On One-Way Functions and Sparse Languages

Yanyi Liu[1](✉) and Rafael Pass[2]

[1] Cornell Tech, New York, USA
ttyl2866@cornell.edu
[2] Tel-Aviv University & Cornell Tech, Tel Aviv-Yafo, Israel
ttrafaelp@tau.ac.il

Abstract. We show equivalence between the existence of one-way functions and the existence of a *sparse* language that is hard-on-average w.r.t. some efficiently samplable "high-entropy" distribution. In more detail, the following are equivalent:

- The existence of a $S(\cdot)$-sparse language L that is hard-on-average with respect to some samplable distribution with Shannon entropy $h(\cdot)$ such that $h(n) - \log(S(n)) \geq 4 \log n$;
- The existence of a $S(\cdot)$-sparse language $L \in \mathsf{NP}$, that is hard-on-average with respect to some samplable distribution with Shannon entropy $h(\cdot)$ such that $h(n) - \log(S(n)) \geq n/3$;
- The existence of one-way functions.

where a language L is said to be $S(\cdot)$-sparse if $|L \cap \{0,1\}^n| \leq S(n)$ for all $n \in \mathbb{N}$. Our results are inspired by, and generalize, results from the elegant recent paper by Ilango, Ren and Santhanam (IRS, STOC'22), which presents similar connections for *specific* sparse languages.

1 Introduction

A *one-way function* [4] (OWF) is a function f that can be efficiently computed (in polynomial time), yet no probabilistic polynomial-time (PPT) algorithm can invert f with inverse polynomial probability for infinitely many input lengths n. Whether one-way functions exist is unequivocally the most important open problem in Cryptography (and arguably the most important open problem in the theory of computation, see e.g., [18]): OWFs are both necessary [15] and sufficient for many of the most central cryptographic primitives and protocols (e.g., pseudorandom generators [2,10], pseudorandom functions [6], private-key encryption [7], digital signatures [22], commitment schemes [20], identification

Supported by a JP Morgan fellowship.
Supported in part by NSF Award CNS 2149305, NSF Award CNS-2128519, NSF Award RI-1703846, AFOSR Award FA9550-18-1-0267, FA9550-23-1-0312, a JP Morgan Faculty Award, the Algorand Centres of Excellence programme managed by Algorand Foundation, and DARPA under Agreement No. HR00110C0086. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government, DARPA or the Algorand Foundation.

G. Rothblum and H. Wee (Eds.): TCC 2023, LNCS 14369, pp. 219–237, 2023.
https://doi.org/10.1007/978-3-031-48615-9_8

protocols [5], coin-flipping protocols [1], and more). These primitives and protocols are often referred to as *private-key primitives*, or "Minicrypt" primitives [13] as they exclude the notable task of public-key encryption [4,21]. Additionally, as observed by Impagliazzo [8,13], the existence of a OWF is equivalent to the existence of polynomial-time method for sampling hard *solved* instances for an NP language (i.e., hard instances together with their witnesses).

A central open question at the intersection of Cryptography and Complexity-theory, however, is whether the existence of just an average-case hard problem in NP suffices to get the existence of OWFs:

> *Does the existence of a language in* NP *that is hard-on-average imply the existence of OWFs?*

(In Impagliazzo's language, can we rule out "Pessiland"—a world where NP is hard on average but OWFs do not exist.) There has been some recent progress towards this question. Most notably, Liu and Pass [19] recently showed that (mild) average-case hardness, w.r.t. the uniform distribution of instances, of a particular natural problem in NP—the time-bounded Kolmogorov Complexity problem [9,16,17,23,24]—characterizes the existence of OWFs. This problem, however is not average-case complete for NP so it does not resolve the above question.

In this work, our goal is to identify properties of languages such that their average-case hardness implies OWFs:

> *Can we identify simple/natural properties of a distribution-language pair* $(\mathcal{D}, L)$ *such that average-case hardness of* L *with respect to distribution* $\mathcal{D}$ *implies the existence of OWFs?*

Our starting point towards answering this problem is an elegant recent work by Ilango, Ren and Santhanam [11,12] (IRS). IRS first show that the existence of OWFs is equivalent to average-case hardness of a Gap version of the Kolmogorov complexity problem w.r.t. *any* efficiently computable distribution. In a second step, they next show that average-case hardness of some specific sparse languages implies average-case hardness of this Gap problem.

In more detail, their first step shows that OWFs exist iff there exists some samplable distribution $\mathcal{D}$ and efficiently computable thresholds t_0, t_1, $t_1(n) - t_0(n) > \omega(\log n)$, so that it is hard to decide whether $K(x) > t_1(|x|)$ or $K(x) < t_0(|x|)$. Let us highlight that this characterization differs from the one in [19] in three aspects: (1) it considers unbounded, as opposed to time-bounded Kolmogorov complexity, (2) hardness holds w.r.t. to a gap problem, as opposed to a decisional problem, and (3) it considers hardness w.r.t. *any* efficient distribution, as opposed to the uniform distribution considered in [19]. (In particular, this result does not provide a candidate distribution for which the Gap problem may be hard—and it is provably *easy* with respect to the uniform distribution.) In the second step, they present some concrete languages (k-SAT and t-Clique) such that average-case hardness of these languages with respect to high-entropy distributions implies (but does not characterize) the existence of OWFs.

In this work, we show how to generalize the results obtained in the second step and to demonstrate that the existence of OWFs is equivalent to the existence of a *sparse* language that is hard-on-average w.r.t. some efficiently samplable "high-entropy" distribution. In more details, the Shannon entropy of the sampler needs to be just slightly bigger than the logarithm of the density of the language.

As a result of independent interest, we additionally show how to generalize the results of IRS in their Step 1 with respect to K-complexity (but note that the results with respect to sparse languages no longer pass through this result).[1]

1.1 Our Results

To formalize the statements of our results, let us briefly state some preliminaries.

Preliminaries We say that a language $L \subset \{0,1\}^*$ is $S(\cdot)$-*sparse* if for all $n \in \mathbb{N}$, $|L_n| \leq S(n)$, where $L_n = |L \cap \{0,1\}^n|$. Given a language L, we abuse the notation and let $L(x) = 1$ iff $x \in L$. For a random variable X, let $H(X) = \mathsf{E}[\log \frac{1}{\Pr[X=x]}]$ denote the Shannon entropy of X. A function μ is said to be *negligible* if for every polynomial $p(\cdot)$ there exists some n_0 such that for all $n > n_0$, $\mu(n) \leq \frac{1}{p(n)}$.

We say that $\mathcal{D} = \{D_n\}_{n \in \mathbb{N}}$ is an *ensemble* if for all $n \in \mathbb{N}$, D_n is a probability distribution over $\{0,1\}^n$. We say that an ensemble $\mathcal{D} = \{D_n\}_{n \in \mathbb{N}}$ is *samplable* if there exists a probabilistic polynomial-time Turing machine S such that $S(1^n)$ samples D_n; we use the notation $S(1^n; r)$ to denote the algorithm S with randomness fixed to r. We say that an ensemble $\mathcal{D}$ has entropy $h(\cdot)$ if for all sufficiently large $n \in \mathbb{N}$, $H(D_n) \geq h(n)$.

