Cryptographic Hardness Under Projections for Time-Bounded Kolmogorov Complexity

Eric Allender
Rutgers University, Piscataway, NJ, USA

John Gouwar
Northeastern University, Boston, USA

Shuichi Hirahara
National Institute of Informatics, Japan

Caleb Robelle
MIT, Boston, USA

Abstract

A version of time-bounded Kolmogorov complexity, denoted KT, has received attention in the past several years, due to its close connection to circuit complexity and to the Minimum Circuit Size Problem MCSP. Essentially all results about the complexity of MCSP hold also for MKTP (the problem of computing the KT complexity of a string). Both MKTP and MCSP are hard for SZK (Statistical Zero Knowledge) under BPP-Turing reductions; neither is known to be NP-complete.

Recently, some hardness results for MKTP were proved that are not (yet) known to hold for MCSP. In particular, MKTP is hard for DET (a subclass of P) under nonuniform $\leq_m^{NC^0}$ reductions. In this paper, we improve this, to show that $\overline{\text{MKTP}}$ is hard for the (apparently larger) class NISZK_L under not only $\leq_m^{NC^0}$ reductions but even under projections. Also $\overline{\text{MKTP}}$ is hard for NISZK under $\leq_m^{P/poly}$ reductions. Here, NISZK is the class of problems with non-interactive zero-knowledge proofs, and NISZK_L is the non-interactive version of the class SZK_L that was studied by Dvir et al.

As an application, we provide several improved worst-case to average-case reductions to problems in NP, and we obtain a new lower bound on MKTP (which is currently not known to hold for MCSP).

2012 ACM Subject Classification Theory of computation → Complexity classes; Theory of computation → Problems, reductions and completeness; Theory of computation → Circuit complexity

Keywords and phrases Kolmogorov Complexity, Interactive Proofs, Minimum Circuit Size Problem, Worst-case to Average-case Reductions

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2021.54

Related Version *Full Version*: https://eccc.weizmann.ac.il/report/2021/010/

Funding JG and CR were supported in part by the DIMACS 2020 REU program under NSF grants CCF-1852215 and CCF-1836666.
Eric Allender: Supported in part by NSF Grants CCF-1909216 and CCF-1909683.

Acknowledgements We thank Rahul Santhanam, Oded Goldreich, Salil Vadhan, Harsha Tirumala, Dieter van Melkebeek and Andrew Morgan for helpful discussions. We also thank the anonymous reviewers who provided helpful comments.

1 Introduction

The study of time-bounded Kolmogorov complexity is tightly connected to the study of circuit complexity. Indeed, the measure that we study most closely in this paper, denoted KT, was initially defined in order to capitalize on the framework of Kolmogorov complexity in investigations of the Minimum Circuit Size Problem (MCSP) [4]. If f is a bit string of length 2^k representing the truth-table of a k-ary Boolean function, then $KT(f)$ is polynomially

related to the size of the smallest circuit computing f. Thus the problem of computing KT complexity (denoted MKTP) was initially viewed as a more-or-less equivalent encoding of MCSP, and it is still the case that all theorems that have been proved about the complexity of MCSP hold also for MKTP (such as those in [5, 9, 10, 17, 21–24, 30, 31, 33, 34]).

In recent years, however, a few hardness results were proved for MKTP that are not yet known to hold for MCSP [7, 8]. We believe that these results can be taken as an indication of what is likely to be true also for MCSP. The present work gives significantly improved hardness results for MKTP.

Reducibility and completeness are the most effective tools in the arsenal of complexity theory for giving evidence of intractability. However, it is not clear whether MCSP or MKTP is NP-complete; neither can be shown to be NP-complete – or even hard for ZPP – under the usual $\leq_{\mathrm{m}}^{\mathsf{P}}$ reductions without first showing that EXP $\neq$ ZPP, a long-standing open problem [17, 31].

The strongest hardness results that have been proved thus far for MCSP and MKTP are that both are hard for SZK under BPP-Turing reductions [5]. SZK is the class of problems that have Statistical Zero Knowledge Interactive Proofs, and contains many problems of interest to cryptographers. Indeed, if MCSP (or MKTP) is in P/poly, then there are no cryptographically-secure one-way functions [26].

Our main results involve improving the hardness results for MKTP, by reducing the number of queries from polynomially-many, to one. In the paragraphs that follow, we explain the sense in which we accomplish this goal. Along the way, we also obtain a new circuit lower bound for MKTP; it remains unknown whether this circuit lower bound also holds for MCSP.

