilde{\mathcal{D}}^{(t)}$ is γ_t-close to some distribution $\mathcal{D}^{(t)}$ which has $K^{(t)}$ rows, each coming independently from a distribution $\mathcal{W}^{(t)}$ with $\mathrm{H}_\infty(\mathcal{W}^{(t)}) \geq k - t(md+1)$, for some $\gamma_t \leq K^{2^{md}(2^t-1)} \leq 2^{\sqrt{K}}$.

The following lemma, which will be proved later, shows that when conditioned on $E^{(t)}$, the probability of $B^{(t)}$ is indeed small.

Lemma 6: For any t such that $0 \leq t \leq T-1$,

$$\Pr[B^{(t)} \mid E^{(t)}] \leq 2^{-\Omega(k)}.$$

Next, we would like to show that $E^{(t)}$ happens with high probability. Note that for $t = 0$, the event $E^{(0)}$ always happens because the initial distribution $\tilde{\mathcal{D}}^{(0)}$ has the nice property itself, so we have $\mathcal{D}^{(0)} = \tilde{\mathcal{D}}^{(0)}$ and $\gamma_0 = 1$. For $1 \leq t \leq T-1$, we use induction to show that

$$\begin{aligned}\Pr\left[\neg E^{(t)}\right] &\leq \Pr\left[\neg E^{(t)} \mid E^{(t-1)}\right] + \Pr\left[\neg E^{(t-1)}\right]\\ &\leq \sum_{\tau=1}^{t} \Pr\left[\neg E^{(\tau)} \mid E^{(\tau-1)}\right],\end{aligned}$$

and then we rely on the following lemma, which will be proved later.

Lemma 7: For any t such that $1 \leq t \leq T-1$,

$$\Pr\left[\neg E^{(t)} \mid E^{(t-1)}\right] \leq 2^{-\Omega(K)}.$$

From these two lemmas, we have that for any t such that $1 \leq t \leq T-1$,

$$\begin{aligned}&\Pr\left[B^{(t)}\right]\\ &\leq \Pr\left[B^{(t)} \mid E^{(t)}\right] + \Pr\left[\neg E^{(t)}\right]\\ &\leq \Pr\left[B^{(t)} \mid E^{(t)}\right] + \sum_{\tau=1}^{t} \Pr\left[\neg E^{(\tau)} \mid E^{(\tau-1)}\right]\\ &\leq 2^{-\Omega(k)}.\end{aligned}$$

For $t = 0$, we have

$$\Pr\left[B^{(0)}\right] = \Pr\left[B^{(0)} \mid E^{(0)}\right] \leq 2^{-\Omega(k)}.$$

As a result, a union bound gives us

$$\Pr\left[\bigvee_{t=0}^{T-1} B^{(t)}\right] \leq \sum_{t=0}^{T-1} \Pr\left[B^{(t)}\right] \leq T \cdot 2^{-\Omega(k)} = 2^{-\Omega(k)},$$

which proves Lemma 4. Thus, it remains to prove Lemma 6 and Lemma 7, which we do in the next two subsections.

C. Proof of Lemma 6

Let us first count the number of solutions z such that

$$\Pr_{w \in \mathcal{W}^{(t)}}\left[q^{(t)}(w) = \langle z, w\rangle\right] \geq 1/2^m + \delta_t/2.$$

Let Z denote the set of such z's. Note that $\mathcal{W}^{(t)}$ is a source over $\mathbb{F}^{\ell-td} = \{0,1\}^{(\ell-td)m}$ with $\mathrm{H}_\infty(\mathcal{W}^{(t)}) \geq k - t(md+1)$. Thus by Theorem 1 and Lemma 1, we have

$$\begin{aligned}|Z| &\leq 2^{(\ell-td)m+m+2\log(2/\delta_t)-2-(k-t(md+1))}\\ &= 2^{n-k+m+t+2\log(1/\delta_t)}\\ &\leq L.\end{aligned}$$

This means that for the event $B^{(t)}$ to happen, some $z \notin Z$ must be δ_t-good.

Consider any restriction $R^{(1)},\ldots,R^{(t)}$ such that the event $E^{(t)}$ happens. If we sample the matrix $W^{(t)}$ according to the distribution $\mathcal{D}^{(t)}$, which has each row coming independently from $\mathcal{W}^{(t)}$, then any fixed $z \notin Z$ is δ_t-good (satisfying at least $1/2^m + \delta_t$ fraction of equations in $[W^{(t)}|q^{(t)}]$) with probability at most $2^{-\Omega(\delta_t^2 K^{(t)})}$ by a Chernoff bound, and a union bound shows that

$$\begin{aligned}\Pr_{\mathcal{D}^{(t)}}\left[B^{(t)}\right] &\leq \Pr_{\mathcal{D}^{(t)}}\left[\exists z \notin Z : z \text{ is } \delta_t\text{−good}\right]\\ &\leq 2^n \cdot 2^{-\Omega(\delta_t^2 K^{(t)})}\\ &\leq 2^{-\Omega(K^{0.8})}.\end{aligned}$$

Now if we sample $W^{(t)}$ according to the distribution $\tilde{\mathcal{D}}^{(t)} = (\mathcal{S}^{(t)}|I_{R^{(1)},\ldots,R^{(t)}})$, which is γ_t-close to $\mathcal{D}^{(t)}$ (given that $E^{(t)}$ happens), the probability is only scaled up by a factor γ_t. Thus, we have

$$\begin{aligned}\Pr_{\tilde{\mathcal{D}}^{(t)}}\left[B^{(t)}\right] &\leq \gamma_t \cdot 2^{-\Omega(K^{0.8})}\\ &\leq 2^{\sqrt{K}} \cdot 2^{-\Omega(K^{0.8})}\\ &\leq 2^{-\Omega(k)}.\end{aligned}$$

Since the bound holds for any restriction $R^{(1)},\ldots,R^{(t)}$ such that the event $E^{(t)}$ happens, we have the lemma.

D. Proof of Lemma 7

Let us consider any restriction $R^{(1)},\ldots,R^{(t-1)}$ such that the event $E^{(t-1)}$ happens, and we will show that $E^{(t)}$ happens with high probability, over the selection of $R^{(t)}$. More precisely, the assumption that $E^{(t-1)}$ happens means that we start iteration t from the distribution $\tilde{\mathcal{D}}^{(t-1)}$ which is close to some nice distribution $\mathcal{D}^{(t-1)}$, and our task is to show that with high probability over the selection of $R^{(t)}$, the resulting conditional distribution $\tilde{\mathcal{D}}^{(t)}$ after iteration t is close to another nice distribution $\mathcal{D}^{(t)}$, so that $E^{(t)}$ happens. For this, we need to figure out which of these $R^{(t)}$'s make $E^{(t)}$ happen.

Note that for a restriction $R^{(t)}$, the corresponding distribution $\tilde{\mathcal{D}}^{(t)}$ is obtained by applying Steps 1(c) and 1(d) on the matrix $W^{(t-1)}$ sampled from $\tilde{\mathcal{D}}^{(t-1)}$ under the condition that it is consistent with $R^{(t)}$. The restriction $R^{(t)}$ fixes some $r \leq 2^{md}$ rows

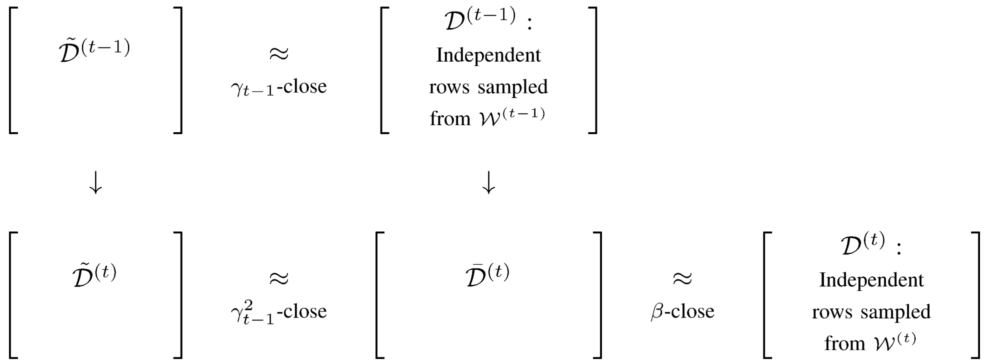


Fig. 3. If $\tilde{\mathcal{D}}^{(t-1)}$ is close to $\mathcal{D}^{(t-1)}$, then $\tilde{\mathcal{D}}^{(t)}$ is close to $\mathcal{D}^{(t)}$, conditioned on $I_{R^{(t)}}$.

of the matrix $W^{(t-1)}$ as pivots and it has the effect on the distribution $\tilde{\mathcal{D}}^{(t-1)}$ that all the rows of $W^{(t-1)}$ must belong to the r groups of those r rows. We would like the effect to be small, and we consider the following event, over the selection of $R^{(t)}$.

