Electronic Colloquium on Computational Complexity, Report No. 120 (2024)

Consequences of Randomized Reductions from SAT to Time-Bounded Kolmogorov Complexity

Halley Goldberg[*] Valentine Kabanets[†]

July 15, 2024

Abstract

A central open question within meta-complexity is that of NP-hardness of problems such as MCSP and $\mathsf{MK}^t\mathsf{P}$. Despite a large body of work giving consequences of and barriers for NP-hardness of these problems under (restricted) deterministic reductions, very little is known in the setting of randomized reductions. In this work, we give consequences of randomized NP-hardness reductions for both approximating and exactly computing time-bounded and time-unbounded Kolmogorov complexity.

In the setting of *approximate* $\mathsf{K}^{\mathsf{poly}}$ complexity, our results are as follows.

1. Under a derandomization assumption, for any constant $\delta > 0$, if approximating K^t complexity within n^δ additive error is hard for SAT under an honest randomized non-adaptive Turing reduction running in time polynomially less than t, then $\mathsf{NP} = \mathsf{coNP}$.
2. Under the same assumptions, the worst-case hardness of NP is equivalent to the existence of one-way functions.

Item 1 above may be compared with a recent work of Saks and Santhanam [SS22], which makes the same assumptions except with $\omega(\log n)$ additive error, obtaining the conclusion $\mathsf{NE} = \mathsf{coNE}$.

In the setting of *exact* $\mathsf{K}^{\mathsf{poly}}$ complexity, where the barriers of Item 1 and [SS22] do not apply, we show:

3. If computing K^t complexity is hard for SAT under reductions as in Item 1, then the average-case hardness of NP is equivalent to the existence of one-way functions. That is, "Pessiland" is excluded.

Finally, we give consequences of NP-hardness of *exact time-unbounded* Kolmogorov complexity under randomized reductions.

4. If computing Kolmogorov complexity is hard for SAT under a randomized many-one reduction running in time t_R and with failure probability at most $1/(t_R)^{16}$, then coNP is contained in non-interactive statistical zero-knowledge; thus $\mathsf{NP} \subseteq \mathsf{coAM}$. Also, the worst-case hardness of NP is equivalent to the existence of one-way functions.

We further exploit the connection to NISZK along with a previous work of Allender et al. [AHT23] to show that hardness of K complexity under randomized many-one reductions is highly robust with respect to failure probability, approximation error, output length, and threshold parameter.

[*] Simon Fraser University; halley_goldberg@sfu.ca. Supported by NSERC CGS D.

[†] Simon Fraser University; kabanets@sfu.ca. Supported by NSERC Discovery research grant.

ISSN 1433-8092

Contents

1 Introduction

Meta-complexity aims to determine the computational complexity of the tasks to compute various intrinsic complexity measures of given binary strings. Two prominent examples of such complexity measures are the minimum circuit size of a given truth table of a Boolean function, and the minimum time-bounded Kolmogorov complexity (denoted K^t) of a given binary string. The corresponding meta-complexity problems are the *Minimum Circuit Size Problem* (MCSP):

> given a binary string $x \in \{0,1\}^{2^n}$ and a parameter $s \leq 2^n$, decide if there is an n-input boolean circuit of size at most s whose truth table equals x,

and the *Minimum* K^t *Problem* ($\mathsf{MK}^t\mathsf{P}$):

> given a binary string $x \in \{0,1\}^n$, and a parameter $s \leq n$, decide if there is a binary input w of length at most s such that some fixed universal Turing machine U on input w prints x within t time steps.

The history of these two problems goes back to at least the 1950s and '60s. In the Soviet Union, during that period, those involved in 'theoretical cybernetics' were keenly interested in problems related to switching circuits and Kolmogorov's new theory of the complexity of strings. It was widely suspected that one could not avoid *perebor* (exhaustive search) in the solution of the corresponding minimization problems. Levin's interest in perebor, culminating in his discovery of NP-completeness in the early 1970s, was motivated in particular by questions about the complexity of time-bounded Kolmogorov complexity [Tra84]. Since then, both MCSP and $\mathsf{MK}^t\mathsf{P}$ have resisted categorization as efficiently decidable or as NP-complete, a somewhat uncommon state of affairs for natural problems in NP.

In 2000, Kabanets and Cai took up the study of circuit minimization again, with a result suggesting that NP-hardness of MCSP may be very difficult to resolve: if MCSP is NP-hard under a deterministic many-one reduction such that output length depends only on input length, then one gets the lower bound $\mathsf{E} \not\subseteq \mathsf{P/poly}$ [KC00]. At least, if MCSP is NP-hard, then showing its hardness would seem to require different techniques than those applied in the past, barring any further major breakthroughs. A line of work has continued to push further in this negative direction, progressively obtaining (1) "stronger" consequences, and (2) consequences of NP-hardness under more powerful forms of reducibility. An example of the former is a result of Murray and Williams, which obtains $\mathsf{NP} \not\subseteq \mathsf{P/poly}$ from NP-hardness of MCSP under log-time uniform AC^0 reductions [MW15]. An example of the latter is a result of Hitchcock and Pavan, which obtains $\mathsf{EXP} \neq \mathsf{ZPP}$ from NP-hardness of MCSP under deterministic non-adaptive Turing reductions [HP15]. There are many more examples of this kind of work relying essentially on the determinism of the reductions in question; see, e.g., [AH19; SS20; AHK17; HW16].[1]

In contrast to the negative line of work for deterministic reductions, there is a positive line of work obtaining NP-hardness of variants of MCSP and $\mathsf{MK}^t\mathsf{P}$ that seem to come progressively closer to the standard definitions of these problems. Examples include [Ila23; Hir22; ILO20; Ila20]. A common feature of these results is their employment of randomness in the NP-hardness reductions. An impressive example of such a result is Hirahara's recent proof of NP-hardness of partial-function

[1] One result of [HW16] deals with one-query randomized reductions to $\mathsf{MCSP}^{\mathcal{O}}$ working for *every* oracle $\mathcal{O}$, which may be seen as an exception. Other results of that work give consequences of deterministic reductions to, for example, approximating circuit size and Levin's Kt complexity.

versions of MCSP and $\mathsf{MK}^t\mathsf{P}$ [Hir22]. Additionally, from [AD17], MCSP is hard for SZK (statistical zero-knowledge) under randomized reductions, which is the strongest unconditional hardness known for MCSP. All of this begs the question whether randomness is the key ingredient for the hardness of problems in meta-complexity: most barriers apply to deterministic reductions, whereas most progress has been made via randomized reductions.

As for the negative direction for randomized reductions, there has been far less headway. In fact, prior to this work, only two such results were known for MCSP and $\mathsf{MK}^t\mathsf{P}$. Murray and Williams ruled out NP-hardness of MCSP in the very restrictive setting of poly-logarithmic-time randomized projections [MW15]. More recently, Saks and Santhanam showed that $\mathsf{NE} = \mathsf{coNE}$ if approximating K^t-complexity is NP-hard under randomized non-adaptive polynomial-time reductions (with some caveats, including a derandomization assumption and that the time-bound t in the superscript of K^t must be greater than the running time of the reduction) [SS22].

Of course, any NP-hardness of $\mathsf{MK}^t\mathsf{P}$ or MCSP would be a major breakthrough for complexity theory, including hardness under a non-black-box reduction. In that sense, the *kind* of reduction in question is hardly important in itself. That being said, obtaining consequences of restricted forms of reduction can certainly help guide the "search for NP-hardness". For example, a recent work of Ilango proved that approximating K^t within $\Omega(n)$ additive error is NP-hard in the random oracle model [Ila23]. As mentioned in that paper, the reduction circumvents the barrier of [SS22] by requiring more time than the superscript t. As with much of complexity theory, one can always take negative results as putting into focus the space for positive progress.

In this paper, we advance in the negative direction for randomized reductions, obtaining results with stronger consequences and from reductions to harder problems compared to prior work.

1.1 Results

We show a number of consequences of the assumptions that there exist restricted randomized NP-hardness reductions for the exact and approximate variants of the problem to determine the (time-bounded) Kolmogorov complexity of a given binary string.

In addition to the problem $\mathsf{MK}^t\mathsf{P}$ introduced above, we shall also consider its time-unbounded version, MKP, where given a binary string $x \in \{0,1\}^n$ and a threshold parameter $s \leq n$, one needs to decide if there is a string $w \in \{0,1\}^{\leq s}$ such that a fixed universal TM $U(w)$ outputs x. We also consider the probabilistic variant of K^t, denoted by pK^t, where $\mathsf{pK}^t(x)$ is defined as the minimum length s such that, for each of at least $2/3$ of random strings r, there exists some input $w_r \in \{0,1\}^{\leq s}$ such that $U(w_r, r)$ outputs x within t time steps. The corresponding Minimum pK^t Problem is denoted by $\mathsf{MpK}^t\mathsf{P}$. For $g : \mathbb{N} \to \mathbb{N}$ and $\mu \in \{\mathsf{K}, \mathsf{pK}\}$, Approx_g-μ^t refers to the problem of approximating μ^t complexity of a given $x \in \{0,1\}^n$ to within a $g(n)$ additive error. Approx_g-$\mathsf{K}[s]$ refers to the problem of approximating K complexity except with threshold parameter fixed to s. (See Section 2 for precise definitions.)

Consequences of showing the NP-hardness of an approximation to pK^t or K^t. Informally, our first results show that NP-hardness of $\mathsf{Approx}_{n^\delta}$-$\mathsf{pK}^t$ under honest non-adaptive randomized reductions with runtime sufficiently smaller than t implies that

- $\mathsf{NP} \subseteq \mathsf{coAM}$ (and hence, the polynomial-time hierarchy collapses [BHZ87]), and
- if, in addition, no one-way functions exist, then $\mathsf{NP} \subseteq \mathsf{BPP}$;

here 'honest' reductions are those that make queries of length at least some polynomial of the input to the reduction. We also get a similar result for $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{K}^t$, under a derandomization assumption that E requires exponential-size nondeterministic circuits.

More precisely, we show that under the same NP-hardness assumptions, there is a black-box non-adaptive reduction from SAT to inverting an auxiliary input one-way function.[2] Moreover, this reduction is of a restricted form in which the oracle only needs to invert the function on auxiliary input φ, where φ is the input to SAT; this is called a "fixed-auxiliary-input reduction" [ABX08]. The "γ-honesty" condition below means that all queries $q \in \{0,1\}^*$ made by the reduction are such that $|q| \geq n^\gamma$, where n is the length of the input to the reduction.

Theorem 1.1 (Collapsing the Polynomial Hierarchy)**.** *For any constants $\delta, \gamma > 0$, there is a polynomial p such that, for any $t, t_R : \mathbb{N} \to \mathbb{N}$ satisfying $p(t_R(n)) \leq t(n)$ for all $n \in \mathbb{N}$, we have the following.*

1. *If $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{pK}^t$ is hard for SAT under a γ-honest non-adaptive randomized reduction running in time t_R, then there is a black-box non-adaptive fixed-auxiliary-input reduction from SAT to inverting an auxiliary-input OWF. The latter implies that*

$$\mathsf{NP} \subseteq \mathsf{coAM}.$$

2. *Assume $\mathsf{E} \not\subseteq \mathsf{io}\text{-}\mathsf{NSIZE}[2^{o(n)}]$. If $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{K}^t$ is hard for SAT under an honest non-adaptive randomized reduction running in time t_R, then*

$$\mathsf{NP} = \mathsf{coNP}.$$

As a consequence of the above non-adaptive black-box reduction from SAT to inverting an auxiliary-input one-way function, we further obtain from the hypothesis of Theorem 1.1 that the existence of a standard one-way function can be based on the worst-case hardness of NP. That is, proving NP-hardness of $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{K}^t$ (under restricted randomized reductions) is as hard as achieving the 'holy grail of cryptography'.

We obtain both adaptive black-box and non-adaptive BPP-black-box[3] reductions from SAT to the problem of inverting a standard OWF. The former follows immediately from our Theorem 1.1 and a recent work of Nanashima [Nan21] (see Theorem 2.27 below), and the latter is implicit in [Hir23], though we provide a short, self-contained proof building on Theorem 1.1.

Theorem 1.2 (Excluding Pessiland and Heuristica)**.** *For any constants $\delta, \gamma > 0$, there is a polynomial p such that, for any $t_R, t : \mathbb{N} \to \mathbb{N}$ satisfying $p(t_R(n)) \leq t(n)$ for all $n \in \mathbb{N}$, we have the following.*

1. *If $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{pK}^t$ is hard for SAT under a γ-honest non-adaptive randomized reduction running in time t_R, then there exist both (I) a black-box adaptive randomized polynomial-time reduction, and (II) a BPP-black-box non-adaptive randomized polynomial-time reduction, from SAT to inverting a OWF. As a consequence, we get*

$$\mathsf{NP} \not\subseteq \mathsf{BPP} \iff \exists\, \mathsf{OWF}.$$

[2] We consider auxiliary input functions $f = \{f_\varphi\}_{\varphi \in \{0,1\}^*}$ as defined in [OW93]; see Section 2.3

[3] As defined by [GT07], a BPP-black-box reduction R from a problem L to a problem L' is an efficient oracle Turing machine that correctly decides L, given any oracle $A \in \mathsf{BPP}$ such that A decides L'; see Section 2.

2. *Assume* $\mathsf{E} \not\subseteq \mathsf{io\text{-}NSIZE}[2^{o(n)}]$*. If* $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{K}^t$ *is hard for* SAT *under an honest non-adaptive randomized reduction running in time* t_R*, then*

$$\mathsf{NP} \neq \mathsf{P} \iff \exists\, \mathsf{OWF}.$$

We also obtain similar results for Levin's Kt complexity. See Corollary 3.3.

Consequences of showing the NP-hardness of K^t. Though the conclusions of Theorems 1.1 and 1.2 are incomparable, one may find $\mathsf{NP} \subseteq \mathsf{coAM}$ unbelievable, in which case Theorem 1.2 would not appear to yield a promising route for actually excluding Pessiland and Heuristica. Indeed, the earlier barrier result of [SS22] was part of Hirahara's motivation to introduce a harder 'distributional' variant of K^t complexity in a recent work [Hir23], delineating an intact positive approach for excluding Impagliazzo's worlds via NP-hardness of meta-complexity.

As a counterpoint, building on a work of Liu and Pass [LP23], we show that NP-hardness of *exact* K^t complexity would still suffice to exclude Pessiland while circumventing the barrier of Theorem 1.1 (and [SS22]). As noted in [LP23], problems of exact and approximate K^t complexity are qualitatively different: approximating K^t within $\omega(\log n)$ additive error is unconditionally easy on average (in the 'error-prone' sense) over the uniform distribution, but the argument fails in the setting of exact K^t. Thus, there is still room for optimism with regard to excluding Pessiland via NP-hardness of standard K^t complexity.

Theorem 1.3 (Excluding Pessiland)**.** *There is a polynomial* p *such that, for any* $t, t_R : \mathbb{N} \to \mathbb{N}$ *satisfying* $t(n) \geq p(t_R(n))$ *for all* $n \in \mathbb{N}$*, we have the following.*

1. *If* $\mathsf{MpK}^t\mathsf{P}$ *is hard for* SAT *under an honest non-adaptive randomized reduction running in time* t_R*, then there is a black-box average-case reduction from* SAT *to inverting OWFs. As a consequence, we get that*
$$\mathsf{DistNP} \not\subseteq \mathsf{HeurBPP} \iff \exists\, \mathsf{OWF}.$$

2. *Assume* $\mathsf{E} \not\subseteq \mathsf{io\text{-}NSIZE}[2^{o(n)}]$*. If* $\mathsf{MK}^t\mathsf{P}$ *is hard for* SAT *under an honest non-adaptive randomized reduction running in time* t_R*, then*
$$\mathsf{DistNP} \not\subseteq \mathsf{HeurP} \iff \exists\, \mathsf{OWF}.$$

Consequences of showing the NP-hardness of K. Finally, we show that NP-hardness of Kolmogorov complexity under randomized many-one reductions would imply $\mathsf{NP} \subseteq \mathsf{coAM}$ and a collapse of the polynomial hierarchy. To the best of our knowledge, this is the first evidence against NP-hardness of exact Kolmogorov complexity under randomized many-one reductions. We also get under the same assumption that if $\mathsf{NP} \not\subseteq \mathsf{BPP}$ then one-way functions exist.

Theorem 1.4 (Collapsing the Polynomial Hierarchy)**.** *There is a polynomial* p *such that, for any* $t_R : \mathbb{N} \to \mathbb{N}$*, we have the following. If* MKP *is hard for* SAT *under a randomized polynomial time many-one reduction running in time* $t_R(n)$ *and with failure probability at most* $1/p(t_R(n))$*, then*

$$\mathsf{NP} \subseteq \mathsf{coAM}.$$

If, in addition, no one-way functions exist, then $\mathsf{NP} \subseteq \mathsf{BPP}$*.*

Robustness of reductions to K. In fact, we can get a stronger result than that stated above: namely, we show that if a decidable language L reduces to MKP as in Theorem 1.4, then $\overline{L} \subseteq \mathsf{NISZK}$, where NISZK is the class of promise problems admitting *non-interactive* statistical zero-knowledge proofs. Since it is known that $\mathsf{NISZK} \subseteq \mathsf{SZK} \subseteq \mathsf{AM} \cap \mathsf{coAM}$ [GSV99; For89; AH91], where SZK is the class of problems admitting statistical zero-knowledge proofs, this captures Theorem 1.4. It also yields improvements on a previous work of Allender et al. [AHT23] (see Section 7).

Combining the above with a converse provided in [AHT23], we show that hardness of MKP under randomized many-one reductions (with sufficiently small failure probability) is remarkably robust with respect to approximation error, failure probability, honesty, and threshold parameter (fixed or unfixed). For instance, if MKP is NP-hard under a $t_R(n)$-time many-one reduction with failure probability $1/\mathsf{poly}(t_R(n))$, then it is also NP-hard under a polynomial-time many-one reduction with exponentially small failure probability.

More specifically,

Theorem 1.5. *There is a polynomial p such that for any decidable language L and polynomial t_R, the following are equivalent.*

1. $\overline{L} \subseteq \mathsf{NISZK}$*;*
2. MKP *is hard for L under a randomized many-one reduction running in time $t_R(n)$ and with two-sided failure probability at most $1/p(t_R(n))$;*
3. $\mathsf{Approx}_{n^{o(1)}}\text{-}\mathsf{K}[n/2]$ *is hard for L under an honest randomized many-one reduction with one-sided failure probability at most $2^{-\mathsf{poly}(n)}$.*

1.2 Related Work

Saks and Santhanam obtain a barrier result similar to our Theorem 1.1, Item 2, for the regime of super-logarithmic additive error. Specifically, they prove the following.

Theorem 1.6 ([SS22])**.** *Assume $\mathsf{E} \not\subseteq \text{io-}\mathsf{NSIZE}[2^{o(n)}]$. There is a polynomial p satisfying the following. For any $t, t_R : \mathbb{N} \to \mathbb{N}$ such that $p(t_R(n)) \leq t(n)$, if $\mathsf{Approx}_{\omega(\log n)}\text{-}\mathsf{K}^t$ is hard for SAT under an honest, fixed query length, non-adaptive randomized reduction running in time t_R, then $\mathsf{NE} = \mathsf{coNE}$.*

Here, "fixed query length" means that the lengths of all queries made in the reduction are identical and depend only on the length of the input to the reduction, independent of randomness. In comparison, at the cost of increasing the approximation error term from $\omega(\log n)$ to n^{δ} for any constant $\delta > 0$, we obtain the stronger (and presumably less believable) consequence $\mathsf{NP} = \mathsf{coNP}$. Moreover, we do not require that the reduction have fixed query length: in our case, the length of queries need not be the same, and they can depend on the input and the randomness of the reduction. The honesty condition is identical in this work and [SS22]. We also note that our proof techniques can be made to capture the regime of $\omega(\log n)$ additive error, in which case we recover the statement of [SS22] improved to reductions without fixed query length; see Corollary 3.4.