We say that a language $L \subset \{0,1\}^*$ is $\alpha(\cdot)$ *hard-on-average (α-HoA) on an ensemble* $\mathcal{D} = \{D_n\}_{n \in \mathbb{N}}$ if for all probabilistic polynomial-time heuristics $\mathcal{H}$, for all sufficiently large $n \in \mathbb{N}$,

$$\Pr[x \leftarrow D_n : \mathcal{H}(x) = L(x)] < 1 - \alpha(n).$$

We simply say that L is *hard-on-average (HoA)* on $\mathcal{D}$ if for every c, $\alpha(n) = \frac{1}{2} - \frac{1}{n^c}$, L is α-HoA.

Let $f : \{0,1\}^* \to \{0,1\}^*$ be a polynomial-time computable function. f is said to be a *one-way function (OWF)* if for every PPT algorithm $\mathcal{A}$, there exists a negligible function μ such that for all $n \in \mathbb{N}$,

$$\Pr[x \leftarrow \{0,1\}^n; y = f(x) : A(1^n, y) \in f^{-1}(f(x))] \leq \mu(n)$$

Main Theorem We are now ready to state our main theorem.

Theorem 11 *The following are equivalent:*

1. *The existence of a $S(\cdot)$-sparse language L that is $(\frac{1}{2} - \frac{1}{4n})$-HoA with respect to some samplable distribution with Shannon entropy $h(\cdot)$ such that $h(n) - \log(S(n)) \geq 4 \log n$;*

[1] It appears that a similar generalization was concurrently and independently obtained by IRS in the proceedings version [12]; see Sect. 5 for more details.

2. *The existence of a* $S(\cdot)$*-sparse language* $L \in \mathsf{NP}$*, that is HoA with respect to some samplable distribution with Shannon entropy* $h(\cdot)$ *such that* $h(n) - \log(S(n)) \geq n/3$*;*
3. *The existence of one-way functions.*

Theorem 11 is proven by, in Sect. 2 showing that (1) implies (3), and in Sect. 3 showing that (3) implies (2); the fact that (2) implies (1) is trivial. We present some corollaries of Theorem 11 in Sect. 4. In Sect. 5, we finally present some result of independent interest that generalize the result in [11] with respect to the particular K-complexity problem—in particular, we strengthen the result from [11] to show that that it suffices to assume hardness of approximating K-complexity (as opposed to assuming hardness of deciding a threshold version of a Gap-K problem).

1.2 Proof Overview

To explain the proof of our results, and to put it in context, let us start by reviewing the results of Ilango, Ren and Santhanam (IRS) [11].

IRS Part 1: OWFs from Hardness of Gap-K. As mentioned, IRS first show that OWFs exist iff there exists some samplable distribution $\mathcal{D}$ and efficiently computable thresholds t_0, t_1 where $t_1(n) - t_0(n) > \omega(\log n)$ so that it is hard to decide whether $K(x) > t_1(|x|)$ or $K(x) < t_0(|x|)$ when sampling x from $\mathcal{D}$. We here focus only on the "if" direction which will be most relevant to us.[2] On a high level, the IRS result is obtained by showing that any sampler that makes this Gap problem hard must itself be a OWF. In more detail, they first appeal to the result of [14,15] showing that if OWFs do not exist, then approximate counting is possible on average. They next show how to use an approximate counter to solve the Gap-K problem: Given an instance x, approximately count the number of random strings r that lead the sampler $\mathcal{D}$ (given randomness r) to generate x. If the number is "small"—we refer to such strings x as rare, where "small" is appropriately defined as a function of $t_1(|x|)$ (which, recall, is required to be efficiently computable), then output NO (i.e., that the K-complexity is large), and otherwise (i.e., if x is common) output YES.

It remains to analyze that this deciding algorithm works (on average). The key observation is that common instances x must be YES-instances: their K-complexity must be small simply by enumerating all common strings (since there can only be a small number of them!). Thus (whenever the approximate counter is correct), the decider always gives the right answer on NO-instances. On the other hand, since YES-instances are sparse, it directly follows by a Union bound, that the probability that we sample a YES-instance that is rare must be small. Consequently, the decider will also give the right answer on YES-instances with high probability. This concludes the existence of OWFs assuming the hardness of the Gap-K problem.

[2] The only-if direction is a direct consequence of [10].

IRS Part 2: OWF from Specific Sparse Languages. In the second part of their paper, IRS next present some concrete languages—k-SAT and t-Clique—and show that average-case hardness with respect to *high-entropy* distributions of these concrete languages imply hardness of Gap-K (which in turn by the first result implies OWFs). This argument relies on the following three steps:

- **Step 1** (Language-specific): Proving—*using language specific structures*—that YES instances have small K-complexity.
- **Step 2** (Generic): Rely on a generic counting argument (following a similar statement in [19]) to show that elements sampled from any high-entropy distribution need to have high K-complexity with reasonable (roughly $1/n$) probability.
- **Step 3** (Language-specific): Finally, to argue that average-case hardness of these languages w.r.t. any high-entropy distribution implies average-case hardness of Gap-K, we additionally need to argue that the thresholds t_0, t_1 for the K-complexity problem are efficiently computable. Another language specific argument is used to show that the number of YES-instances in these languages can be efficiently estimated and this can be used to give the thresholds.

Towards Sparse Languages: A Warm-Up We start by observing that Step 1, in fact, holds for *any* sparse language that is *decideable*, or even *recursively enumerable*: If the language is sparse and recursively enumerable, then we can simply compress an instance by writing down its index, so YES-instances need to have small K-complexity. We additionally note that if the sparsity threshold, $S(\cdot)$, is efficiently computable, the thresholds t_0, t_1 for the Gap-K problem also become efficiently computable and we can also carry out Step 3 (and Step 2 is obviously generic). Thus, relying on these observations, we can directly obtain a *weaker* version of Theorem 11 by appealing to the results of [11]. Let us highlight, however, that this version is weaker in two important ways:

1. We require the sparse language to be recursively enumerable (to deal with Step 1).
2. We require the sparsity threshold to be efficiently computable (to deal with Step 3).

Proving the Full Result To remove the above two restrictions, our key observation is that passing through K-complexity may not be the right approach. Rather, we can directly redo Part 1 of IRS (i.e., decide the language using an approximate counter) for any sparse language w.r.t. to a high-entropy distribution. Our decider proceeds as follows given an instance x:

- Just as IRS, use the approximate counter to check if the string x is rare (i.e., that there is a small number of random coins r for $\mathcal{D}$ that generate x).
- If x is deemed rare, then output NO, and otherwise output a *random guess* (as opposed to outputting YES as in IRS).

In the above approach, we still need to define the threshold for what counts as rare. To do this, we note that we can use approximate counting to estimate the Shannon entropy of any efficiently sampleable distribution (see Lemma 25), and we can use the (estimated) Shannon entropy as the threshold for determining when to deem a string rare. More precisely, we call a string x rare if it is sampled by $\mathcal{D}$ with probability $\leq 2^{-h+3}$, where h is the Shannon entropy of $\mathcal{D}$.

To argue that this approach works, we proceed as follows:

- We first note that any distribution $\mathcal{D}$ needs to output strings that are rare (where recall, rare is defined w.r.t. the Shannon-entropy of $\mathcal{D}$) with probability $1/n$ (See Lemma 22). (This statement is a stronger version of a result shown in [19], and relies on essentially the same proof as used in [11] to argue that high-entropy distributions output strings with high K-complexity with reasonable probability.)
- We next argue that conditioned on $\mathcal{D}$ sampling a rare instance, our decider succeeds with high probability. First, note that the decider always outputs NO on rare instances (unless the approximate counter fails, which happens with small probability so we can ignore this event). Next, by the sparsity of the language and the Union bound, we have that the probability that $\mathcal{D}$ samples a YES-instance that is rare is tiny (technically, $\leq 1/n^2$) (see Lemma 23). But since the probability that $\mathcal{D}$ samples a rare instance is a lot larger, we have that our decider succeeds with high probability conditioned on rare instances.
- On common instances, our decider succeeds with probability $1/2$ (again, as long as the approximate counter does not fail, which happens with tiny probability). So, we conclude that the decider succeeds with probability roughy $1/2 + 1/(2n)$.