- $G^{(t)}$: those elements in the support of $\mathcal{W}^{(t-1)}$ which would belong to those r groups of $R^{(t)}$ when selected as rows of $W^{(t-1)}$ (i.e., those with their first blocks matching one of the first blocks of the r rows in $R^{(t)}$) have a combined measure of $\rho \geq 1/2$ in the distribution $\mathcal{W}^{(t-1)}$.

We will show that if $G^{(t)}$ happens then $E^{(t)}$ happens. For this, let us consider any fixed restriction $R^{(t)}$ such that $G^{(t)}$ happens, and let us use $I_{R^{(t)}}$ to denote the event that the pivots chosen in iteration t match those in $R^{(t)}$. Our approach is illustrated in Fig. 3.

First, let us consider the case of starting iteration t from the nice distribution $\mathcal{D}^{(t-1)}$, instead of $\tilde{\mathcal{D}}^{(t-1)}$, conditioned on $I_{R^{(t)}}$, and let $\bar{\mathcal{D}}^{(t)}$ be the resulting distribution after iteration t. The following claim shows that $\bar{\mathcal{D}}^{(t)}$ is in fact close to a nice distribution.

Claim 1: For some $\beta \leq K^{2^{md}}$, the distribution $\bar{\mathcal{D}}^{(t)}$ is β-close to some nice distribution $\mathcal{D}^{(t)}$ described in the event $E^{(t)}$ (i.e., $\bar{\mathcal{D}}^{(t)}$ has $K^{(t)}$ rows, each coming independently from a distribution $\mathcal{W}^{(t)}$ with $\mathrm{H}_\infty(\mathcal{W}^{(t)}) \geq k - t(md+1)$).

Next, let us go back to the actual situation of starting iteration t from the distribution $\tilde{\mathcal{D}}^{(t-1)}$, instead of $\mathcal{D}^{(t-1)}$ as we did in the above claim. Using the assumption that $\tilde{\mathcal{D}}^{(t-1)}$ is close to $\mathcal{D}^{(t-1)}$, our next claim shows that when we start iteration t from the distribution $\tilde{\mathcal{D}}^{(t-1)}$ conditioned on $I_{R^{(t)}}$, the resulting distribution $\tilde{\mathcal{D}}^{(t)}$ is close to the distribution $\bar{\mathcal{D}}^{(t)}$.

Claim 2: The distribution $\tilde{\mathcal{D}}^{(t)}$ is γ_{t-1}^2-close to the distribution $\bar{\mathcal{D}}^{(t)}$.

From these two claims, we can conclude that $\tilde{\mathcal{D}}^{(t)}$ is γ_t-close to $\mathcal{D}^{(t)}$, for $\gamma_t = \gamma_{t-1}^2\beta \leq \gamma_{t-1}^2 K^{2^{md}}$, which by induction is at most

$$K^{2^{md}(2^t-2)}K^{2^{md}} \leq K^{2^{md}(2^t-1)} \leq 2^{\sqrt{K}}.$$

This implies that for any restriction $R^{(t)}$ such that the event $G^{(t)}$ happens, the event $E^{(t)}$ must happen as well. Therefore, the probability that $E^{(t)}$ does not happen is at most the probability that $G^{(t)}$ does not happen, which we bound by the following claim.

Claim 3: The probability over the selection of $R^{(t)}$ that $G^{(t)}$ does not happen is at most $2^{-\Omega(K)}$.

We have shown that for any restriction $R^{(1)}, \ldots, R^{(t-1)}$ such that the event $E^{(t-1)}$ happens, the probability, over the selection of $R^{(t)}$, that the event $E^{(t)}$ does not happen is at most $2^{-\Omega(K)}$. This implies that $\Pr[\neg E^{(t)} \mid E^{(t-1)}] \leq 2^{-\Omega(K)}$, which proves Lemma 7. Thus, it remains to prove the three claims above, which we do next.

Proof: (of Claim 1)

Recall that we have fixed a restriction $R^{(t)}$ which fixes some r rows as pivots such that the event $G^{(t)}$ happens, and we use $I_{R^{(t)}}$ to denote the event that the pivots selected during iteration t match those in the restriction $R^{(t)}$. In this claim, we consider the situation of starting iteration t from the nice distribution $\mathcal{D}^{(t-1)}$ conditioned on the event $I_{R^{(t)}}$.

First, let us see how the distribution $\mathcal{D}^{(t-1)}$ is affected by the conditioning on $I_{R^{(t)}}$. Consider any fixed matrix M of $K^{(t)} = K^{(t-1)} - r$ rows, insert the rows of $R^{(t)}$ at the proper places to get a fixed matrix $W^{(t-1)}$ of $K^{(t-1)}$ rows, and let us use $I_{W^{(t-1)}}$ to denote the event that a randomly sampled matrix from $\mathcal{D}^{(t-1)}$ equals this matrix $W^{(t-1)}$. If the matrix has a row not in the r groups of $R^{(t)}$, then $\Pr_{\mathcal{D}^{(t-1)}}[I_{W^{(t-1)}} \mid I_{R^{(t)}}] = 0$. Otherwise, $\Pr_{\mathcal{D}^{(t-1)}}[I_{W^{(t-1)}} \mid I_{R^{(t)}}]$ is

$$\frac{\left(\prod_{j=1}^{K^{(t)}} \mathcal{W}^{(t-1)}(M_j)\right) \cdot \left(\prod_{i=1}^{r} \frac{1}{\ell'_i+1}\right)}{\sum_{\ell_1+\cdots+\ell_r=K^{(t)};\ell_i\geq 0} \binom{K^{(t)}}{\ell_1,\cdots,\ell_r} \cdot \left(\prod_{i=1}^{r} \rho_i^{\ell_i}\right) \cdot \left(\prod_{i=1}^{r} \frac{1}{\ell_i+1}\right)},$$

where $\mathcal{W}^{(t-1)}(M_j)$ is the measure of the j'th row of M in $\mathcal{W}^{(t-1)}$, ℓ'_i is the number of rows of M in group i, and ρ_i is the measure of group i in $\mathcal{W}^{(t-1)}$. Note that for some $\alpha_1, \alpha_2 \in [K^{-r}, 1]$, the numerator equals

$$\left(\prod_{j=1}^{K^{(t)}} \mathcal{W}^{(t-1)}(M_j)\right) \cdot \alpha_1,$$

while the denominator equals

$$\begin{aligned}\sum_{\ell_1+\cdots+\ell_r=K^{(t)};\ell_i\geq 0} &\binom{K^{(t)}}{\ell_1,\cdots,\ell_r} \cdot \left(\prod_{i=1}^{r} \rho_i^{\ell_i}\right) \cdot \alpha_2 \\ &= \left(\sum_{i=1}^{r} \rho_i\right)^{K^{(t)}} \cdot \alpha_2 \\ &= \rho^{K^{(t)}} \cdot \alpha_2,\end{aligned}$$

where $\sum_{i=1}^{r} \rho_i = \rho \geq 1/2$ as we assume that the event $G^{(t)}$ happens. As a result, for $\beta = \frac{\alpha_1}{\alpha_2} \in [K^{-r}, K^{r}]$, we have

$$\Pr_{\mathcal{D}^{(t-1)}} [I_{W^{(t-1)}} \mid I_{R^{(t)}}] = \left(\prod_{j=1}^{K^{(t)}} \frac{\mathcal{W}^{(t-1)}(M_j)}{\rho} \right) \cdot \beta.$$

Note that the first factor above can be seen as the probability when we sample each row of the matrix independently according a new distribution $\tilde{\mathcal{W}}^{(t-1)}$, which is the distribution $\mathcal{W}^{(t-1)}$ restricted to those r groups of $R^{(t)}$ and normalized by their measure ρ. Thus, although the conditioning on the event $I_{R^{(t)}}$ may destroy the independence so that we can no longer see each row as coming independently from $\mathcal{W}^{(t-1)}$, we can somehow have the independence restored by considering another distribution $\tilde{\mathcal{W}}^{(t-1)}$ with some distortion factor β. More precisely, we have shown that the distribution $\mathcal{D}^{(t-1)}$ conditioned on the event $I_{R^{(t)}}$ is β-close to a nice distribution, denoted as $\hat{D}^{(t-1)}$, which has each of its remaining row (not fixed by $R^{(t)}$) coming independently from $\tilde{\mathcal{W}}^{(t-1)}$, with

$$\begin{aligned} \mathrm{H}_\infty(\tilde{\mathcal{W}}^{(t-1)}) &\geq \mathrm{H}_\infty(\mathcal{W}^{(t-1)}) - \log(1/\rho) \\ &\geq k - (t-1)(md+1) - 1. \end{aligned}$$

