Computational Analogues of Entropy

Boaz Barak[1], Ronen Shaltiel[1], and Avi Wigderson[2]

[1] Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel. Emails: {boaz,ronens}@wisdom.weizmann.ac.il
[2] School of Mathematics, Institute for Advanced Study, Princeton, NJ and Hebrew University, Jerusalem, Israel. Email: avi@ias.edu

Abstract. Min-entropy is a statistical measure of the amount of randomness that a particular distribution contains. In this paper we investigate the notion of *computational min-entropy* which is the computational analog of statistical min-entropy. We consider three possible definitions for this notion, and show equivalence and separation results for these definitions in various computational models.
We also study whether or not certain properties of statistical min-entropy have a computational analog. In particular, we consider the following questions:

1. Let X be a distribution with high computational min-entropy. Does one get a pseudo-random distribution when applying a "randomness extractor" on X?
2. Let X and Y be (possibly dependent) random variables. Is the computational min-entropy of (X, Y) at least as large as the computational min-entropy of X?
3. Let X be a distribution over $\{0,1\}^n$ that is "weakly unpredictable" in the sense that it is hard to predict a constant fraction of the coordinates of X with a constant bias. Does X have computational min-entropy $\Omega(n)$?

We show that the answers to these questions depend on the computational model considered. In some natural models the answer is false and in others the answer is true. Our positive results for the third question exhibit models in which the "hybrid argument bottleneck" in "moving from a distinguisher to a predictor" can be avoided.

1 Introduction

One of the most fundamental notions in theoretical computer science is that of *computaional indistinuishability* [1, 2]. Two probability distributions are deemed close if no *efficient*[3] test can tell them apart - this comes in stark contrast to the information theoretic view which allows *any* test whatsoever. The discovery [3, 2, 4] that simple computational assumptions (namely the existance of one-way functions) make the computational and information theoretic notions completely

[3] What is meant by "efficient" can naturally vary by specifying machine models and resource bounds on them

different has been one of the most fruitful in CS history, with impact on cryptography, complexity theory and computational learning theory.

The most striking result of these studies has been the efficient construction of nontrivial *pseudorandom* distributions, namely ones which are information theoretically very far from the uniform distribution, but are nevertheless indistinguishable from it. Two of the founding papers [2, 4] found it natural to extend information theory more generally to the computational setting, and attempt to define its most fundamental notion of entropy[4]. The basic question is the following: when should we say that a distribution has (or is close to having) computational entropy (or pseudoentropy) k?. Interestingly, these two papers give two very different definitions! This point may be overlooked, since for the most interesting special case, the case of pseudorandomness (i.e., when the distributions are over n-bit strings and $k = n$), the two definitions coincide. This paper is concerned with the other cases, namely $k < n$, attempting to continue the project of building a computational analog of information theory.

1.1 Definitions of pseudoentropy

To start, let us consider the two original definitions. Let X be a probability distribution over a set S.

A definition using "compression". Yao's definition of pseudoentropy [2] is based on compression. He cites Shannon's definition [5], defining $H(X)$ to be the minimum number of bits needed to describe a typical element of X. More precisely, one imagines the situation of Alice having to send Bob (a large number of) samples from X, and is trying to save on communication. Then $H(X)$ is the smallest k for which there are a compression algorithm A (for Alice) from S into k-bit strings, and a decompression algorithm B (for Bob) from k-bit strings into S, such that $B(A(x)) = x$ (in the limit, for typical x from X). Yao take this definition verbatim, adding the crucial computational constraint that both compression and decompression algorithms must be efficient. This notion of efficient compression is further studied in [6].

A definition using indistinguishability. Hastad et al's definition of pseudoentropy [4] extends the definition of pseudorandomness syntactically. As a distribution is said to be pseudorandom if it is indistinguishable from a distribution of maximum entropy (which is unique), they define a distribution to have pseudoentropy k is it is indistinguishable from a distribution of Sahnnon entropy k (for which there are many possibilities).

It turns out that the two definitions of pseudoentropy above can be very different in natural computational settings, despite the fact that in the information theoretic setting they are identical for any k. Which definition, then, is the "natural one" to choose from? This question is actually more complex, as another natural point of view lead to yet another definition.

[4] While we will first mainly talk about Shannon's entropy, we later switch to min-entropy and stay with it throughout the paper. However the whole introduction may be read when regarding the term "entropy" with any other of its many formal variants, or just as well as the informal notion of "information content" or "uncertainty"

A definition using a natural metric space. The computational viewpoint of randomness may be thought of as endowing the space of *all* probability distributions with new, interesting metrics.

For every event (=test) T in our probability space we define: $d_T(X, Y) = |\Pr_X[T] - \Pr_Y[T]|$. In words, the distance between X and Y is the difference (in absolute value) of the probabilities they assign to T.[5]

Note that given a family of metrics, their maximum is also a metric. An information theoretic metric on distributions, the *statistical distance*[6] (which is basically $\frac{1}{2}L_1$-distance) is obtained by taking the maximum over the T-metrics above for *all* possible tests T. A natural computational metric, is given by taking the maximum over any class $\mathcal{C}$ of efficient tests. When should we say that a distribution X is indistinguishable from having Shannon entropy k? Distance to a set is the distance to the closest point in it, so X has to be close in this metric to *some* Y with Shannon entropy k.

A different order of quantifiers. At first sight this may look identical to the "indistinguishability" definition in [4]. However let us parse them to see the difference. The [4] definition say that X has pseudoentropy k if *there exists* a distribution Y of Shannon entropy k, such that *for all* tests T in $\mathcal{C}$, T has roughly the same probability under both X and Y. The metric definition above reverses the quantifiers: X has pseudoentropy k if *for every* a distribution Y of Shannon entropy k, *there exists* a test T in $\mathcal{C}$, which has roughly the same probability under both X and Y. It is easy to see that the metric definition is more liberal - it allows for at least as many distributions to have pseudoentropy k. Are they really different?