SZK is not known to be contained in NP; until such a containment can be established, there is no hope of improving the BPP-Turing reduction of [5] to a $\leq_{\mathrm{m}}^{\mathsf{P}}$ reduction. But we come close in this paper. NISZK is the "non-interactive" subclass of SZK; it contains intractable problems if and only if SZK does [18]. We show that $\overline{\mathsf{MKTP}}$ is hard for NISZK under $\leq_{\mathrm{m}}^{\mathsf{P/poly}}$ reductions. (Thus, instead of asking many queries, as in [5], a single query suffices.[1]) Our proof also shows that MKTP is hard for NISZK under BPP reductions that ask only one query. Combined with [18], this shows that MKTP is hard for SZK under *non-adaptive* BPP reductions, yielding a modest improvement over [5]; this has implications regarding the study of worst-case to average-case reductions. (See Section 1.1.)

But $\leq_{\mathrm{m}}^{\mathsf{P/poly}}$ reductions are still quite powerful. There is great interest currently in proving lower bounds for MCSP, MKTP, and related problems such as MKtP (the problem of computing a different kind of time-bounded Kolmogorov complexity, due to Levin [28]) on very limited classes of circuits and formulae, as part of the "hardness magnification" program. For instance, if modest lower bounds can be shown on the size required to compute MKtP on de Morgan formulae augmented with PARITY gates at the leaves, then EXP is not contained in non-uniform NC^1 [32]. Also, there is great interest in finding lower bounds against a variety of other models, such as depth-three threshold gates, or circuits consisting of polynomial threshold gates [27]. If a lower bound is known against one of these limited classes of circuits for some problem A that is reducible to, say, MKTP or MKtP under $\leq_{\mathrm{m}}^{\mathsf{P/poly}}$ reductions, it implies nothing about the complexity of MKTP or MKtP, since the circuitry involved in computing the reduction is much more powerful than the circuitry in the class of circuits for which the lower bound is known.

[1] Some readers may have mistakenly believed that we view our work as a step toward showing that MKTP (or MCSP) is hard for SZK under (uniform) $\leq_{\mathrm{m}}^{\mathsf{P}}$ reductions. We do not. In fact, some of us doubt that hardness under uniform deterministic reductions holds.

Thus there is a great deal of interest in considering reductions that are much less powerful than $\leq_m^{\mathsf{P/poly}}$ reductions. For extremely weak (uniform) notions of reducibility (such as log-time reductions), it is known that MCSP and MKTP are *not* hard for any complexity class that contains the PARITY function [31]. However, this non-hardness result relies on uniformity; it was later shown that MKTP is hard for the complexity class DET under *nonuniform* $\leq_m^{\mathsf{NC}^0}$ reductions [8].

However, even $\leq_m^{\mathsf{NC}^0}$ reductions are too powerful a tool, when one is interested in lower bounds against the classes of circuits discussed above, since they do not seem to be closed under $\leq_m^{\mathsf{NC}^0}$ reductions. This motivates consideration of the most restrictive type of reduction that we will be considering: projections.

A projection is a reduction that is computed by a circuit consisting only of wires and NOT gates. Each output bit is either a constant, or is connected by a wire to a (possibly negated) input bit. All of the classes of circuits mentioned above (and – indeed – most conceivable classes of circuits) are closed under projections.

Prior to our work, the result of [8] showing that MKTP is hard for DET under $\leq_m^{\mathsf{NC}^0}$ reductions was improved, to show that MKTP is hard for DET even under projections [3]. Since DET is a subclass of P, this provides little ammunition when one is seeking to prove that MKTP is intractable. One of our main contributions is to show that $\overline{\mathsf{MKTP}}$ is hard for $\mathsf{NISZK_L}$ under projections. As a corollary, we obtain that MKTP cannot be computed by THRESHOLD$\circ$MAJORITY circuits of size $2^{n^{o(1)}}$. This lower bound relies on the fact that MKTP is hard under projections.

The reader will not be familiar with $\mathsf{NISZK_L}$; this complexity class makes its first appearance in the literature here. It is the "non-interactive" counterpart to the complexity class $\mathsf{SZK_L}$ that was studied previously by Dvir et al. [15], and was shown there to contain several important natural problems of interest to cryptographers (such as Discrete Log and Decisional Diffie-Hellman). $\mathsf{NISZK_L}$ contains intractable problems if and only if $\mathsf{SZK_L}$ does (see Section 2). Thus, for the first time, we show that MKTP is hard under projections for a complexity class that is widely believed to contain intractable problems. Our hardness results carry over immediately to MKtP and to similar problems defined in terms of general Kolmogorov complexity; no hardness results under projections had been known previously for those problems. We present some complete problems for $\mathsf{NISZK_L}$ and establish some other basic facts about this class in Section 4.