This concludes that (1) implies (3) in Theorem 11. To show that (3) implies (2) we simply note that one-way functions imply pseudo-random generators (PRG) by [10], and by considering the language of images of the PRG (which is extremely sparse) and the distribution that with probability $1/2$ samples a random string and with probability $1/2$ samples an image of the PRG (which has Shannon-entropy entropy $1/2n$); this language is hard-on-average on this distribution by the security of the PRG.

Concluding Corollaries for Concrete Languages We finally observe—using standard arguments—that the languages considered in [11] (k-SAT and t-Clique) are sparse, and so is the language of strings with small K-complexity. See Sect. 4 for more details.

In our view, these results show that for many of the corollaries of [11], K-complexity was perhaps a mirage, and in our eyes, sparsity is the central feature. We note that a similar phenomena actually happened also with respect to the vein of work on "hardness magnification", as shown in the elegant work by Chen, Jin, Williams [3].

Musings on the Relevance of our Results The reader may wonder why it matters to deal with non-recursively enumerable languages and with nonefficiently computable sparsity. After all, the natural sparse languages we consider in Sect. 4 are both recursively enumerable and have efficiently computable sparsity. In our opinion, the difference is significant. In particular, removing these restriction opens up for the possibility of using a probabilistic argument to define a candidate language that is hard-on-average. Probabilistic arguments are typically used for proving lower-bounds and our hope is that our result opens up the avenue for using such techniques.

Back to K-complexity Motivated by the above results, one may wonder whether the efficient computability condition in the results of [11] w.r.t. K-complexity is inherent (i.e., whether the efficient computability of the thresholds t_0, t_1 in the Gap-K problem is inherent). As a result of independent interest, we show how to strengthen the result of IRS to show that it suffices to assume average-case hardness of *approximately computing* K-complexity within an additive term of $\omega(\log n)$ to deduce the existence of one-way function (i.e., that hardness of the search version suffices, and thus we no longer need to consider any thresholds).[3]

Theorem 12 *The following are equivalent:*

- *One-way functions exist;*
- *There exists some efficiently samplable distribution $\mathcal{D}$ such that K-complexity is mildly hard to approximate within an additive term of $\omega(\log n)$.*
- *There exists some efficiently samplable distribution $\mathcal{D}$ such that K-complexity is hard to approximate within an additive term of $n - n^{o(1)}$.*

Let us first compare this result to IRS; the result is strictly stronger as our hardness of approximating K-complexity assumption is trivially implied the decisional Gap-K hardness condition considered in IRS. In fact, as a corollary of this result (of independent interest), we get a decision-to-search reduction for K-complexity (for efficiently computable thresholds); See Theorem 51 for more details.

It is also worthwhile to compare it to the results of [19]; here the results is incomparable. [19] shows that mild average-case hardness of *time-bounded* Kolmogorov complexity (even to approximate) with respect to the *uniform distribution* characterizes OWF. We note that K-complexity (and also time-bounded K-complexity) is *easy* to approximate within an additive factor of $\omega(\log n)$ with overwhelming probability with respect to the uniform distribution so it was crucial for [19] that an approximate factor of $O(\log n)$ was employed. Theorem 12 thus cannot hold w.r.t. the uniform distribution, and just as the result in IRS, it gives no indication of what the hard distribution may be—in fact, as mentioned before, the distribution $\mathcal{D}$ gives the OWF.

[3] As mentionned above, it appears that a similar generalization was concurrently and independently obtained by IRS in the proceedings version [12]; see Sect. 5 for more details.

2 OWFs from Avg-Case Hardness of Sparse Languages

Theorem 21 *Let $S(\cdot)$ be a function, let $h(n) \geq \log S(n) + 4 \log n$, and let L be a $S(\cdot)$-sparse language. Assume there exists some samplable ensemble $\mathcal{D}$ with entropy $h(\cdot)$ such that L is $(\frac{1}{2} - \frac{1}{4n})$-HoA on $\mathcal{D}$. Then, one-way functions exist.*

Before proving the theorem, we will state some useful lemmas.

Lemma 22 (Implicit in [11,19]) *Let D_n be a distribution over $\{0,1\}^n$ with entropy at least h. Then, with probability at least $\frac{1}{n}$ over $x \leftarrow D_n$, it holds that*

$$\Pr[D_n = x] \leq 2^{-h+3}$$

Proof. Assume for contradiction that with probability less than $\frac{1}{n}$ over $x \leftarrow D_n$, $\Pr[D_n = x] \leq 2^{-h+3}$. Let Freq denote the set of strings $x \subseteq \{0,1\}^n$ such that $\Pr[D_n = x] > 2^{-h+3}$, and let Rare denote the set of strings $\subseteq \{0,1\}^n$ such that $\Pr[D_n = x] \leq 2^{-h+3}$. Let flag be a binary random variable such that $\mathsf{flag} = 0$ if $D_n \in \mathsf{Freq}$ and 1 otherwise (i.e. if $D_n \in \mathsf{Rare}$). Let p_{Freq} be the probability that $D_n \in \mathsf{Freq}$ and p_{Rare} be the probability that $D_n \in \mathsf{Rare}$. By the chain rule for entropy, it holds that

$$\begin{aligned} H(D_n) &\leq H(D_n, \mathsf{flag}) \\ &= H(\mathsf{flag}) + p_{\mathsf{Freq}} H(D_n \mid D_n \in \mathsf{Freq}) + p_{\mathsf{Rare}} H(D_n \mid D_n \in \mathsf{Rare}) \end{aligned}$$

In the RHS, the first term is at most 1 (since flag is a binary variable). The second term is at most $h-3$ since $|\mathsf{Freq}| \leq 2^{h-3}$. Recall that by assumption, we have that $p_{\mathsf{Rare}} < \frac{1}{n}$; furthermore, $H(D_n \mid D_n \in \mathsf{Rare}) \leq n$ (since $|\mathsf{Rare}| \leq 2^n$) and thus the last term of the RHS is at most 1. Therefore, $H(D_n) \leq 1 + (h-3) + 1 < h$, which is a contradiction.

Lemma 23 *Let $L_n \subset \{0,1\}^n$ be a set of strings such that $|L_n| \leq S(n)$. Let D_n be a distribution over $\{0,1\}^n$. Let ε be any number satisfying $\varepsilon \leq \frac{1}{S(n)n^2}$. Then, the following holds:*

$$\Pr_{x \leftarrow D_n}[L_n(x) = 1 \wedge \Pr[D_n = x] \leq \varepsilon] \leq \frac{1}{n^2}$$

Proof. By taking a union bound over the at most $S(n)$ instances in L_n, it follows that $\Pr_{x \leftarrow D_n}[L_n(x) = 1 \wedge \Pr[D_n = x] \leq \varepsilon]$ is bounded by $S(n) \times \frac{1}{S(n)n^2} = \frac{1}{n^2}$.