Next, let us see what the resulting distribution $\bar{\mathcal{D}}^{(t)}$ will be when Steps 1(c) and 1(d) are performed on the distribution $\mathcal{D}^{(t-1)}$ conditioned on $I_{R^{(t)}}$. Again, we first consider the case of applying the two steps on the nice distribution $\hat{\mathcal{D}}^{(t-1)}$ instead. When we perform Step 1(c) to subtract from each row its corresponding pivot, which is a fixed value, each resulting row still remains independent from others. However, the distribution of each resulting row is now changed to another distribution which may have a smaller min-entropy than that of $\tilde{\mathcal{W}}^{(t-1)}$, because different initial rows after subtracting their corresponding pivots may result in the same value. Still, the number of such initial rows can be at most 2^{md} since no two such rows can come from the same group, which implies that the min-entropy only decreases by at most md. Then after performing Step 1(d) to remove the pivots and delete the first blocks, the resulting matrix has each row coming independently from some distribution $\mathcal{W}^{(t)}$ with min-entropy at least

$$\mathrm{H}_\infty(\tilde{\mathcal{W}}^{(t-1)}) - md \geq k - t(md+1).$$

That is, after performing Steps 1(c) and 1(d) on the distribution $\hat{\mathcal{D}}^{(t-1)}$, the resulting distribution, denoted as $\mathcal{D}^{(t)}$, satisfies the condition in event $E^{(t)}$. Finally, let us get back to the actual case of starting with the distribution $\mathcal{D}^{(t-1)}$ conditioned on $I_{R^{(t)}}$. Since it is β-close to $\hat{\mathcal{D}}^{(t-1)}$, the resulting distribution $\bar{\mathcal{D}}^{(t)}$ after applying the two steps is β-close to the corresponding resulting distribution $\mathcal{D}^{(t)}$, which proves the claim. ■

Proof: (of Claim 2)

In this claim, we go back to the actual situation of starting iteration t from the distribution $\tilde{\mathcal{D}}^{(t-1)}$, instead of $\mathcal{D}^{(t-1)}$ as we just did. We would like to show that the resulting distribution $\tilde{\mathcal{D}}^{(t)}$ when starting from $\tilde{\mathcal{D}}^{(t-1)}$ is γ_{t-1}^2-close to the distribution $\bar{\mathcal{D}}^{(t)}$ when starting from $\mathcal{D}^{(t-1)}$. For this, it suffices to show that for any event A, the probabilities of $\Pr_{\mathcal{D}^{(t-1)}}[A \mid I_{R^{(t)}}]$ and $\Pr_{\tilde{\mathcal{D}}^{(t-1)}}[A \mid I_{R^{(t)}}]$ are within a multiplicative factor of γ_{t-1}^2. This is true because from the fact that $\mathcal{D}^{(t-1)}$ and $\tilde{\mathcal{D}}^{(t-1)}$ are γ_{t-1}-close, we know that $\Pr_{\mathcal{D}^{(t-1)}}[I_{R^{(t)}}]$ and $\Pr_{\tilde{\mathcal{D}}^{(t-1)}}[I_{R^{(t)}}]$ are within a multiplicative factor of γ_{t-1}, and so are $\Pr_{\mathcal{D}^{(t-1)}}[A \wedge I_{R^{(t)}}]$ and $\Pr_{\tilde{\mathcal{D}}^{(t-1)}}[A \wedge I_{R^{(t)}}]$. ■

Proof: (of Claim 3)

Note that the restriction $R^{(t)}$ can be selected by sampling a matrix $W^{(t-1)}$ according to the distribution $\tilde{\mathcal{D}}^{(t-1)}$ and then applying Steps 1(a) and 1(b) to select the pivots. Thus, the probability that $G^{(t)}$ does not happen is at most the probability that all the $K^{(t-1)}$ rows of $W^{(t-1)}$ lie in some r groups with a combined measure of $\rho \leq 1/2$ in the distribution $\mathcal{W}^{(t-1)}$.

Again, let us first consider the case of sampling $W^{(t-1)}$ according to the distribution $\mathcal{D}^{(t-1)}$, instead of $\tilde{\mathcal{D}}^{(t-1)}$. Note that there are at most $2^{2^{md}}$ ways of choosing the r groups with a combined measure of $\rho \leq 1/2$ in $\mathcal{W}^{(t-1)}$, and the probability that all the $K^{(t-1)} \geq K/2$ independent rows lie in any particular choice of such r groups is at most $(1/2)^{K/2}$. Then a union bound shows that the probability of having $\rho \leq 1/2$ is at most

$$2^{2^{md}} \cdot (1/2)^{K/2} \leq 2^{-\Omega(K)}.$$

Next, let us go back to actual case of sampling $W^{(t-1)}$ according to the distribution $\tilde{\mathcal{D}}^{(t-1)}$. Note that the probability of having $\rho \leq 1/2$ according to $\tilde{\mathcal{D}}^{(t-1)}$ can only be larger than that according to $\mathcal{D}^{(t-1)}$ by at most a factor of γ_{t-1}, and hence it is still at most

$$\gamma_{t-1} \cdot 2^{-\Omega(K)} \leq 2^{\sqrt{K}} \cdot 2^{-\Omega(K)} \leq 2^{-\Omega(K)}.$$

■

References

[1] N. Alon and J. Spencer, *The Probabilistic Method.* : John Wiley, 1992.

[2] B. Barak, R. Impagliazzo, and A. Wigderson, “Extracting randomness using few independent sources,” *SIAM J. Comput.*, vol. 36, no. 4, pp. 1095–1118, 2006.

[3] B. Barak, G. Kindler, R. Shaltiel, B. Sudakov, and A. Wigderson, “Simulating independence: New constructions of condensers, Ramsey graphs, dispersers, and extractors,” in *Proc. 37th Annu. ACM Symp. on Theory of Computing (STOC’05)*, 2005, pp. 1–10.

[4] B. Barak, R. Shaltiel, and A. Wigderson, “Computational analogues of entropy,” in *Proc. 7th Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM’03)*, 2003, pp. 200–215.

[5] A. Blum, A. Kalai, and H. Wasserman, “Noise-tolerant learning, the parity problem, and the statistical query model,” *J. ACM*, vol. 50, no. 4, pp. 506–519, 2003.

[6] J. Bourgain, “More on the sum-product phenomenon in prime fields and its applications,” *Int. J. Numb. Theory*, vol. 1, no. 1, pp. 1–32, 2005.

[7] B. Chor and O. Goldreich, “Unbiased bits from sources of weak randomness and probabilistic communication complexity,” *SIAM J. Comput.*, vol. 17, no. 2, pp. 230–261, Apr. 1988.

[8] B. Chor, O. Goldreich, J. Håstad, J. Friedman, S. Rudich, and R. Smolensky, “The bit extraction problem of t-resilient functions,” in *Proc. 26th Annu. IEEE Symp. Found. Comput. Sci. (FOCS’85)*, pp. 396–407.

[9] V. Feldman, P. Gopalan, S. Khot, and A. Ponnuswami, “On agnostic learning of parities, monomials, and halfspaces,” *SIAM J. Comput.*, vol. 39, no. 2, pp. 606–645, 2009.

[10] A. Gabizon, R. Raz, and R. Shaltiel, “Deterministic extractors for bit-fixing sources by obtaining an independent seed,” *SIAM J. Comput.*, vol. 36, no. 4, pp. 1072–1094, 2006.

[11] O. Goldreich and L. A. Levin, “A hard-core predicate for all one-way functions,” in *Proc. 21st Annu. ACM Symp. Theory Comput. (STOC’89)*, pp. 25–32.

[12] O. Goldreich, R. Rubinfeld, and M. Sudan, "Learning polynomials with queries: The highly noisy case," *SIAM J. Discrete Math.*, vol. 13, no. 4, pp. 535–570, 2000.

[13] V. Guruswami, C. Umans, and S. Vadhan, "Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes," *J. ACM*, vol. 56, no. 4, 2009, Art. 20.

[14] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby, "A pseudorandom generator from any one-way function," *SIAM J. Comput.*, vol. 28, no. 4, pp. 1364–1396, 1999.

[15] C.-Y. Hsiao, C.-J. Lu, and L. Reyzin, "Conditional computational entropy, or toward separating pseudoentropy from compressibility," in *Proc. Adv. Cryptol.—EUROCRYPT*, 2007, pp. 169–186.

[16] A. Kalai, Y. Mansour, and E. Verbin, "On agnostic boosting and parity learning," in *Proc. 40th Annu. ACM Symp. Theory Comput. (STOC'08)*, pp. 629–638.

[17] J. Kamp, A. Rao, S. Vadhan, and D. Zuckerman, "Deterministic extractors for small-space sources," in *Proc. 38th Annu. ACM Symp. Theory Comput. (STOC'06)*, pp. 691–700.

[18] J. Kamp and D. Zuckerman, "Deterministic extractors for bit-fixing sources and exposure-resilient cryptography," *SIAM J. Comput.*, vol. 36, no. 5, pp. 1231–1247, 2007.