Our Theorem 1.2 is related to a recent work of Hirahara [Hir23], which introduces a "distributional" variant of K^t complexity, denoted dK^t, defined as follows: for a string $x \in \{0,1\}^*$, a time bound $t \in \mathbb{N}$, and a distribution $\mathcal{D}$,

$$\mathsf{dK}^t(x \mid \mathcal{D}) = \min_{s \in \mathbb{N}} \left\{ \exists d \in \{0,1\}^s \;\middle|\; \Pr_{r \sim \mathcal{D}}[U(d,r) \text{ halts and outputs } x \text{ within } t \text{ steps}] \geq 2/3 \right\}.$$

Using the techniques of that work, it is possible to recover a part of our Theorem 1.2 exactly: namely, the existence of a BPP-black-box non-adaptive reduction from SAT to inverting a OWF. This is essentially due to the fact that if, for example, approximating K^t is NP-hard, then approximating dK^t is also NP-hard, since dK^t captures K^t when the provided distribution $\mathcal{D}$ always outputs the empty string. A probabilistic variant of dK^t is also introduced in [Hir23], which similarly generalizes pK^t.

However, our proof of Theorem 1.2 takes a partly different approach to that implicit in [Hir23]. In particular, though both our proof and that work employ a non-black-box worst-case to average-case reduction as in [Hir18; Hir20a; Gol+22], the latter approach would use this kind of reduction in two places: once to reduce NP to inverting an auxiliary-input one-way function, and once to obtain $\mathsf{NP} \nsubseteq \mathsf{BPP} \implies \mathsf{DistNP} \nsubseteq \mathsf{AvgBPP}$. To accommodate the reduction to inverting an auxiliary-input OWF, Hirahara introduces a new kind of mildly black-box reduction, which is more restrictive than the standard notion of a class-specific black-box reduction [GT07]. In contrast, as an intermediate step, we obtain a completely black-box non-adaptive reduction from NP to inverting an auxiliary-input OWF. We employ a class-specific worst-to-average reduction only to obtain $\mathsf{NP} \nsubseteq \mathsf{BPP} \implies \mathsf{DistNP} \nsubseteq \mathsf{AvgBPP}$.

As noted above, we could alternatively simply combine our Theorem 1.1 with [Nan21] to obtain the statement

$$\mathsf{NP} \nsubseteq \mathsf{BPP} \implies \exists \mathsf{OWF}.$$

However, we provide in Section 4 a self-contained proof of a BPP-black-box non-adaptive reduction. This is for completeness and to clarify the connection to Theorem 1.1.

Finally, we mention a few previous works related to our Theorem 1.4. Interestingly, by Allender et al., computing Kolmogorov complexity is known to be hard for PSPACE under deterministic adaptive Turing reductions [All+06]. This was improved by Hirahara to show that Kolmogorov complexity is hard for $\mathsf{EXP}^{\mathsf{NP}}$ under deterministic adaptive Turing reductions and hard for NEXP under randomized non-adaptive reductions [Hir20b]. Thus, Theorem 1.4 indicates a sharp contrast between the power of randomized many-one reductions and more powerful reductions with respect to the hardness of Kolmogorov complexity. Saks and Santhanam also prove that NP-hardness of *approximating* Kolmogorov complexity within $\omega(\log n)$ additive error under honest randomized non-adaptive reductions would imply $\mathsf{NP} \subseteq \mathsf{coAM}$ [SS22]. Note that Theorem 1.4 does not assume honesty.

1.3 Techniques

Proof sketch of Theorem 1.1. As a warm-up, first consider the case of a deterministic length-increasing many-one reduction. In particular, let R be such a reduction from SAT to $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{K}^t$ mapping inputs $\varphi \in \{0,1\}^n$ to outputs $(x, 1^s)$ with $|x| \geq n^{2/\delta}$ and with the superscript t greater than the running time of R. It is easy to see that, for any output $(x, 1^s)$ of $R(\varphi)$,

$$\begin{aligned}\mathsf{K}^t(x) &\leq |\varphi| + O(\log n)\\ &\leq |x|^\delta\\ &\leq s + |x|^\delta.\end{aligned}$$

This follows from the procedure that, given φ hard-coded, simulates $R(\varphi)$ and returns its output. Accordingly, a reduction of this kind cannot exist: since all of its outputs are Yes-instances, it would imply $\varphi \in \mathsf{SAT}$ for every formula φ.

When moving to the more general case of a randomized many-one reduction, one can think of $R(\varphi)$ as a distribution over instances of $\mathsf{Approx}_{n^\delta}$-$\mathsf{K}^t$, and a given output x is made with probability according to $R(\varphi)$. Observe that in the deterministic case, it held trivially that with high probability over $x \sim R(\varphi)$,

$$\mathsf{K}^t(x) \lesssim s \iff \Pr[R(\varphi) = x] > \beta,$$

for any choice of $\beta \in (0, 1)$. We would like to show that something similar is true in the randomized setting. That is, there is still a correspondence between the K^t complexity of outputs and their probability under $R(\varphi)$. This means that $\mathsf{Approx}_{n^\delta}$-$\mathsf{K}^t$ (and thereby SAT) will reduce to a problem of probability estimation.

There exists unconditionally a coAM protocol A that, given (φ, x, β) as input, accepts iff $\Pr[R(\varphi) = x]$ is roughly greater than β, with high probability over $x \sim R(\varphi)$ [For89; BT06b]; see also [HW15] Appendix A. Under our derandomization assumption, A can be implemented in coNP. For simplicity, assume that every output $(x, 1^s)$ of R has the same threshold parameter $s \in \mathbb{N}$, so we may omit this part of the outputs. Define a parameter

$$\beta = \frac{1}{2^s \cdot \mathsf{poly}(n)}.$$

We claim that for every $\varphi \in \{0, 1\}^n$, $A(\varphi, x, \beta)$ will work well at deciding $\mathsf{Approx}_{n^\delta}$-$\mathsf{K}^t$ on outputs x of $R(\varphi)$.

On one hand, we will show that with high probability over $x \sim R(\varphi)$, if $\mathsf{K}^t(x) \leq s$, then $\Pr[R(\varphi) = x] > \beta$. The idea is to use a counting argument, giving an upper bound on x such that $\mathsf{K}^t(x) \leq s$, to show that $R(\varphi)$ must be "concentrated" on these inputs. In particular, the probability over $x \sim R(\varphi)$ that $\mathsf{K}^t(x) \leq s$ and $\Pr[R(\varphi) = x] \leq \beta$ is roughly at most

$$2^s \cdot \beta = \frac{1}{\mathsf{poly}(n)}.$$

So, with high probability over $x \sim R(\varphi)$, if x is a Yes-instance of $\mathsf{Approx}_{n^\delta}$-$\mathsf{K}^t$, then $\Pr[R(\varphi) = x] > \beta$, in which case $A(\varphi, x, \beta)$ correctly outputs 1.

On the other hand, we will show that if an output x has probability greater than β under $R(\varphi)$, then x must have K^t complexity roughly upper-bounded by s. In the realm of time-unbounded Kolmogorov complexity, we could rely on the well-known Coding Theorem to prove a statement of this kind. Namely, for any samplable distribution D, it holds that

$$\mathsf{K}(x) \leq \log(1/D(x)) + O(\log n).$$

Similarly, if D is samplable given some non-uniform input φ, then

$$\mathsf{K}(x) \leq \log(1/D(x)) + |\varphi| + O(\log n).$$

Observe that our distribution $R(\varphi)$ is samplable in polynomial time given φ as input. Thus, if x is samplable with probability greater than β under $R(\varphi)$, then it holds that

$$\begin{aligned}\mathsf{K}(x) &< \log(1/\beta) + |\varphi| + O(\log n)\\ &\leq s + |\varphi| + O(\log n)\\ &\leq s + |x|^\delta.\end{aligned}$$

Of course, bounding K-complexity does not suffice for our purposes. Instead, we apply a recent work of Lu, Oliveira, and Zimand [LOZ22], which gives unconditionally a coding theorem for *probabilistic* K^t complexity, denoted pK^t. Specifically, we use a version of the coding theorem for distributions samplable in polynomial time given an auxiliary non-uniform input. For some polynomial p_{sc} and time-bound $t_0 = \mathsf{poly}(n)$ at least the running time of R, this yields

$$\mathsf{pK}^{p_{sc}(t_0)}(x) \leq s + |\varphi| + O(\log n).$$

Roughly speaking, pK^t-complexity refers to the time-bounded Kolmogorov complexity of a string in the presence of some uniform randomness. This notion is in some sense intermediate between K^t complexity and K complexity. Moreover, under the derandomization assumption $\mathsf{E} \not\subseteq \mathsf{io\text{-}NSIZE}[2^{o(n)}]$, pK^t and K^t turn out to be nearly equal: for some polynomial p_0, $\mathsf{K}^{p_0(t)}(x) \leq \mathsf{pK}^t(x) + \log p_0(t)$ [Gol+22]. So, for $t \geq p_0(p_{sc}(t_0))$, the above implies

$$\begin{aligned} \mathsf{K}^t(x) &\leq s + |\varphi| + O(\log n) \\ &\leq s + |x|^{\delta}. \end{aligned}$$

To summarize, with a sufficiently large $t = \mathsf{poly}(n)$ and a derandomization assumption, we obtain an auxiliary-input coding theorem for K^t complexity. This yields the required converse, namely, that high probability under $R(\varphi)$ implies bounded K^t.[4]

We conclude that the coNP procedure A can be used to decide SAT. Therefore, $\mathsf{NP} \subseteq \mathsf{coAM} = \mathsf{coNP}$.

To obtain Theorem 1.1 for *honest* reductions rather than polynomially length-increasing reductions, we can simply rely on the "paddability" of SAT. That is, given a SAT-instance $\varphi \in \{0,1\}^n$, it is trivial to append some terms to φ in a way that does not affect its satisfiability but increases its length as desired. Since our assumed reduction R is honest, for some constant $\gamma > 0$, for any query x of $R(\varphi)$, it holds that $|x| \geq |\varphi|^{\gamma}$. If we let R' be the reduction that, on input $\varphi \in \{0,1\}^n$, pads to obtain $\varphi' \in \{0,1\}^{n^{c/\gamma}}$ and then runs $R(\varphi')$ to obtain x, we will now have $|x| \geq |\varphi'|^{\gamma} = n^c$. To summarize, if there is an honest reduction from SAT to some language L, then there is also a polynomially length-increasing reduction from SAT to L.[5]

For the full statement of Theorem 1.1, we need techniques that can handle randomized non-adaptive Turing reductions. We exploit the fact from [IL90] that the non-existence of a one-way function would provide an algorithm A for probability estimation as described above. In particular, for any distribution $D \in \mathsf{PSAMP}$, for some poly-time computable function f, there is an oracle algorithm A such that $A^I(x)$ outputs an estimate of $\Pr[D = x]$ with high probability over $x \sim D$, where I is any inverter for f. Thus, in the presence of a non-adaptive reduction from SAT to $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{K}^t$, we also get a non-adaptive reduction from SAT to the inversion of a one-way function. It was shown in [Aka+06; Aka+10], with the construction of a sophisticated protocol building on techniques from [FF93; BT06b], that such a reduction would imply $\mathsf{SAT} \in \mathsf{coAM}$. However, as mentioned above, our distributions of interest $R(\varphi)$ are not in PSAMP, but require φ as a non-uniform input. Luckily, a result of [ABX08] transposes [Aka+06] to this non-uniform setting. Specifically, we have a reduction from SAT to the inversion of an *auxiliary-input* function $f = \{f_\varphi\}_{\varphi \in \{0,1\}^*}$, where on input φ to SAT, the reduction only needs to invert f on auxiliary

[4] We note that the use of the coding theorem for pK^t is the main reason why we need to require that the runtime of our randomized NP-hardness reductions for $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{K}^t$ must be polynomially smaller than the parameter t.

[5] A similar application of padding is in [Hir23].

input φ; given this, [ABX08] yields $\mathsf{SAT} \in \mathsf{coAM}$. This completes our overview of the proof of Theorem 1.1.

Proof Sketch of Theorem 1.2. Our proof of Theorem 1.2 builds on that of Theorem 1.1, making use of a few more ideas to obtain a reduction from NP to inversion of a standard OWF. The first idea is the fact that any inverter for an appropriate function can be used as an *errorless average-case* inverter for a desired auxiliary-input function. In particular, let $f = \{f_x\}_{x\in\mathbb{N}}$ be an auxiliary-input function, and define g to be the function that randomly samples x from a distribution D' and then applies f_x to a uniformly random input z. It is not hard to show by an averaging argument that any inverter for g works as an inverter for f_x with high probability over $x \sim D'$. Moreover, crucially, if the inverter fails to invert some f_x, then it can be made to output a special failure symbol $\perp$ when given the auxiliary input x, with high probability. This is due to the fact that successful inversion can be verified in poly-time: given a candidate pre-image y of some string z under f_x, simply run $f_x(y)$ to verify; see Lemma 2.20. This, along with a reduction from SAT to inverting an auxiliary-input OWF, yields an errorless randomized heuristic for SAT over any distribution $D' \in \mathsf{PSAMP}$.

The final piece of Theorem 1.2 is a worst-case to average-case reduction. The goal is to obtain

$$(\mathsf{SAT}, D') \in \mathsf{AvgBPP} \implies \mathsf{SAT} \in \mathsf{BPP},$$

which will complete the proof given the discussion above. To that end, we employ tools from [Hir18] and follow-up works. A difficulty is that, from $(\mathsf{SAT}, D') \in \mathsf{AvgBPP}$, the available worst-case to average-case reductions only yield

$$\mathsf{Gap}_{\tau,n^\delta}\mathsf{pK}^\mathsf{t} \in \mathsf{BPP}.$$

The promise-problem $\mathsf{Gap}_{\tau,n^\delta}\mathsf{pK}^\mathsf{t}$ is potentially easier than $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{pK}^t$, since it involves a polynomial gap τ between time-bounds in Yes-instances and No-instances. As a result, the gap version may not be NP-hard, so its easiness would not yield $\mathsf{SAT} \in \mathsf{BPP}$. Fortunately, by a different application of the coding theorem for pK^t, we are able to show that NP-hardness of $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{pK}^t$ implies NP-hardness of $\mathsf{Gap}_{\tau,n^\delta}\mathsf{pK}^\mathsf{t}$. Roughly, with high probability over the randomness of the reduction from SAT to $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{pK}^t$, the pK^t complexity of queried strings will be somewhat close to their time-unbounded K complexity. Thus, granted the leeway of the n^δ approximation term, the difference in time-bounds between t and $\tau(t)$ does not affect the correctness of the (slightly modified) reduction when we use $\mathsf{Gap}_{\tau,n^\delta}\mathsf{pK}^\mathsf{t}$ as an oracle in lieu of $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{pK}^t$. A more technical outline of the proof is given in Section 4.

Proof Sketch of Theorem 1.3. For the proof of Theorem 1.3 in the setting of exact pK^t and K^t, the approach discussed above does not work; recall that the approximation term n^δ was critical at a number of points. Thus, our starting point is the following statement from a recent work of Liu and Pass [LP23].

> Assuming $\mathsf{E} \not\subseteq \text{io-}\mathsf{NSIZE}[2^{o(n)}]$, if $\{\mathsf{MK}^t\mathsf{P}\} \times \mathsf{SAMP}[t_D(n)] \not\subseteq \mathsf{HeurP}$ for some time bound t_D polynomially less than t, then one-way functions exist.

That is, the average-case hardness of $\mathsf{MK}^t\mathsf{P}$ with respect to *any* distribution samplable within some polynomial running time smaller than t would suffice to imply one-way functions.

Our goal now is to show that if $\mathsf{MK}^t\mathsf{P}$ is NP-hard, then $\{\mathsf{MK}^t\mathsf{P}\} \times \mathsf{SAMP}[t_D(n)]$ is "hard for distributional NP": namely, if $\mathsf{MK}^t\mathsf{P}$ is easy on average over every distribution D samplable in time t_D, then every distributional problem $(L, D') \in \mathsf{NP} \times \mathsf{PSAMP}$ is likewise easy on average. Combining this with the statement from [LP23], we would get

$$\begin{aligned} \mathsf{DistNP} \not\subseteq \mathsf{HeurP} &\implies \{\mathsf{MK}^t\mathsf{P}\} \times \mathsf{SAMP}[t_D(n)] \not\subseteq \mathsf{HeurP} \\ &\implies \exists \mathsf{OWF}. \end{aligned}$$

To show the distributional NP-hardness of $\mathsf{MK}^t\mathsf{P}$, we reduce from an arbitrary distributional problem $(L, D') \in \mathsf{DistNP}$. Under the assumed NP-hardness of $\mathsf{MK}^t\mathsf{P}$, there is a randomized non-adaptive reduction R from L to $\mathsf{MK}^t\mathsf{P}$. With a large enough choice of the polynomial t, we can ensure that the reduction from L to $\mathsf{MK}^t\mathsf{P}$ runs in time polynomially less than t. In particular, we get that the following distribution Q is samplable in time at most t_D:

Sample $x \sim D'$, and then output a sample from the query distribution of $R(x)$.

From there, it is not too hard to show that, if H is a heuristic for $\mathsf{MK}^t\mathsf{P}$ working over Q, then the algorithm R^H (that simulates R and answers any oracle queries with H) is a heuristic for L over D'. This yields the desired result.

Proof Sketch of Theorem 1.4. Finally, the proof of Theorem 1.4 proceeds along the lines of that of Theorem 1.1, but with several important changes.[6] The main challenge is that the Coding Theorem for K only gives us an *approximate* equality between $\mathsf{K}(x)$ and $\log(1/D(x))$ for x's sampled from a distribution D. This was not a problem for Theorem 1.1 as it dealt with an *approximate* version of K^t, and we could absorb some slack of the Coding Theorem into an approximation error of K^t. But Theorem 1.4 is for the *exact* version of K, and we cannot apply the same strategy here. Instead, we show that this slack can be absorbed by a different argument, crucially relying on the fact that the randomized reductions R in the assumption of Theorem 1.4 are *many-one* and have the error probability *inverse-polynomially small* in their runtime t_R.

Namely, for $\varphi \in \{0,1\}^n$, consider the distribution of queries $(x, 1^s)$ made by the reduction $R(\varphi)$. We call such a query "heavy" if its probability (according to $R(\varphi)$) is at least $1/(\mathsf{poly}(t_R(n)) \cdot 2^s)$.

Our SAT algorithm (using a probability estimation protocol as in Theorem 1.1) essentially behaves as follows:

On input φ, sample a query $(x, 1^s)$ according to $R(\varphi)$, and accept if $(x, 1^s)$ is heavy.

For $\varphi \notin \mathsf{SAT}$ (which is the difficult case to analyze), heavy queries will cause our SAT algorithm to make a mistake by incorrectly accepting φ. We bound the error probability of our SAT algorithm by upperbounding the total probability mass of such heavy queries.

Roughly speaking, we upperbound the total probability mass of "heavy" queries $(x, 1^s)$ by

$$\mathsf{poly}(t_R(n)) \cdot \Pr[\mathsf{K}(x) \le s].$$

Note that, since $\varphi \notin \mathsf{SAT}$, we have by the condition of correctness of the many-one reduction R that $R(\varphi)$ must place a very small γ probability on its queries that are Yes-instances of MKP, i.e., $\Pr[\mathsf{K}(x) \le s] \le \gamma$. Hence, the error probability of our SAT algorithm is at most $\mathsf{poly}(t_R(n)) \cdot \gamma$, which can be made sufficiently small if the error probability γ of the reduction R is inverse-polynomially small in the runtime $t_R(n)$.

[6] We actually give two different proofs of Theorem 1.4. We describe the first one here. The second one uses techniques building on a work of Allender et al. [AHT23] to obtain $\mathsf{coNP} \subseteq \mathsf{NISZK} \subseteq \mathsf{AM} \cap \mathsf{coAM}$. More details may be found in Section 7.