We will rely on the following important lemma showing that approximate counting can be efficiently done infinitely often if one-way functions do not exist.

Lemma 24 **([11,14,15])** *Assume that one-way functions do not exist. Then, for any samplable ensemble $\mathcal{D} = \{D_n\}_{n \in \mathbb{N}}$ and any constant $q \geq 1$, there exist a* PPT *algorithm $\mathcal{A}$ and a constant $\delta > 0$ such that for infinitely many n,*

$$\Pr_{x \leftarrow D_n}[\delta \cdot p_x \leq \mathcal{A}(x) \leq p_x] \geq 1 - \frac{1}{n^q}$$

where $p_x = \Pr[D_n = x]$.

In addition, we observe that if approximate counting can be done, the Shannon entropy of any samplable distribution $\mathcal{D}$ can be estimated efficiently.

Lemma 25 *Let $\mathcal{D} = \{D_n\}_{n\in\mathbb{N}}$ be a samplable ensemble, let* Samp *be the corresponding sampler, and let $m(\cdot)$ be a polynomial such that $m(n)$ is greater than the number of random coins used by* $\mathsf{Samp}(1^n)$*. Assume that there exist a* PPT *algorithm $\mathcal{A}$, a constant δ, and an infinite set $I \subseteq \mathbb{N}$ such that for all $n \in I$,*

$$\Pr_{x\leftarrow D_n}[\delta \cdot p_x \leq \mathcal{A}(x) \leq p_x] \geq 1 - \frac{1}{m(n)}$$

where $p_x = \Pr[D_n = x]$. Then, there exist a PPT *algorithm* est *and a constant δ' such that for all $n \in I$, with probability at least $1 - \frac{1}{n^2}$,*

$$|\mathsf{est}(1^n) - H(D_n)| \leq \delta'$$

Proof. Let $n \in I$ be a sufficiently large input length on which $\mathcal{A}$ succeeds, and let $m = m(n)$. Let p_x denote $\Pr[D_n = x]$. Let $\mathcal{A}'$ be the algorithm defined as $\mathcal{A}'(x) = \max(2^{-m}, \min(1, \mathcal{A}(x)))$. $\mathcal{A}'$ will have the same property that $\mathcal{A}$ has in the assumption since for all x in the support of D_n, it holds that $2^{-m} \leq p_x \leq 1$. We first claim that

$$|\mathsf{E}_{x\leftarrow D_n}[-\log \mathcal{A}'(x)] - H(D_n)| \leq -\log\delta + 1 \tag{1}$$

If this holds, note that $\mathcal{D}$ is samplable and $\mathcal{A}'$ runs in PPT, it follows that we can empirically estimate $\mathsf{E}_{x\leftarrow D_n}[-\log \mathcal{A}'(x)]$ in polynomial time by collecting at least polynomially many samples and taking the average. By Hoeffding's inequality, the difference between this estimation and the real expectation value is at most 1 with very high probability ($\geq 1 - \frac{1}{n^2}$).

Thus, it remains to show that inequality 1 holds. Notice that

$$\begin{aligned}
&|\mathsf{E}_{x\leftarrow D_n}[-\log \mathcal{A}'(x)] - H(D_n)| \\
=&|\mathsf{E}_{x\leftarrow D_n}[-\log \mathcal{A}'(x)] - \mathsf{E}_{x\leftarrow D_n}[-\log p_x]| \\
\leq&\mathsf{E}_{x\leftarrow D_n}[|-\log \mathcal{A}'(x) - (-\log p_x)|] \\
=&\Pr_{x\leftarrow D_n}[\mathcal{A}' \text{ succeeds}] \cdot \mathsf{E}_{x\leftarrow D_n}[|-\log \mathcal{A}'(x) - (-\log p_x)| \mid \mathcal{A}' \text{ succeeds}] \\
&+ \Pr_{x\leftarrow D_n}[\mathcal{A}' \text{ fails}] \cdot \mathsf{E}_{x\leftarrow D_n}[|\log \mathcal{A}'(x) - (-\log p_x)| \mid \mathcal{A}' \text{ fails}] \\
\leq&\mathsf{E}_{x\leftarrow D_n}[|\log \frac{p_x}{\mathcal{A}'(x)}| \mid \mathcal{A}' \text{ succeeds}] + \frac{1}{m} \cdot m \\
\leq&\mathsf{E}_{x\leftarrow D_n}[-\log\delta \mid \mathcal{A}' \text{ succeeds}] + 1 \\
\leq&-\log\delta + 1
\end{aligned}$$

Now we are ready to prove Theorem 21.

Proof (Proof of Theorem 21). Assume for contradiction that one-way functions do not exist. Then, by Lemma 24, there exist a PPT algorithm $\mathcal{A}$ and a constant δ such that for infinitely many n,

$$\Pr_{x\leftarrow D_n}[\delta \cdot p_x \leq \mathcal{A}(x) \leq p_x] \geq 1 - \frac{1}{n^2}$$

where $p_x = \Pr[D_n = x]$. By Lemma 25, there exist a PPT algorithm est and a constant δ' such that for all n on which $\mathcal{A}$ succeeds, with probability at least $1 - \frac{1}{n^2}$,

$$|\mathsf{est}(1^n) - H(D_n)| \le \delta' \tag{2}$$

Consider some sufficiently large input length n on which $\mathcal{A}$ succeeds. Let the random variable

$$\varepsilon = 2^{-\mathsf{est}(1^n)+\log n}$$

We are now ready to describe our heuristic $\mathcal{H}$ for L. On input $x \leftarrow D_n$, $\mathcal{H}$ computes ε and outputs 0 if $\mathcal{A}(x) \le \varepsilon$; otherwise, $\mathcal{H}$ outputs a random guess $b \in \{0,1\}$. We will show that $\mathcal{H}$ solves L with probability $\frac{1}{2} + \frac{1}{4n}$ on the input length n (whenever n is sufficiently large).

Towards this, let us first assume we have access to a "perfect" approximate-counter algorithm $\mathcal{O}$ such that $\delta \cdot p_x \le \mathcal{O}(x) \le p_x$ with probability 1 when x sampled from D_n; let us also assume we have access to a "perfect" entropy-estimator algorithm est^* such that $|\mathsf{est}^*(1^n) - H(D_n)| \le \delta'$ with probability 1; consider the heuristic $\mathcal{H}'$ that behaves just as $\mathcal{H}$ except that $\mathcal{H}'$ uses $\mathcal{O}$ and est^* instead of $\mathcal{A}$ and est.