[19] C.-J. Lee, C.-J. Lu, and S.-C. Tsai, "Deterministic extractors for independent-symbol sources," in *Proc. 33rd Int. Colloq. Automata, Lang., Program. (ICALP 2006)*, pp. 84–95.

[20] C.-J. Lee, C.-J. Lu, S.-C. Tsai, and W.-G. Tzeng, "Extracting randomness from multiple independent sources," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 2224–2227, Jun. 2005.

[21] C.-J. Lu, O. Reingold, S. Vadhan, and A. Wigderson, "Extractors: Optimal up to constant factors," in *Proc. 35th Annu. ACM Symp. Theory Comput. (STOC'03)*, pp. 602–611.

[22] N. Nisan and D. Zuckerman, "Randomness is linear in space," *J. Comput. Syst. Sci.*, vol. 52, no. 1, pp. 43–52, 1996.

[23] A. Rao, "Extractors for a constant number of polynomially small min-entropy independent sources," *SIAM J. Comput.*, vol. 39, no. 1, pp. 168–194, 2009.

[24] R. Raz, "Extractors with weak random seeds," in *Proc. 37th Annu. ACM Symp. Theory Comput. (STOC'05)*, pp. 11–20.

[25] R. Shaltiel, "Recent developments in explicit constructions of extractors," *Bull. Eur. Assoc. Theor. Comput. Sci.*, vol. 77, pp. 67–95, 2002.

[26] A. Ta-Shma and D. Zuckerman, "Extractor codes," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3015–3025, Dec. 2004.

[27] L. Trevisan, "Extractors and pseudorandom generators," *J. ACM*, vol. 48, no. 4, pp. 860–879, 2001.

[28] L. Trevisan and S. Vadhan, "Extracting randomness from samplable distributions," in *Proc. 41st Annu. IEEE Symp. Found. Comput. Sci. (FOCS'00)*, pp. 32–42.

[29] A. C. Yao, "Theory and applications of trapdoor functions," in *Proc. 23rd Annu. IEEE Symp. Found. Comput. Sci. (FOCS'82)*, pp. 80–91.

[30] D. Zuckerman, "General weak random sources," in *Proc. 31st Annu. IEEE Symp. Found. Comput. Sci. (FOCS'90)*, pp. 534–543.

Chia-Jung Lee received the B.S. degree from the National Taiwan Normal University, Taipei, Taiwan, in 2000, and the Ph.D. degree in computer science from the National Chiao-Tung University, Hsinchu, Taiwan, in 2010. She is now doing postdoctoral research at the Institute of Information Science, Academia Sinica, Taipei, Taiwan. Her research interests are randomness in computation, cryptography, and theoretical computer science.

Chi-Jen Lu received his B.S. and M.S. degrees from National Taiwan University, Taiwan, in 1988 and 1990 respectively, and his Ph.D. degree from University of Massachusetts at Amherst, USA, in 1999, all in computer science. He is currently a research fellow in the Institute of Information Science, Academia Sinica, Taiwan. His research interests include randomness in computation, computational complexity, cryptography, game theory, and machine learning.

Shi-Chun Tsai (M'06) received his B.S. and M.S. degrees in computer science and information engineering from National Taiwan University, Taiwan, in 1984 and 1988, respectively, and the Ph.D. degree in computer science from the University of Chicago, USA, in 1996. During 1993–1996, he served as a Lecturer in the Computer Science Department, University of Chicago. During 1996–2001, he was Associate Professor of Information Management Department, and Computer Science and Information Engineering Department, National Chi Nan University, Taiwan. He has been with the Department of Computer Science, National Chiao Tung University, Taiwan since 2001, and was promoted to full Professor in 2007. He is currently serving as the Director of the Information Technology Service Center of National Chiao Tung University. His research interests include computational complexity, algorithms, coding theory, and combinatorics.

Computational Randomness from Generalized Hardcore Sets

Chia-Jung Lee[1], Chi-Jen Lu[1], and Shi-Chun Tsai[2]

[1] Institute of Information Science, Academia Sinica, Taipei, Taiwan
{leecj,cjlu}@iis.sinica.edu.tw
[2] Department of Computer Science, National Chiao-Tung University,
Hsinchu, Taiwan
sctsai@csie.nctu.edu.tw

Abstract. The seminal hardcore lemma of Impagliazzo states that for any mildly-hard Boolean function f, there is a subset of input, called the hardcore set, on which the function is extremely hard, almost as hard as a random Boolean function. This implies that the output distribution of f given a random input looks like a distribution with some statistical randomness. Can we have something similar for hard functions with several output bits? Can we say that the output distribution of such a general function given a random input looks like a distribution containing several bits of randomness? If so, one can simply apply any statistical extractor to extract computational randomness from the output of f. However, the conventional wisdom tells us to apply extractors with some additional *reconstruction* property, instead of just any extractor. Does this mean that there is no analogous hardcore lemma for general functions?

We show that a general hard function does indeed have some kind of hardcore set, but it comes with the price of a security loss which is proportional to the number of output values. More precisely, consider a hard function $f : \{0,1\}^n \to [V] = \{1, \ldots, V\}$ such that any circuit of size s can only compute f correctly on at most $\frac{1}{L}(1-\gamma)$ fraction of inputs, for some $L \in [1, V-1]$ and $\gamma \in (0,1)$. Then we show that for some $I \subseteq [V]$ with $|I| = L+1$, there exists a hardcore set $H_I \subseteq f^{-1}(I)$ with density $\gamma/\binom{V}{L+1}$ such that any circuit of some size s' can only compute f correctly on at most $\frac{1+\varepsilon}{L+1}$ fraction of inputs in H_I. Here, s' is smaller than s by some poly$(V, 1/\varepsilon, \log(1/\gamma))$ factor, which results in a security loss of such a factor. We show that it is basically impossible to guarantee a much larger hardcore set or a much smaller security loss. Finally, we show how our hardcore lemma can be used for extracting computational randomness from general hard functions.

1 Introduction

Impagliazzo's hardcore lemma [9] is a fundamental result in complexity theory which states that any mildly-hard function has a subset of inputs on which it is extremely hard. More precisely, consider a function $f : \{0,1\}^n \to \{0,1\}$ such that any circuit of size s disagrees with f on at least δ fraction of inputs, and

O. Owe, M. Steffen and J.A. Telle (Eds.): FCT 2011, LNCS 6914, pp. 78–89, 2011.

we call such a function (δ, s)-hard where the parameter δ is called the hardness of f. Then the hardcore lemma asserts that there exists a subset $H \subseteq \{0,1\}^n$ of density δ such that any circuit of size s' must disagree with f on at least $\frac{1-\varepsilon}{2}$ fraction of inputs from H, for some s' slightly smaller than s. This means that given a random input x in H, although the value of $f(x)$ is fixed and thus has no randomness at all in a statistical sense, it still looks like a random bit to small circuits. Because of this nice property, the hardcore lemma has become an important tool in the study of pseudo-randomness. For example, it was used in [9] for an alternative proof of Yao's XOR lemma [20], used in [17] for constructing a pseudo-random generator directly from a mildly-hard function without going through the XOR lemma, and more recently used in [15,18,19,6] for amplifying hardness of functions in NP. The parameters of the hardcore lemma were later improved by [10,7,2].

Note that Impagliazzo's hardcore lemma works for *Boolean* functions. It says that the output of a hard function given a random input looks like a random bit and thus contains statistical randomness, when the input falls in the hardcore set. When using the lemma, the hard function is usually evaluated at several inputs in order to obtain several output bits, which together can be argued to contain some sufficient amount of randomness. Usually, the amount of randomness in a distribution is measured by its min-entropy, where a distribution has min-entropy at least k if every element occurs with probability at most 2^{-k}. Then from a distribution with some min-entropy, one applies a so-called randomness extractor [21,14] to extract a distribution which looks almost random.

On the other hand, there are natural functions with many output bits which are believed to be hard, such as factoring and discrete logarithm, and one may be able to extract several bits at once from one output value. This is also related to the problem of extracting randomness from sources with computational randomness, studied in [3,8,12]. One may wonder if there is an analogous hardcore lemma for a general non-Boolean function, which can guarantee that the output distribution given a random input will look like one with some min-entropy, hopefully much larger than one. For example, assume that a one-way permutation $g : \{0,1\}^n \to \{0,1\}^n$ exists, whose inverse function $f = g^{-1}$ is hard to compute by small (say, polynomial-size) circuits. Then, if one could show that the distribution of $x = f(y)$ given a random y looks like having some min-entropy to small circuits, one could simply apply *any* extractor on x. However, the conventional wisdom does not suggest so and the following counter example seems to be known as a folklore. Given an efficiently-computable extractor E and a one-way permutation g, the function $\textsc{Ext}$ defined as $\textsc{Ext}(x, u) = \mathrm{E}(g(x), u)$ is still an extractor, but its output can be easily computed (and hence does not look random at all) given $y = g(x)$ and u. To extract such computational randomness, previous works all resorted to extractors with some *reconstruction* property, which roughly corresponds to error correcting codes with efficient decoders (see, e.g., [16] for a definition).