1.4 Organization

In the next section, we give some technical preliminaries and notation. In Sections 3, 4, 5, and 6, we prove Theorems 1.1, 1.2, 1.3, and 1.4, respectively. In Section 7, we exhibit a connection between the hardness of K complexity and NISZK, which we use to prove Theorem 1.5. In Section 8, we discuss consequences of NP-hardness of promise problems of the form (K^t vs. $\mathsf{K}^{t'}$) in light of Hirahara's recent proof that the "partial function" versions of these promise problems are indeed NP-hard under randomized many-one reductions [Hir22]. We conclude in Section 9 with some open questions.

2 Preliminaries

2.1 Meta-complexity

Fix a universal Turing Machine U. The following three complexity measures are with respect to this choice.

Definition 2.1 (Kolmogorov Complexity)**.** *The Kolmogorov complexity of a string $x \in \{0,1\}^*$, denoted $\mathsf{K}(x)$, is equal to*

$$\min_{s \in \mathbb{N}} \{\, \exists d \in \{0,1\}^s \mid U(d) \text{ halts and outputs } x \,\}.$$

Definition 2.2 (Time-bounded Kolmogorov Complexity)**.** *For a time bound $t \in \mathbb{N}$ and a string $x \in \{0,1\}^*$, the t-time-bounded Kolmogorov complexity of x, denoted $\mathsf{K}^t(x)$, is equal to*

$$\min_{s \in \mathbb{N}} \{\, \exists d \in \{0,1\}^s \mid U(d) \text{ halts and outputs } x \text{ within } t \text{ steps} \,\}.$$

For a function $\tau : \mathbb{N} \to \mathbb{N}$, $\mathsf{K}^\tau(x)$ denotes $\mathsf{K}^{\tau(|x|)}(x)$.

Definition 2.3 (Probabilistic Time-bounded Kolmogorov Complexity [Gol+22])**.** *For a time bound $t \in \mathbb{N}$ and a string $x \in \{0,1\}^*$, the t-time-bounded probabilistic Kolmogorov complexity of x, denoted $\mathsf{pK}^t(x)$, is equal to*

$$\min_{s \in \mathbb{N}} \left\{ \Pr_{r \sim \mathcal{U}_t} [\, \exists d \in \{0,1\}^s \mid U(d,r) \text{ halts and outputs } x \text{ within } t \text{ steps} \,] \geq 2/3 \right\}.$$

For a function $\tau : \mathbb{N} \to \mathbb{N}$, $\mathsf{pK}^\tau(x)$ denotes $\mathsf{pK}^{\tau(|x|)}(x)$.

Definition 2.4 (Levin's Kt Complexity)**.** *For a string $x \in \{0,1\}^*$, the Kt complexity of x is equal to*

$$\min \{\, |d| + \log t \mid U(d) \text{ halts and outputs } x \text{ within } t \text{ steps} \,\}.$$

Definition 2.5 (Kolmogorov complexity approximation problems)**.** *For any $g, t : \mathbb{N} \to \mathbb{N}$, for $\mu \in \{\mathsf{K}^t, \mathsf{pK}^t, \mathsf{K}, \mathsf{Kt}\}$, Approx_g-μ is the following promise-problem (Π_Y, Π_N):*

- $\Pi_Y = \{(x, 1^s) \mid \mu(x) \leq s\}$
- $\Pi_N = \{(x, 1^s) \mid \mu(x) > s + g(|x|)\}$

For $s : \mathbb{N} \to \mathbb{N}$, we also consider a "fixed threshold" version, denoted Approx_g-$\mu[s]$, as follows.

- $\Pi_Y = \{x \mid \mu(x) \le s(|x|)\}$
- $\Pi_N = \{x \mid \mu(x) > s(|x|) + g(|x|)\}$

Definition 2.6 ($\mathsf{Gap}_{\tau,g}\mathsf{K}^{\mathsf{t}}$ and $\mathsf{Gap}_{\tau,g}\mathsf{pK}^{\mathsf{t}}$)**.** *For each $\mu \in \{\mathsf{K}, \mathsf{pK}\}$, and for any $g, \tau : \mathbb{N} \to \mathbb{N}$, $\mathsf{Gap}_{\tau,g}\mu^{\mathsf{t}}$ is the following promise-problem (Π_Y, Π_N):*

- $\Pi_Y = \{(x, 1^s, 1^t) \mid \mu^t(x) \le s\}$
- $\Pi_N = \{(x, 1^s, 1^t) \mid \mu^{\tau(t)}(x) > s + g(|x|)\}$

Definition 2.7 (MKP, $\mathsf{MK}^t\mathsf{P}$, and $\mathsf{MpK}^t\mathsf{P}$)**.** MKP *denotes the language $\{(x, 1^s) \mid \mathsf{K}(x) \le s\}$. For any $t \colon \mathbb{N} \to \mathbb{N}$, and for each $\mu \in \{\mathsf{K}, \mathsf{pK}\}$, $\mathsf{M}\mu^t\mathsf{P}$ denotes the language $\{(x, 1^s) \mid \mu^t(x) \le s\}$.*

Fact 2.8. *For every $t \in \mathbb{N}$ and $x \in \{0,1\}^*$, it holds that*

$$\mathsf{K}(x) \le \mathsf{pK}^t(x) \le \mathsf{K}^t(x).$$

Lemma 2.9 ([Gol+22])**.** *If $\mathsf{E} \not\subseteq \text{io-}\mathsf{NSIZE}[2^{o(n)}]$, then there is a polynomial p_0 such that*

$$\mathsf{K}^{p_0(t(n))}(x) \le \mathsf{pK}^{t(n)}(x) + \log p_0(t(n))$$

for every $t : \mathbb{N} \to \mathbb{N}$, $n \in \mathbb{N}$, and $x \in \{0,1\}^n$.

The Coding Theorem for $\mathsf{pK}^{\mathsf{poly}}$, due to Lu, Oliveira, and Zimand, is easily extended to the statement below, which allows for distributions samplable in polynomial time given an auxiliary non-uniform input.

Lemma 2.10 (Unconditional Coding Theorem for pK^t [LOZ22])**.** *There exists a polynomial p_{sc} such that for every $t, a : \mathbb{N} \to \mathbb{N}$, $D = \{D_z\}_{z \in \{0,1\}^*}$, where for every $z \in \{0,1\}^{a(n)}$, D_z is a distribution over $\{0,1\}^{\le n}$ that is samplable by a randomized $t(n)$-time algorithm on input $z \in \{0,1\}^{a(n)}$, we have the following: for every $n \in \mathbb{N}$, $z \in \{0,1\}^{a(n)}$, and $x \in \{0,1\}^{\le n}$ in the support of D_z,*

$$\mathsf{pK}^{p_{sc}(t(n))}(x) \le \log\left(1/\Pr[D_z = x]\right) + \log p_{sc}(t(n)) + a(n).$$

The following is similar to Lemma 35 of [SS22], adapted to $\mathsf{pK}^{\mathsf{poly}}$ and the case that the distribution is sampled non-uniformly.

Corollary 2.11. *Let p_{sc} be the polynomial of Lemma 2.10. For $t, a : \mathbb{N} \to \mathbb{N}$, $D = \{D_z\}$, where for every $z \in \{0,1\}^{a(n)}$, D_z is over $\{0,1\}^{\le n}$ and is samplable by a randomized $t(n)$-time algorithm given input z, we have for every $n \in \mathbb{N}$, $z \in \{0,1\}^{a(n)}$, and $1/2^{2n} < \varepsilon < 1$,*

$$\Pr_{x \sim D_z}\left[\mathsf{pK}^{p_{sc}(t(n))}(x) \le \mathsf{K}(x) + a(n) + \log p_{sc}(t(n)) + \log(16 \cdot n/\varepsilon)\right] \ge 1 - \varepsilon.$$

Proof. By Lemma 2.10, for any $z \in \{0,1\}^{a(n)}$ and $x \in \{0,1\}^{\le n}$ in the support of D_z,

$$\mathsf{pK}^{p_{sc}(t(n))}(x) \le \log(1/\Pr[D_z = x]) + \log p_{sc}(t(n)) + a(n). \tag{1}$$

We partition $\{0,1\}^{\le n}$ into subsets S_i, where for $i \in [4n-1]$,

$$S_i = \left\{ x \in \{0,1\}^{\le n} \;\middle|\; \frac{1}{2^i} < \Pr[D_z = x] \le \frac{1}{2^{i-1}} \right\},$$

and

$$S_{4n} = \left\{ x \in \{0,1\}^{\leq n} \;\middle|\; \Pr[D_z = x] \leq \frac{1}{2^{4n}} \right\}.$$

For $i \in [4n-1]$, call a string x *i-bad* iff $x \in S_i$ and $\mathsf{K}(x) < i - \log(16 \cdot n/\varepsilon)$. Observe that for $x \in S_i$, if

$$\mathsf{pK}^{p_{sc}(t(n))}(x) > \mathsf{K}(x) + a(n) + \log p_{sc}(t(n)) + \log(16 \cdot n/\varepsilon),$$

then by Eq. (1),

$$\begin{aligned}\mathsf{K}(x) &< \log(1/\Pr[D_z = x]) - \log(16 \cdot n/\varepsilon)\\ &\leq i - \log(16 \cdot n/\varepsilon),\end{aligned}$$

so x is i-bad; note that x may be i-bad only for $i \geq \log(16 \cdot n/\varepsilon)$, as otherwise $\mathsf{K}(x)$ would be negative, which is impossible. By a counting argument, for any $i \in [4n-1]$, the probability over D_z that a string is i-bad is at most

$$\frac{2^{i-\log(16 \cdot n/\varepsilon)}}{2^{i-1}} = \frac{\varepsilon}{8n}.$$

By a union bound over $i \in [4n-1]$, the total probability that the output of D_z is i-bad for some i is less than $\varepsilon/2$.

Finally, since there are at most 2^{n+1} strings of length at most n, the probability over $x \sim D_z$ that x belongs to S_{4n} is at most $1/2^{3n-1} < \varepsilon/2$. Applying another union bound, we conclude that

$$\Pr_{x \sim D_z}\left[\mathsf{pK}^{p_{sc}(t(n))}(x) > \mathsf{K}(x) + a(n) + \log p_{sc}(t(n)) + \log(4n/\varepsilon)\right] < \varepsilon,$$

as desired. □

We also have a version of the above for $\mathsf{K}^{\mathsf{poly}}$, under a derandomization assumption.

Corollary 2.12. *Assume* $\mathsf{E} \not\subseteq \text{io-}\mathsf{NSIZE}[2^{o(n)}]$. *There is a polynomial q_{sc} with the following property. For $t, a : \mathbb{N} \to \mathbb{N}$, $D = \{D_z\}$, where for every $z \in \{0,1\}^{a(n)}$, D_z is over $\{0,1\}^n$ and is samplable by a randomized $t(n)$-time algorithm given input z, we have for every $n \in \mathbb{N}$, $z \in \{0,1\}^{a(n)}$, and $0 < \varepsilon < 1$,*

$$\Pr_{x \sim D_z}\left[\mathsf{K}^{q_{sc}(t(n))}(x) \leq \mathsf{K}(x) + a(n) + \log q_{sc}(t(n)) + \log(4n/\varepsilon)\right] \geq 1 - \varepsilon.$$

Proof. Let q_{sc} in the statement of this corollary be such that $q_{sc}(n) = 2 \cdot p_0(p_{sc}(n))$ for all $n \in \mathbb{N}$, where p_{sc} is as in Corollary 2.11 and p_0 is as in Lemma 2.9. By Lemma 2.9, for any threshold $s \in \mathbb{N}$, if

$$\mathsf{pK}^{p_{sc}(t)}(x) \leq s,$$

then

$$\mathsf{K}^{p_0(p_{sc}(t))}(x) \leq s + \log p_0(p_{sc}(t)).$$

Thus, Corollary 2.11 implies that

$$\begin{aligned}\mathsf{K}^{q_{sc}(t)}(x) &\leq \mathsf{K}(x) + a(n) + \log p_{sc}(t) + \log(4n/\varepsilon) + \log p_0(p_{sc}(t))\\ &\leq \mathsf{K}(x) + a(n) + \log q_{sc}(t) + \log(4n/\varepsilon)\end{aligned}$$

with probability at least $1 - \varepsilon$ over $x \sim D_z$, as desired. □

2.2 Reductions

Definition 2.13 (Honest Reductions)**.** *For a constant $\gamma > 0$, a reduction is called* γ-honest *if, on inputs of length $n \in \mathbb{N}$, it only makes queries of length at least n^{γ}. In particular, if queries are parameterized in the form $(x, 1^t)$ for some $x \in \{0,1\}^*$ and $t \in \mathbb{N}$, γ-honesty requires that $|x| \geq n^{\gamma}$.*

We will simply say that a reduction is honest *if it is γ-honest for some constant $\gamma > 0$.*

Definition 2.14 (Length-increasing Reductions)**.** *A reduction is* n^c-length-increasing *if for any constant $c \in \mathbb{N}$, on inputs of length $n \in \mathbb{N}$, it only makes queries of length at least n^c.*

Lemma 2.15. *For any language L, if there is a γ-honest non-adaptive reduction from* SAT *to L running in time $t_R(n)$, then for any $c \in \mathbb{N}$, there is an n^c-length-increasing non-adaptive reduction from* SAT *to L running in time at most $O(t_R(n^{c/\gamma}))$.*

Proof. Let R be an honest non-adaptive reduction from SAT to L. For any $c \in \mathbb{N}$, let $R_c : \{0,1\}^n \to \{0,1\}^{n^{c/\gamma}}$ be the mapping that, given a SAT-instance $\varphi \in \{0,1\}^n$, outputs a SAT-instance $\varphi' \in \{0,1\}^{n^{c/\gamma}}$ such that $\varphi \in$ SAT iff $\varphi' \in$ SAT. This may be done in time at most $O(n^{c/\gamma})$, by the paddability of SAT.

Let R' be the reduction obtained by composing R_c and R. Namely,

> On input $\varphi \in \{0,1\}^n$, apply $R_c(\varphi)$ to obtain $\varphi' \in \{0,1\}^{n^{c/\gamma}}$. Then run the reduction $R(\varphi')$, which will make a number of queries to L.

By honesty, $R(\varphi')$ only makes queries of length at least $|\varphi'|^{\gamma} = n^c$. Thus, R' is a n^c-length-increasing reduction from SAT to L. Moreover, R' runs in time at most

$$O(n^{c/\gamma}) + t_R(n^{c/\gamma}) = O(t_R(n^{c/\gamma})),$$

and it is easy to see that R' is non-adaptive. □

Definition 2.16 (Class-specific Black-box Reductions [GT07])**.** *Consider languages L and L' and a complexity class C. A PPT oracle machine R is a C-black-box reduction from L to L' if, for every oracle $A \in C$ such that A decides L', it holds that R^A decides L.*

Note that if C is the class of all functions, then R is a (standard) black-box reduction from L to L'.

We also consider class-specific reductions to inverting (auxiliary-input) one-way functions. In particular for a (auxiliary-input) function f, R is a C-black-box reduction from L to inverting f if, for every oracle $I \in C$ that inverts f (as defined in Section 2.3), R^I decides L.

2.3 One-way Functions, Inversion, and Average-case Complexity

Definition 2.17 (HeurP and HeurBPP [BT06a])**.** *Let (L, D) be a distributional problem. An algorithm A is a* randomized heuristic scheme *for (L, D) if, for any $\delta > 0$, $n \in \mathbb{N}$, and x in the support of D_n, $A(x; n, \delta)$ runs in randomized time* $\mathsf{poly}(n/\delta)$*, and with probability at least $1-\delta$ over $x \sim D_n$,*

$$\Pr_A[A(x; n, \delta) = L(x)] \geq 2/3.$$

If A is deterministic, we simply call it a heuristic scheme *for (L, D).*

Define HeurP *to be the class of distributional problems with a heuristic scheme, and define* HeurBPP *to be those with a randomized heuristic scheme.*

Note that we will typically omit the parameters n and δ from the input to a heuristic scheme when they are clear from context.

Definition 2.18 (Inversion of functions)**.** *Consider a poly-time computable function $f = \{f_n : \{0,1\}^{s(n)} \to \{0,1\}^{\ell(n)}\}_{n\in\mathbb{N}}$, for polynomials s and ℓ, and a PPT algorithm I. For $\delta \in [0,1]$, we say that I* inverts f with failure probability δ *if for all sufficiently large $n \in \mathbb{N}$,*

$$\Pr_{I\ ;\ x\sim\mathcal{U}_{s(n)}}[\, f_n(I(1^n, f_n(x))) \neq f_n(x)\,] \leq \delta.$$

We say f is weakly invertible *if, for some PPT algorithm I and $b \in \mathbb{N}$, I inverts f with failure probability $1 - 1/n^b$. We say f is* strongly invertible *if, for every $b \in \mathbb{N}$ there exists some PPT algorithm I such that I inverts f with failure probability $1/n^b$.*

It is well known from the work of Yao that if every poly-time computable function is weakly invertible, then every such function is strongly invertible [Yao82]. Throughout this paper, we will use the statement that "one-way functions do not exist" to mean that every poly-time computable function $f = \{f_n\}_{n\in\mathbb{N}}$ is strongly invertible.

We also consider the inversion of auxiliary-input functions

$$f = \{f_x : \{0,1\}^{s(|x|)} \to \{0,1\}^{\ell(|x|)}\}_{x\in\{0,1\}^*},$$

for polynomials s and ℓ, in the sense of Ostrovsky and Wigderson [OW93], where $f_x(y)$ is computable in time $\mathsf{poly}(n)$ given $x \in \{0,1\}^n$ and $y \in \{0,1\}^{s(n)}$ as input.

Definition 2.19 (Inversion of auxiliary-input functions)**.** *Consider a poly-time computable auxiliary-input function $f = \{f_x\}_{x\in\{0,1\}^*}$ and a PPT algorithm I. For $x \in \{0,1\}^n$ and $\delta \in [0,1]$, we say that I* inverts f_x with failure probability δ *if*

$$\Pr_{z\sim\mathcal{U}_{s(n)}}[f_x(I(x, f_x(z))) \neq f_x(z)] \leq \delta.$$

The following lemma states that inversion of a standard one-way function implies errorless average-case inversion of an auxiliary input one-way function. This idea is implicit in [Nan21], and a similar statement is made explicit in [Hir23, Theorem 10.3]. We refer the reader to the proof in [Hir23].

Lemma 2.20. *There is a PPT non-adaptive oracle machine M satisfying the following. For any distribution $D' = \{D'_n\}_{n\in\mathbb{N}} \in \mathsf{PSAMP}$ and poly-time computable auxiliary-input function $f = \{f_x\}_{x\in\{0,1\}^*}$, there is a poly-time computable function $g = \{g_n\}_{n\in\mathbb{N}}$ such that for any $n \in \mathbb{N}$ and constant $d \in \mathbb{N}$, if I is a n^{-4d}-inverter for g_n, then M^I n^{-d}-inverts f_x with probability at least $1 - n^{-d}$ over $x \sim D'_n$. Moreover, if x does not have this property, then for all $z \in \{0,1\}^*$,*

$$\Pr_{M,I}\left[\, M^I(x,z) \text{ outputs } \perp \,\right] \geq 1 - 1/2^n.$$

2.4 Probability Estimation Protocols

For a probability distribution D over $\{0,1\}^m$, and for any $x \in \{0,1\}^m$, we denote by $D(x)$ the probability of x in D.

Lemma 2.21 (Leftover Hash Lemma [ILL89])**.** *Let X be a distribution over $\{0,1\}^t$ such that for every $z \in \{0,1\}^t$, the probability assigned by X to z is at most 2^{-m}. Let $\mathcal{H}_{t,k}$ be a universal family of hash functions mapping t bits to k bits. Then the following distributions have statistical distance at most $\sqrt{2^k/2^m}$:*

1. *$(h, h(x))$ for $h \sim \mathcal{H}_{t,k}$ and $x \sim X$, and*
2. *(h, v) for $h \sim \mathcal{H}_{t,k}$ and $v \sim \mathcal{U}_k$.*

The lemma below follows from [IL90], modified for the case that samplers for the distributions D_x require a non-uniform input x. It gives a protocol that estimates, to within a constant multiplicative factor, the probability of a given sample z from a given distribution D_x, assuming oracle access to an inverter for an auxiliary-input function f_x defined from the distribution D_x.