Relations between the three definitions. As all these definitions are natural and well-motivated, it makes sense to study their relationship. In the information theoretic world (when ignoring the "efficiency" constraints) all definitions are equivalent. It is easy to verify that regardless of the choice of a class $\mathcal{C}$ of "efficient" tests, they are ordered in permisiveness (allowing more distributions to have pseudoentropy k). The "indistinguishability" definition of [4] is the most stringent, then the "metric definition", and then the "compression" definition of [2]. What is more interesting is that we can prove collapses and separations for different computational settings and assumptions. For example, we show that the first two definitions drastically differ for logspace observers, but coincide for polynomial time observers (both in the uniform and nonuniform settings). The proof of the latter statement uses the "min-max" Theorem of [7] to "switch" the order of quantifiers. We can show some weak form of equivalence between all three definitions for circuits. We show that the "metric" coincides with the "compression" definition if $\mathbf{NP} \subseteq \mathbf{BPP}$. More precisely, we give a *non-deterministic* reduction showing the equivalence of the two definitions. This reduction guarantees high min-entropy according to the "metric" definition if the distribution has

[5] This isn't precisely a metric as there may be different X and Y such that $d_T(X, Y) = 0$. However it is symmetric and satisfies the triangle inequality.

[6] Another basic distance measure is the so called KL-divergence, but for our purposes, which concern very close distributions, is not much different than statistical distance

high min-entropy according to the "compression" distribution with respect to an **NP** oracle. A clean way to state this is that all three definitions are equivalent for **PH**/poly. We refer to this class as the class of poly-size **PH**-circuits. Such circuits are poly-size circuits which are allowed to compute an arbitrary function in the polynomial-hierarchy (**PH**). We remark that similar circuits (for various levels of the **PH** hierarchy) arise in related contexts in the study of "computational randomness": They come up in conditional "derandomization" results of **AM** [8–10] and "extractors for samplable distributions" [11].

1.2 Pseudoentropy versus information theoretic entropy

We now move to another important part of our project. As these definitions are supposed to help establish a computational version of information theory, we attempt to see which of them respect some natural properties of information-theoretic entropy.

Using randomness extractors. In the information theoretic setting, there are *randomness extractors* which convert a high entropy[7] distribution into one which is statistically close to uniform. The theory of extracting the randomness from such distributions is by now quite developed (see surveys [12–14]). It is natural to expect that applying these randomness extractors on high pseudoentropy distributions produces a pseudorandom distribution. In fact, this is the motivation for pseudoentropy in some previous works [15, 4, 16].

It is easy to see that the the "indistinguishability" definition of [4] has this property. This also holds for the "metric" definition by the equivalence above. Interestingly, we do not know whether this holds for the "compression" definition. Nevertheless, we show that some extractor constructions in the literature (the ones based on Trevisan's technique [17–20, 10]) do produce a pseudorandom distribution when working with the "compression" definition.

The information in two dependent distributions. One basic principle in information theory is that two (possibly dependent) random variables have at least as much entropy as any one individually, e.g. $H(X,Y) \geq H(X)$. A natural question is whether this holds when we replace information-theoretic entropy with pseudoentropy. We show that the answer depends on the model of computation. If there exist one-way functions, then the answer is *no* for the standard model of polynomial-time distinguishers. On the other hand, if $\mathbf{NP} \subseteq \mathbf{BPP}$, then the answer is *yes*. Very roughly speaking, the negative part follows from the existence of pseudorandom generators, while the positive part follows from giving a *nondeterministic* reduction which relies on nondeterminism to perform approximate counting. Once again, this result can be also stated as saying that the answer is positive for poly-size **PH**-circuits. We remark that the positive result holds for (nonuniform) online space-bounded computation as well.

[7] It turns out that a different variant of entropy called "min-entropy" is the correct measure for this application. The min-entropy of a distribution X is $\log_2(\min_x 1/\Pr[X = x])$. This should be compared with Shannon's entropy in which the minimum is replaced by averaging.

Entropy and unpredictability. A deeper and interesting connection is the one between entropy and unpredictability. In the information theoretic world, a distribution which is unpredictable has high entropy.[8] Does this relation between entropy and unpredictability holds in the computational world?

Let us restrict ourselves here for a while to the metric definition of pseudoentropy. Two main results we prove is that this connection indeed holds in two natural computational notions of efficient observers. One is for logspace observers. The second is for **PH**-circuits. Both results use one mechanism - a different characterization of the metric definition, in which distinuguishers accept very few inputs (less than 2^k when the pseudoentropy is k). We show that predictors for the accepted set are also good for any distribution "caught" by such a distinguisher. This direction is promising as it suggests a way to "bypass" the weakness of the "hybrid argument".

The weakness of the hybrid argument. Almost all pseudorandom generators (whether conditional such as the ones for small circuits or unconditional such as the ones for logspace) use the hybrid argument in their proof of correctness. The idea is that if the output distribution can be efficiently distinguished from random, some bit can be efficiently predicted with nontrivial advantage. Thus, pseudorandomness is established by showing unpredictability.

However, in standard form, if the distinughishability advantage is ϵ, the prediction advantage is only ϵ/n. In the results above, we manage (for these two computational models) to avoid this loss and make the prediction advantage $\Omega(\epsilon)$ (just as information theory suggests).

While we have no concrete applications, this seem to have potential to improve various constructions of pseudorandom generators. To see this, it suffices to observe the consequences of the hybrid argument loss. It requires every output bit of the generator to be very unpredictable, for which a direct cost is paid in the seed length (and complexity) of the generator. For generators against circuits, a long sequence of works [2, 21, 22, 16] resolved it optimally using efficient *hardness amplification*. These results allow constructing distributions which are unpredictable even with advantage $1/\text{poly}(n)$. The above suggests that sometimes this amplification may not be needed. One may hope to construct a pseudorandom distribution by constructing an unpredictable distribution which is only unpredictable with constant advantage, and then use a randomness extractor to obtain a pseudorandom distribution.[9]

This problem is even more significant when constructing generators against logspace machines [24, 25]. The high unpredictability required seems to be the bottleneck for reducing the seed length in Nisan's generator [24] and its refine-

[8] We consider two different forms of prediction tests: The first called "next bit predictor" attempts to predict a bit from previous bits, whereas the second called "complement predictor" has access to all the other bits, both previous and latter.