Does this mean that there is no analogous hardcore lemma for general functions? If we consider a hard function $f : \{0,1\}^n \to \{0,1\}^2$ with two, instead of one, output bits, it may be hard to believe that we can no longer have any kind of hardcore lemma for it. But can we guarantee the existence of a hardcore set H such that $f(x)$, for a random $x \in H$, looks like a random value in $\{0,1\}^2$? The answer is no in general because f may in fact have at most three possible output values, so we have to settle for something weaker. One approach is to see each output bit of a (δ, s)-hard function $f : \{0,1\}^n \to \{0,1\}^d$ as a Boolean function, so some of these d Boolean functions must have hardness $\Omega(\delta/d)$ and they have Boolean hardcore sets (with two output values) of density $\Omega(\delta/d)$ using Impagliazzo's lemma. Unfortunately, this only gives a very weak result because even if f is extremely hard, with δ close to $1 - 2^{-d}$, one may still only be able to guarantee one bit of randomness in the output of f if those Boolean hardcore sets are disjoint.

We are looking for something stronger, in which more bits of randomness can be guaranteed. We consider a general (δ, s)-hard function f of the form $f : \{0,1\}^n \to [V] = \{1, \ldots, V\}$. We discover that a good way to see its hardness is to express it in the form of $\delta = 1 - \frac{1}{L}(1 - \gamma)$, for some $L \in [1, V-1]$ and $\gamma \in (0,1)$, and we obtain the following results.

First, we show that any function with such hardness has a hardcore set with $L+1$ output values. More precisely, we show that for such a hard function f, there exist some $I \subseteq [V]$ with $|I| = L+1$ and some $H_I \subseteq f^{-1}(I)$ of density $|H_I|/2^n \geq \gamma/\binom{V}{L+1}$ such that any circuit of size s' can only compute f correctly on $\frac{1+\varepsilon}{L+1}$ fraction of the inputs in H_I, where s' is smaller than s by a factor of $\mathrm{poly}(V, 1/\varepsilon, \log(1/\gamma))$. Let us call such a set H_I an (I, ε, s') hardcore set, and let us take a close look at what our result says as L varies. At one end of the spectrum with $L = V-1$, our result guarantees the existence of a hardcore set H_I, with $I = [V]$, such that f restricting to the set H_I has almost the largest possible hardness and it looks like a random function from H_I to $[V]$. Note that when $V = 2$ (and $L = 1$), we have Impagliazzo's hardcore lemma as our special case. As L becomes smaller, the hardness decreases, and it is no longer possible to always have a hardcore set with V output values. Nevertheless, our result shows that one can still have a hardcore set H_I with $|I| = L+1$ output values, such that f restricting to H_I looks like a random function from H_I to I.

Notice that in our first result, we can only guarantee a hardcore set of density $\gamma/\binom{V}{L+1}$, and one may wonder if it is possible to guarantee a larger one. Our second result shows that this is basically impossible. More precisely, we show the existence of a (δ, s)-hard function $f : \{0,1\}^n \to [V]$, with $\delta = 1 - \frac{1}{L}(1-\gamma)$ and $s \geq \mathrm{poly}(\gamma, 1/L, 2^n)$, which has no (I, ε, s') hardcore set of density $4(L+1)\gamma/\binom{V}{L+1}$ for any $I \subseteq [V]$ with $|I| = L+1$, where $s' = \mathrm{poly}(n)$. Note that the density achieved by out first result and that ruled out by our second result are off by an $O(L)$ factor, and we believe that the bound of our second result may be improved. On the other hand, our second result is strong in the sense that even when we start from a function which is hard against very large circuits of exponential size, it is still impossible to have a hardcore set of a small density against small circuits of polynomial size.

With a small hardcore set, one can only say that the output of a hard function f looks somewhat random when the input falls into that small set. This alone is not good enough for the purpose of randomness extraction because the vast majority of inputs are outside of the hardcore set and may contribute a large error. Our next result shows that in fact we can have not just one but a collection of disjoint hardcore sets, and they together cover all but a small fraction of the inputs, which implies that the output of f looks somewhat random for most input. More precisely, we show that for a (δ, s)-hard function, with $\delta \geq 1 - 2^{-k}$, its output distribution given a random input looks close, within some distance ε, to a distribution with min-entropy $\Omega(k)$, by circuits of size $s' = s/\text{poly}(V, 1/\varepsilon)$. This implies that we can simply apply any seeded statistical extractor to extract computational randomness from the output of f as long as s is large (say, super-polynomial) or V is small (say, polynomial). This also works for seedless multi-source extractors, and in particular, it fits nicely with the setting of independent-symbol sources studied in [12] in which each symbol is considered to come from a small set. Therefore, we can generalize the result of [12] from a statistical setting to a computational one: given multiple independent sources, over a small set of symbols, which look slightly random to polynomial-size circuits but may have no min-entropy at all, the statistical extractor there can be used to produce an output which looks almost random to polynomial-size circuits.

Note that in our hardcore set result, there is a security loss of some factor $\text{poly}(V, 1/\varepsilon)$ in circuit size. That is, starting from a function which is hard against circuits of size s, we can only guarantee the hardness of a hardcore set against circuits of size s', with s' smaller than s by that factor. Consequently, with $s = \text{poly}(n)$, we can only extract randomness from a function with $V \leq \text{poly}(n)$ (or equivalently, with $O(\log n)$ output bits). One may wonder if such a security loss of circuit size can be avoided. Our final result shows that this is basically impossible, if the proof is done in a certain black box way. Here, we use the notion of black-box proofs for hardcore sets introduced in [13]. Informally speaking, a black-box proof is realized by an oracle algorithm R such that for any function f and any collection G of circuits, if G breaks the hardcore set condition, then R breaks the hardness of f by using G only as an oracle. In this black-box model, we show that any algorithm R must make at least $q = \Omega((Vk/\varepsilon^2)\log(1/\delta))$ queries in order to show the existence of a hardcore set with k output values. This translates to a loss of a q factor in circuit size, because the resulting circuit of R^G is larger than those in G by this factor. This explains the need of using *reconstructive* extractors, instead of just any extractors, on the input of a one-way permutation discussed before, since there we have a large $V = 2^n$. Finally, we would like to clarify a potential confusion with the security loss of using *reconstructive* extractors in previous works. When applying reconstructive extractors on the output of a hard function f, previous results also suffered some loss of circuit size in the same sense: the outputs of extractors only look random to smaller circuits compared to those which the hardness of f is measured against. However, the loss is in terms of the output length m of extractors, instead of the output length of f. More precisely, the loss factor is $\mathrm{poly}(2^m)$, which again limits us to extracting $O(\log n)$ bits when f is hard against circuits of size $s = \mathrm{poly}(n)$.

2 Preliminaries

For $n \in \mathbb{N}$, let $[n]$ denote the set $\{1, \ldots, n\}$, and let $\mathcal{U}_n$ denote the uniform distribution over $\{0,1\}^n$. For a set X, we let $|X|$ denote the number of elements in X, and for a subset $S \subseteq X$, we say that S has density $|S|/|X|$ in X. For a set X and an integer $n \in \mathbb{N}$, we use the notation $\binom{X}{n}$ to denote the collection of subsets $S \subseteq X$ such that $|S| = n$. When we sample from a finite set, the default distribution is the uniform one. All logarithms used in this paper will have base two. Let $\mathsf{SIZE}(s)$ be the class of functions computable by circuits of size s. We measure the hardness of computing a function in the following way.

Definition 1. *A function f is (δ, s)-hard if any circuit in $\mathsf{SIZE}(s)$ must fail to compute f correctly for at least a δ fraction of inputs.*

Impagliazzo [9] considered Boolean functions and show that any hard function must have a hardcore set such that the function restricted to the hardcore set is extremely hard. In this paper, we consider general functions of the form $f : X \to [V]$, for some input set X and for the output set $[V]$ with integer $V \geq 2$. For such general functions, we introduce our notion of generalized hardcore sets as follows.

Definition 2. *For a function $f : X \to [V]$ and some $I \subseteq [V]$, we say that a subset of inputs $H \subseteq f^{-1}(I)$ is an (I, ε, s) hardcore set if for any circuit $C \in \mathsf{SIZE}(s)$, $\Pr_{x \in H}[C(x) = f(x)] \leq (1+\varepsilon)/|I|$. We say that such an H is a hardcore set with $|I|$ output values.*

Note that $f(x) \in I$ for any $x \in H$, so the above probability bound says that f restricted to H looks like a random function from H to I. We say that a distribution looks like another one if there is no distinguisher for them, defined as follows.