Lemma 2.22. *Let $D = \{D_x\}_{x \in \{0,1\}^*}$ be such that each D_x is samplable in time $t_D(n)$ given $x \in \{0,1\}^n$ as input. There exist an auxiliary-input function $f = \{f_x\}_{x \in \{0,1\}^*}$ and a randomized algorithm A as follows. For any $n \in \mathbb{N}$, $x \in \{0,1\}^n$, constant $d \in \mathbb{N}$, and oracle I that inverts f_x with failure probability at most $1/(10 \cdot n^d)$,*

$$\Pr_{z \sim D_x\ ;\ A}[(1/c) \cdot p_z \leq A^I(x,z) \leq 2 \cdot p_z] \geq 1 - \frac{1}{n^d},$$

where $p_z = D_x(z)$ and $c \in \mathbb{N}$ is a universal constant. Moreover, A runs in time $\mathsf{poly}(n)$ and queries its oracle non-adaptively.

Proof. We follow a proof given in [IRS21], modified appropriately for the auxiliary-input setting.

Let M be a machine that samples D_x, for $x \in \{0,1\}^n$, in time $t := t_D(n)$ given x as input. For $k \in \mathbb{N}$, let $\mathcal{H}_{t,k}$ be a universal hash function family mapping t bits to k bits. Consider the auxiliary-input function $f = \{f_x\}_{x \in \{0,1\}^*}$ defined as

$$f_x(r,k,h) = (M(x,r), k, h, h(r)),$$

where $r \in \{0,1\}^t$ is the randomness used by M, and $h \in \mathcal{H}_{t,k}$. Let I be an inverter for f_x.

Define an algorithm A_0 with oracle access to I as follows.

> On input (x,z,k), sample $h \sim \mathcal{H}_{t,k}$ and $w \sim \mathcal{U}_k$, and simulate $I(z,k,h,w)$. If I finds a valid pre-image, accept. Otherwise, reject.

Let

$$R_z = \{r \in \{0,1\}^t \mid M(x,r) = z\}.$$

Let $k \in [t+2]$ be arbitrary. First, consider the case that $p_z = D_x(z) \leq 2^{k-t}/4$. Then we have that $|R_z| \leq 2^k/4$. Thus, for a fixed $h \in \mathcal{H}_{t,k}$, the number of buckets $w \in \{0,1\}^k$ such that $h(r) = w$ for some $r \in R_z$ is at most $2^k/4$. The probability that A_0 samples some such $w \in \{0,1\}^k$ is at most

$$\frac{2^k}{4} \cdot \frac{1}{2^k} = \frac{1}{4}.$$

Hence, the probability (over its internal randomness) that $A_0(x,z,k)$ rejects is at least $3/4$.

Next consider the case that $p_z \geq c \cdot 2^{k-t}/4$, where we choose $c = 256$. Then, we have $|R_z| \geq c \cdot 2^k/4$. Consider the distribution X over $\{0,1\}^t$ that samples a uniformly random string from R_z.

By the Leftover Hash Lemma (Lemma 2.21) applied to X, we have that the distributions $(h, h(r))$ and (h, v), for $h \sim \mathcal{H}_{t,k}$, $r \sim R_z$, and $v \sim \mathcal{U}_k$, have statistical distance at most

$$\sqrt{\frac{2^k}{c \cdot 2^k/4}} = \frac{1}{8}.$$

Hence, for every $z \in \{0,1\}^t$ and $k \in [t]$ with $|R_z| \geq c \cdot 2^k/4$, the following two distributions have statistical distance at most 1/8:

- (z, k, h, v), for $h \sim \mathcal{H}_{t,k}$ and $v \sim \mathcal{U}_k$;
- $(z, k, h, h(r))$, for $r \sim R_z$ and $h \sim \mathcal{H}_{t,k}$.

Sampling $r \sim \mathcal{U}_t$, $h \sim \mathcal{H}_{t,k}$, and outputting $(M(x,r), k, h, h(r))$ is equivalent to sampling $z \sim D_x$, $r \sim R_z$, $h \sim \mathcal{H}_{t,k}$, and outputting $(z, k, h, h(r))$. Then, by an averaging argument, we get that, with probability at least $1 - 1/(2n^d)$ over $z \sim D_x$,

$$\Pr_{r \sim R_z \; ; \; h \sim \mathcal{H}_{t,k}}[I \text{ fails to invert } (z, k, h, h(r))] \leq \frac{1}{8}. \tag{2}$$

So, with probability at least $1 - 1/(2n^d)$ over $z \sim D_x$, we have that if $p_z \geq c \cdot 2^{k-t}/4$, then the probability (over its internal randomness) that $A_0(x, z, k)$ rejects is at most the failure probability of I in (2) plus the statistical distance between the distribution in (2) and the distribution sampled by $A_0(x, z, k)$ (with its internal randomness). The former is at most 1/8 by (2), and the latter is also at most 1/8 by the argument above based on the Leftover Hash Lemma. Hence, in this case, $A_0(x, z, k)$ rejects with probability at most $1/8 + 1/8 = 1/4$.

Let A be the algorithm that, for every $1 \leq k \leq t + 2$, repeats A_0 for n times on independent choices of h and w, and outputs $2^{k-t}/4$ for the smallest choice of k such that $A_0(x, z, k)$ rejects more than half the time.

For all but $1/(2n^d)$ of $z \sim D_x$, we have by the Chernoff bounds that this k must be such that $p_z \leq c \cdot 2^{k-t}/4$, with probability at least $1 - 2^{-\Omega(n)}$ over the internal randomness of A. On the other hand, let $1 \leq k' \leq t + 2$ be the maximum value such that $p_z \leq 2^{k-t}/4$, and hence $2^{k'-t-1}/4 < p_z$. Again by the Chernoff bounds, A outputs $k \leq k'$, with probability at least $1 - 2^{-\Omega(n)}$ over its internal randomness. It follows, that the estimate $2^{k-t}/4$ output by $A(x, z)$ is such that

$$p_z/c \leq 2^{k-t}/4 \leq 2 \cdot p_z,$$

with probability at least $1 - 1/(2n^d) - 2 \cdot 2^{-\Omega(n)} \geq 1 - 1/n^d$, where the probability is over $z \sim D_x$ and the internal randomness of A. The lemma follows. □

Finally, we will need the lemma that gives a coAM protocol for any language that is reducible (in a certain particular way) to the task of inverting auxiliary-input one-way functions.

We need the following definition.

Definition 2.23 (Fixed-auxiliary-input reduction [ABX08]). *Consider a language L, an auxiliary-input function $f = \{f_x\}_{x \in \{0,1\}^*}$, a randomized poly-time non-adaptive oracle machine R, and some $\varepsilon \in (0, 1)$. For any string $x \in \{0,1\}^*$, we say that R is a* non-adaptive fixed-auxiliary-input reduction *from $L(x)$ to ε-inverting f_x if, for some polynomial p, for every oracle I such that I inverts f_x with failure probability ε,*

$$\Pr_R[R^I(x) = L(x)] \geq \frac{1}{2} + \frac{1}{p(n)}.$$

If R satisfies the above for every sufficiently large $x \in \{0,1\}^*$, we simply say that R is a non-adaptive fixed-auxiliary-input reduction from L to ε-inverting f.

Lemma 2.24 (AM∩coAM protocol from fixed-auxiliary-input reduction [ABX08; Aka+06])**.** *Consider a language L, a poly-time computable auxiliary-input function $f = \{f_x\}_{x \in \{0,1\}^*}$, and a constant $d \in \mathbb{N}$. If there is a non-adaptive fixed-auxiliary-input reduction R from L to n^{-d}-inverting f, then $L \in \mathsf{AM} \cap \mathsf{coAM}$.*

The protocol above was adapted for the auxiliary-input setting by [ABX08] from [Aka+06]. We include a sketch of the proof below for completeness. The proof relies on the lower-bound protocol of Goldwasser and Sipser [GS86], and the upper-bound protocol of Fortnow [For89] (see also [AH91]), briefly described below. We consider sets $S \subseteq \{0,1\}^n$ such that membership can be verified in poly-time by a machine V given an auxiliary input string z.

Very roughly, in the [GS86] lower-bound protocol, on input $(s; z, 1^n)$, the verifier randomly selects a pairwise-independent hash function h and sends it to the prover. The prover then sends to the verifier some strings $\{x_1, \ldots, x_\ell\}$. The verifier checks that each $V(x_i, z) = 1$ and that $h(x_i) = 0$. By repeating the above a number of times to amplify the gap in probabilities, we obtain the following statement.

Lemma 2.25 (Lower-bound Protocol [GS86])**.** *Let $\delta > 0$ be a constant. There is an Arthur-Merlin protocol such that, on input $(s; z, 1^n)$,*

- *If $s \leq |S|$, then some prover causes the protocol to accept with probability $1 - \delta$.*
- *If $s > (1 + \delta) \cdot |S|$, then all provers cause the protocol to reject with probability $1 - \delta$.*

The idea behind the [For89; AH91] (see also [BT06b]) upper-bound protocol is as follows. On input $(s; z, 1^n)$, the verifier first obtains a secret uniformly random member x of S. Then, it randomly selects a hash function h, and sends to the prover both h and $h(x)$. The prover sends the verifier some strings $\{x_1, \ldots, x_l\}$. The verifier checks that each $V(x_i, z) = 1$ and that $x \in \{x_1, \ldots, x_l\}$.

Lemma 2.26 (Upper-bound Protocol [For89; AH91; BT06b])**.** *Let $\delta > 0$ be a constant. There is an Arthur-Merlin protocol such that, on input $(s; z, 1^n)$, if the verifier has access to secret uniformly random elements of S, the following holds.*

- *If $s \geq |S|$, then some prover causes the protocol to accept with probability $1 - \delta$.*
- *If $s < (1 - \delta) \cdot |S|$, then all provers cause the protocol to reject with probability $1 - \delta$.*

Proof sketch of Lemma 2.24. The prover in the AM∩coAM protocol will take the place of the oracle I in a simulation of $R(x)$. Let Q_x denote the query distribution and $t_R(n)$ the running time of R on input $x \in \{0,1\}^n$. In order to force the prover not to cheat by saying that some query cannot be inverted, the verifier will ask the prover to provide the pre-image sizes $|f_x^{-1}(y)|$ for all queries y

made by $R(x)$. In order to enforce these sizes, one would like to use the protocols of Goldwasser and Sipser [GS86] and Aiello and Håstad [AH91]. However, the latter can only be applied when the verifier has access to secret (i.e., hidden from the prover) uniformly random elements of $f_x^{-1}(y)$, which will not be possible in general for $y \sim Q_x$.

To remedy this issue, the verifier asks for [GS86] lower-bound proofs of the $|f_x^{-1}(y)|$ and also for the expectation $\mathbb{E}[|f_x^{-1}(Q_x)|]$. By a Chernoff bound, with very high probability, an empirical average of $|f_x^{-1}(y)|$ over many samples $y \sim Q_x$ should be close to this value; if it is not, the verifier can abort. This together with the lower bound proofs will ensure that the prover can neither falsely over- nor under-estimate the sizes of these sets frequently.

However, we still need a protocol for the prover and verifier to agree on $\mathbb{E}[|f_x^{-1}(Q_x)|]$, and the aforementioned obstacle for [AH91] still applies here: the verifier has no way to sample random preimages of $y \sim Q_x$. In contrast, [AH91] *can* be applied to give an upper bound on $\mathbb{E}[|f_x^{-1}(f_x(\mathcal{U}_n)|]$: the verifier can privately sample random $x_1, \ldots, x_m \sim \mathcal{U}_n$ and send the values $y_1 = f(x_1), \ldots, y_m = f(x_m)$ to the prover for upper- and lower-bound proofs. To take advantage of this, one divides queries into 'heavy' and 'light' categories, where for a threshold t, a query y is 't-heavy' iff

$$\Pr[Q_x = y] > t \cdot \Pr[f_x(\mathcal{U}_n) = y].$$

In the case that all queries are t-light for $t = \mathsf{poly}(n)$, the verifier can use a 'hiding protocol', sending the prover hidden $y \sim Q_x$ randomly interspersed with $y \sim f_x(\mathcal{U}_n)$. One can show that if the prover frequently cheats on $y \sim Q_x$, then it is likely also to violate [AH91] on one of the $y \sim f_x(\mathcal{U}_n)$. In this way, the expectation

$$\mathbb{E}_{y \sim Q(x)}[\ \Pr[f_x(\mathcal{U}_n) = y] \mid y \text{ is light }]$$

can be verified. See [Aka+06] for more details.

Moreover, observe from the definition of 't-heavy' that the probability over $y \sim f_x(\mathcal{U}_n)$ that y is t-heavy is at most $1/t$. Thus, choosing a uniformly random $t \sim [n^{d+1}, n^{d+2}]$ we can let the prover answer $\perp$ on all t-heavy queries, and since the prover still n^{-d}-inverts f_x, the reduction should still succeed. Thus, we would like an $\mathsf{AM} \cap \mathsf{coAM}$ protocol to tell apart t-heavy and t-light queries, and then we will do size-verification only on the t-light queries.

In light of the foregoing discussion, we arrive at the following protocol.

- Sample a uniformly random $t \sim [n^{d+1}, n^{d+2}]$.
- Sample m sets of queries $(y_1^{(1)}, \ldots, y_k^{(1)}), \ldots, (y_1^{(m)}, \ldots, y_k^{(m)})$ by simulating $R(x)$ on independent random strings $r_1, \ldots, r_m \sim \mathcal{U}_{t_R(n)}$, where k is the number of queries made in an execution of $R(x)$ and m is a sufficiently large polynomial in n.
- For each $(i, j) \in [m] \times [k]$, ask the prover for $\Pr[Q_x = y_j^{(i)}]$, and use [GS86] to enforce lower bounds.
- For each $(i, j) \in [m] \times [k]$, ask the prover for $\Pr[f_x(\mathcal{U}_n) = y_j^{(i)}]$, and use [GS86] to enforce lower bounds. Along with the above, this entails a commitment from the prover as to which queries are t-light and which are t-heavy.
- Take the average
$$\mu_R := \mathbb{E}_{i,j}[\Pr[Q_x = y_j^{(i)}]].$$

Take the average

$$\mu_f := \mathbb{E}_{i,j}[\Pr[f_x(\mathcal{U}_n) = y_j^{(i)}] \mid y_j^{(i)} \text{ is claimed } t\text{-light }].$$

- Using [GS86] and [AH91], obtain estimates of

$$\mu'_R := \mathbb{E}_{y \sim Q_x}[\Pr[Q_x = y]].$$

- Using a hiding protocol as described above, obtain estimates of

$$\mu'_f := \mathbb{E}_{y \sim Q_x}[\ \Pr[f_x(\mathcal{U}_n) = y] \mid y \text{ is } t\text{-light }].$$

- Abort if μ_R or μ_f deviate significantly from μ'_R or μ'_f respectively.
- For a random $j \sim [m]$, simulate $R(x)$ on randomness r_j and queries $(y_1^{(j)}, \ldots, y_k^{(j)})$. For each $i \in [k]$, if $y_i^{(j)}$ is claimed t-heavy or if the provided value $\Pr[f_x(\mathcal{U}_n) = y_j^{(i)}]$ is equal to 0, answer the query with $\perp$. Otherwise, answer the query with a pre-image of $y_i^{(j)}$ under f_x, which the prover provides and the verifier checks. Output the output of $R(x)$.

To see the correctness of the protocol, first note that with high probability, almost all t-heavy and t-light queries are correctly classified as such. This is due to the fact that we have correct upper- and lower-bounds for each $\Pr[Q_x = y_j^{(i)}]$. On one hand, since we have correct lower-bounds for each $\Pr[f_x(\mathcal{U}_n) = y_j^{(i)}]$, all t-light queries have proofs of being t-light. On the other hand, by the use of [GS86], the only possible inaccuracies of the provided $\Pr[f_x(\mathcal{U}_n) = y_j^{(i)}]$ are overestimations: thus, a t-light query is never claimed t-heavy.

Next, since the verifier obtains correct values of $\Pr[f_x(\mathcal{U}_n) = y_j^{(i)}]$ on all light queries $y_j^{(i)}$, the prover cannot make many false claims of light queries being non-invertible. By the above reasoning that answering $\perp$ on all heavy queries does not violate the condition of the reduction, the lemma follows. □

We also need the following result of Nanashima [Nan21].

Theorem 2.27 ([Nan21])**.** *For any poly-time computable auxiliary-input function $f = \{f_z\}_{z \in \{0,1\}^*}$ and any constant $d \in \mathbb{N}$, if there exists a black-box non-adaptive reduction from SAT to n^{-d}-inverting f, then there is a black-box adaptive reduction from SAT to the task of inverting one-way functions. Hence, if no one-way functions exist, then $\mathsf{NP} \subseteq \mathsf{BPP}$.*

3 Proof of Theorem 1.1

The following lemma comprises the first part of Theorem 1.1, Item 1. The conclusion $\mathsf{NP} \subseteq \mathsf{coAM}$ follows by combining Lemma 3.1 and Lemma 2.24.

Lemma 3.1. *For any constants $\delta, \gamma > 0$, there is a polynomial p_1 such that, for any $t, t_R : \mathbb{N} \to \mathbb{N}$ satisfying $p_1(t_R(n)) \leq t(n)$ for all $n \in \mathbb{N}$, we have the following. If $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{pK}^t$ is hard for SAT under a γ-honest non-adaptive randomized reduction running in time t_R, then there exist a poly-time computable auxiliary-input function $f = \{f_\varphi\}_{\varphi \in \{0,1\}^*}$, a constant $d \in \mathbb{N}$, and a black-box non-adaptive fixed-auxiliary-input reduction from SAT to n^{-d}-inverting f.*

Proof. Let R_1 be the assumed γ-honest $t_R(n)$-time reduction from SAT to $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{pK}^t$, for $\gamma, \delta > 0$. Let R_2 be the $n^{2/\delta}$-length-increasing randomized reduction from SAT to $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{pK}^t$ of Lemma 2.15 applied to R_1. The reduction R_2 runs in time less than

$$t'(n) := t_R(n^{4/(\delta\cdot\gamma)}).$$

For any $\varphi \in \{0,1\}^n$, let Q_φ denote the query distribution of R_2 on input φ.[7] Note that Q_φ is samplable by a randomized $t'(n)$-time algorithm given input $\varphi \in \{0,1\}^n$. Also note that, by honesty of the reduction and the definition of $\mathsf{Approx}_g\text{-}\mathsf{pK}^t$, each query $(x, 1^s)$ of R_2 on input $\varphi \in \{0,1\}^n$ is asking if $\mathsf{pK}^t(x) \leq s$, or if $\mathsf{pK}^t(x) > s + n^2$. Since $\mathsf{pK}^t(x) \leq |x| + c_u$, for some universal constant $c_u \in \mathbb{N}$ dependent on the choice of the universal Turing machine, we may assume that s is such that $s + n^2 \leq |x| + c_u \leq t'(n) + c_u$, since otherwise we can always correctly answer such a query $(x, 1^s)$ as 'Yes'. Below we shall always assume that a query $(x, 1^s)$ of R_2 is such that $s \leq t'(n) - n^2 + c_u$.

Let $f = \{f_\varphi\}_{\varphi\in\{0,1\}^*}$ be the auxiliary-input function, A the non-adaptive oracle algorithm, and $c \in \mathbb{N}$ the constant of Lemma 2.22 applied to $\{Q_\varphi\}_{\varphi\in\{0,1\}^*}$ and a constant $d \in \mathbb{N}$ such that $n^d \geq t'(n)^2$.

Let $m = t'(n)$. For any $s \in \mathbb{N}$, let

$$\beta_s = \frac{1}{c \cdot m^3 \cdot 2^{s+1}}.$$

Define a reduction as follows.