We first show that $\mathcal{H}'$ solves L with high probability on D_n. Note that

$$\begin{aligned}
&\Pr_{x\leftarrow D_n}[\mathcal{H}'(x) = L(x)]\\
=&\Pr_{x\leftarrow D_n}[\mathcal{H}'(x) = L(x) \mid \mathcal{O}(x) > \varepsilon]\Pr[\mathcal{O}(x) > \varepsilon]\\
&+\Pr_{x\leftarrow D_n}[\mathcal{H}'(x) = L(x) \mid \mathcal{O}(x) \le \varepsilon]\Pr[\mathcal{O}(x) \le \varepsilon]\\
=&\frac{1}{2}(1 - \Pr[\mathcal{O}(x) \le \varepsilon]) + \left(1 - \Pr_{x\leftarrow D_n}[\mathcal{H}'(x) \ne L(x) \mid \mathcal{O}(x) \le \varepsilon]\right)\Pr[\mathcal{O}(x) \le \varepsilon]\\
=&\frac{1}{2}(1 - \Pr[\mathcal{O}(x) \le \varepsilon]) + \left(1 - \Pr_{x\leftarrow D_n}[L(x) = 1 \mid \mathcal{O}(x) \le \varepsilon]\right)\Pr[\mathcal{O}(x) \le \varepsilon]\\
=&\frac{1}{2} + \frac{1}{2}\Pr[\mathcal{O}(x) \le \varepsilon] - \Pr_{x\leftarrow D_n}[L(x) = 1 \mid \mathcal{O}(x) \le \varepsilon]\Pr[\mathcal{O}(x) \le \varepsilon]\\
=&\frac{1}{2} + \frac{1}{2}\Pr[\mathcal{O}(x) \le \varepsilon] - \Pr_{x\leftarrow D_n}[L(x) = 1 \wedge \mathcal{O}(x) \le \varepsilon]
\end{aligned}$$

Note that $p_x \le \varepsilon$ implies $\mathcal{O}(x) \le \varepsilon$ (since $\mathcal{O}$ is a prefect approximate-counter). In addition, for sufficiently large n, $p_x \le 2^{-H(D_n)+3}$ implies $p_x \le \varepsilon$ since

$$2^{-H(D_n)+3} \le 2^{-\mathsf{est}^*(1^n)+\delta'+3} \le 2^{-\mathsf{est}^*(1^n)+\log n} = \varepsilon.$$

Thus,

$$\Pr[\mathcal{O}(x) \le \varepsilon] \ge \Pr_{x\leftarrow D_n}[p_x \le \varepsilon] \ge \Pr_{x\leftarrow D_n}[p_x \le 2^{-H(D_n)+3}] \ge \frac{1}{n}$$

where the last inequality follows from by Lemma 22.

Next, observe that $\varepsilon/\delta \leq \frac{1}{S(n)n^2}$ (for sufficiently large n). This follows since if n is sufficiently large, we have:

$$\begin{aligned}\varepsilon = 2^{-\mathsf{est}^*(1^n)+\log n} \leq & 2^{-H(D_n)+\delta'+\log n} = 2^{-H(D_n)+\log n} \cdot 2^{\delta'} \\ \leq & 2^{-H(D_n)+\log n} \cdot \delta n = 2^{-H(D_n)+2\log n}\delta \\ \leq & 2^{-h(n)+2\log n}\delta \leq 2^{-\log S(n)-4\log n+2\log n}\delta \\ = & \frac{\delta}{S(n)n^2}\end{aligned}$$

Finally, since $p_x \leq \mathcal{O}(x)/\delta$ holds with probability 1, it follows that

$$\Pr_{x\leftarrow D_n}[L(x)=1 \wedge \mathcal{O}(x) \leq \varepsilon] \leq \Pr_{x\leftarrow D_n}[L(x)=1 \wedge p_x \leq \varepsilon/\delta] \leq \frac{1}{n^2}$$

where the last inequality follows from Lemma 23 and the fact that $\varepsilon/\delta \leq \frac{1}{S(n)n^2}$. Thus, we conclude that

$$\Pr_{x\leftarrow D_n}[\mathcal{H}'(x) = L(x)] \geq \frac{1}{2} + \frac{1}{2}\cdot\frac{1}{n} - \frac{1}{n^2}$$

We now turn to analyzing $\mathcal{H}$ as opposed to $\mathcal{H}'$ and note that $\mathcal{H}$ and $\mathcal{H}'$ work identically the same except when either $\mathcal{A}$ or est "fail". Observe that the probability that $\mathcal{A}(x) \neq \mathcal{O}(x)$ on x sampled from D_n is at most $\frac{1}{n^2}$. Additionally, the probability that $|\mathsf{est}(1^n) - H(D_n)| > \delta'$ is at most $\frac{1}{n^2}$. Thus, by a union bound, we have that

$$\Pr_{x\leftarrow D_n}[\mathcal{H}(x) = L(x)] \geq \frac{1}{2} + \frac{1}{2n} - \frac{3}{n^2} \geq \frac{1}{2} + \frac{1}{4n}$$

on infinitely many $n \in \mathbb{N}$, which is a contradiction.

3 Avg-Case Hardness of Sparse Languages from OWFs

Theorem 31 *Assume the existence of one-way functions. Let $S(n) = 2^{n/10}$ and $h(n) = n/2$. Then there exists a $S(\cdot)$-sparse language $L \in \mathsf{NP}$ and a samplable ensemble $\mathcal{D}$ with entropy $h(\cdot)$ such that L is HoA on $\mathcal{D}$.*

Proof. Assume the existence of OWFs. By [10], there exists some pseudorandom generator $g : \{0,1\}^{n/10} \to \{0,1\}^n$. Consider the NP-language $L = \{g(s) \mid s \in \{0,1\}^*\}$. Note that L is $S(\cdot)$-sparse for $S(n) = 2^{n/10}$. Let $\mathcal{D} = \{D_n\}_{n\in\mathbb{N}}$ be an ensemble such that D_n samples from $g(\mathcal{U}_{n/10})$ with probability 1/2 and from $\mathcal{U}_n$ with probability 1/2. Note that $\mathcal{D}$ has entropy at least $h(n) = n/2$ (since with probability 1/2, we sample from $\mathcal{U}_n$). Finally, it follows from the pseudorandomness property of g (using a standard argument) that L is HoA over $\mathcal{D}$.

4 Corollaries

In this section, we present some direct corollaries that follow by applying our main theorem to known sparse languages. For convenience of the reader, we recall the (standard) proofs that these languages are sparse.

4.1 Kolmogorov Complexity

The Kolmogorov complexity (K-complexity) of a string $x \in \{0,1\}^*$ is defined to be the length of the shortest program Π that outputs the string x. More formally, let U be a fixed Universal Turing machine, for any string $x \in \{0,1\}^*$, we define $K(x) = \min_{\Pi \in \{0,1\}^*} \{|\Pi| : U(\Pi) = x\}$. Let $\mathsf{MINK}[s]$ denote the language of strings x having the property that $K(x) \leq s(|x|)$. We observes that $\mathsf{MINK}[s]$ is a sparse language when $s(n)$ is slightly below n.

Lemma 41 *For all $n \in \mathbb{N}$, $|\mathsf{MINK}[s] \cap \{0,1\}^n| \leq 2^{s(n)+1}$.*

Proof. The lemma directly follows from the fact that the number of strings with length $\leq s(n)$ is at most $2^{s(n)+1}$.

Combining Lemma 41, we get:

Corollary 42 *Let $s(n) \leq n - 4\log n - 1$ be a function. Assume that there exists some samplable ensemble $\mathcal{D}$ with entropy $h(n) \geq s(n) + 4\log n + 1$ such that $\mathsf{MINK}[s]$ is $(\frac{1}{2} - \frac{1}{4n})$-HoA on $\mathcal{D}$. Then, one-way functions exist.*

Proof. By Lemma 41, the number of n-bit YES instances is at most $S(n) = 2^{s(n)+1}$. Since D_n has entropy $h(n) \geq s(n) + 1 + 4\log n$, the corollary follows directly from Theorem 11.

4.2 k-SAT

Let k, c be two positive integers. The language $k\text{-}\mathsf{SAT}(m, cm)$ is defined to consist of all satisfiable k-CNF formulas on m variables with cm clauses. We recall the well-known fact that $k\text{-}\mathsf{SAT}(m, cm)$ is a sparse language when $c \geq 2^{k+1}$.