Definition 3. *A function $D : X \to \{0,1\}$ is an ε-distinguisher for two distributions $\mathcal{X}$ and $\mathcal{Y}$ over X if $|\Pr[D(\mathcal{X}) = 1] - \Pr[D(\mathcal{Y}) = 1]| \geq \varepsilon$, and we call such a function D an (ε, s)-distinguisher if $D \in \mathsf{SIZE}(s)$.*

We measure the amount of randomness in a distribution $\mathcal{X}$ by its *min-entropy*, and we say that $\mathcal{X}$ has min-entropy at least k, denoted by $\mathrm{H}_\infty(\mathcal{X}) \geq k$, if for any x, $\Pr[\mathcal{X} = x] \leq 2^{-k}$. From a source which is only weakly random (with some min-entropy), we would like to have a procedure called an extractor to extract a distribution which is almost random. When trying to extract randomness from a single source, one usually needs an additional short seed, and such an extractor is called a seeded one, which is defined as follows.

Definition 4. *A function $\textsc{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is called a (seeded) (k, ε)-extractor if for any distribution $\mathcal{X}$ over $\{0,1\}^n$ with $\mathrm{H}_\infty(\mathcal{X}) \geq k$, there is no ε-distinguisher for the distributions $\textsc{Ext}(\mathcal{X}, \mathcal{U}_d)$ and $\mathcal{U}_m$.*

When there are at least two independent sources which are weakly random, it becomes possible to have a seedless extractor, which is defined as follows.

Definition 5. *A function* $\textsc{Ext} : (\{0,1\}^n)^t \to \{0,1\}^m$ *is called a (seedless) t-source (k, ε)-extractor if for any t independent distributions $\mathcal{X}_1, \ldots, \mathcal{X}_t$ over $\{0,1\}^n$ with $\sum_{i\in[t]} \mathrm{H}_\infty(\mathcal{X}_i) \geq k$, there is no ε-distinguisher for the distributions* $\textsc{Ext}(\mathcal{X}_1, \ldots, \mathcal{X}_t)$ *and* $\mathcal{U}_m$.

3 Generalized Hardcore Set

In this section, we generalize Impagliazzo's hardcore lemma [9] from the case of Boolean functions to the case of general functions. More precisely, we have the following.

Lemma 1. *Let $f : X \to [V]$ be a (δ, s)-hard function, with $\delta \geq 1 - \frac{1}{L}(1-\gamma)$ for some $\gamma \in (0,1)$ and some integer $L \in [V-1]$. Then for any $\varepsilon > 0$, there exist $s' = s/\mathrm{poly}(V, 1/\varepsilon, \log(1/\gamma))$ and $I \in \binom{[V]}{L+1}$ such that f has an (I, ε, s')-hardcore set H_I of density $|H_I|/|X| \geq \gamma/\binom{V}{L+1}$.*

To prepare for the proof of Lemma 1, let us first recall Nisan's proof of Impagliazzo's hardcore lemma (for Boolean functions) described in [9]. The proof is by contradiction, which starts by assuming that a (ρ, s)-hard function $f : X \to I$, with $|I| = 2$, has no hardcore set of density ρ. Then the key step there is to use the min-max theorem of von Neumann to show the existence of a subset of inputs $T \subseteq X$ of density less than ρ and a collection of circuits $A_I \subseteq \mathsf{SIZE}(s')$ with $|A_I| \leq O((1/\varepsilon^2)\log(1/\rho))$ such that for any $x \notin T$,

$$\Pr_{A\in A_I}[A(x) = f(x)] > \frac{1}{2}.$$

Then by letting C be the circuit computing the majority of those circuits in A_I, one has $C(x) = f(x)$ for every $x \notin T$, which contradicts the fact that f is (ρ, s)-hard.

We would like to extend this idea to a general function $f : X \to [V]$, with $V \geq 3$. First, it is straightforward to verify that a similar argument using the min-max theorem can also prove the following lemma.

Lemma 2. *Suppose $f : X \to [V]$ does not have an (I, ε, s')-hardcore set of density ρ in X, for some $I \subseteq [V]$. Then there exist a subset of inputs $T_I \subseteq f^{-1}(I)$ of density less than ρ in X and a collection of circuits $A_I \subseteq \mathsf{SIZE}(s')$ with $|A_I| \leq O((1/\varepsilon^2)\log(1/\rho))$ such that for any $x \in f^{-1}(I) \setminus T_I$,*

$$\Pr_{A\in A_I}[A(x) = f(x)] > \frac{1}{|I|}.$$

However, unlike the Boolean case, it is not clear how to construct a circuit C to approximate f from these collections of circuits. This is because for an input

x with $\Pr_{A\in A_I}[A(x)=f(x)] > \frac{1}{|I|}$, it is still possible that the majority value of $A(x)$, for $A \in A_I$, differs from $f(x)$. Moreover, given an input x, we do not know which subset I contains $f(x)$, so we do not even know which collection A_I of circuits might help. A more careful analysis is needed, and now we proceed to prove Lemma 1.

Proof. Assume for the sake of contradiction that there is no (I, ε, s')-hardcore set of density $\gamma/\binom{V}{L+1}$ in X for any $I \in \binom{[V]}{L+1}$. Then we know from Lemma 2 that for any $I \in \binom{[V]}{L+1}$, there exist a collection $A_I \subseteq \mathsf{SIZE}(s')$ with $|A_I| \leq O((1/\varepsilon^2)\log(\binom{V}{L+1}/\gamma))$ and a subset T_I of inputs with density less than $\gamma/\binom{V}{L+1}$ in X such that for any $x \in f^{-1}(I) \setminus T_I$, $\Pr_{A\in A_I}[A(x)=f(x)] > \frac{1}{L+1}$. Let T be the union of all such T_I's, and we have $\Pr_{x\in X}[x \in T] < \binom{V}{L+1} \cdot \gamma/\binom{V}{L+1} = \gamma$. Note that for any $x \notin T$, we can rule out the value v as a candidate for $f(x)$ if v is contained in some $I \in \binom{[V]}{L+1}$ such that the following condition holds:

$$\Pr_{A\in A_I}[A(x)=v] \leq \frac{1}{L+1}. \tag{1}$$

This suggests the randomized algorithm R described in Figure 1, which tries to compute $f(x)$ by ruling out the candidates one by one.

```
1. Let $Q = [V]$.
2. While $|Q| \geq L+1$ do the following:
   (a) Choose any $I \subseteq Q$ with $|I| = L+1$.
   (b) Delete from $Q$ any $v \in I$ such that the condition (1) holds.
3. Output a random element in $Q$.
```

Fig. 1. The randomized algorithm R

Observe that each iteration of the while loop in algorithm R has at least one v deleted from Q, because it is impossible that all the $L+1$ outcomes have probability more than $\frac{1}{L+1}$. Thus, R exits the while loop after at most V iterations, and for any input $x \notin T$, the value $f(x)$ remains in the final Q, with $|Q| \leq L$, which implies that R outputs $f(x)$ correctly with probability at least $\frac{1}{L}$. By an averaging argument, one can fix the randomness of R to obtain a deterministic circuit C such that $C(x) = f(x)$ for at least $\frac{1}{L}$ fraction of $x \notin T$. As a result, we have

$$\Pr_{x\in X}[C(x)=f(x)] \geq \Pr_{x\in X}[x \notin T] \cdot \Pr_{x\in X}[C(x)=f(x) \mid x \notin T] > (1-\gamma)\cdot\frac{1}{L} \geq 1-\delta.$$

On the other hand, the size of the circuit C is at most

$$\mathrm{poly}(V) \cdot O\left((1/\varepsilon^2)\log\left(\binom{V}{L+1}/\gamma\right)\right) \cdot s' \leq s,$$

for some $s' = s/\mathrm{poly}(V, 1/\varepsilon, \log(1/\gamma))$, which contradicts the hardness condition of f. This implies that the assumption of no hardcore set at the beginning is false, which proves Lemma 1. □

4 Density of Hardcore Sets

For the generalized hardcore set lemma in Section 3, one may wonder whether it is possible to guarantee the existence of a much larger hardcore set. In this section, we show that this is basically impossible. Formally, we have the following.

Theorem 1. *For any $\delta = 1 - \frac{1}{L}(1-\gamma)$, with $\gamma \in (0, 1/2(L+1))$ and $L \le V-1$, there is a (δ, s)-hard function $f : \{0,1\}^n \to [V]$, for some $s \ge \mathrm{poly}(\gamma, 1/L, 2^n)$, such that the following condition holds:*

- *For any $I \in \binom{[V]}{L+1}$ and $\varepsilon < \frac{1}{2L}$, there exists some $s' \le \mathrm{poly}(n)$ such that f has no (I, ε, s')-hardcore set of density $4(L+1)\gamma/\binom{V}{L+1}$ in $\{0,1\}^n$.*

Note that the theorem says that even for a function which is hard against very large circuits of exponential size, one can only guarantee a hardcore set of a small density against small circuits of polynomial size. However, there is a gap of a $4(L+1)$ factor between the density of a hardcore set ruled out by Theorem 1 and the density achievable by our Lemma 1.