[9] This approach was used in [16]. They show that even "weak" hardness amplification suffices to construct a high pseudoentropy distribution using the pseudo-random generator construction of [23]. However, their technique relies on the properties of the specific generator and cannot be applied in general.

ments from $O((\log n)^2)$ bits to the optimal $O(\log n)$ bits (that will result in $BPL = L$). The argument above gives some hope that for fooling logspace machines (or even just constant-width oblivious branching programs) the suggested approach may yield substantial improvements. However, in this setup there is another hurdle: In [26] it was shown that randomness extraction cannot be done by one pass log-space machines. Thus, in this setup it is not clear how to move from pseudoentropy to pseudorandomness.

1.3 Organization of the paper

In Section 2 we give some basic notation. Section 3 formally defines our three basic notions of pseudoentropy, and proves a useful characterization of the metric definition. In Sections 5 and 6 we prove equivalence and separations results between the various definitions in several natural computational models. Section 7 is devoted to our results about computational analogs of information theory for concatenation and unpredictability of random variables. Because of space limitations many of the proofs do not appear in this version.

2 Preliminaries

Let X be a random variable over some set S. We say that X has *(statistical) min-entropy* at least k, denoted $H^\infty(X) \geq k$, if for every $x \in S$, $\Pr[X = x] \leq 2^{-k}$. We use U_n to denote the uniform distribution on $\{0,1\}^n$.

Let X, Y be two random variables over a set S. Let $f : S \to \{0,1\}$ be some function. The *bias* of X and Y with respect to f, denoted $\mathsf{bias}_f(X,Y)$, is defined by $\big|\mathbb{E}[f(X)] - \mathbb{E}[f(Y)]\big|$. Since it is sometimes convenient to omit the absolute value, we denote $\mathsf{bias}^*_f(X,Y) = \mathbb{E}[f(X)] - \mathbb{E}[f(Y)]$.

The *statistical distance* of X and Y, denoted $\mathsf{dist}(X,Y)$, is defined to be the maximum of $\mathsf{bias}_f(X,Y)$ over all functions f. Let $\mathcal{C}$ be a class of functions from S to $\{0,1\}$ (e.g., the class of functions computed by circuits of size m for some integer m). The *computational distance* of X and Y w.r.t. $\mathcal{C}$, denoted $\mathsf{comp\text{-}dist}_{\mathcal{C}}(X,Y)$, is defined to be the maximum of $\mathsf{bias}_f(X,Y)$ over all $f \in \mathcal{C}$. We will sometimes drop the subscript $\mathcal{C}$ when it can be inferred from the context.

Computational models. In addition to the standard model of uniform and non-uniform polynomial-time algorithms, we consider two additional computational models. The first is the model of **PH**-*circuits.* A **PH**-circuit is a boolean circuit that allows queries to a language in the polynomial hierarchy as a basic gate.[10] The second model is the model of *bounded-width read-once oblivious branching programs.* A width-S read once oblivious branching program P is a directed graph with Sn vertices, where the graph is divided into n layers, with S vertices in each layer. The edges of the graph are only between from one layer to the next one, and each edge is labelled by a bit $b \in \{0,1\}$ which is thought of as a variable. Each vertex has two outgoing edges, one labelled 0 and the other labelled 1. One of the vertices in the first layer is called the *source* vertex, and some of the vertices in the last layer are called the **accepting vertices**. A computation of

[10] Equivalently, the class languages accepted by poly-size **PH**-circuits is **PH**/poly.

the program P on input $x \in \{0,1\}^n$ consists of walking the graph for n steps, starting from the source vertex, and in step i taking the edge labelled by x_i. The output of $P(x)$ is 1 iff the end vertex is accepting. Note that variables are read in the natural order and thus width-S read once oblivious branching programs are the non-uniform analog of one-pass (or online) space-$\log S$ algorithms.

3 Defining computational min-entropy

In this section we give three definitions for the notion of computational (or "pseudo") min-entropy. In all these definitions, we fix $\mathcal{C}$ to be a class of functions which we consider to be efficiently computable. Our standard choice for this class will be the class of functions computed by a boolean circuit of size $p(n)$, where n is the circuit's input length and $p(\cdot)$ is some fixed polynomial. However, we will also be interested in instantiations of these definitions with respect to different classes $\mathcal{C}$. We will also sometimes treat $\mathcal{C}$ as a class of *sets* rather then functions, where we say that a set D is in $\mathcal{C}$ iff its characteristic function is in $\mathcal{C}$. We will assume that the class $\mathcal{C}$ is closed under complement.

3.1 HILL-type pseudoentropy: using indistinguishability

We start with the standard definition of computational (or "pseudo") min-entropy, as given by [4]. We call this definition *HILL-type pseudoentropy*.

Definition 1. *Let X be a random variable over a set S. Let $\epsilon \geq 0$. We say that X has* **ϵ-HILL-type pseudoentropy** *at least k, denoted $H^{\mathbf{HILL}}_{\epsilon}(X) \geq k$, if there exists a random variable Y with (statistical) min-entropy at least k such that the computational distance (w.r.t. $\mathcal{C}$) of X and Y is at most ϵ.*

We will usually be interested in ϵ-pseudoentroy for ϵ that is a small constant. In these cases we will sometimes drop ϵ and simply say that X has (HILL-type) pseudoentropy at least k (denoted $H^{\mathbf{HILL}}(X) \geq k$).

3.2 Metric-type pseudoentropy: using a metric space

In Definition 1 the distribution X has high pseudoentropy if there *exists* a high min-entropy Y such that X and Y are indistinguishable. As explained in the introduction, it is also natural to reverse the order of quantifiers: Here we allow Y to be a function of the "distinguishing test" f.