Lemma 43 *The number of satisfiable k-CNF formulas on m variables with cm clauses is at most $2^m \left((2^k-1)\binom{m}{k}\right)^{cm}$, and the number of all such k-CNF formulas is $\left((2^k)\binom{m}{k}\right)^{cm}$.*

Proof. We first show that there are $((2^k)\binom{m}{k})^{cm}$ k-CNF formulas on m variables with cm clauses. Note that are $2^k\binom{m}{k}$ choices for a single k-clause; therefore, the number of cm k-clauses is $((2^k)\binom{m}{k})^{cm}$.

We then show that there are at most $2^m((2^k-1)\binom{m}{k})^{cm}$ satisfiable k-CNF formulas on m variables with cm clauses. Consider any possible assignment x; the number of k-clauses that is satisfied by x is at most $(2^k-1)\binom{m}{k}$ since given the choice of k variables, there are at most 2^k-1 possible choices of the polarities. Finally, since there are cm such k-clauses with m variables, we have that the total number of satisfiable formulas is at most $2^m((2^k-1)\binom{m}{k})^{cm}$

To consider average-case hardness of this problem, we need to have a way to encode formulas as strings. We use the following standard encoding scheme for k-SAT from [11]: a m-variable cm-clause k-CNF is represented by using $n(m,k,c) = cm(k\lceil \log m \rceil + k)$ bits to describe a sequence of cm clauses (and here n denotes the length of the input bit string encoding the instance). In each clause, we specify k literals one-by-one, and each of them takes $\lceil \log m \rceil$ bits to specify the index of a variable and 1 bit to fix the polarity. When n is not of the form $n(m,k,c)$, for an input of length n, we ignore as few bits as possible in the end of the input such that the prefix of the input is of length $n(m,k,c)$ for some m. Following [11], let the *entropy deficiency* of a distribution D_n over n bits denote the difference between n and $H(D_n)$. The following corollary implies [11, Theorem 4, Term 1].

Corollary 44 *Let k, c be two integers such that $c \geq 2^{k+2}$. Let $m = m(n)$ be a polynomial. Assume that there exists some samplable ensemble $\mathcal{D} = \{D_n\}_{n\in\mathbb{N}}$ with entropy deficiency at most $cm(n)/2^{k+1}$ distributed over k-CNF formulas on $m(n)$ variables and $cm(n)$ clauses such that k-SAT is $(\frac{1}{2} - \frac{1}{4n})$-HoA on $\mathcal{D}$. Then, one-way functions exist.*

Proof. Recall that k-CNF formulas are represented by binary strings using the standard encoding scheme. Let $n' = n(m(n), k, c)$ (be the length of the input without padding); by the encoding scheme, it follows that every $m(n)$-variable $cm(n)$-clause k-CNF formula will be encoded by $2^{n-n'}$ n-bit strings. By Lemma 43, it follows that n' is at least

$$\log\left(\left((2^k)\binom{m}{k}\right)^{cm}\right) = cm \log 2^k + cm \log \binom{m}{k}$$

Since D_n has entropy deficiency at most $cm/2^{k+1}$, it follows that D_n has entropy lower bounded by:

$$n' + (n - n') - cm/2^{k+1} \geq cm\left(\log 2^k - \frac{1}{2^{k+1}} + \log\binom{m}{k}\right) + (n - n')$$

By Lemma 43, the number of n-bit YES instances is at most

$$S(n) = 2^m\left((2^k - 1)\binom{m}{k}\right)^{cm} \times 2^{n-n'}$$

It follows that

$$\begin{aligned}H(D_n) - \log S(n) \geq & cm\left(\log 2^k - \frac{1}{2^{k+1}} + \log\binom{m}{k}\right) + (n-n') \\ & - \log\left(2^m\left((2^k-1)\binom{m}{k}\right)^{cm} \times 2^{n-n'}\right) \\ = & m(c\log 2^k - c\log(2^k-1) - \frac{c}{2^{k+1}} - 1) \\ \geq & m(\frac{c}{2^k} - \frac{c}{2^{k+1}} - 1) \\ \geq & m \\ \geq & 4\log n.\end{aligned}$$

where the second inequality follows from the standard inequality that $\log x - \log(x-1) \geq \frac{1}{x}$ for all $x \geq 2$, the third one from the fact that, by assumption, $c \geq 2^{k+2}$, and the fourth one inequality follows from the fact that due to the encoding scheme, $m \geq \Omega(\sqrt{n})$.

4.3 t-Clique

Let $t : \mathbb{N} \to \mathbb{N}$ be a function and let $t\text{-}\mathsf{Clique}(m)$ be the set of graphs on m vertices having a clique of size at least $t(m)$. We recall the well-known fact that $t\text{-}\mathsf{Clique}(m)$ is sparse when $t(\cdot)$ is large enough.

Lemma 45 *The number of m-vertex graphs with at least a t-size clique is at most $\binom{m}{t}2^{\binom{m}{2}-\binom{t}{2}}$. However, the number of m-vertex graphs is $2^{\binom{m}{2}}$.*

Proof. There are $\binom{m}{2}$ edges in a m-vertex graph, and thus the number of possible graphs is $2^{\binom{m}{2}}$. There are $\binom{m}{t}$ choices of cliques in a graph, and after fixing a clique, there are $\binom{m}{2}-\binom{t}{2}$ edges in the rest of the graph and therefore the number of graphs with at least 1 clique is at most $\binom{m}{t}2^{\binom{m}{2}-\binom{t}{2}}$.

We use the standard encoding scheme for t-Clique from [11]. A m-vertex graph is encoded by a $(n = n(m) = \binom{m}{2})$-bit string where the i-th bit is 1 iff the i-th edge appears in the graph. For input lengths n that are not of the form $n(m)$, we ignore as few bits as possible at the end of the input such that the prefix of the input is of length $n(m)$ for some m.

Corollary 46 *Let $m(n), t(n) \in \omega(\log m)$ be two polynomials. Assume that there exists some samplable ensemble $\mathcal{D} = \{D_n\}_{n\in\mathbb{N}}$ with entropy deficiency at most $0.99\binom{t(n)}{2}$ distributed over $m(n)$-vertex graphs such that $t\text{-}\mathsf{Clique}(m)$ is $(\frac{1}{2} - \frac{1}{4n})$-HoA on $\mathcal{D}$. Then, one-way functions exist.*

Proof. Recall that graphs are represented by binary strings using the standard encoding scheme. Let $n' = n(m(n))$ (be the length of the input without padding);

by the encoding scheme, it follows that every $m(n)$-vertex graph will be encoded by at least $2^{n-n'}$ n-bit strings. By Lemma 45, it follows that n' is at least

$$\log 2^{\binom{m}{2}} = \binom{m}{2}$$

Since D_n has entropy deficiency $0.99\binom{t}{2}$, it follows that D_n has entropy lower bounded by:

$$n' + (n - n') - 0.99\binom{t}{2} \geq \binom{m}{2} - 0.99\binom{t}{2} + (n - n')$$

By Lemma 45, the number of n-bit YES instances is at most

$$S(n) = \binom{m}{t} 2^{\binom{m}{2} - \binom{t}{2}} \times 2^{n-n'}$$

It follows that

$$\begin{aligned}
H(D_n) - \log S(n) \geq & \binom{m}{2} - 0.99\binom{t}{2} + (n - n') \\
& - \log\left(\binom{m}{t} 2^{\binom{m}{2} - \binom{t}{2}} \times 2^{n-n'}\right) \\
\geq & \binom{m}{2} - 0.99\binom{t}{2} - \log\binom{m}{t} - \left(\binom{m}{2} - \binom{t}{2}\right) \\
\geq & \binom{t}{2} - 0.99\binom{t}{2} - t\log m \\
\geq & 4\log n
\end{aligned}$$

since $t(n) = \omega(\log m)$.