Proof. We show the existence of such a function f by a probabilistic method. Let T denote the first $2(L+1)\gamma$ fraction of the input space $\{0,1\}^n$, and let us divide T into $\binom{V}{L+1}$ disjoint parts of equal size (assuming for simplicity of presentation that T can be divided evenly), denoted by T_I, for $I \in \binom{[V]}{L+1}$. Then we choose the function $f : \{0,1\}^n \to [V]$ randomly in the way such that independently for each input x,

$$f(x) = \begin{cases} \text{a random value in } I, & \text{if } x \in T_I \text{ for some } I \in \binom{[V]}{L+1}; \\ \text{a random value in } [L], & \text{if } x \notin T. \end{cases}$$

We need the following lemma; the proof is by a standard probabilistic argument and is omitted here due to the page limit.

Lemma 3. $\Pr_f$ [*f is not (δ, s)-hard*] < 1, *for some $s = \mathrm{poly}(\gamma, 1/L, 2^n)$.*

This lemma implies the existence of a function f which is (δ, s)-hard, and let us fix one such f. It remains to show that this f satisfies the condition of the theorem. For any $I \in \binom{[V]}{L+1}$ and any $H \subseteq f^{-1}(I)$ of density $4(L+1)\gamma/\binom{V}{L+1}$ in $\{0,1\}^n$, consider the algorithm A which outputs a random value in $I \cap J$ when $x \in T_J$ for some $J \in \binom{[V]}{L+1}$, and outputs a random value in $[L]$ when $x \notin T$. Then the probability, over $x \in H$ and the randomness of A, that $A(x) = f(x)$ is at least

$$\Pr_{x \in H}[x \in T_I] \cdot \frac{1}{L+1} + \Pr_{x \in H}[x \notin T_I] \cdot \frac{1}{L} = \frac{1}{L+1} + \Pr_{x \in H}[x \notin T_I] \cdot \left(\frac{1}{L} - \frac{1}{L+1}\right)$$

which is at least $\frac{1}{L+1} + \frac{1}{2} \cdot \frac{1}{L(L+1)} > \frac{1+\varepsilon}{L+1}$ for any $\varepsilon < \frac{1}{2L}$. This means that there exists a fixing of the randomness of A to get a deterministic circuit which preserves the above bound. Since we can do this for every I and H, the condition of the theorem is satisfied, which proves the theorem. □

5 Extracting Computational Randomness

In this section, we show that one can extract randomness from the output of a hard function. For this, we first show that the output of a hard function looks somewhat random, even given the input. More precisely, we have the following.

Lemma 4. *Let $f : X \to [V]$ be a (δ, s)-hard function, with $\delta \geq 1 - 2^{-k}$ for some positive $k \in \mathbb{R}$. Let $\mathcal{X}$ be the uniform distribution over X. Then for any $\varepsilon \in (0,1)$, there exist some $s' \geq s/\mathrm{poly}(V, 1/\varepsilon)$ and a distribution $\mathcal{V}$ (correlated with $\mathcal{X}$) such that the following two conditions hold.*

- *The distributions $(\mathcal{X}, f(\mathcal{X}))$ and $(\mathcal{X}, \mathcal{V})$ have no (ε, s')-distinguisher.*
- $\Pr_{x \in \mathcal{X}} \left[\mathrm{H}_\infty(\mathcal{V} | \mathcal{X} = x) < \lceil k/3 \rceil \right] \leq 2^{-k/3}$.

The proof of Lemma 4 is omitted here due to the page limit. The basic idea is that by applying Lemma 1 repeatedly, we can find a collection of disjoint hardcore sets covering a large fraction of inputs. Then by extending the idea in [17], we can show that when a randomly sampled input x falls into one of these hardcore sets, its output looks like a random value from some set.

According to Lemma 4, the output distribution of a hard function looks like one with some min-entropy, given a randomly selected input. This suggests the possibility that one can simply apply any extractor to extract randomness from the output. Formally, we have the following theorem, for the case of seeded extractors. Due to the page limit, the proof is omitted here.

Theorem 2. *Let $f : X \to \{0,1\}^\ell$ be a (δ, s)-hard function with $\delta \geq 1 - 2^{-k}$, and let $\mathcal{X}$ be the uniform distribution over X. Then for any seeded $(k/3, \varepsilon)$-extractor* Ext : $\{0,1\}^\ell \times \{0,1\}^d \to \{0,1\}^m$ *computable in* $\mathsf{SIZE}(s_0)$, *the distributions $(\mathcal{X}, \textsc{Ext}(f(\mathcal{X}), \mathcal{U}_d))$ and $(\mathcal{X}, \mathcal{U}_m)$ have no $(\bar{\varepsilon}, \bar{s})$-distinguisher, for some $\bar{\varepsilon} \leq 2\varepsilon + 2^{-k/3}$ and $\bar{s} \geq s/\mathrm{poly}(2^\ell, 1/\varepsilon) - s_0$.*

Note that many constructions of seeded extractors are in fact computable by small circuits, of size $s_0 \leq \mathrm{poly}(\ell/\varepsilon)$. Then, for example, when $k \geq \Omega(\ell)$ and $\varepsilon = \mathrm{poly}(2^{-\ell})$, we have $\bar{\varepsilon} \leq 2^{-\Omega(\ell)}$ and $\bar{s} \geq s/2^{O(\ell)}$. This means that as long as s is large enough (or ℓ is small enough), any single-source seeded extractor can be used to extract randomness in the computational setting.

For the case of seedless extractors, we have the following. Due to the page limit, the proof is omitted here.

Theorem 3. *For $i \in [t]$, let $f^{(i)} : X^{(i)} \to \{0,1\}^\ell$ be a $(\delta^{(i)}, s)$-hard function with $\delta^{(i)} \geq 1 - 2^{-k^{(i)}}$, and let $k = \sum_{i \in [t]} k^{(i)}$. Let $\mathcal{X} = (\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(t)})$, where each $\mathcal{X}^{(i)}$ is an independent uniform distribution over $X^{(i)}$, and let $f(\mathcal{X}) = (f^{(1)}(\mathcal{X}^{(1)}), \ldots, f^{(t)}(\mathcal{X}^{(t)}))$. Then for any seedless t-source $(k/7, \varepsilon)$-extractor* Ext : $(\{0,1\}^\ell)^t \to \{0,1\}^m$ *computable in* $\mathsf{SIZE}(s_0)$, *the distributions $(\mathcal{X}, \textsc{Ext}(f(\mathcal{X})))$ and $(\mathcal{X}, \mathcal{U}_m)$ have no $(\bar{\varepsilon}, \bar{s})$-distinguisher, for some $\bar{\varepsilon} \leq (t+1)\varepsilon + 2^{-\Omega(k^2/t\ell^2)}$ and $\bar{s} \geq s/\mathrm{poly}(2^\ell, 1/\varepsilon) - s_0$.*

Let Ext be the seedless t-source extractor in [11], which is computable by a small circuit, with $s_0 \leq \mathrm{poly}(t\ell/\varepsilon)$. Then, for example, when $t\varepsilon = \mathrm{poly}(2^{-\ell})$

and $k \geq \ell^c$ for a large enough constant c, we have $\bar{\varepsilon} \leq 2^{-\Omega(\ell)}$ and $\bar{s} \geq s/2^{O(\ell)}$. Again, this means that when s is large enough (or ℓ is small enough), any seedless multi-source extractor can also work in the computational setting.

6 Loss of Circuit Size

Recall that in our generalized hardcore set lemma (Lemma 1), there is a loss of circuit size by a factor of poly(V) for functions with V output values. That is, from a (δ, s)-hard function, we can only guarantee the existence of an (I, ε, s')-hardcore set with $s' \leq s/\text{poly}(V)$. In this section, we show that such a loss of circuit size is in fact unavoidable, if the proof is done in a black-box way. Before we can formally state our result, we need to introduce some definitions. Let $F_{X,V}$ denote the collection of functions from X to $[V]$.

Definition 6. *Given a collection $G \subseteq F_{X,V}$, we say that a function $f \in F_{X,V}$ is $(k, \rho, \varepsilon, G)$-easy if for any $I \in \binom{[V]}{k}$ and any $H \subseteq f^{-1}(I)$ of density $|H|/|X| \geq \rho$, there is a function $g \in G$ such that $\Pr_{x \in H}[g(x) = f(x)] \geq \frac{1+\varepsilon}{k}$.*

Next, we define our notion of a black-box proof, which is realized by some oracle algorithm $R^{(\cdot)}$. We allow R to be non-uniform and randomized, and we use the notation $R_r^{G;\alpha}(x)$ to denote that R is given an oracle G, an advice string α, a random string r, and an input x.