> On input $\varphi \in \{0,1\}^n$, simulate $R_2(\varphi)$, obtaining queries to $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{pK}^t$. Accept a query $(x, 1^s)$ iff $A^I(\varphi, (x, 1^s)) \geq \beta_s$ (which will require a number of non-adaptive queries to an inverter I for f_φ). Accept φ iff R_2 accepts.

We claim that the above is a black-box fixed-auxiliary-input reduction from SAT to $(n^{-d}/10)$-inverting f. To that end, consider any $\varphi \in \{0,1\}^n$. We will bound the probability over queries $(x, 1^s) \sim Q_\varphi$ that the query is answered incorrectly by evaluating $A^I(\varphi, (x, 1^s)) \geq \beta_s$, where I is an arbitrary inverter for f_φ with failure probability $n^{-d}/10$. There are two ways for this to happen: either $\mathsf{pK}^t(x) \leq s$ but $A^I(\varphi, (x, 1^s)) < \beta_s$, or $\mathsf{pK}^t(x) > s + n^2$ but $A^I(\varphi, (x, 1^s)) \geq \beta_s$.

We start by bounding the probability of the first kind of failure. By Lemma 2.22,

$$\begin{aligned}
\Pr_{(x,1^s)\sim Q_\varphi\,;\,A}[\,\mathsf{pK}^t(x) \leq s \ \wedge\ A^I(\varphi, (x, 1^s)) < \beta_s\,] &\leq \Pr[\,\mathsf{pK}^t(x) \leq s \ \wedge\ Q_\varphi(x, 1^s) \leq c \cdot \beta_s\,] + \frac{1}{m^2} \\
&\leq \sum_{s=0}^{m-n^2+c_u} \sum_{x\in\{0,1\}^{\leq m}\,:\ \mathsf{pK}^t(x)\leq s} c \cdot \beta_s + \frac{1}{m^2} \\
&\leq \sum_{s=0}^{m-n^2+c_u} 2^{s+1} \cdot c \cdot \beta_s + \frac{1}{m^2} \\
&\leq \frac{m}{m^3} + \frac{1}{m^2} \\
&\leq \frac{2}{m^2}.
\end{aligned}$$

[7] Without loss of generality, we may assume the queries of R_1 are identically (though not necessarily independently) distributed. It is always possible to obtain a reduction with this property by applying a random permutation to the queries of a non-adaptive reduction.

To bound the probability of the second kind of failure, observe that for all $(x, 1^s)$ in the support of Q_φ, we have

$$\begin{aligned} \mathsf{pK}^{2\cdot p_{sc}(t'(n))}(x) &\le \mathsf{pK}^{p_{sc}(t'(n))}(x, 1^s) + O(\log n) \\ &\le \log(1/Q_\varphi(x, 1^s)) + |\varphi| + O(\log n), \qquad \text{(Lemma 2.10)} \end{aligned}$$

where the polynomial p_{sc} is as in Lemma 2.10. In particular, if $Q_\varphi(x, 1^s) \ge \beta_s/2$, then for $t \ge 2 \cdot p_{sc}(t'(n))$,

$$\begin{aligned} \mathsf{pK}^t(x) &\le \log(2/\beta_s) + |\varphi| + O(\log n) \\ &\le s + |\varphi| + O(\log n) \\ &\le s + n^2. \end{aligned} \tag{3}$$

Taking the contrapositive,

$$\mathsf{pK}^t(x) > s + n^2 \implies Q_\varphi(x, 1^s) < \beta_s/2.$$

Define p_1 in the statement of this theorem as

$$p_1(n) = 2 \cdot p_{sc}(n^{4/(\delta\cdot\gamma)}).$$

This ensures that $p_1(t_R(n)) \ge 2 \cdot p_{sc}(t'(n))$. Then, for $t \ge p_1(t_R(n))$,

$$\begin{aligned} &\Pr_{(x,1^s)\sim Q_\varphi\,;\,A}[\ \mathsf{pK}^t(x) > s + n^2 \ \wedge\ A^I(\varphi, (x, 1^s)) \ge \beta_s\ ] \\ &\le \Pr[\ \mathsf{pK}^t(x) > s + n^2 \ \wedge\ Q_\varphi(x, 1^s) \ge \beta_s/2\ ] + \frac{1}{m^2} \qquad \text{(Lemma 2.22)} \\ &\le \frac{1}{m^2}. \end{aligned}$$

Overall, by a union bound, the probability over Q_φ that *some* query of R_2 is answered incorrectly by A^I is at most $2/m$. When A^I answers all queries of $R_2(\varphi)$ correctly with respect to $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{pK}^t$, the reduction outputs $\mathsf{SAT}(\varphi)$ with probability at least 2/3. This completes the proof. □

We can also prove Item 2 of Theorem 1.1 via Lemma 3.1, by making use of the derandomization assumption. Specifically, we show that if $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{K}^t$ is hard for SAT under an honest reduction, then the same is true of $\mathsf{Approx}_{n^{\delta/2}}\text{-}\mathsf{pK}^{t'}$ for $t = \mathsf{poly}(t')$.

Lemma 3.2. *Assume* $\mathsf{E} \not\subseteq \text{io-}\mathsf{NSIZE}[2^{o(n)}]$. *Let* p_0 *be the polynomial of Lemma 2.9 and* p_{sc} *the polynomial of Lemma 2.10. For any constants* $\delta, \gamma > 0$ *and* $t, t_R : \mathbb{N} \to \mathbb{N}$ *satisfying* $t(n) \ge p_0(p_{sc}(t_R(n^{4/(\gamma\cdot\delta)})))$ *for all* $n \in \mathbb{N}$, *we have the following. If* $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{K}^t$ *is hard for* SAT *under an honest non-adaptive randomized reduction running in time* t_R, *then* $\mathsf{Approx}_{n^{\delta/2}}\text{-}\mathsf{pK}^{p_0^{-1}(t)}$ *is hard for* SAT *under a* $n^{2/\delta}$*-length-increasing non-adaptive randomized reduction running in time* $t_R(n^{4/(\gamma\cdot\delta)})$.

Proof. By Lemma 2.15, there is a $n^{2/\delta}$-length-increasing randomized reduction R from SAT to $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{K}^t$, samplable in time at most $t_R(n^{4/(\gamma\cdot\delta)})$. On input $\varphi \in \{0,1\}^n$, R makes queries to $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{K}^t$ of the form $(x, 1^s)$ with $|x| \ge n^{2/\delta}$.

Given any $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{K}^t$ query $(x, 1^s)$ in the course of the reduction R, we will instead query $(x, 1^{s'})$ to an oracle for $\mathsf{Approx}_{n^{\delta/2}}\text{-}\mathsf{pK}^{t'}$, where $s' := s + 4n$ and $t'(n) := p_0^{-1}(t(n))$ for all $n \in \mathbb{N}$.

First consider the case that $(x, 1^s)$ is a Yes-instance of $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{K}^t$, so

$$\mathsf{K}^t(x) \leq s.$$

Adding $(\mathsf{K}^{t'}(x) - \mathsf{K}^t(x))$ to both sides of the equation, it follows that

$$\begin{aligned}
\mathsf{pK}^{t'}(x) &\leq \mathsf{K}^{t'}(x) \\
&\leq s + (\mathsf{K}^{t'}(x) - \mathsf{K}^t(x)) \\
&\leq s + (\mathsf{K}^{t'}(x) - \mathsf{K}(x)) \\
&\leq s + 4n, && \text{(Corollary 2.12)}
\end{aligned}$$

where the last line holds with probability at least $1 - 2^{-n}$ over the internal randomness of R. In this case, $(x, 1^{s'})$ is also a Yes-instance of $\mathsf{Approx}_{n^{\delta/2}}\text{-}\mathsf{pK}^{t'}$.

Now suppose $(x, 1^s)$ is a No-instance of $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{K}^t$, so

$$\mathsf{K}^t(x) > s + |x|^\delta.$$

This implies that

$$\begin{aligned}
\mathsf{pK}^{t'}(x) &= \mathsf{pK}^{p_0^{-1}(t)}(x) \\
&\geq \mathsf{K}^t(x) - O(\log n) && \text{(Lemma 2.9)} \\
&> s + |x|^\delta - O(\log n) \\
&> s' + |x|^{\delta/2}
\end{aligned}$$

Thus, $(x, 1^{s'})$ is a No-instance of $\mathsf{Approx}_{n^{\delta/2}}\text{-}\mathsf{pK}^{t'}$. □

Item 2 of Theorem 1.1 now follows from Lemma 3.1, Lemma 2.24, and Lemma 3.2.

We also obtain the following statement for Levin's Kt complexity.

Corollary 3.3. *For any constant $\delta > 0$, we have the following. Assume $\mathsf{E} \not\subseteq \text{io-}\mathsf{NSIZE}[2^{o(n)}]$. If $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{Kt}$ is hard for SAT under an honest non-adaptive randomized reduction, then $\mathsf{NP} = \mathsf{coNP}$. Moreover, if no one-way functions exist, then $\mathsf{NP} \subseteq \mathsf{P}$.*

Proof. Argue as in Lemma 3.1. In "bounding the probability of the first kind of failure", simply replace pK^t with Kt and argue analogously. For the "second kind of failure", apply the derandomization assumption and Lemma 2.9 at Eq. (3) to obtain that

$$\mathsf{K}^{p_0(t)}(x) \leq s + |\varphi| + O(\log n),$$

and therefore

$$\mathsf{Kt}(x) \leq s + |\varphi| + O(\log n).$$

The rest of Lemma 3.1 proceeds analogously. The first part of the corollary then follows from Lemma 2.24 and the derandomization assumption. The second part follows from Theorem 2.27. □

Finally, we remark that using the techniques of Lemma 3.1, we can recover the result of [SS22] for logarithmic additive error, while doing away with the requirement of "fixed query length".

Corollary 3.4. *There is a polynomial p such that, for any constant $\delta > 0$ and $t, t_R : \mathbb{N} \to \mathbb{N}$ satisfying $p(t_R(n)) \leq t(n)$ for all $n \in \mathbb{N}$, we have the following. Assume $\mathsf{E} \not\subseteq \mathsf{io\text{-}NSIZE}[2^{o(n)}]$. If $\mathsf{Approx}_{\omega(\log n)}\text{-}\mathsf{K}^t$ is hard for SAT under an honest non-adaptive randomized reduction running in time t_R, then $\mathsf{NE} = \mathsf{coNE}$.*

Proof. Argue as in Theorem 1.1, Item 2, but for the case of unary languages in NP. In Lemma 3.1, we now consider unary inputs $\varphi = 1^n$ for $n \in \mathbb{N}$. In this case, the query distribution Q_φ can be sampled in time $t'(n)$ given $\log n$ bits encoding n. That is, $Q_\varphi \in \mathsf{PSAMP}$. Applying Lemma 2.10, at Eq. (3), we now obtain $\mathsf{pK}^t(x) \leq s + O(\log n)$. The rest of the proof proceeds analogously. We conclude that $\mathsf{NP} = \mathsf{coNP}$ for unary languages, which is equivalent to $\mathsf{NE} = \mathsf{coNE}$. □

4 Proof of Theorem 1.2

We start with a brief outline of the proof of Theorem 1.2.

1. By Lemma 3.1 (from the proof of Theorem 1.1), we get a black-box non-adaptive fixed-auxiliary input reduction from SAT to inverting an auxiliary-input function $f = \{f_\varphi\}_{\varphi \in \{0,1\}^*}$.

2. By Lemma 2.20, under our assumption of the non-existence of OWFs, we get, for any polynomial-time samplable distribution D, a PPT machine that inverts f_φ with high probability over $\varphi \sim D$. Combined with step (1), this yields that $(\mathsf{SAT}, D) \in \mathsf{AvgBPP}$.

3. From the worst-case to average-case reduction of [Hir18] (and subsequent works [Hir21] and [Gol+22]), for some distribution $D' \in \mathsf{PSAMP}$, there is a BPP-black-box non-adaptive randomized polynomial-time reduction from $\mathsf{Gap}_{\tau,O(\log n)}\mathsf{pK^t}$ to the average-case problem of solving SAT over D'. That is,

$$(\mathsf{SAT}, D') \in \mathsf{AvgBPP} \implies \mathsf{Gap}_{\tau,O(\log n)}\mathsf{pK^t} \in \mathsf{BPP}$$

   for a sufficiently large polynomial τ depending on the running time of the heuristic for SAT. (See Corollary 4.2.) Combined with step (2), we get that $\mathsf{Gap}_{\tau,O(\log n)}\mathsf{pK^t} \in \mathsf{BPP}$.

4. For a sufficiently large t, if $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{pK}^t$ is NP-hard, then $\mathsf{Gap}_{\tau,O(\log n)}\mathsf{pK^t}$ is also NP-hard. (See Lemma 4.3.) Combined with step (3), this yields $\mathsf{NP} \subseteq \mathsf{BPP}$.

We provide more details on steps (3) and (4) of the proof outline above. We start with step (3).

Lemma 4.1 ([Gol+22])**.** *There exist a language $L \in \mathsf{NP}$ and a PPT non-adaptive oracle machine M_1 satisfying the following. If B is a PPT errorless heuristic for L over the uniform distribution, then for some polynomial τ depending on the running time of B, M_1^B decides $\mathsf{Gap}_{\tau,O(\log n)}\mathsf{pK^t}$ with high probability in the worst case.*

The NP-hardness of SAT yields the following.

Corollary 4.2. *There exist a distribution $D' \in \mathsf{PSAMP}$ and a PPT non-adaptive oracle machine M_1 satisfying the following. If B is a PPT errorless heuristic for SAT over D', then for some polynomial τ depending on the running time of B, M_1^B decides $\mathsf{Gap}_{\tau,O(\log n)}\mathsf{pK^t}$ with high probability in the worst case.*

Proof sketch. We use a t-time heuristic for SAT over the distribution $D' := R(\mathcal{U})$, where R is a t_L-time reduction from L to SAT, to work as a $(t \circ t_L)$-time heuristic for $(L, \mathcal{U})$. Then we apply Lemma 4.1. □

Next we provide the details on step (4).

Lemma 4.3. *Consider any constant $\delta > 0$, and let p_{sc} be the polynomial of Lemma 2.10. Let polynomials $t, t_R : \mathbb{N} \to \mathbb{N}$ be such that $t(n) \geq p_{sc}(t_R(n))$ for all $n \in \mathbb{N}$, and let R be a $n^{2/\delta}$-length-increasing black-box non-adaptive t_R-time randomized reduction from SAT to $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{pK}^t$. Then, there is a black-box non-adaptive reduction R' that, for any polynomial τ, reduces SAT to $\mathsf{Gap}_{\tau, n^{\delta/2}}\mathsf{pK^t}$.*

Proof. We will show that querying $(x, 1^s, 1^t)$ to a $\mathsf{Gap}_{\tau, n^{\delta/2}}\mathsf{pK^t}$ oracle would answer any query $(x, 1^s)$ to R correctly with respect to $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{pK}^t$. First consider the case that a query $(x, 1^s)$ is a Yes-instance of $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{pK}^t$. That is,

$$\mathsf{pK}^t(x) \leq s.$$

Clearly, $(x, 1^s, 1^t)$ is also Yes-instance of $\mathsf{Gap}_{\tau, n^{\delta/2}}\mathsf{pK^t}$, and the query is answered correctly.

Now suppose $(x, 1^s)$ is a No-instance of $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{pK}^t$. That is,

$$\mathsf{pK}^t(x) > s + |x|^\delta.$$

Adding $\mathsf{pK}^{\tau(t)}(x) - \mathsf{pK}^t(x)$ to both sides of the inequality, we obtain

$$\begin{aligned} \mathsf{pK}^{\tau(t)}(x) &> s + |x|^\delta + (\mathsf{pK}^{\tau(t)}(x) - \mathsf{pK}^t(x)) \\ &\geq s + |x|^\delta - (\mathsf{pK}^t(x) - \mathsf{K}(x)). \end{aligned}$$

By our definition of t and Corollary 2.11, with probability at least $1 - 2^{-n}$ over the internal randomness of R, we have that

$$\mathsf{pK}^t(x) - \mathsf{K}(x) \leq 4n,$$

so by the above,

$$\begin{aligned} \mathsf{pK}^{\tau(t)}(x) &\geq s + |x|^\delta - 4n \\ &\geq s + |x|^{\delta/2}. \end{aligned}$$

Thus, $(x, 1^s, 1^t)$ is a No-instance of $\mathsf{Gap}_{\tau, n^{\delta/2}}\mathsf{pK^t}$, and the query is answered correctly. □

We are now ready to prove Theorem 1.2, restated below for convenience.

Theorem 4.4 (Restatement of Theorem 1.2, Item 1-(II))**.** *For any constants $\delta, \gamma > 0$, there is a polynomial p satisfying the following. If $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{pK}^t$ is hard for SAT under a γ-honest non-adaptive randomized reduction running in time t_R, where $t_R, t : \mathbb{N} \to \mathbb{N}$ satisfy $t(n) \geq p(t_R(n))$ for all $n \in \mathbb{N}$, then there is a BPP-black-box non-adaptive reduction from SAT to inverting a one-way function.*

Proof. Let p_{sc} be the polynomial of Lemma 2.10. We choose p in the statement of this theorem such that

$$p(n) = 2 \cdot p_{sc}(n^{4/(\delta\cdot\gamma)}).$$

Note that $p(n) = p_1(n) = 2 \cdot p_{sc}(n^{4/(\delta\cdot\gamma)})$, where p_1 is the polynomial of Lemma 3.1.

Let $d \in \mathbb{N}$ and $f = \{f_\varphi\}_{\varphi\in\{0,1\}^*}$ be as in Lemma 3.1, and let M_0 be the black-box fixed-auxiliary-input reduction guaranteed by that lemma. Let $D' \in \mathsf{PSAMP}$ be as in Corollary 4.2.

Let I be any PPT n^{-4d}-inverter for f, and let M and g be as in Lemma 2.20 applied to f and D'. By that lemma, M^I n^{-d}-inverts f_φ with probability at least $1 - n^{-d}$ over the choice of $\varphi \sim D'$. Moreover, if M^I does not invert f on some φ, then on input φ, it outputs $\perp$ with probability at least $1 - 1/2^n$.

Consider the algorithm $H := M_0^{M^I}$ that simulates M_0 and answers its queries with M^I, outputting $\perp$ if any of the queries are answered with $\perp$. By Lemma 3.1, we have that H is an errorless heuristic for SAT over D'. Let M_1 be the PPT oracle machine of Corollary 4.2. By that corollary, M_1^H decides $\mathsf{Gap}_{\tau,O(\log n)}\mathsf{pK^t}$ in the worst case, for some polynomial τ depending on the running time of H.

By Lemma 2.15, there is a $n^{2/\delta}$-length-increasing reduction from SAT to $\mathsf{Approx}_{n^\delta}\text{-}\mathsf{pK}^t$ running in time at most $t_R(n^{4/(\delta\cdot\gamma)})$. By Lemma 4.3, there is a PPT non-adaptive oracle machine M_2 reducing SAT to $\mathsf{Gap}_{\tau,O(\log n)}\mathsf{pK^t}$. Thus, the machine that simulates M_2 and answers its oracle queries with M_1^H decides SAT.

To summarize, we have shown that the composition of machines M_2, M_1, M_0, and M is a BPP-black-box reduction from SAT to inverting f. This concludes the proof of the theorem. □

Item 2 of Theorem 1.2 now follows from Item 1 and Lemma 3.2.

5 Proof of Theorem 1.3

We will make use of the following statements from Liu and Pass [LP23]. We observe that these statements are proved in that work by exhibiting black-box average-case reductions from $\mathsf{MpK}^t\mathsf{P}$ and $\mathsf{MK}^t\mathsf{P}$ to the inversion of one-way functions. Note that the heuristics below may err in the sense of Definition 2.17.