5 OWF from Hardness of Approximating K-Complexity

We turn to showing how to (slightly) generalize the result in [11] with respect to K-complexity, and show that the hardness of approximating K-complexity (even with respect to unknown thresholds) is equivalent to the existence of OWFs. We refer the reader to Sect. 4.1 for a formal definition of the notion of K-complexity.

It appears that a similar generalization was concurrently and independently obtained by IRS in the proceedings version [12] for a general class of complexity measures satisfying a so-called "coding theorem"—see Theorem 2.2 in [12]—but their full version has not appeared yet as far as we can tell. The corollary of this Theorem 2.2 to K-complexity is still stated w.r.t. a gap problem with a computable threshold.

We start by recalling what it means for a function to be hard on average to approximate. We say that a function $f : \{0,1\}^* \to \mathbb{N}$ is $\alpha(\cdot)$ *hard-on-average (α-HoA) to $\beta(\cdot)$-approximate on an ensemble $\mathcal{D} = \{D_n\}_{n\in\mathbb{N}}$* if for all probabilistic polynomial-time heuristics $\mathcal{H}$, for all sufficiently large $n \in \mathbb{N}$,

$$\Pr[x \leftarrow D_n : |\mathcal{H}(x) - f(x)| \leq \beta(n)] < 1 - \alpha(n).$$

We simply say that f is *mildly hard-on-average (mildly HoA) to approximate* on $\mathcal{D}$ if there exists a polynomial $p(\cdot)$ such that f is $\frac{1}{p}$-HoA to approximate; We say that f is *hard-on-average (HoA) to approximate* on $\mathcal{D}$ if for every c, $\alpha(n) = \frac{1}{2} - \frac{1}{n^c}$, L is α-HoA to approximate.

The hardness notion above is defined with respect to the search version of approximating the function f and when considering K-complexity, it asserts that approximating the value of the K-complexity is hard. We can also consider its decisional version, parametrized by two efficiently computable thresholds $t_0(\cdot), t_1(\cdot)$, where we aim at deciding whether the input string x is of K-complexity below $t_0(|x|)$ or above $t_1(|x|)$. Let $\mathsf{GapK}[t_0, t_1]$ be a promise problem where YES-instances are strings $x \in \Pi_{\mathsf{YES}}$ such that $K(x) \leq t_0(|x|)$, and NO-instances are strings $x \in \Pi_{\mathsf{NO}}$ such that $K(x) \geq t_1(|x|)$. We say that $\mathsf{GapK}[t_0, t_1]$ is *mildly hard on average (mildly HoA)* on an ensemble $\mathcal{D} = \{D_n\}_{n \in \mathbb{N}}$ if there exists a polynomial $p(\cdot)$ such that for all probabilistic polynomial-time heuristics $\mathcal{H}$, for all sufficiently large $n \in \mathbb{N}$,

$$\Pr[x \leftarrow D_n : (x \in \Pi_{\mathsf{YES}} \wedge \mathcal{H}(x) = 0) \vee (x \in \Pi_{\mathsf{NO}} \wedge \mathcal{H}(x) = 1)] \geq 1/p(n).$$

The result in [11] showed that the existence of a samplable distribution $\mathcal{D}$ and efficiently computable $t_0, t_1, t_1(n) - t_0(n) \in \omega(\log n)$ such that $\mathsf{GapK}[t_0, t_1]$ is mildly HoA on $\mathcal{D}$ is equivalent to the existence of OWFs. We show in the following Theorem that it suffices to assume hardness with respect to the search version (with an additive factor $\omega(\log n)$) to obtain OWFs, therefore giving a search to decision reduction for this problem by going through the notion of OWFs.

We are not aware of any "direct" way of showing such a decision-to-search reduction. While one direction is trivial (hardness of decision—with respect to *efficiently computable* thresholds—to hardness of search), it is not clear how to show the converse direction.[4]

Theorem 51 (Theorem 12, restated) *The following are equivalent:*

1. *One-way functions exist;*
2. *There exists some efficiently samplable distribution such that K-complexity is mildly hard to approximate within an additive term of $\omega(\log n)$.*
3. *There exists some efficiently samplable distribution such that K-complexity is hard to approximate within an additive term of $n - n^{o(1)}$.*
4. *There exist some efficiently samplable distribution and efficiently computable thresholds $t_0, t_1, t_1(n) - t_0(n) = \omega(\log n)$ such that $\mathsf{GapK}[t_0, t_1]$ is mildly HoA.*

[4] The naive approach to try to prove such a result would be to simply try running the decision heuristic on different thresholds. There are several problems with this approach. First, for every threshold $t = (t_0, t_1)$, there may exist a *different* heuristic H_t that solves the decision problem for that threshold, so it's not clear how to get a uniform search heuristic. Next, its not even clear how to define efficient threshold functions as we require n/Gap thresholds to approximate within an additive term of *Gap*. Finally, it is not a-prior clear how to use a Gap-K heuristic to approximate K given that the Gap-K heuristic only works on *average*.

Proof (of Theorem 12). (2) ⇒ (1) follows from Theorem 52 (stated and proved below). The implications (1) ⇒ (3) and (1) ⇒ (4) essentially follow from the argument proving Theorem 31 (and see also [11]). (3) ⇒ (2) trivially holds. (4) ⇒ (2) follows from the following argument. Assume that there exists a heuristic $\mathcal{H}$ for approximating K-complexity within $(t_1 - t_0)/2$. To solve $\mathsf{GapK}[t_0, t_1]$ on input x, we simply output 1 if $\mathcal{H}(x) \leq t_0(|x|) + (t_1(|x|) - t_0(|x|))/2$. Note that if $\mathcal{H}$ succeeds on x (with some probability), our algorithm also succeeds in solving $\mathsf{GapK}[t_0, t_1]$ on x (with the same probability). This concludes our proof.

Theorem 52 *For any constant $\gamma \geq 3$, there exists a polynomial p such that if there exists a samplable ensemble $\mathcal{D}$ on which K-complexity is $\frac{1}{p}$-HoA to $(\gamma \log n)$-approximate, then OWFs exist.*

Proof. Consider some fixed constant $\gamma \geq 3$ and let $p(n) = n^{\gamma-2}$. We assume for contradiction that OWFs do not exist. Then, by Lemma 24, there exist a constant δ and an approximate counter $\mathcal{A}$ for $\mathcal{D} = \{D_n\}$ with an (multiplicative) approximate factor δ and an error probability $\leq \frac{1}{2p(n)}$. We will use $\mathcal{A}$ to compute the K-complexity of strings sampled by $\mathcal{D}$.

On input $x \leftarrow D_n$, our heuristic $\mathcal{H}$ simply outputs $-\lfloor \log \mathcal{A}(x) \rfloor$ as (our estimate of) $K(x)$. $\mathcal{H}$ runs in polynomial time since $\mathcal{A}$ is a PPT machine. We next show that $\mathcal{H}(x)$ approximates $K(x)$ with probability at least $1 - \frac{1}{p(n)}$ (over $x \sim D_n$). Fix some input length n on which $\mathcal{A}$ succeeds (and there are infinitely many such input lengths). Let us first assume that $\mathcal{A}$ is a "perfect" approximate counter and $\delta \cdot p_x \leq \mathcal{A}(x) \leq p_x$ with probability 1 (where p_x is defined to be $\Pr[D_n = x]$). The following two claims will show that $\mathcal{H}$ approximate K with high probability.