Definition 2. *Let X be a random variable over a set S. Let $\epsilon \geq 0$. We say that X has* **ϵ-metric-type pseudoentropy** *at least k, denoted $H^{\mathbf{Metric}}_{\epsilon}(X) \geq k$, if for every test f on S there exists a Y which has (statistical) min-entropy at least k and $\mathsf{bias}_f(X,Y) < \epsilon$.*

It turns out that metric-pseudoentropy is equivalent to a different formulation. (Note that the condition below is only meaningful for D such that $|D| < 2^k$). The proof of Lemma 3 appears in the full version.

Lemma 3. *For every class $\mathcal{C}$ which is closed under complement and for every $k \leq \log|S| - 1$ and ϵ, $H^{\mathbf{Metric}}_{\epsilon}(X) \geq k$ if and only if for every set $D \in \mathcal{C}$, $\Pr[X \in D] \leq \frac{|D|}{2^k} + \epsilon$*

3.3 Yao-type pseudoentropy: using compression

Let $\mathcal{C}$ be a class of functions which we consider to efficiently computable. Recall that we said that a set D is a member of $\mathcal{C}$ if its characteristic function was in $\mathcal{C}$. That is, a set D is in $\mathcal{C}$ if it is *efficiently decidable*. We now define a family $\mathcal{C}_{\mathsf{compress}}$ of sets that are **efficiently compressible**. That is, we say that a set $D \subseteq S$ is in $\mathcal{C}_{\mathsf{compress}}(\ell)$ if there exist functions $c, d \in \mathcal{C}$ ($c : S \to \{0,1\}^{\ell}$ stands for *compress* and $d : \{0,1\}^{\ell} \to S$ for *decompress*) such that $D = \{x | d(c(x)) = x\}$. Note that every efficiently compressible set is also efficiently decidable (assuming the class $\mathcal{C}$ is closed under composition). Yao-type pseudoentropy is defined by replacing the quantification over $D \in \mathcal{C}$ in the alternative characterization of metric-type pseudoentropy (Lemma 3) by a quantification over $D \in \mathcal{C}_{\mathsf{compress}}(\ell)$ for all $\ell < k$. The resulting definition is the following:

Definition 4. *Let X be a random variable over a set S. X has ϵ-*Yao-type pseudoentropy *at least k, denoted $H_{\epsilon}^{\mathbf{Yao}}(X) \geq k$, if for every $\ell < k$ and every set $D \in \mathcal{C}_{\mathsf{compress}}(\ell)$,* $\Pr[X \in D] \leq 2^{l-k} + \epsilon$

4 Using randomness extractors

An extractor uses a short seed of truly random bits to extract many bits which are (close to) uniform.

Definition 5 ([27]). *A function $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k, ϵ)-extractor if for every distribution X on $\{0,1\}^n$ with $H^{\infty}(x) \geq k$, the distribution $Z = E(X, U_d)$ has $\mathsf{dist}(Z, U_m) < \epsilon$.*

We remark that there are explicit (polynomial time computable) extractors with seed length polylog(n/ϵ) and $m = k$. The reader is referred to survey papers on extractors [12–14]. The following standard lemma says that if a distribution X has HILL-type pseudoentropy at least k with respect to circuits, then for every randomness extractor the distribution $E(X, U_d)$ is pseudorandom.

Lemma 6. *Let $\mathcal{C}$ be the class of polynomial size circuits. Let X be a distribution with $H_{\epsilon}^{\mathbf{HILL}}(X) \geq k$ and let E be a (k, ϵ)-extractor computable in time* poly(n) *then* $\mathsf{comp\text{-}dist}_{\mathcal{C}}(E(X, U_d), U_m) < 2\epsilon$.

Note that by Theorem 8 the same holds for the metric definition. Interestingly, we do not know whether this holds for Yao-type pseudoentropy. We can however show that this holds for the extractor of Trevisan [17]. Trevisan's extractor $E^{\mathbf{Tre}} : \{0,1\}^n \times \{0,1\}^{O(\log^2 n / \log k)} \to \{0,1\}^{\sqrt{k}}$ is a $(k, 1/n)$-extractor

Lemma 7. *Let $\mathcal{C}$ be the class of polynomial size circuits. Let X be a distribution with $H_{\epsilon}^{\mathbf{Yao}}(X) \geq k$, then* $\mathsf{comp\text{-}dist}_{\mathcal{C}}(E^{\mathbf{Tre}}(X, U_d), U_m) < 2\epsilon$.

The proof of Lemma 7 appears in the full version. Loosely speaking, the correctness proof of Trevisan's extractor (and some later constructions, c.f., [14]) shows that if the output of the extractor isn't close to uniform, then the distribution X can be compressed (which is impossible for a distribution of sufficiently high min-entropy). For the lemma, one only needs to observe that in this argument an *efficient* distinguisher gives rise to an *efficient* compressing algorithm. Thus, running the extractor on an "incompressible" distribution gives a pseudorandom distribution.

5 Relationships between definitions

5.1 Equivalence between HILL-type and metric-type

The difference between HILL-type and metric-type pseudoentropy is in the order of quantifiers. HILL-type requires that there exist a unique "reference distribution" Y with $H^{\infty}(Y) \geq k$ such that for every D, $\mathsf{bias}_D(X,Y) < \epsilon$, whereas metric-type allows Y to depend on D, and only requires that for every D there exists such a Y. It immediately follows that for every class $\mathcal{C}$ and every X, $H^{\mathbf{Metric}}(X) \geq H^{\mathbf{HILL}}(X)$. In this section we show that the other direction also applies (with small losses in ϵ and time/size) for small circuits.

Theorem 8 (Equivalence of HILL-type and metric-type for circuits). *Let X be a distribution over $\{0,1\}^n$. For every $\epsilon, \delta > 0$ and k, if $H^{\mathbf{Metric}}_{\epsilon-\delta}(X) \geq k$ (with respect to circuits of size $O(ns/\delta^2)$ then $H^{\mathbf{HILL}}_{\epsilon}(X) \geq k$ (with respect to circuits of size s)*

The proof of Theorem 8 appears only in the full version. We now provide a sketch of the argument. It is sufficient to show that if $H^{\mathbf{HILL}}_{\epsilon}(X) < k$ then then $H^{\mathbf{Metric}}_{\epsilon-\delta}(X) < k$. Suppose indeed that $H^{\mathbf{HILL}}_{\epsilon}(X) < k$. This implies that for every Y with $H^{\infty}(Y) \geq k$ there is a small circuit $D \in \mathcal{C}$ such that $\mathsf{bias}_D(X,Y) \geq \epsilon$.