Lemma 5.1 ([LP23])**.** *There exists a polynomial q satisfying the following. Consider polynomials $t, t_D : \mathbb{N} \to \mathbb{N}$ such that $t(n) \geq q(t_D(n))$ for all sufficiently large $n \in \mathbb{N}$, and any distribution $D = \{D_n\}_{n\in\mathbb{N}} \in \mathsf{SAMP}[t_D(n)]$.*

1. *There is a non-adaptive PPT oracle machine M and a function f such that, for any oracle I that inverts f, M^I is a heuristic for $\mathsf{MpK}^t\mathsf{P}$ over D.*
2. *Assume that $\mathsf{E} \not\subseteq \mathsf{io\text{-}NSIZE}[2^{o(n)}]$. There is a non-adaptive PPT oracle machine M and a function f such that, for any oracle I that inverts f, M^I is a heuristic for $\mathsf{MK}^t\mathsf{P}$ over D.*

Theorem 5.2 (Restatement of Theorem 1.3, Item 1)**.** *There is a polynomial p satisfying the following. Let $t, t_R : \mathbb{N} \to \mathbb{N}$ be such that $t(n) \geq p(t_R(n))$ for all $n \in \mathbb{N}$. If $\mathsf{MpK}^t\mathsf{P}$ is hard for SAT under an honest non-adaptive randomized reduction running in time t_R, then there is a black-box average-case non-adaptive randomized reduction from every problem in DistNP to the problem of inverting a one-way function. It follows that*

$$\mathsf{DistNP} \not\subseteq \mathsf{HeurBPP} \implies \exists\mathsf{OWF}.$$

Proof. Assume that $\mathsf{MpK}^t\mathsf{P}$ is hard for SAT in the sense described in the statement of this theorem. Consider any language $L \in \mathsf{NP}$ and distribution $D = \{D_n\}_{n\in\mathbb{N}} \in \mathsf{PSAMP}$.

Suppose D is samplable in time at most n^a and L is reducible to SAT in time at most n^a via an honest deterministic reduction, for some constant $a \in \mathbb{N}$. Note that there is a randomized black-box non-adaptive reduction from L to $\mathsf{MpK}^t\mathsf{P}$ running in time at most $t_R(n^a)$ and only making queries of length at least n^γ for some constant $\gamma > 0$. That is, there is a $\mathsf{MpK}^t\mathsf{P}$-oracle algorithm R that runs in time at most $t_R(n^a)$, and for every $x \in \{0,1\}^n$,

$$\Pr_{r\sim\mathcal{U}_{t_R(n^a)}}[L(x) = R^{\mathsf{MpK}^t\mathsf{P}}(x,r)] \geq \frac{2}{3}.$$

Now, since R is honest, it only queries strings of length at least $\ell(n) := n^\gamma$. Let $Q_{\ell(n)}$ denote the query distribution of R on input $x \sim D_n$ and randomness $r \sim \mathcal{U}_{t_R(n^a)}$. Since $Q_{\ell(n)}$ may be implemented in time at most $n^a + n^a + t_R(n^a) =: t_D(\ell(n))$, we have $Q \in \mathsf{SAMP}[t_D(n)]$. Choose p in the statement of this theorem such that $p(t_R(n)) \geq q(t_D(n))$, where q is as in Lemma 5.1.

Let M be the reduction and f the function of Lemma 5.1, Item 1. Consider any $c \in \mathbb{N}$, and let I be any oracle that inverts f (with sufficiently small inverse polynomial failure probability). By Lemma 5.1, M^I is a randomized poly-time heuristic for $\mathsf{MpK}^t\mathsf{P}$ over the distribution Q with failure probability at most $1/(t_R(n^a) \cdot n^{c+1})$.

Let $M' := R^{M^I}$ be the machine that, on input (x,r), simulates $R(x,r)$ and answers all oracle queries with M^I. By a union bound and an averaging argument, with probability at least $1 - 1/n^c$ over $x \sim D_n$, it holds that

$$\Pr_r[L(x) = M'(x,r)] \geq \frac{2}{3} - \frac{1}{n}.$$

This completes the proof of the theorem. □

Theorem 5.3 (Restatement of Theorem 1.3, Item 2)**.** *Assume* $\mathsf{E} \not\subseteq \mathsf{io\text{-}NSIZE}[2^{o(n)}]$. *There is a polynomial p satisfying the following. Let $t, t_R : \mathbb{N} \to \mathbb{N}$ be such that $t(n) \geq p(t_R(n))$ for all $n \in \mathbb{N}$. If* $\mathsf{MK}^t\mathsf{P}$ *is hard for* SAT *under an honest non-adaptive randomized reduction running in time t_R, then there is a black-box average-case non-adaptive deterministic reduction from every problem in* DistNP *to the problem of inverting a one-way function. It follows that*

$$\mathsf{DistNP} \not\subseteq \mathsf{HeurP} \implies \exists\mathsf{OWF}.$$

Proof sketch. The proof is analogous to that of Theorem 5.2, applying Item 2 of Lemma 5.1 instead of Item 1. Assuming that $\mathsf{E} \not\subseteq \mathsf{io\text{-}NSIZE}[2^{o(n)}]$, the machine M' may be derandomized. □

6 Proof of Theorem 1.4

Below, we give a first proof of Theorem 1.4, which proceeds somewhat like the proof of Theorems 1.1 and 1.2: namely, we exhibit a non-adaptive reduction from SAT to inverting an auxiliary-input one-way function.

Theorem 6.1 (Restatement of Theorem 1.4)**.** *For any polynomial t_R, if* MKP *is hard for* SAT *under a randomized many-one reduction running in time $t_R(n)$ and with failure probability γ, where $\gamma \leq 1/(t_R(n))^7$, then there is a non-adaptive fixed-auxiliary-input reduction from* SAT *to $1/(10\cdot n)$-inverting an auxiliary-input function f (dependent on R). Hence, we get that*

1. $\mathsf{NP} \subseteq \mathsf{coAM}$*; and*

2. if, in addition, no one-way functions exist, then $\mathsf{NP} \subseteq \mathsf{BPP}$*.*

Proof. Let R be a randomized poly-time reduction from SAT to MKP, running in time at most $t_R(n)$ on inputs of length n, and with failure probability at most γ. For $\varphi \in \{0,1\}^n$, let Q_φ be the $t_R(n)$-time samplable distribution of K-queries $(x, 1^s)$ induced by $R(\varphi)$. Note that for any output $(x, 1^s)$ of R, we may assume that $s < t_R(n)$. If not, then it must be that $|x| \leq O(1)$ in order to print the output within the running time of R, in which case the MKP instance can easily be answered as 'Yes'. Below, we assume $s < t_R(n)$ in every possible output of R.

Let $f = \{f_\varphi\}_{\varphi \in \{0,1\}^*}$ be the auxiliary-input function, A the non-adaptive oracle algorithm, and $c \in \mathbb{N}$ the constant of Lemma 2.22 applied to $\{Q_\varphi\}_{\varphi \in \{0,1\}^*}$ with $d = 1$.

For any given $s \in [m-1]$, where $m := t_R(n)$, define

$$\beta_s := \frac{1}{8c \cdot m \cdot 2^{s+1}}.$$

Let I be any algorithm inverting f, and consider the following algorithm B for SAT:

On input $\varphi \in \{0,1\}^n$, simulate $R(\varphi)$ to get a query $(x, 1^s)$. Accept iff $A^I(\varphi, (x, 1^s)) \geq \beta_s$.

Toward a contradiction, suppose B fails to decide SAT at infinitely many input lengths; let $n \in \mathbb{N}$ be some such input length, and let $\varphi \in \{0,1\}^n$ be the lexicographically first string of length n such that

$$\Pr_B[B(\varphi) \neq \mathsf{SAT}(\varphi)] \geq 1/3.$$

First consider the case that $\varphi \in \mathsf{SAT}$. By the correctness of R, with probability at least $1 - \gamma$ over the choice of $(x, 1^s) \sim R(\varphi)$, it will hold that $\mathsf{K}(x) \leq s$.

Arguing as in Lemma 3.1,

$$\begin{aligned}
\Pr_{(x,1^s)\sim Q_\varphi}[\mathsf{K}(x) \leq s \ \wedge\ A^I(\varphi, (x, 1^s)) < \beta_s] &\leq \sum_{s=0}^{m-1} \sum_{x \in \{0,1\}^{\leq m}:\ \mathsf{K}(x) \leq s} c \cdot \beta_s + \frac{1}{n} && \text{(Lemma 2.22)} \\
&\leq m \cdot 2^{s+1} \cdot c \cdot \beta + \frac{1}{n} \\
&< \frac{1}{8} + \frac{1}{n}.
\end{aligned}$$

Thus, for $\varphi \in \mathsf{SAT}$, the overall probability that $B(\varphi) \neq \mathsf{SAT}(\varphi)$ is at most

$$\frac{1}{8} + \frac{1}{n} + \gamma < \frac{1}{3}.$$

Now suppose $\varphi \notin \mathsf{SAT}$. For each $0 \leq s \leq m-1$, let $x_1^s, x_2^s, \ldots, x_{2^m}^s$ be the elements in the support of the conditional distribution

$$Q_{\varphi,s}(x) := \frac{Q_\varphi(x, 1^s)}{\sum_y Q_\varphi(y, 1^s)},$$

listed in the non-increasing order of their probabilities in $Q_{\varphi,s}$. For a set $S \subseteq [2^m]$, let $X^s(S) = \{x_j^s \mid j \in S\}$.

Observe that for each $0 \le s \le m-1$ and $i \in [2^m]$,

$$\mathsf{K}(x_i^s) \le \log i + 2\log(nm).$$

This bound follows from the procedure that, on input (i, n, s), finds and defines as φ the lexicographically first string of length n such that $\Pr[B(\varphi) \neq \mathsf{SAT}(\varphi)] \ge 1/3$ and then determines the i^{th} heaviest element of the distribution $Q_{\varphi,s}$ and outputs it.

It follows that, for each $0 \le s \le m-1$, we have

$$\Pr_{x \sim Q_{\varphi,s}}[\mathsf{K}(x) \le s] \ge \Pr_{x \sim Q_{\varphi,s}}[x \in X^s([2^s/(nm)^2])]. \tag{4}$$

By partitioning $X^s([2^s])$ into $(nm)^2$ consecutive intervals of size $2^s/(nm)^2$, we get

$$\Pr_{x \sim Q_{\varphi,s}}[x \in X^s([2^s])] \le (nm)^2 \cdot \Pr_{x \sim Q_{\varphi,s}}[x \in X^s([2^s/(nm)^2])]. \tag{5}$$

Consider any $0 \le s^* < m$ such that the marginal probability

$$Q_\varphi(s^*) := \sum_y Q_\varphi(y, 1^{s^*}) \ge \frac{1}{4m}.$$

Observe that, for every such s^*, some elements of $X^{s^*}([2^{s^*}])$ must have probability *less than* $\beta_{s^*}/2$ with respect to Q_{φ,s^*}, that is,

$$Q_{\varphi,s^*}(x) \ge \beta_{s^*}/2 \implies x \in X^{s^*}([2^{s^*}]). \tag{6}$$

Indeed, suppose that for every $x \in X^{s^*}([2^{s^*}])$ we have $Q_{\varphi,s^*}(x) \ge \beta_{s^*}/2$. Then we get that

$$\begin{aligned}
\gamma &\ge \Pr_{(x,1^s)\sim Q_\varphi}[\mathsf{K}(x) \le s] && \text{(correctness of } R(\varphi)\text{)}\\
&\ge \frac{1}{4m} \cdot \Pr_{x \sim Q_{\varphi,s^*}}[\mathsf{K}(x) \le s^*] && (Q_\varphi(x,1^s) = Q_\varphi(s) \cdot Q_{\varphi,s}(x))\\
&\ge \frac{1}{4m} \cdot \Pr_{x \sim Q_{\varphi,s^*}}[x \in X^{s^*}([2^{s^*}/(nm)^2])] && \text{(Eq. (4))}\\
&\ge \frac{1}{4m(nm)^2} \cdot \Pr_{x \sim Q_{\varphi,s^*}}[x \in X^{s^*}([2^{s^*}])] && \text{(Eq. (5))}\\
&\ge 2^{s^*} \cdot \frac{\beta_{s^*}}{2} \cdot \frac{1}{4m(nm)^2}\\
&= \frac{1}{128 \cdot c \cdot n^2 \cdot m^4}\\
&> \frac{1}{m^7},
\end{aligned}$$

contradicting the choice of $\gamma \le 1/m^7$.

It follows that

$$\begin{aligned}
&\Pr_{(x,1^s)\sim Q_\varphi}[K(x) > s \ \wedge \ A^I(\varphi,(x,1^s)) \geq \beta_s) \\
&\qquad \leq \Pr_{(x,1^s)\sim Q_\varphi}[Q_\varphi(x,1^s) \geq \beta_s/2)] + \frac{1}{n} && \text{(Lemma 2.22)} \\
&\qquad \leq \sum_{s\colon Q_\varphi(s)\geq 1/(4m)} Q_\varphi(s)\cdot \Pr_{x\sim Q_{\varphi,s}}[x \in X^s([2^s])] && \text{(Eq. (6))} \\
&\qquad\qquad + \sum_{s\colon Q_\varphi(s)< 1/(4m)} Q_\varphi(s) + \frac{1}{n} \\
&\qquad \leq \sum_{s\colon Q_\varphi(s)\geq 1/(4m)} Q_\varphi(s)\cdot \Pr_{x\sim Q_{\varphi,s}}[x \in X^s([2^s])] + \frac{1}{4} + \frac{1}{n} \\
&\qquad \leq \sum_s Q_\varphi(s)\cdot \Pr_{x\sim Q_{\varphi,s}}[x \in X^s([2^s])] + \frac{1}{4} + \frac{1}{n} \\
&\qquad \leq (nm)^2 \cdot \sum_s Q_\varphi(s)\cdot \Pr_{x\sim Q_{\varphi,s}}[\mathsf{K}(x)\leq s] + \frac{1}{4} + \frac{1}{n} && \text{(Eqs. (4) and (5))} \\
&\qquad \leq (nm)^2 \cdot \Pr_{(x,1^s)\sim Q_\varphi}[\mathsf{K}(x)\leq s] + \frac{1}{4} + \frac{1}{n} && (Q_\varphi(x,1^s) = Q_\varphi(s)\cdot Q_{\varphi,s}(x)) \\
&\qquad \leq (nm)^2 \cdot \gamma + \frac{1}{4} + \frac{1}{n}. && \text{(correctness of } R(\varphi))
\end{aligned}$$

Thus, for $\varphi \notin \mathsf{SAT}$, the overall probability that $B(\varphi) \neq \mathsf{SAT}(\varphi)$ is at most

$$\gamma + (nm)^2\cdot\gamma + \frac{1}{4} + \frac{1}{n} < \frac{1}{3}.$$

We conclude that $B(\varphi) \neq \mathsf{SAT}(\varphi)$ with probability less than 1/3, contradicting the definition of φ.

Applying Lemma 2.24, we conclude that $\mathsf{NP} \subseteq \mathsf{coAM}$, obtaining Item 1 of the theorem. Applying Theorem 2.27 yields Item 2 of the theorem. □

7 NP-hardness of K complexity and NISZK

In this section, we build on techniques from Allender et al. [AHT23] to give an alternative proof of (a strengthening of) Theorem 1.4. In particular, we prove the following.

Theorem 7.1. *For any polynomial t_R and decidable language L, if* MKP *is hard for L under a randomized many-one reduction running in time $t_R(n)$ and with failure probability at most $1/t_R(n)^{16}$, then $L \subseteq$* coNISZK*.*

On one hand, this yields an improvement on the following statement implicit in Theorem 15 of [AHT23]. Recall the definition of fixed-threshold Approx_g-$\mathsf{K}[s]$ from Section 2: namely, for $g, s : \mathbb{N} \to \mathbb{N}$,

- $\Pi_Y = \{x \mid \mathsf{K}(x) \leq s(|x|)\}$;

- $\Pi_N = \{x \mid \mathsf{K}(x) > s(|x|) + g(|x|)\}$.

Theorem 7.2 ([AHT23])**.** *For any decidable language L, if $\mathsf{Approx}_{\omega(\log n)}\text{-}\mathsf{K}[n/2]$ is hard for L under an honest randomized many-one reduction with failure probability at most $1/n^{\omega(1)}$, then $L \subseteq \mathsf{coNISZK}$.*

Note that Theorem 7.1 improves on Theorem 7.2 above in 3 respects: we do not require the reduction to be honest, we do not require an $\omega(\log n)$ approximation term, and we do not require the threshold parameter to be fixed.[8] Also note that, setting $L = \mathsf{SAT}$, Theorem 7.1 captures Item 1 of Theorem 6.1, since, by known results, $\mathsf{NISZK} \subseteq \mathsf{SZK} \subseteq \mathsf{AM} \cap \mathsf{coAM}$ [GSV99; For89; AH91]. One can also obtain Item 2 of Theorem 6.1 from Theorem 7.1 by combining ideas from [Ost91; OW93] and [Nan21]. (See [HN24] for a further explanation.)

On the other hand, [AHT23] also provides a converse to Theorem 7.2, namely:

Theorem 7.3. *For a language L, suppose $L \subseteq \mathsf{coNISZK}$. Then, for any polynomial p, there is an honest randomized many-one reduction from L to $\mathsf{Approx}_{n^{o(1)}}\text{-}\mathsf{K}[n/2]$ with (one-sided) failure probability $2^{-p(n)}$.*

Thus, we obtain the following result, indicating a surprising robustness of NP-hardness of K complexity with respect to many-one reductions.

Theorem 7.4 (Restatement of Theorem 1.5)**.** *Consider any decidable language L and polynomials t_R and p. The following are equivalent.*

1. *$L \subseteq \mathsf{coNISZK}$;*

2. *MKP is hard for L under a randomized many-one reduction running in time $t_R(n)$ and with two-sided failure probability at most $1/t_R(n)^{16}$;*

3. *$\mathsf{Approx}_{n^{o(1)}}\text{-}\mathsf{K}[n/2]$ is hard for L under an honest randomized many-one reduction with one-sided failure probability at most $2^{-p(n)}$.*

We now prove Theorem 7.1. To understand the proofs in this section, the reader doesn't need to know the definitions of the statistical zero-knowledge class SZK and its non-interactive version NISZK. We just need the following problem that is known to be complete for NISZK.

Definition 7.5 (Entropy Approximation problem)**.** *The* entropy approximation problem*, denoted* EA*, is defined as follows. Let D be a probability distribution sampled by a circuit C, and let $H(D)$ denote the entropy of D. Then EA is the following promise problem:*

- *$\Pi_Y = \{(C, k) \mid H(D) > k + 1\}$, and*

- *$\Pi_N = \{(C, k) \mid H(D) < k - 1\}$.*

The following lemma was originally stated for the case of a deterministic reduction, but the proof is easily modified to give the same conclusion from a randomized reduction with exponentially small failure probability.

[8] Allender et al. do argue for robustness with respect to the fixed threshold parameter in the case of an n^{ε} approximation gap, for $0 < \varepsilon < 1$ [AHT23, Propositions 3 and 5]. However, they do not consider the case that the threshold parameter can depend on the randomness of the reduction, which we allow in a reduction to MKP.

Lemma 7.6 ([GSV99])**.** *Suppose a language L reduces to* EA *under a randomized many-one reduction with failure probability at most $2^{-\Omega(n)}$. Then, $L \in \mathsf{NISZK}$.*

We now state and prove the following strengthening of Theorem 1.4. Theorem 7.1 follows by combining Lemmas 7.6 and 7.7.

Lemma 7.7. *For any decidable language L and polynomial t_R, if* MKP *is hard for L under a randomized many-one reduction running in time $t_R(n)$ and with failure probability at most γ, where $\gamma < 1/t_R(n)^{16}$, then $\overline{\mathsf{EA}}$ is hard for L under a randomized polynomial-time many-one reduction with failure probability $2^{-\Omega(n)}$.*

Proof. Let R be a randomized poly-time reduction from L to MKP, running in time at most $t_R(n)$ on inputs of length n, and with failure probability at most γ. For $\varphi \in \{0,1\}^n$, let Q_φ be the $t_R(n)$-time samplable distribution of K-queries $(x, 1^s)$ induced by $R(\varphi)$. As in the proof of Theorem 6.1, we may assume without loss of generality that $s < m$ in every possible output of R, where $m := t_R(n)$.