Claim 1 $K(x) \leq \mathcal{H}(x) + \gamma \log n$ holds with probability 1.

Proof. We will show that $K(x) \leq -\lfloor \log p_x \rfloor + 2\log(n) + O(1)$ with probability 1. Note that $\mathcal{H}(x) = -\lfloor \log \mathcal{A}(x) \rfloor \geq -\lfloor \log p_x \rfloor$ (due to the correctness of $\mathcal{A}$) and $\gamma \geq 3$, the claim follows. For any string $x \in \{0,1\}^n$, let $S = \{y \in \{0,1\}^n : -\lfloor \log p_y \rfloor = -\lfloor \log p_x \rfloor\}$. Note that for each $y \in S$, it holds that $\Pr[D_n = y] = p_y \geq 2^{\lfloor \log p_x \rfloor}$. So S is of size at most $2^{-\lfloor \log p_x \rfloor}$. Membership of S can be checked by using an exponential time algorithm computing p_y (enumerating all randomness used in D_n) with the values $-\lfloor \log p_x \rfloor$ and n. Therefore, we can compress each element in S (including x) into $-\lfloor \log p_x \rfloor + 2\log(n) + O(1)$ bits by hardwiring its index and running an exhaustive search with the membership checker, which shows that $K(x) \leq -\lfloor \log p_x \rfloor + 2\log(n) + O(1)$.

Claim 2 $K(x) \geq \mathcal{H}(x) - \gamma \log n$ holds with probability at least $1 - \frac{1}{2p(n)}$

Proof. Towards this, we show that $\mathcal{H}(x) > K(x) + \gamma \log n$ with probability at most $\frac{1}{2p(n)}$. This follows from a union bound.

$$\begin{aligned}
&\Pr_{x \leftarrow D_n}[\mathcal{H}(x) > K(x) + \gamma \log n] \\
=&\sum_{w=1}^{n+O(1)} \Pr_{x \leftarrow D_n}[K(x) = w \wedge \mathcal{H}(x) > w + \gamma \log n] \\
\leq&\sum_{w=1}^{n+O(1)} \Pr_{x \leftarrow D_n}[K(x) = w \wedge \Pr[D_n = x] < \frac{1}{\delta} \cdot 2^{-w-\gamma \log n}] \\
\leq&\sum_{w=1}^{n+O(1)} 2^w \cdot \frac{1}{\delta} \cdot 2^{-w-\gamma \log n} \\
\leq&\frac{1}{2p(n)}.
\end{aligned}$$

where the second to last line follows from a union bound.

Finally, we note that $\mathcal{A}$ is not necessarily a perfect approximate counter and $\mathcal{A}$ fails with probability $\frac{1}{2p(n)}$. By a union bound, it follows that

$$\Pr_{x \leftarrow D_n}[|\mathcal{H}(x) - K(x)| \leq \gamma \log n] \geq 1 - \frac{1}{2p(n)} - \frac{1}{2p(n)} \geq 1 - \frac{1}{p(n)}$$

on infinitely many n.

References

1. Blum, M.: Coin flipping by telephone - a protocol for solving impossible problems. In: COMPCON'82, Digest of Papers, Twenty-Fourth IEEE Computer Society International Conference, San Francisco, California, USA, February 22–25, 1982, pp. 133–137. IEEE Computer Society (1982)
2. Blum, M., Micali, S.: How to generate cryptographically strong sequences of pseudo-random bits. SIAM J. Comput. **13**(4), 850–864 (1984)
3. Chen, L., Jin, C., Williams, R.R.: Hardness magnification for all sparse np languages. In: 2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS), pp. 1240–1255. IEEE (2019)
4. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Trans. Inf. Theory **22**(6), 644–654 (1976)
5. Feige, U., Shamir, A.: Witness indistinguishable and witness hiding protocols. In: STOC '90, pp. 416–426 (1990). http://doi.acm.org/10.1145/100216.100272
6. Goldreich, O., Goldwasser, S., Micali, S.: On the cryptographic applications of random functions. In: CRYPTO, pp. 276–288 (1984)
7. Goldwasser, S., Micali, S.: Probabilistic encryption. J. Comput. Syst. Sci. **28**(2), 270–299 (1984)
8. Gurevich, Y.: The challenger-solver game: variations on the theme of p=np. In: Logic in Computer Science Column, The Bulletin of EATCS (1989)

9. Hartmanis, J.: Generalized kolmogorov complexity and the structure of feasible computations. In: 24th Annual Symposium on Foundations of Computer Science (sfcs 1983), pp. 439–445 (1983). https://doi.org/10.1109/SFCS.1983.21
10. Håstad, J., Impagliazzo, R., Levin, L.A., Luby, M.: A pseudorandom generator from any one-way function. SIAM J. Comput. **28**(4), 1364–1396 (1999)
11. Ilango, R., Ren, H., Santhanam, R.: Hardness on any samplable distribution suffices: New characterizations of one-way functions by meta-complexity. Electron. Colloquium Comput. Complex. **28**, 82 (2021)
12. Ilango, R., Ren, H., Santhanam, R.: Robustness of average-case meta-complexity via pseudorandomness. In: Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing, pp. 1575–1583 (2022)
13. Impagliazzo, R.: A personal view of average-case complexity. In: Structure in Complexity Theory '95, pp. 134–147 (1995)
14. Impagliazzo, R., Levin, L.A.: No better ways to generate hard NP instances than picking uniformly at random. In: 31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22–24, 1990, Volume II, pp. 812–821 (1990)
15. Impagliazzo, R., Luby, M.: One-way functions are essential for complexity based cryptography (extended abstract). In: 30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989, pp. 230–235 (1989)
16. Ko, K.: On the notion of infinite pseudorandom sequences. Theor. Comput. Sci. **48**(3), 9–33 (1986). https://doi.org/10.1016/0304-3975(86)90081-2
17. Kolmogorov, A.N.: Three approaches to the quantitative definition of information. Int. J. Comput. Math. **2**(1–4), 157–168 (1968)
18. Levin, L.A.: The tale of one-way functions. Problems of Information Transmission **39**(1), 92–103 (2003). https://doi.org/10.1023/A:1023634616182
19. Liu, Y., Pass, R.: On one-way functions and Kolmogorov complexity. In: 61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16–19, 2020, pp. 1243–1254. IEEE (2020)
20. Naor, M.: Bit commitment using pseudorandomness. J. Cryptol. **4**(2), 151–158 (1991). https://doi.org/10.1007/BF00196774
21. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems (reprint). Commun. ACM **26**(1), 96–99 (1983). https://doi.org/10.1145/357980.358017
22. Rompel, J.: One-way functions are necessary and sufficient for secure signatures. In: STOC, pp. 387–394 (1990)
23. Sipser, M.: A complexity theoretic approach to randomness. In: Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25–27 April, 1983, Boston, Massachusetts, USA, pp. 330–335. ACM (1983)
24. Trakhtenbrot, B.A.: A survey of Russian approaches to perebor (brute-force searches) algorithms. Ann. Hist. Comput. **6**(4), 384–400 (1984)