We consider a game between two players. The "circuit player" Alice chooses a small circuit D and the "distribution player" Bob chooses a "flat" distribution Y with $H^{\infty}(Y) \geq k$.[11] (Note that both players have a finite number of strategies in the game.) After the choices are made, Bob pays $\mathsf{dist}_D(X,Y)$ dollars to Alice. Our assumption says that if Alice plays after Bob then she can always win ϵ dollars. Loosely speaking, the "min-max" theorem of [7] allows to switch the order of quantifiers and assert that Alice can guarantee the same amount even when playing first.[12] More formally, we conclude that there exists a *distribution* $\bar{D}$ over circuits for Alice such that she expects to get ϵ dollars for every reply Y of Bob. Note that we were able to switch the order of quantifiers to that of the "metric" definition. We are left with the task of converting $\hat{D}$ into a single circuit. This is done by sampling sufficiently many circuits $D_1, \cdots, D_t$ from $\hat{D}$

[11] A "flat" distribution is a distribution which is uniformly distributed over a subset of S.

[12] There is a subtlety here. In order to apply the theorem, Alice must be able to win ϵ dollars even when Bob plays a *mixed strategy* (i.e., a convex combination of his choices). However, a convex combination of flat distributions with min-entropy k also has min-entropy k.

and taking their average. By a union bound there exists a choice of $D_1, \cdots, D_t$ which is good for every distribution Y.[13]

In the full version we also prove equivalence for *uniform* polynomial time machines.[14]

5.2 Equivalence between all types for PH-circuits.

We do not know whether the assumption that $H^{\mathbf{Yao}}_{\epsilon}(X) \geq k$ for circuits implies that $H^{\mathbf{Metric}}_{\epsilon}(X) \geq k'$ for slightly smaller k' and circuit size (and in fact, we conjecture that it's false). However, we can prove it assuming the circuits for the Yao-type definition have access to an NP-oracle.

Theorem 9. *Let $k' = k+1$ There is a constant c so that if $H^{\mathbf{Yao}}_{\epsilon}(X) \geq k'$ (with respect to circuits of size $max(s, n^c)$ that use an NP-oracle) then $H^{\mathbf{Metric}}_{\epsilon}(X) \geq k$ (with respect to circuits of size s).*

The proof of Theorem 9 appears in the full version. The reduction in the proof of Theorem 9 uses an NP-oracle. The class of polynomial size **PH**-circuits are closed under the use of NP-oracles ($\mathbf{PH}^{NP}/poly = \mathbf{PH}/poly$). Applying the argument of Theorem 9 give the following corollary.

Corollary 10. *Let $\mathcal{C}$ be the class of polynomial size* **PH***-circuits. If $H^{\mathbf{Yao}}_{\epsilon}(X) \geq 2k$ then $H^{\mathbf{Metric}}_{\epsilon}(X) \geq k$.*

6 Separation between types

Given the results of the previous section it is natural to ask if HILL-type and metric-type pseudoentropy are equivalent in **all** natural computational models? We give a negative answer and prove that there's large gap between HILL-type and metric-type pseudoentropy in the model of bounded-width read-once oblivious branching programs.

Theorem 11. *For every constant $\epsilon > 0$ and sufficiently large $n \in \mathbb{N}$, and , there exists a random X variable over $\{0,1\}^n$ such that $H^{\mathbf{Metric}}_{\epsilon} X \geq (1-\epsilon)n$ with respect to width-S read once oblivious branching programs, but $H^{\mathbf{HILL}}_{1-\epsilon}(X) \leq \mathrm{polylog}(n, S)$ with respect to width-4 oblivious branching programs.*

Theorem 11 follows from the following two lemmas, whose proofs appear in the full version:

---

[13] It is crucial that this union bound is not performed over the $\binom{2^n}{2^k}$ choices for Y but rather on the 2^n inputs. More precisely, we show that there exist $D_1, \cdots, D_t$ such that for all inputs x, $\frac{1}{t}\sum D_i(x) \approx \mathbb{E}[\hat{D}(x)]$.

[14] We find this surprising because the argument above seems to exploit the non-uniformity of circuits: The "min-max theorem" works only for *finite* games and is non-constructive - it only shows existence of a distribution $\hat{D}$ and gives no clue to its complexity. The key idea is the observation that pseudoentropy with respect to uniform Turing machines implies also pseudoentropy for "slightly non-uniform" Turing machines. Exact details appear in the full version.

Lemma 12 (Based on [28]). *Let $\epsilon > 0$ be some constant and $S \in \mathbb{N}$ such that $S > \frac{1}{\epsilon}$. Let $l = \frac{10}{\epsilon} \log S$ and consider the distribution $X = (U_l, U_l, \ldots, U_l)$ over $\{0,1\}^n$ for some $n < S$ which is a multiple of l. Then, $H_\epsilon^{\mathbf{Metric}}(X) \geq (1-\epsilon)n$ with respect to width-S oblivious branching programs.*

Lemma 13. *Let $\epsilon > 0$ be some constant, and X be the random variable $(U_l, U_l, \ldots, U_l)$ over $\{0,1\}^n$ (where $l > \log n$). Then, $H_{(1-\epsilon)}^{\mathbf{HILL}}(X) \leq \frac{100}{\log(1/\epsilon)} l^3$ with respect to width-4 oblivious branching programs.*

7 Analogs of information-theoretic inequalities

7.1 Concatenation lemma

A basic fact in information theory is that for every (possibly correlated) random variables X and Y, the entropy of (X, Y) is at least as large as the entropy of X. We show that if one-way-functions exist then this does not hold for all types of pseudoentropy with respect to polynomial time circuits. On the other hand, we show that the fact above does hold for polynomial-sized **PH**-circuits and for bounded-width oblivious branching programs.[15]