For $0 \leq s < m$ and $x, y \in \{0,1\}^*$, denote the marginal probability

$$Q_\varphi(s) := \sum_y Q_\varphi(y, 1^s),$$

and denote the conditional probability

$$Q_{\varphi,s}(x) := \frac{Q_\varphi(x, 1^s)}{Q_\varphi(s)}.$$

Given any L-instance φ, by random sampling from Q_φ, one may find a parameter $0 \leq s^* < m$ such that

$$Q_\varphi(s^*) \geq \frac{1}{4m}. \tag{7}$$

By a Chernoff bound, this may be done in polynomial time and with failure probability at most 2^{-n}. Then, if s^* has the property in (7), one may compute from φ and s^* a poly-size circuit C that outputs the string 0^n with probability at most 2^{-n} and otherwise outputs strings distributed according to Q_{φ,s^*}. Namely, C is defined to sample at most m^2 pairs $(x, 1^s) \sim Q_\varphi$; if it finds one with $s = s^*$, it outputs x, and if not, it outputs 0^n. Let Q'_{φ,s^*} denote the distribution sampled by C. Let R' be the reduction from L to $\overline{\mathsf{EA}}$ that computes s^* and C as described above and then outputs $(C, s^* + 4\log m)$.

We will now show the correctness of R'. For a contradiction, suppose R' fails at some input length $n \in \mathbb{N}$, and let $\varphi \in \{0,1\}^n$ be the lexicographically first string of its length such that $\overline{\mathsf{EA}}(R'(\varphi)) \neq L(\varphi)$ with probability greater than 2^{-n}. Assume the parameter s^* computed by R' has the property in (7). Note that since Q'_{φ,s^*} is computable given parameters n and s^*, by the coding theorem for Kolmogorov complexity, it holds that[9]

$$\left| \mathop{\mathbb{E}}_{x \sim Q'_{\varphi,s^*}} [\mathsf{K}(x)] - H(Q'_{\varphi,s^*}) \right| \leq 3 \log m. \tag{8}$$

[9] See [LV19, Theorem 8.1.1]. We have chosen the statement of Eq. (8) to ensure that it holds both for prefix-free and plain Kolmogorov complexity, regardless of the choice of universal TM. See also [LV19, Example 3.1.5].

First consider the case that $\varphi \in L$. By the correctness of R,

$$\begin{aligned}\gamma &\geq \Pr_{(x,1^s)\sim Q_\varphi}[\mathsf{K}(x) > s] \\ &\geq \frac{1}{4m} \cdot \Pr_{x\sim Q_{\varphi,s^*}}[\mathsf{K}(x) > s^*] && (Q_\varphi(x,1^s) = Q_\varphi(s)\cdot Q_{\varphi,s}(x))\end{aligned}$$

Rearranging the above, by definition of γ,

$$\Pr_{x\sim Q_{\varphi,s^*}}[\mathsf{K}(x) > s^*] < \frac{1}{m^{14}}, \tag{9}$$

and so

$$\begin{aligned}H(Q'_{\varphi,s^*}) &\leq \mathop{\mathbb{E}}_{x\sim Q'_{\varphi,s^*}}[\mathsf{K}(x)] + 3\log m && \text{(Eq. (8))} \\ &\leq \left(1-\frac{1}{2^n}\right)\cdot \mathop{\mathbb{E}}_{x\sim Q_{\varphi,s^*}}[\mathsf{K}(x)] + \frac{1}{2^n}\cdot O(\log n) + 3\log m && \text{(definition of } Q'_{\varphi,s^*}) \\ &\leq \mathop{\mathbb{E}}_{x\sim Q_{\varphi,s^*}}[\mathsf{K}(x)] + 3\log m + o(1) \\ &< s^* + 3\log m + 1. && \text{(Eq. (9))}\end{aligned}$$

It follows that $(C, s^*+4) \notin \mathsf{EA}$ in this case, as desired.

Now suppose $\varphi \notin L$. By the correctness of R and the definition of s^*, we may show as above that

$$\Pr_{x\sim Q_{\varphi,s^*}}[\mathsf{K}(x) \leq s^*] < \frac{1}{m^{14}}. \tag{10}$$

Let $x_1, x_2, \ldots, x_{2^m}$ be the elements in the support of the conditional distribution $Q_{\varphi,s^*}(x)$ listed in the non-increasing order of their probabilities in Q_{φ,s^*}. For a set $S \subseteq [2^m]$, let $X(S) = \{x_j \mid j \in S\}$.

Observe that, by the decidability of L, for each $i \in [2^m]$,

$$\mathsf{K}(x_i) \leq \log i + 4\log m.$$

This bound follows from the procedure that, on input (i, n, s^*), finds and defines as φ the lexicographically first string of length n such that $\Pr[\ \overline{\mathsf{EA}}(R'(\varphi)) \neq L(\varphi)\ ] \geq 2^{-n}$ and then determines the i^{th} heaviest element of the distribution Q_{φ,s^*} and outputs it.

It follows that

$$\Pr_{x\sim Q_{\varphi,s^*}}[x \in X([2^{s^*}/m^4])] \leq \Pr_{x\sim Q_{\varphi,s^*}}[\mathsf{K}(x) \leq s^*]. \tag{11}$$

By partitioning $X([m^8\cdot 2^{s^*}])$ into m^{12} consecutive intervals of size $2^{s^*}/m^4$, we get

$$\begin{aligned}\Pr_{x\sim Q_{\varphi,s^*}}[x \in X([m^8\cdot 2^{s^*}])] &\leq m^{12}\cdot \Pr_{x\sim Q_{\varphi,s^*}}[x \in X([2^{s^*}/m^4])] \\ &\leq \frac{1}{m^2}. && \text{(Eqs. (10) and (11))}\end{aligned}$$

Since there are at most $m^8 \cdot 2^{s^*}$ strings with Kolmogorov complexity at most $s^* + 8 \log m - 1$, and since the cumulative probability of those strings under Q_{φ,s^*} is at most that of $X([m^8 \cdot 2^{s^*}])$,

$$\begin{aligned}\Pr_{x \sim Q_{\varphi,s^*}}[\mathsf{K}(x) \leq s^* + 8 \log m - 1] &\leq \Pr_{x \sim Q_{\varphi,s^*}}[x \in X([m^8 \cdot 2^{s^*}])] \\ &\leq \frac{1}{m^2}.\end{aligned}$$

Thus,

$$\begin{aligned}H(Q'_{\varphi,s^*}) &\geq \mathop{\mathbb{E}}_{x \sim Q'_{\varphi,s^*}}[\mathsf{K}(x)] - 3 \log m & \text{(Eq. (8))} \\ &\geq (1 - 2^{-n}) \cdot \mathop{\mathbb{E}}_{x \sim Q_{\varphi,s^*}}[\mathsf{K}(x)] - 3 \log m & \text{(definition of } Q'_{\varphi,s^*}\text{)} \\ &> s^* + 4 \log m + 1.\end{aligned}$$

We get that $(C, s^* + 4 \log m) \in \mathsf{EA}$ in this case, as desired. We conclude that on all sufficiently large L-instances φ, R' is correct with probability at least $1 - 2^{-n}$ over its internal randomness. □

8 NP-hardness of $(\mathsf{K}^t$ vs. $\mathsf{K})$ and $(\mathsf{K}^t$ vs. $\mathsf{K})^*$

8.1 Randomized Reductions

In this section, we examine promise problems of the form $(\mathsf{K}^t$ vs. $\mathsf{K}^{t'})$, for time bounds $t, t' \in \mathbb{N}$, in comparison with the 'partial function' versions $(\mathsf{K}^t$ vs. $\mathsf{K}^{t'})^*$ recently shown NP-complete by Hirahara [Hir22]. While NP-hardness of $(\mathsf{K}^t$ vs. $\mathsf{K})$ would imply $\mathsf{NP} \subseteq \mathsf{coAM}$ via our proof techniques above, the consequence does not seem to follow in the partial setting, as we discuss further below. We then show that NP-hardness via *deterministic Turing* reductions of either $(\mathsf{K}^t$ vs. $\mathsf{K}^{t'})$ or $(\mathsf{K}^t$ vs. $\mathsf{K}^{t'})^*$ (with appropriate settings of t and t') would imply $\mathsf{NP} = \mathsf{P}$. It follows that these problems are NP-intermediate with respect to deterministic Turing reductions, provided the existence of one-way functions.

We start with formal definitions of the partial version of K^t complexity and the promise problems mentioned above.

Definition 8.1 (Partial (Time-bounded) Kolmogorov Complexity)**.** *For a time bound $t \in \mathbb{N}$, a string $x \in \{0, 1, *\}^*$, and a complexity measure $\mu \in \{\mathsf{pK}^t, \mathsf{K}^t, \mathsf{K}\}$, the* partial ($t$-time-bounded, probabilistic) Kolmogorov complexity *of x, denoted $(\mu)^*(x)$, is equal to*

$$\min \left\{ \mu(x') \mid x' \textit{ consistent with } x \right\},$$

where a string $x' \in \{0,1\}^*$ is said to be consistent with *$x \in \{0, 1, *\}^*$ if $|x'| = |x|$ and, for every index $i \in [|x|]$ such that $x[i] \neq *$, it holds that $x[i] = x'[i]$.*

Definition 8.2 ($(\mathsf{K}^t$ vs. $\mathsf{K}^{t'})$)**.** *Let $t, t' : \mathbb{N} \to \mathbb{N}$. For $\mu_1 \in \{\mathsf{K}^t, \mathsf{pK}^t\}$ and $\mu_2 \in \{\mathsf{K}^{t'}, \mathsf{pK}^{t'}, \mathsf{K}\}$, $(\mu_1$ vs. $\mu_2)$ is the following promise problem.*

- $\Pi_Y = \{(x, 1^s) \mid \mu_1(x) \leq s\}$
- $\Pi_N = \{(x, 1^s) \mid \mu_2(x) > s\}$

$(\mu_1$ vs. $\mu_2)^*$ is defined analogously, with the partial complexity measures $(\mu_1)^*$ and $(\mu_2)^*$ in place of the standard ('complete function') ones.

By a proof analogous to that of Theorem 1.4, we get the following statement.

Lemma 8.3. *Let $t : \mathbb{N} \to \mathbb{N}$ be arbitrary and $t_R : \mathbb{N} \to \mathbb{N}$ a polynomial. If $(\mathsf{K}^t$ vs. $\mathsf{K})$ is NP-hard under a randomized many-one reduction running in time $t_R(n)$ and with failure probability at most $1/(t_R(n))^7$, then $\mathsf{NP} \subseteq \mathsf{coAM}$.*

One may contrast Lemma 8.3 with Hirahara's recent proof that $(\mathsf{K}^t$ vs. $\mathsf{K})^*$ is in fact NP-hard under a randomized many-one reduction with the same properties. This suggests that the techniques of [Hir22] will not extend to the setting of standard $(\mathsf{K}^t$ vs. $\mathsf{K})$ without leveraging some more powerful notion of reducibility. Viewed another way, to obtain NP-hardness of $\mathsf{MK}^t\mathsf{P}$ complexity under randomized many-one reductions, one would need techniques that apply more narrowly to smaller-gap versions of the problem.

Note that the statement gives NP-hardness of $\mathsf{MK}^t\mathsf{P}^*$ under a randomized reduction even when $t \in \mathbb{N}$ is arbitrarily larger than the running time of the reduction. In the case of a randomized reduction, it is not unreasonable to make the assumption that $t \gg t_R$, as is done in [SS22] and in this work. This is because randomized reductions may easily sample strings of maximum Kolmogorov complexity, so it is easy to generate No-instances of $\mathsf{MK}^t\mathsf{P}$ (or $\mathsf{MK}^t\mathsf{P}^*$) within time t_R. Note that this would be impossible for a deterministic reduction.

Lemma 8.4 (Implicit in [Hir22])**.** *There exists a polynomial $t_R : \mathbb{N} \to \mathbb{N}$ such that for any constant $c \in \mathbb{N}$ and any sufficiently large polynomial $t : \mathbb{N} \to \mathbb{N}$, $(\mathsf{K}^t$ vs. $\mathsf{K})^*$ is NP-hard under a randomized many-one reduction running in time $t_R(n)$ and with failure probability at most $1/t_R(n)^c$.*

Proof sketch. One needs to verify that the failure probability of the reduction is at most $1/n^c$ for an arbitrary large constant $c \in \mathbb{N}$. Recall that in the proof of [Hir22] Lemma 8.3, the reduction samples random strings $f_i \sim \{0,1\}^{\lambda \cdot w(i)}$ for $i \in [n]$, where $n \in \mathbb{N}$ is the number of variables in the input CMMSA instance, $w : [n] \to \mathbb{N}$ is a weight function, and λ is some fixed polynomial in n. The reduction succeeds provided, for every $T \subseteq [n]$, for some constant $c \in \mathbb{N}$,

$$\mathsf{K}(f_T) \geq \lambda \cdot w(T) - c \cdot |T| \cdot \log n. \tag{12}$$

This is used in the 'soundness' part of the proof to argue that the set $B \subseteq [n]$ is not authorized. In particular, one must prove that $w(B) < \theta$ from the fact that $\mathsf{K}(f_B) \leq o(\lambda \cdot w(B)) + |M|$, where $|M|$ is an arbitrary program of size $\lambda\theta/2$. To see that Eq. (12) is sufficient for this purpose, observe that for any $c \in \mathbb{N}$,

$$\begin{aligned}\lambda \cdot w(B) - c \cdot |B| \cdot \log n &\leq \mathsf{K}(f_B) \\ &\leq o(\lambda \cdot w(B)) + |M|\end{aligned}$$

implies that

$$\begin{aligned}\lambda \cdot w(B) &\leq c \cdot |B| \cdot \log n + o(\lambda \cdot w(B)) + |M| \\ &\leq o(\lambda \cdot w(B)) + |M|,\end{aligned}$$

since $c \cdot |B| \cdot \log n \leq cn \log n = o(\lambda)$. Thus,

$$\begin{aligned}w(B) \cdot \lambda \cdot (1 - o(1)) &\leq |M| \\ &\leq \lambda \cdot \theta/2,\end{aligned}$$

which implies that $w(B) < \theta$, as desired.

Now we will show that, for any $c \in \mathbb{N}$, Eq. (12) holds with probability at least $1 - 1/n^{c-2}$. First observe that by a standard counting argument, with probability $1 - 1/n^{c-2}$,

$$\mathsf{K}(f_{[n]}) \geq \lambda \cdot w([n]) - (c-2) \cdot \log n.$$

Moreover,

$$\mathsf{K}(f_{[n]}) \leq \mathsf{K}(f_T) + \lambda \cdot w([n] \backslash T) + 2 \cdot |T| \cdot \log n,$$

since one may describe $f_{[n]}$ by describing f_T, hard-wiring $f_{[n]\backslash T}$, and describing the set $T \subseteq [n]$ itself. Thus,

$$\begin{aligned} \mathsf{K}(f_T) &\geq \mathsf{K}(f_{[n]}) - \lambda \cdot w([n] \backslash T) - 2 \cdot |T| \cdot \log n \\ &\geq \lambda \cdot w([n]) - (c-2) \cdot \log n - \lambda \cdot w([n] \backslash T) - 2 \cdot |T| \cdot \log n \\ &\geq \lambda \cdot w(T) - c \cdot |T| \cdot \log n, \end{aligned}$$

so the reduction does not fail in this case. □

One may wonder why the barrier of Lemma 8.3 does not apply to the partial K^t setting. The primary issue is that a correspondence between the compressibility of queries and their probability under the query distribution Q_φ appears to be missing. As a result, we cannot apply our central proof technique of reducing meta-complexity to a problem of probability estimation.

Roughly speaking, there is a difference between the Kolmogorov complexity $\mathsf{K}(z)$ of the *description* of a query $z := (x, 1^s)$ with $x \in \{0,1,*\}^*$ and the *partial complexity* $\mathsf{K}^*(x)$ of x. By the Coding Theorem for K, we still have an approximate correspondence between the logarithm of the inverse probability of (the description of the query) z output by the randomized reduction and the complexity $\mathsf{K}(z)$. However, $\mathsf{K}^*(x)$ can differ significantly from $\mathsf{K}(z)$. For example, consider a string $y = 0^n$, and let y' be a uniformly random string in $\{0,*\}^n$. Since y' is a uniformly random string over the binary alphabet $\{0,*\}$, it's almost certainly true that $\mathsf{K}(y') \geq n - O(\log n)$. On the other hand, $\mathsf{K}^*(y') \leq \mathsf{K}(y) \leq O(\log n)$.

More concretely, for example, consider a reduction from SAT to the problem of approximating $(\mathsf{K}^t)^*$ (with a fixed threshold parameter $s \in \mathbb{N}$). Here, the queries $x \in \{0,1,*\}^*$ may contain unspecified '$*$' positions. On one hand, we can use a standard coding theorem (adapted appropriately) to show that a query x having probability greater than $\beta \approx 1/(2^s \cdot \mathsf{poly}(n))$ under the query distribution Q_φ would imply that $(\mathsf{K}^t)^*(x) \lesssim s$.

However, the converse does not seem to hold. Previously we showed that, for strings queried in the reduction, it was unlikely for a string to be both of low complexity and low probability. This followed from a counting argument and a union bound: there are roughly at most 2^s strings $x \in \{0,1\}^*$ with $\mathsf{K}^t(x) \leq s$, so the cumulative probability of strings with both this property and $Q_\varphi(x) \leq \beta$ is at most $1/\mathsf{poly}(n)$. In the case of partial K^t, it is no longer true that there are 'few' strings of low complexity. In particular, any one short description $d \in \{0,1\}^s$ can witness $(\mathsf{K}^t)^*(x) \leq s$ for 2^n distinct strings $x \in \{0,1,*\}^n$ (unlike standard K^t, where one description only 'maps' to one string). Thus, partial K^t complexity is not readily connected to probability under efficiently samplable distributions, which was the key connection exploited in the previous sections.

8.2 Deterministic Reductions

As another point of comparison, in Lemmas 8.6 and 8.7, we show that if either of $(\mathsf{K}^t \text{ vs. } \mathsf{K}^{t'})$ or $(\mathsf{K}^t \text{ vs. } \mathsf{K}^{t'})^*$ is NP-hard with respect to deterministic adaptive Turing reductions (for a sufficiently large exponential function t'), then one obtains the stronger consequence that $\mathsf{NP} = \mathsf{P}$. This implies that if one-way functions exist, $(\mathsf{K}^t \text{ vs. } \mathsf{K}^{t'})$ and $(\mathsf{K}^t \text{ vs. } \mathsf{K}^{t'})^*$ are both NP-intermediate with respect to deterministic Turing reductions.[10]

Note that Lemmas 8.6 and 8.7 hold for Turing reductions with *arbitrary* polynomial running time (i.e., less than or greater than the time-bound t), and there is no honesty requirement. After this, we show similar results for honest reductions and superpolynomial t'.

We will use the 'dream-breaker' of Bogdanov et al. [BTW10].