Definition 7. *We say that an oracle algorithm $R^{(\cdot)}$ realizes a $(\delta, k, \rho, \varepsilon, S)$ black-box proof of hardcore sets for functions in $F_{X,V}$, if the following holds. For any $f \in F_{X,V}$ and any $G \subseteq F_{X,V}$ with $|G| = S$, if f is $(k, \rho, \varepsilon, G)$-easy, then there exists some advice α such that*

$$\Pr_{x,r}\left[R_r^{G;\alpha}(x) \neq f(x)\right] < \delta.$$

Here we allow R to make adaptive queries, but for simplicity we consider only the case that R on input x queries functions in the oracle at all x. That is, R may first queries $g_i(x)$ for some $g_i \in G$, and depending on the answer, R next queries $g_j(x)$ for some $g_j \in G$, and so on. Note that our proof for the generalized hardcore sets is done in this black-box way, and so do all the known proofs for Impagliazzo's hardcore set lemma. Our result in this section shows that any algorithm realizing such a black-box proof must make many queries to the oracle.

Theorem 4. *Suppose $V \geq \omega(1)$, $0 < \delta \leq 1 - (4 \log V)/V$, $0 < \varepsilon \leq 1/3$, $0 < \rho < 1$, and $S \geq \Omega((V^{k+1}k^3/\varepsilon^2)\log(1/\rho))$. Consider any oracle algorithm which uses an advice of length $\tau \leq o(\delta|X|)$ and realizes a $(\delta, k, \varepsilon, S, \rho)$ black-box proof of hardcore sets for functions in $F_{X,V}$. Then it must make at least $\Omega((Vk/\varepsilon^2)\log(1/\delta))$ oracle queries.*

Note that the theorem says that even if we start from a very hard function, with δ close to one, and even if we only want a hardcore set with $k = 2$ output values, any algorithm realizing such a black-box proof still need to make many queries, which corresponds to a large loss of circuit size. In particular, a loss by a V factor is unavoidable. Now let us prove the theorem.

Proof. Consider any R which realizes such a black-box proof. Assume that R makes at most $q = o((Vk/\varepsilon^2)\log(1/\delta))$ oracle queries, and we will show that this leads to a contradiction. In particular, we will show the existence of a function f and a collection of functions $G = \{g_{I,i} : I \in \binom{[V]}{k}$ and $i \in [T]\}$, for some $T = \Omega((Vk^3/\varepsilon^2)\log(1/\rho))$, such that f is (k,ρ,ε,G)-easy but $\Pr_{x,r}[R_r^{G;\alpha}(x) \neq f(x)] \geq \delta$, for any advice α, which violates the requirement for a black-box proof. We will prove the existence of such f and G by a probabilistic argument.

We choose $f \in F_{X,V}$ randomly such that independently for each input x, $f(x)$ takes a uniformly random value in $[V]$. Then we choose each $g_{I,i}$ in the following way:

- Independently for each input $x \in f^{-1}(I)$, $g_{I,i}(x)$ takes the value $f(x)$ with probability $(1+2\varepsilon)/k$ and each other value in I with probability $(1-2\varepsilon/(k-1))/k$.
- Independently for each input $x \notin f^{-1}(I)$, $g_{I,i}(x)$ takes each value in I with probability $1/k$.

Our key lemma is the following.

Lemma 5. *For any advice α and any input x, we have $\Pr_{f,G,r}[R_r^{G;\alpha}(x) \neq f(x)] \geq \sqrt{\delta}$.*

Due to the page limit, we omit the formal proof of Lemma 5 and only sketch the proof idea here. Consider any advice α and any input x. Recall that in our model, any query made by $R^{G;\alpha}(x)$ is of the form $g_{I,i}(x)$, for some $g_{I,i} \in G$, and the outcome of such a query has a distribution close to the uniform over the k values in I, which is independent of f. When R makes only a small number of queries, we can show that the distribution of the sequence of outcomes corresponding to the queries is still close to a distribution which is independent of f, with the outcome of each query being independent and uniform over some k values. That is, such an R cannot fully exploit the small correlation between f and G, and hence it behaves similarly when the useful oracle G is replaced by a useless one which is independent of f. However, without a useful oracle, R cannot possibly predict a random f well, which implies that even given G, R cannot predict f well either.

Using this lemma together with a Hoeffding bound, one can show that for any advice α,

$$\Pr_{f,G,r}\left[\Pr_x\left[R_r^{G;\alpha}(x) \neq f(x)\right] < \delta\right] \leq 2^{-\Omega(\delta|X|)}.$$

Then, using a union bound, we have

$$\Pr_{f,G,r}\left[\exists\alpha \in \{0,1\}^\tau : \Pr_x\left[R_r^{G;\alpha}(x) \neq f(x)\right] < \delta\right] \leq 2^\tau \cdot 2^{-\Omega(\delta|X|)} \leq o(1), \quad (2)$$

when $\tau \leq o(\delta|X|)$. Next, we need the following; the proof is by a simple probabilistic argument and is omitted here due to the page limit.

Lemma 6. $\Pr_{f,G}[f$ *is not* (k,ρ,ε,G)*-easy*$] \leq o(1)$.

From Lemma 6 and the bound in (2), we can conclude the existence of some f and G such that f is $(k, \rho, \varepsilon, G)$-easy but $\Pr_{f,G,r}[R_r^{G;\alpha}(x) \neq f(x)] \geq \delta$ for any advice α, which contradicts the requirement for a black-box proof of hardcore sets. Therefore, any R realizing such a black-box proof must make at least $\Omega((Vk/\varepsilon^2)\log(1/\delta))$ queries, which proves Theorem 4. □

References

1. Auer, P., Cesa-Bianchi, N., Freund, Y., Schapire, R.: The non-stochastic multi-armed bandit problem. SIAM J. Comput. 32(1), 48–77 (2002)
2. Barak, B., Hardt, M., Kale, S.: The Uniform Hardcore Lemma via Approximate Bregman Projectionss. In: SODA 2008, pp. 1193–1200 (2008)
3. Barak, B., Shaltiel, R., Wigderson, A.: Computational analogues of entropy. In: Proc. APPROX-RANDOM, pp. 200–215 (2003)
4. Cover, T., Thomas, J.: Elements of Information Theory. Wiley, Chichester (1991)
5. Goldreich, O., Rubinfeld, R., Sudan, M.: Learning polynomials with queries: the highly noisy case. SIAM J. Disc. Math. 13(4), 535–570 (2000)
6. Healy, A., Vadhan, S., Viola, E.: Using nondeterminism to amplify hardness. SIAM J. Comput. 35(4), 903–931 (2006)
7. Holenstein, T.: Key agreement from weak bit agreement. In: STOC 2005, pp. 664–673 (2005)
8. Hsiao, C.-Y., Lu, C.-J., Reyzin, L.: Conditional computational entropy, or toward separating pseudoentropy from compressibility. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 169–186. Springer, Heidelberg (2007)
9. Impagliazzo, R.: Hard-core distributions for somewhat hard problems. In: FOCS 1995, pp. 538–545 (1995)
10. Klivans, A., Servedio, R.A.: Boosting and hard-core sets. Machine Learning 51(3), 217–238 (2003)
11. Lee, C.-J., Lu, C.-J., Tsai, S.-C.: Deterministic extractors for independent-symbol sources. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4051, pp. 84–95. Springer, Heidelberg (2006)
12. Lee, C.-J., Lu, C.-J., Tsai, S.-C.: Extracting computational entropy and learning noisy linear functions. In: Ngo, H.Q. (ed.) COCOON 2009. LNCS, vol. 5609, pp. 338–347. Springer, Heidelberg (2009)
13. Lu, C.-J., Tsai, S.-C., Wu, H.-L.: On the complexity of hard-core set constructions. In: Arge, L., Cachin, C., Jurdziński, T., Tarlecki, A. (eds.) ICALP 2007. LNCS, vol. 4596, pp. 183–194. Springer, Heidelberg (2007)
14. Nisan, N., Zuckerman, D.: Randomness is linear in space. J. Comput.Syst. Sci. 52(1), 43–52 (1996)
15. O'Donnell, R.: Hardness amplification within NP. In: STOC, pp. 751–760 (2002)
16. Shaltiel, R.: Recent developments in explicit constructions of extractors. Bulletin of the EATCS 77, 67–95 (2002)
17. Sudan, M., Trevisan, L., Vadhan, S.: Pseudorandom generators without the XOR lemma. J. Comput.Syst. Sci. 62(2), 236–266 (2001)
18. Trevisan, L.: List decoding using the XOR lemma. In: FOCS, pp. 126–135 (2003)
19. Trevisan, L.: On uniform amplification of hardness in NP. In: STOC, pp. 31–38 (2005)
20. Yao, A.: Theory and applications of trapdoor functions. In: FOCS 1982, pp. 80–91 (1982)
21. Zuckerman, D.: General weak random sources. In: FOCS, pp. 534–543 (1990)