Negative result for standard model. Our negative result is the following easy lemma, whose proof is omitted:

Lemma 14. *Let $G : \{0,1\}^l \to \{0,1\}^n$ be a (poly-time computable) pseudorandom generator.[16] Let (X, Y) be the random variables $(G(U_l), U_l)$. Then $H_\epsilon^{\mathbf{HILL}}(X) = n$ (for a negligible ϵ) but $H_{1/2}^{\mathbf{Yao}}(X, Y) \leq l + 1$.*

*Positive result for **PH**-circuits.* Our positive result for **PH**-circuits is stated in the following lemma, whose proof appears in the full version:

Lemma 15. *Let X be a random variable over $\{0,1\}^n$ and Y be a random variable over $\{0,1\}^m$. Suppose that $H_\epsilon^{\mathbf{Yao}}(X) \geq k$ with respect to s-sized **PH**-circuits. Then $H_\epsilon^{\mathbf{Yao}}(X, Y) \geq k$ with respect to $O(s)$-sized **PH**-circuits.*

Applying the results of Section 5.2, we obtain that with respect to **PH**-circuit, the concatenation property is satisfied also for HILL-type and Metric-type pseudoentropy.

Positive result for bounded-width oblivious branching programs. We also show that the concatenation property holds also for metric-type pseudoentropy with respect to bounded-width read-once oblivious branching programs. This is stated in Lemma 16, whose proof appears in the full version. Note that the quality of this statement depends on the order of the concatenation (i.e., whether we consider (X, Y) or (Y, X)).

[15] With respect to the latter, we only prove that concatenation holds for metric-type pseudoentropy.

[16] We mean here a pseudorandom generator in the "cryptographic" sense of Blum, Micali and Yao [3, 2]. That is, we require that G is polynomial time computable.

Lemma 16. *Let X be a random variable over $\{0,1\}^n$ and Y be a random variable over $\{0,1\}^m$. Suppose that $H_\epsilon^{\mathbf{Metric}}(X) \geq k$ with respect to width-S read-once oblivious branching programs. Then, $H_\epsilon^{\mathbf{Metric}}(X,Y) \geq k$ and $H_{2\epsilon S}^{\mathbf{Metric}}(Y,X) \geq k - \log(1/\epsilon)$ with respect to such algorithms.*

7.2 Unpredictability and entropy

Loosely speaking, a random variable X over $\{0,1\}^n$ is δ**-unpredictable** is for every index i, it is hard to predict X_i from $X_{[1,i-1]}$ (which denotes $X_1, \ldots, X_{i-1}$) with probability better than $\frac{1}{2} + \delta$.

Definition 17. *Let X be a random variable over $\{0,1\}^n$. We say that X is* δ-unpredictable in index i *with respect to a class of algorithms $\mathcal{C}$ if for every $P \in \mathcal{C}$, $\Pr[P(X_{[1,i-1]}) = X_i] < \frac{1}{2} + \delta$. X is* δ-unpredictable *if for every $P \in \mathcal{C}$ $\Pr[P(i, X_{[1,i-1]}) = X_i] < \frac{1}{2} + \delta$ where this probability is over the choice of X and over the choice of $i \leftarrow_R [n]$. We also define* complement *unpredictability by changing $X_{[1,i-1]}$ to $X_{[n]\setminus\{i\}}$ in the definition above.*

Yao's Theorem [2] says that if X is δ-unpredictable in all indices by polynomial-time (uniform or non-uniform) algorithms, then it is $n\delta$-indistinguishable from the uniform distribution. Note that this theorem can't be used for a constant $\delta > 0$. This loss of a factor of n comes from the use of the "hybrid argument" [1, 2]. In contrast, in the context of information theory it is known that if a random variable X is δ-unpredictable (w.r.t. to all possible algorithms) for a small constant δ and for a constant fraction of the indices, then $H^\infty(X) \geq \Omega(n)$. Thus, in this context it is possible to extract $\Omega(n)$ bits of randomness even from δ-unpredictable distributions where δ is a *constant* [20].

In this section we consider the question of whether or not there exists a computational analog to this information-theoretic statement.

Negative result in standard model. We observe that if one-way functions exist, then the distribution $(G(U_l), U_l)$ where $|G(U_l)| = \omega(l))$ used in Lemma 14 is also a counterexample (when considering polynomial-time distinguishers). That is, this is a distribution that is δ-unpredictable for a negligible δ in almost all the indices, but has low pseudoentropy. We do not know whether or not there exists a distribution that is δ-unpredictable for a *constant* δ for *all* the indices, and has sublinear pseudoentropy.

Positive results. We also show some computational settings in which the information theoretic intuition *does* holds. We show this for **PH**-circuits, and for bounded-width oblivious branching programs using the metric definition of pseudoentropy. We start by considering a special case in which the distinguisher has distinguishing probability 1 (or very close to 1).[17]

[17] Intuitively, this corresponds to applications that use the high entoropy distribution for hitting a set (like a disperser) rather than for approximation of a set (like an extractor).

Theorem 18. *Let X be a random variable over $\{0,1\}^n$. Suppose there exists a size-s **PH**-circuit (width-S oblivious branching program) D such that $|D^{-1}(1)| \leq 2^k$ and $\Pr[D(X) = 1] = 1$. Then there exists a size-$O(s)$ **PH**-circuit (width-S oblivious branching program) P such that $\Pr_{i\in[n],x\leftarrow_R X}[P(x_{[1,i]}) = x_i] \geq 1 - O(\frac{k}{n})$*

The main step in the proof of Theorem 18 is the following lemma:

Lemma 19. *Let $D \subseteq \{0,1\}^n$ be a set such that $|D| < 2^k$. Let $x = x_1 \dots x_{i-1} \in \{0,1\}^{i-1}$, we define N_x to be the number of continuations of x in D (i.e., $N_x = |\{x' \in \{0,1\}^{n-i} \mid xx' \in D\}|$). We define $P(x)$ as follows: $P(x) = 1$ if $\frac{N_{x1}}{N_x} > \frac{2}{3}$ and $P(x) = 1$ if $\frac{N_{x1}}{N_x} < \frac{1}{3}$, where $P(x)$ is undefined otherwise. Then, for every random variable X such that $X \subseteq D$,*

$$\Pr_{i\in[n],x\leftarrow_R X}\left[P(x_{[1,i-1]}) \text{ is defined and equal to } x_i\right] \geq 1 - O\left(\frac{k}{n}\right)$$