Lemma 8.5 ([BTW10]). *Suppose $\mathsf{NP} \neq \mathsf{P}$. There is an algorithm B and a universal constant d with the following properties. Let A be any poly-time algorithm that attempts to solve search-SAT and only errs by incorrectly outputting $\perp$.[11] For infinitely many $n \in \mathbb{N}$, $B(A, 1^n)$ outputs a formula $\varphi \in \{0,1\}^n$ and a witness a such that $\varphi(a) = 1$ but $A(\varphi) = \perp$. Moreover, if A runs in time at most n^b on inputs of length n, then $B(A, 1^n)$ runs in time at most $(n^b)^d$.*

Lemma 8.6. *For every constant c, there is a constant c' with the following property. Let $t, t' : \mathbb{N} \to \mathbb{N}$ be such that for all $n \in \mathbb{N}$, $t(n) \leq n^c$ and $t'(n) \geq 2^{c'n}$. Then $(\mathsf{K}^t \text{ vs. } \mathsf{K}^{t'})$ is NP-hard under deterministic polynomial-time Turing reductions iff $\mathsf{NP} = \mathsf{P}$.*

Proof. Let M be a Turing reduction from search-SAT to $(\mathsf{K}^t \text{ vs. } \mathsf{K}^{t'})$ running in time at most n^b on inputs of length $n \in \mathbb{N}$. Define a machine M' that on input $\varphi \in \{0,1\}^n$ simulates $M(\varphi)$ and answers its queries as follows. If the query $(x, 1^s)$ is such that $s \leq 4b \log n$ and $s \leq 2|x|$, answer the query by brute force; otherwise simply accept the query. Note that M' runs in time at most n^{6bc}.

Let B be the refuter of Lemma 8.5, and let $n \in \mathbb{N}$ and $\varphi \in \{0,1\}^n$ be such that $B(M', 1^n) = (\varphi, a)$ with $M'(\varphi) = \perp$ but $\varphi(a) = 1$.

Clearly, if a query $(x, 1^s)$ is such that $s \leq 4b \log n$ or $2|x| < s$, M' answers it correctly. We now claim that for every query $(x, 1^s)$ of $M'(\varphi)$, it holds that $\mathsf{K}^{t'}(x) \leq 4b \log n$. In particular, one may compute x from advice (n, i), where x is the i^{th} query of $M'(\varphi)$, in time at most

$$\left(n^{6bc}\right)^d + n^{6bc} < 2^{c' \cdot |x|},$$

assuming $2|x| \geq s > 4b \log n$ and choosing $c' = 4cd$, where d is the constant from Lemma 8.5. For $t' : \mathbb{N} \to \mathbb{N}$ such that $t'(m) \geq 2^{c'm}$, this implies

$$\mathsf{K}^{t'}(x) \leq s.$$

Thus, $M'(\varphi)$ answers all of its queries correctly with respect to $(\mathsf{K}^t \text{ vs. } \mathsf{K}^{t'})$, and

$$M'(\varphi) = M^{(\mathsf{K}^t \text{ vs. } \mathsf{K}^{t'})}(\varphi) = \text{search-}\mathsf{SAT}(\varphi),$$

a contradiction. □

[10] Since either of these problems could be used to break a cryptographic PRG, the existence of OWFs means they must not be efficiently decidable.

[11] Note that any poly-time algorithm may be transformed into such an algorithm by verifying any candidate satisfying assignment to the input before returning it.

The following statement for $(\mathsf{K}^t \text{ vs. } \mathsf{K}^{t'})^*$ indicates that Lemma 8.4 makes essential use of randomness, unless $\mathsf{NP} = \mathsf{P}$.

Lemma 8.7. *For every constant c, there is a constant c' with the following property. Let $t, t' : \mathbb{N} \to \mathbb{N}$ be such that for all $n \in \mathbb{N}$, $t(n) \le n^c$ and $t'(n) \ge 2^{c'n}$. Then $(\mathsf{K}^t \text{ vs. } \mathsf{K}^{t'})^*$ is NP-hard under deterministic polynomial-time Turing reductions iff $\mathsf{NP} = \mathsf{P}$.*

Proof sketch. The proof is nearly identical to that of Lemma 8.6. One may still compute a string *consistent* with x from advice (n, i) by simulating the reduction, obtaining the query x, and replacing any '$*$'s in x with '0's. Let $\tilde{x}$ be the string x with all *'s replaced by 0's. It is easy to verify that $(\mathsf{K}^{t'})^*(x) \le \mathsf{K}^{t'}(\tilde{x}) \le 4b \log n$. □

Note that we could prove the above lemmas for $(\mathsf{K}^t \text{ vs. } \mathsf{K})$ and $(\mathsf{K}^t \text{ vs. } \mathsf{K})^*$ (that is, with time-unbounded K and K^* in Π_N) without the use of a dreambreaker. If we additionally assume that the NP-hardness reductions are honest, we obtain the same results but with t' any superpolynomial function.

Lemma 8.8. *Let $t : \mathbb{N} \to \mathbb{N}$ be polynomial and $t' : \mathbb{N} \to \mathbb{N}$ superpolynomial. $(\mathsf{K}^t \text{ vs. } \mathsf{K}^{t'})$ is NP-hard under honest deterministic polynomial-time Turing reductions iff $\mathsf{NP} = \mathsf{P}$.*

Proof. Argue as in Lemma 8.6. Since the reduction is honest, we have

$$|x| \ge n^\gamma$$

for some constant $\gamma > 0$, for any string x queried in the reduction M. Recall that any such x of M may be computed from advice (n, i) in time at most

$$\begin{aligned} \left(n^{6bc}\right)^d + n^{6bc} &< n^{7bcd} \\ &\le |x|^{7bcd/\gamma} \\ &< t'(|x|), \end{aligned}$$

as desired. □

Lemma 8.9. *Let $t : \mathbb{N} \to \mathbb{N}$ be polynomial and $t' : \mathbb{N} \to \mathbb{N}$ superpolynomial. $(\mathsf{K}^t \text{ vs. } \mathsf{K}^{t'})^*$ is NP-hard under honest deterministic polynomial-time Turing reductions iff $\mathsf{NP} = \mathsf{P}$.*

9 Open Questions

We have shown various consequences of (time-bounded) Kolmogorov complexity being NP-hard under randomized notions of reducibility. Some of these consequences may be taken optimistically (Theorem 1.3), while others may be viewed as barriers to the kinds of NP-hardness in question (Theorems 1.1, 1.4), which include kinds of reduction that have previously been used to show NP-hardness of variants of K^t complexity (eg. [Hir22]; see Section 8 above).

This work leaves open a number of directions; here, we indicate a few.

1. Can we remove the requirement, in Theorems 1.1, 1.2, and 1.3, that the time bound t in the superscript be larger than the running time of the reduction? Recall that this requirement was due to our use of the coding theorem for pK^t.

2. Can we show consequences of randomized NP-hardness reductions to MKTP or MCSP (i.e., minimization problems for Allender's KT complexity or boolean circuit size)?

3. Can we extend Theorems 1.1, 1.2, or 1.3 to *adaptive* randomized Turing reductions? Note that this kind of extension is unlikely in the case of Theorem 1.4, given the prior work discussed in Section 1.2 [All+06; Hir20b].

4. Can we improve Theorem 1.4 to hold for randomized many-one reductions with constant failure probability? In particular, can we improve the "robustness" of many-one reductions to K, as in Theorem 1.5, to hold for constant failure probability and exponentially small failure probability?

References

[ABX08] Benny Applebaum, Boaz Barak, and David Xiao. "On Basing Lower-Bounds for Learning on Worst-Case Assumptions". In: *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*. IEEE Computer Society, 2008, pp. 211–220. DOI: 10.1109/FOCS.2008.35.

[AD17] Eric Allender and Bireswar Das. "Zero knowledge and circuit minimization". In: *Inf. Comput.* 256 (2017), pp. 2–8. DOI: 10.1016/J.IC.2017.04.004.

[AH19] Eric Allender and Shuichi Hirahara. "New Insights on the (Non-)Hardness of Circuit Minimization and Related Problems". In: *ACM Trans. Comput. Theory* 11.4 (2019), 27:1–27:27. DOI: 10.1145/3349616.

[AH91] William Aiello and Johan Håstad. "Statistical Zero-Knowledge Languages can be Recognized in Two Rounds". In: *J. Comput. Syst. Sci.* 42.3 (1991), pp. 327–345. DOI: 10.1016/0022-0000(91)90006-Q.

[AHK17] Eric Allender, Dhiraj Holden, and Valentine Kabanets. "The Minimum Oracle Circuit Size Problem". In: *Comput. Complex.* 26.2 (2017), pp. 469–496. DOI: 10.1007/S00037-016-0124-0.

[AHT23] Eric Allender, Shuichi Hirahara, and Harsha Tirumala. "Kolmogorov Complexity Characterizes Statistical Zero Knowledge". In: *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*. Ed. by Yael Tauman Kalai. Vol. 251. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, 3:1–3:19. DOI: 10.4230/LIPICS.ITCS.2023.3.

[Aka+06] Adi Akavia, Oded Goldreich, Shafi Goldwasser, and Dana Moshkovitz. "On basing one-way functions on NP-hardness". In: *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*. Ed. by Jon M. Kleinberg. ACM, 2006, pp. 701–710. DOI: 10.1145/1132516.1132614.

[Aka+10] Adi Akavia, Oded Goldreich, Shafi Goldwasser, and Dana Moshkovitz. "Erratum for: on basing one-way functions on NP-hardness". In: *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*. Ed. by Leonard J. Schulman. ACM, 2010, pp. 795–796. DOI: 10.1145/1806689.1806798.

[All+06] Eric Allender, Harry Buhrman, Michal Koucký, Dieter van Melkebeek, and Detlef Ronneburger. "Power from Random Strings". In: *SIAM J. Comput.* 35.6 (2006), pp. 1467–1493. DOI: 10.1137/050628994.

[BHZ87] Ravi B. Boppana, Johan Håstad, and Stathis Zachos. "Does co-NP Have Short Interactive Proofs?" In: *Inf. Process. Lett.* 25.2 (1987), pp. 127–132. DOI: 10.1016/0020-0190(87)90232-8.

[BT06a] Andrej Bogdanov and Luca Trevisan. "Average-Case Complexity". In: *Found. Trends Theor. Comput. Sci.* 2.1 (2006). DOI: 10.1561/0400000004.

[BT06b] Andrej Bogdanov and Luca Trevisan. "On Worst-Case to Average-Case Reductions for NP Problems". In: *SIAM J. Comput.* 36.4 (2006), pp. 1119–1159. DOI: 10.1137/S0097539705446974.

[BTW10] Andrej Bogdanov, Kunal Talwar, and Andrew Wan. "Hard Instances for Satisfiability and Quasi-one-way Functions". In: *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*. Ed. by Andrew Chi-Chih Yao. Tsinghua University Press, 2010, pp. 290–300.

[FF93] Joan Feigenbaum and Lance Fortnow. "Random-Self-Reducibility of Complete Sets". In: *SIAM J. Comput.* 22.5 (1993), pp. 994–1005. DOI: 10.1137/0222061.

[For89] Lance Fortnow. "The Complexity of Perfect Zero-Knowledge". In: *Adv. Comput. Res.* 5 (1989), pp. 327–343.

[Gol+22] Halley Goldberg, Valentine Kabanets, Zhenjian Lu, and Igor C. Oliveira. "Probabilistic Kolmogorov Complexity with Applications to Average-Case Complexity". In: *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*. Ed. by Shachar Lovett. Vol. 234. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 16:1–16:60. DOI: 10.4230/LIPICS.CCC.2022.16.

[GS86] Shafi Goldwasser and Michael Sipser. "Private Coins versus Public Coins in Interactive Proof Systems". In: *Proceedings of the 18th Annual ACM Symposium on Theory of Computing, May 28-30, 1986, Berkeley, California, USA*. Ed. by Juris Hartmanis. ACM, 1986, pp. 59–68. DOI: 10.1145/12130.12137.

[GSV99] Oded Goldreich, Amit Sahai, and Salil P. Vadhan. "Can Statistical Zero Knowledge Be Made Non-interactive? or On the Relationship of SZK and NISZK". In: *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*. Ed. by Michael J. Wiener. Vol. 1666. Lecture Notes in Computer Science. Springer, 1999, pp. 467–484. DOI: 10.1007/3-540-48405-1_30.

[GT07] Dan Gutfreund and Amnon Ta-Shma. "Worst-Case to Average-Case Reductions Revisited". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 10th International Workshop, APPROX 2007, and 11th International Workshop, RANDOM 2007, Princeton, NJ, USA, August 20-22, 2007, Proceedings*. Ed. by Moses Charikar, Klaus Jansen, Omer Reingold, and José D. P. Rolim. Vol. 4627. Lecture Notes in Computer Science. Springer, 2007, pp. 569–583. DOI: 10.1007/978-3-540-74208-1_41.

[Hir18] Shuichi Hirahara. "Non-Black-Box Worst-Case to Average-Case Reductions within NP". In: *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*. Ed. by Mikkel Thorup. IEEE Computer Society, 2018, pp. 247–258. DOI: 10.1109/FOCS.2018.00032.

[Hir20a] Shuichi Hirahara. "Non-Disjoint Promise Problems from Meta-Computational View of Pseudorandom Generator Constructions". In: *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*. Ed. by Shubhangi Saraf. Vol. 169. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 20:1–20:47. DOI: 10.4230/LIPICS.CCC.2020.20.

[Hir20b] Shuichi Hirahara. "Unexpected hardness results for Kolmogorov complexity under uniform reductions". In: *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*. Ed. by Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy. ACM, 2020, pp. 1038–1051. DOI: 10.1145/3357713.3384251.

[Hir21] Shuichi Hirahara. "Average-case hardness of NP from exponential worst-case hardness assumptions". In: *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*. Ed. by Samir Khuller and Virginia Vassilevska Williams. ACM, 2021, pp. 292–302. DOI: 10.1145/3406325.3451065.

[Hir22] Shuichi Hirahara. "NP-Hardness of Learning Programs and Partial MCSP". In: *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*. IEEE, 2022, pp. 968–979. DOI: 10.1109/FOCS54457.2022.00095.

[Hir23] Shuichi Hirahara. "Capturing One-Way Functions via NP-Hardness of Meta-Complexity". In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*. Ed. by Barna Saha and Rocco A. Servedio. ACM, 2023, pp. 1027–1038. DOI: 10.1145/3564246.3585130.

[HN24] Shuichi Hirahara and Mikito Nanashima. "One-Way Functions and Zero Knowledge". In: *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*. Ed. by Bojan Mohar, Igor Shinkar, and Ryan O'Donnell. ACM, 2024, pp. 1731–1738. DOI: 10.1145/3618260.3649701.

[HP15] John M. Hitchcock and Aduri Pavan. "On the NP-Completeness of the Minimum Circuit Size Problem". In: *35th IARCS Annual Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS 2015, December 16-18, 2015, Bangalore, India*. Ed. by Prahladh Harsha and G. Ramalingam. Vol. 45. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015, pp. 236–245. DOI: 10.4230/LIPICS.FSTTCS.2015.236.

[HW15] Shuichi Hirahara and Osamu Watanabe. "Limits of Minimum Circuit Size Problem as Oracle". In: *Electron. Colloquium Comput. Complex.* TR15-198 (2015). ECCC: TR15-198.

[HW16] Shuichi Hirahara and Osamu Watanabe. "Limits of Minimum Circuit Size Problem as Oracle". In: *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*. Ed. by Ran Raz. Vol. 50. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016, 18:1–18:20. DOI: 10.4230/LIPICS.CCC.2016.18.

[IL90] Russell Impagliazzo and Leonid A. Levin. "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*. IEEE Computer Society, 1990, pp. 812–821. DOI: 10.1109/FSCS.1990.89604.

[Ila20] Rahul Ilango. "Approaching MCSP from Above and Below: Hardness for a Conditional Variant and $AC^0[p]$". In: *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*. Ed. by Thomas Vidick. Vol. 151. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 34:1–34:26. DOI: 10.4230/LIPICS.ITCS.2020.34.

[Ila23] Rahul Ilango. "SAT Reduces to the Minimum Circuit Size Problem with a Random Oracle". In: *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*. IEEE, 2023, pp. 733–742. DOI: 10.1109/FOCS57990.2023.00048.

[ILL89] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. "Pseudo-random Generation from one-way functions (Extended Abstracts)". In: *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washington, USA*. Ed. by David S. Johnson. ACM, 1989, pp. 12–24. DOI: 10.1145/73007.73009.

[ILO20] Rahul Ilango, Bruno Loff, and Igor C. Oliveira. "NP-Hardness of Circuit Minimization for Multi-Output Functions". In: *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*. Ed. by Shubhangi Saraf. Vol. 169. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 22:1–22:36. DOI: 10.4230/LIPICS.CCC.2020.22.

[IRS21] Rahul Ilango, Hanlin Ren, and Rahul Santhanam. "Hardness on any Samplable Distribution Suffices: New Characterizations of One-Way Functions by Meta-Complexity". In: *Electron. Colloquium Comput. Complex.* TR21-082 (2021). ECCC: TR21-082.

[KC00] Valentine Kabanets and Jin-yi Cai. "Circuit minimization problem". In: *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*. Ed. by F. Frances Yao and Eugene M. Luks. ACM, 2000, pp. 73–79. DOI: 10.1145/335305.335314.

[LOZ22] Zhenjian Lu, Igor C. Oliveira, and Marius Zimand. "Optimal Coding Theorems in Time-Bounded Kolmogorov Complexity". In: *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*. Ed. by Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff. Vol. 229. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 92:1–92:14. DOI: 10.4230/LIPICS.ICALP.2022.92.

[LP23] Yanyi Liu and Rafael Pass. "One-Way Functions and the Hardness of (Probabilistic) Time-Bounded Kolmogorov Complexity w.r.t. Samplable Distributions". In: *Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part II*. Ed. by Helena Handschuh and Anna Lysyanskaya. Vol. 14082. Lecture Notes in Computer Science. Springer, 2023, pp. 645–673. DOI: 10.1007/978-3-031-38545-2_21.

[LV19] Ming Li and Paul M. B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications, 4th Edition*. Texts in Computer Science. Springer, 2019. DOI: 10.1007/978-3-030-11298-1.

[MW15] Cody D. Murray and Richard Ryan Williams. "On the (Non) NP-Hardness of Computing Circuit Complexity". In: *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*. Ed. by David Zuckerman. Vol. 33. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015, pp. 365–380. DOI: 10.4230/LIPICS.CCC.2015.365.

[Nan21] Mikito Nanashima. "On Basing Auxiliary-Input Cryptography on NP-Hardness via Nonadaptive Black-Box Reductions". In: *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*. Ed. by James R. Lee. Vol. 185. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, 29:1–29:15. DOI: 10.4230/LIPICS.ITCS.2021.29.

[Ost91] Rafail Ostrovsky. "One-Way Functions, Hard on Average Problems, and Statistical Zero-Knowledge Proofs". In: *Proceedings of the Sixth Annual Structure in Complexity Theory Conference, Chicago, Illinois, USA, June 30 - July 3, 1991*. IEEE Computer Society, 1991, pp. 133–138. DOI: 10.1109/SCT.1991.160253.

[OW93] Rafail Ostrovsky and Avi Wigderson. "One-Way Fuctions are Essential for Non-Trivial Zero-Knowledge". In: *Second Israel Symposium on Theory of Computing Systems, ISTCS 1993, Natanya, Israel, June 7-9, 1993, Proceedings*. IEEE Computer Society, 1993, pp. 3–17. DOI: 10.1109/ISTCS.1993.253489.

[SS20] Michael E. Saks and Rahul Santhanam. "Circuit Lower Bounds from NP-Hardness of MCSP Under Turing Reductions". In: *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*. Ed. by Shubhangi Saraf. Vol. 169. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 26:1–26:13. DOI: 10.4230/LIPICS.CCC.2020.26.

[SS22] Michael E. Saks and Rahul Santhanam. "On Randomized Reductions to the Random Strings". In: *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*. Ed. by Shachar Lovett. Vol. 234. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 29:1–29:30. DOI: 10.4230/LIPICS.CCC.2022.29.

[Tra84] Boris A. Trakhtenbrot. "A Survey of Russian Approaches to Perebor (Brute-Force Searches) Algorithms". In: *IEEE Ann. Hist. Comput.* 6.4 (1984), pp. 384–400. DOI: 10.1109/MAHC.1984.10036.

[Yao82] Andrew Chi-Chih Yao. "Theory and Applications of Trapdoor Functions (Extended Abstract)". In: *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*. IEEE Computer Society, 1982, pp. 80–91. DOI: 10.1109/SFCS.1982.45.