Proof. For $x \in \{0,1\}^n$, we let $bad(x) \subseteq [n]$ denote the set of indices $i \in [n]$ such that $P(x_{[1,i-1]})$ is either undefined or different from x_i. We will prove the lemma by showing that $|bad(x)| \leq O(k)$ for *every* string $x \in D$. Note that an equivalent condition is that $|D| \geq 2^{-\Omega(|bad(x)|)}$. Indeed, we will prove that $|D| \geq (1+\frac{1}{2})^{|bad(x)|}$. Let N_i denote the number of continuations of $x_{[1,i]}$ in D (i.e., $N_i = N_{x_{[1,i]}}$). We define $N_n = 1$. We claim that for every $i \in bad(x)$, $N_{i-1} \geq (1+\frac{1}{2})N_i$. (Note that this is sufficient to prove the lemma). Indeed, $N_{i-1} = N_{x_{[1,i-1]}0} + N_{x_{[1,i-1]}1}$, or in other words, $N_{i-1} = N_i + N_{x_{[1,i-1]}\overline{x_i}}$ (where $\overline{x_i} \stackrel{def}{=} 1 - x_i$). Yet, if $i \in bad(x)$ then $N_{x_{[1,i-1]}\overline{x_i}} \geq \frac{1}{3}(N_i + N_{x_{[1,i-1]}\overline{x_i}}) \geq \frac{1}{2}N_i$. □

We obtain Theorem 18 from Lemma 19 for the case of **PH**-circuits by observing that deciding whether $P(x)$ is equal to 1 or 0 (in the cases that it is defined) can be done in the polynomial-hierarchy (using approximate counting [29]). The case of bounded-width oblivious branching programs is obtained by observing that the state of the width-S oblivious branching program D after seeing $x_1, \dots, x_{i-1}$ completely determines the value $P(x_1, \dots, x_{i-1})$ and so $P(x_1, \dots, x_{i-1})$ can be computed (non-uniformly) from this state.[18]

We now consider the case that $\Pr_{x\leftarrow_R X}[x \in D] = \epsilon$ for an arbitrary constant ϵ (that may be smaller than $\frac{1}{2}$). In this case we are not able to use standard unpredictability and use *complement unpredictability.*

Theorem 20. *Suppose that X is δ-complement-unpredictable for a random index with respect to s-sized **PH**-circuits, where $\frac{1}{2} > \delta > 0$ is some constant. Let $\epsilon > \delta$ be some constant, then $H^{\mathbf{Metric}}_{\epsilon}(X) \geq \Omega(n)$ with respect to $O(s)$-sized **PH**-circuits.*

Proof. We prove the theorem by the contrapositive. Let $\epsilon > \delta$ and suppose that $H^{\mathbf{Metric}}_{\epsilon}(X) < k$ where $k = \epsilon' n$ (for a constant $\epsilon' > 0$ that will be chosen later).

[18] Lemma 19 only gives a predictor given a distinguisher D such that $\Pr_{x\leftarrow_R X}[x \in D] = 1$. However, the proof of Lemma 19 will still yield a predictor with constant bias even if 1 is replaced by $\frac{9}{10}$ (or any constant greater than $\frac{1}{2}$).

This means that there exists a set $D \in \mathcal{C}$ such that $\Pr_{x \leftarrow_{\mathrm{R}} X}[x \in D] \geq \frac{|D|}{2^k} + \epsilon$. In particular, this means that $|D| < 2^k$ and $\Pr_{x \leftarrow_{\mathrm{R}} X}[x \in D] \geq \epsilon$. We consider the following predictor P': On input $i \in [n]$ and $x = x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n \in \{0,1\}^{n-1}$, P' considers the strings x^0, x^1 where $x^b = x_1, \ldots, x_{i-1}, b, x_{i+1}, \ldots, x_n$. If both x^0 and x^1 are not in D, then P' outputs a random bit. If $x^b \in D$ and $x^{\overline{b}} \notin D$ then P' outputs b. Otherwise (if $x^0, x^1 \in D$), P' outputs $P(x_1, \ldots, x_{i-1})$, where P is the predictor constructed from D in the proof of Lemma 19. Let $\mathit{\Gamma}(D)$ denote the set of all strings x such that $x \notin D$ but x is of Hamming distance 1 from D (i.e., there is $i \in [n]$ such that $x_1, \ldots, x_{i-1}, \overline{x_i}, x_{i+1}, \ldots, x_n \in D$). If $S \subseteq \{0,1\}^n$, then let $X_{\upharpoonright S}$ denote the random variable $X | X \in S$. By Lemma 19 $\Pr_{i \in [i], x \leftarrow_{\mathrm{R}} X_{\upharpoonright D}}[P'(x_{[n] \setminus \{i\}}) = x_i] \geq 1 - O(\frac{k}{n})$ while it is clear that $\Pr_{i \in [i], x \leftarrow_{\mathrm{R}} X_{\upharpoonright \{0,1\}^n \setminus (D \cup \mathit{\Gamma}(D))}}[P'(x_{[n] \setminus \{i\}}) = x_i] = \frac{1}{2}$. Thus if it holds that $\Pr[X \in \mathit{\Gamma}(D)] < \epsilon'$ and $k < \epsilon' n$, where ϵ' is some small constant (depending on ϵ and δ) then $\Pr_{i \in [i], x \leftarrow_{\mathrm{R}} X}[P'(x_{[n] \setminus \{i\}}) = x_i] \geq \frac{1}{2} + \delta$ and the proof is finished.

However, it may be the case that $\Pr[X \in \mathit{\Gamma}(D)] \geq \epsilon'$. In this case, we will consider the distinguisher $D^{(1)} = D \cup \mathit{\Gamma}(D)$, and use $D^{(1)}$ to obtain a predictor $P^{(1)\prime}$ in the same way we obtained P' from D. Note that $|D^{(1)}| \leq n|D|$ and that, using non-determinism, the circuit size of $D^{(1)}$ is larger than the circuit size of D by at most a $O(\log n)$ additive factor.[19] We will need to repeat this process for at most $\frac{1}{\epsilon'}$ steps,[20] to obtain a distinguisher $D^{(c)}$ (where $c \leq \frac{1}{\epsilon'}$) such that $|D^{(c)}| \leq n^{O(1/\epsilon')}|D| \leq 2^{k + O(\log n (1/\epsilon'))}$, $\Pr[X \in D^{(c)}] \geq \epsilon$ and $\Pr[X \in \mathit{\Gamma}(D^{(c)})] < \epsilon'$. The corresponding predictor $P^{(c)\prime}$ will satisfy that $\Pr_{i \in [i], x \leftarrow_{\mathrm{R}} X}[P^{(c)\prime}(x_{[n] \setminus \{i\}}) = x_i] \geq \frac{1}{2} + \delta$ thus proving the theorem. □

Acknowledgements We thank Oded Goldreich and the RANDOM 2003 referees for helpful comments.

References

1. Goldwasser, S., Micali, S.: Probabilistic encryption. Journal of Computer and System Sciences **28** (1984) 270–299 Preliminary version in STOC' 82.
2. Yao, A.C.: Theory and applications of trapdoor functions. In: 23rd FOCS. (1982) 80–91
3. Blum, M., Micali, S.: How to generate cryptographically strong sequences of pseudo-random bits. SIAM Journal on Computing **13** (1984) 850–864
4. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM Journal on Computing **28** (1999) 1364–1396 (electronic)
5. Shannon, C.E.: A mathematical theory of communication. Bell System Technical Journal **27** (1948) 379–423, 623–656
6. Goldberg, A.V., Sipser, M.: Compression and ranking. SIAM Journal on Computing **20** (1991) 524–536
7. von Neumann, J.: Zur theorie der gesellschaftsspiele. Math. Ann. **100** (1928) 295–320

[19] To compute $D^{(1)}(x)$, guess $i \in [n], b \in \{0,1\}$ and compute $D(x')$ where x' is obtained from x by changing x_i to b.

[20] Actually, a tighter analysis will show that we only need $O(\log \frac{1}{\epsilon'})$ steps.

8. Klivans, A.R., van Melkebeek, D.: Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. SIAM J. Comput. **31** (2002) 1501–1526 (electronic)
9. Miltersen, P.B., Vinodchandran, N.V.: Derandomizing Arthur-Merlin games using hitting sets. In: 40th FOCS. (1999) 71–80
10. Shaltiel, R., Umans, C.: Simple extractors for all min-entropies and a new pseudo-random generator. In: 42nd FOCS. (2001) 648–657
11. Trevisan, L., Vadhan, S.: Extracting randomness from samplable distributions. In: 41st FOCS. (2000) 32–42
12. Nisan, N.: Extracting randomness: How and why: A survey. In: Conference on Computational Complexity. (1996) 44–58
13. Nisan, Ta-Shma: Extracting randomness: A survey and new constructions. JCSS: Journal of Computer and System Sciences **58** (1999)
14. Shaltiel, R.: Recent developments in explicit constructions of extractors. Bulletin of the European Association for Theoretical Computer Science **77** (2002) 67– Also available on `http://www.wisdom.weizmann.ac.il/~ronens`.
15. Impagliazzo, R., Levin, L.A., Luby, M.: Pseudo-random generation from one-way functions. In: 21st STOC. (1989) 12–24
16. Sudan, M., Trevisan, L., Vadhan, S.: Pseudorandom generators without the XOR lemma. JCSS: Journal of Computer and System Sciences **62** (2001) Preliminary version in STOC' 99. Also published as ECCC Report TR98-074.
17. Trevisan, L.: Construction of extractors using pseudo-random generators. In: 31st STOC. (1999) 141–148
18. Raz, R., Reingold, O., Vadhan, S.: Extracting all the randomness and reducing the error in trevisan's extractors. JCSS: Journal of Computer and System Sciences **65** (2002) Preliminary version in STOC' 99.
19. Impagliazzo, R., Shaltiel, R., Wigderson, A.: Extractors and pseudo-random generators with optimal seed length. In ACM, ed.: 32nd STOC. (2000) 1–10
20. Ta-Shma, A., Zuckerman, D., Safra, S.: Extractors from Reed-Muller codes. In IEEE, ed.: 42nd FOCS. (2001) 638–647
21. Babai, L., Fortnow, L., Nisan, N., Wigderson, A.: BPP has subexponential time simulations unless EXPTIME has publishable proofs. Computational Complexity **3** (1993) 307–318
22. Impagliazzo, R., Wigderson, A.: P = BPP if E requires exponential circuits: Derandomizing the XOR lemma. In: 29th STOC. (1997) 220–229
23. Nisan, N., Wigderson, A.: Hardness vs. randomness. J. Comput. System Sci. **49** (1994) 149–167
24. Nisan, N.: Pseudorandom generators for space-bounded computations. In ACM, ed.: 22nd STOC. (1990) 204–212
25. Impagliazzo, R., Nisan, N., Wigderson, A.: Pseudorandomness for network algorithms. In ACM, ed.: 26th STOC. (1994) 356–364
26. Bar-Yossef, Z., Reingold, O., Shaltiel, R., Trevisan, L.: Streaming computation of combinatorial objects. In: Conference on Computational Complexity (CCC). Volume 17. (2002)
27. Nisan, N., Zuckerman, D.: Randomness is linear in space. Journal of Computer and System Sciences **52** (1996) 43–52 Preliminary version in STOC' 93.
28. Saks, M.: Randomization and derandomization in space-bounded computation. In: Conference on Computational Complexity (CCC). (1996) 128–149
29. Jerrum, M.R., Valiant, L.G., Vazirani, V.V.: Random generation of combinatorial structures from a uniform distribution. Theoretical Computer Science **43** (1986) 169–188