1.1 Average-Case Complexity

Building on the techniques introduced in [20], we are able to establish new insights regarding the relationship between worst-case and average-case complexity. In Theorem 35, capitalizing on the fact that essentially every circuit complexity class $\mathcal{C}$ is closed under projections, we show that if $\mathsf{NISZK_L}$ does not lie in $\mathsf{OR} \circ \mathcal{C}$, then there are problems A in NP that cannot be solved *in the average case* by errorless heuristics in $\mathcal{C}$. For instance, if one were able to show that there is *any* problem $\mathsf{NISZK_L}$ (including, but not limited to, some of the candidate one-way functions believed to reside there) that cannot be solved *in the worst case* by depth-four ACC^0 circuits, it would follow that there are problems in NP that are hard-on-average for depth-three ACC^0 circuits. Such conclusions would *not* follow if our reductions to MKTP had merely been computable in AC^0 or NC^0.

We are also able to shed more light on worst-case to average-case reductions, in the form that they were studied by Bogdanov and Trevisan [14]. Bogdanov and Trevisan showed that there were severe limits on the complexity of problems whose worst-case complexity could be reduced to the average-case complexity of problems in NP via *non-adaptive* reductions; all such problems lie in $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$. But it was not known how large this class of

problems could be. Hirahara showed that every problem in SZK has an *adaptive* worst-case to average-case reduction to a problem in NP [20], but the upper bound of NP/poly ∩ coNP/poly proved by Bogdanov and Trevisan does not apply for adaptive reductions. As a consequence of our Corollary 17, showing that MKTP is hard for SZK under nonadaptive BPP reductions, we are able to show (in Corollary 37) that the class identified by Bogdanov and Trevisan lies in the narrow range between SZK and NP/poly ∩ coNP/poly.

▶ Remark. This is an illustration of the utility of studying MKTP, as an example of a theorem that does not explicitly mention MKTP or MCSP, but which was proved via the study of MKTP. No such argument based on MCSP is known. We believe that MKTP can in fact be viewed as a *particularly convenient* formulation of MCSP, since (a) KT complexity is closely related to circuit size, (b) essentially all theorems known to hold for MCSP also hold for MKTP, (c) some arguments that one might intend to formulate in terms of MCSP elude current approaches, but can instead be successfully carried through by use of MKTP. Furthermore, theorems proved for MKTP may serve as an indication of what is likely to be true for MCSP as well.

The rest of the paper is organized as follows: Our $\leq_{\mathrm{m}}^{\mathsf{P/poly}}$-hardness theorem for MKTP is proved in Section 3. Then, after establishing some basic facts about $\mathsf{NISZK_L}$ in Section 4, in Section 5 we show that $\overline{\mathsf{MKTP}}$ is hard for $\mathsf{NISZK_L}$ under projections. We present applications of our reductions and implications for average-case complexity in Section 6.

Due to space limitations, some proofs have been omitted from the version of this work that appears in the ISAAC proceedings. The interested reader is encouraged to consult [6] for complete details.

2 Preliminaries

2.1 Complexity Classes and Reducibilities

We assume familiarity with the complexity classes P, NP, L, BPP, and P/poly. We also make use of the circuit complexity classes $\mathsf{AC^0}$ and $\mathsf{NC^0}$. For the purposes of this paper, $\mathsf{AC^0}$ can be understood as the set of problems for which there is a family of circuits $\{C_n : n \in \mathbb{N}\}$ with unbounded-fan-in AND and OR gates (and NOT gates of fan-in 1) of polynomial size and constant depth. $\mathsf{NC^0}$ is defined similarly, but with AND and OR gates of bounded fan-in (and thus each output bit depends on only a constant number of bits of the input). We deal primarily with the "nonuniform" versions of these complexity classes (which means that the mapping $n \mapsto C_n$ need not be computable).

Branching programs are a circuit-like model of computation that can be used to characterize logspace computation. A *branching program* is a directed acyclic graph with a single source and two sinks labeled 1 and 0, respectively. Each non-sink node in the graph is labeled with a variable in $\{x_1, \ldots, x_n\}$ and has two edges leading out of it: one labeled 1 and one labeled 0. A branching program computes a Boolean function f on input $x = x_1 \ldots x_n$ by first placing a pebble on the source node. At any time when the pebble is on a node v labeled x_i, the pebble is moved to the (unique) vertex u that is reached by the edge labeled 1 if $x_i = 1$ (or by the edge labeled 0 if $x_i = 0$). If the pebble eventually reaches the sink labeled b, then $f(x) = b$. Branching programs can also be used to compute functions $f : \{0,1\}^m \to \{0,1\}^n$, by concatenating n branching programs $p_1, \ldots, p_n$, where p_i computes the function $f_i(x) =$ the i-th bit of $f(x)$. For more information on the definitions, backgrounds, and nuances of these complexity classes, circuits, and branching programs, see the text by Vollmer [35].

A *promise problem* Π is a pair of disjoint sets (Π_{YES}, Π_{NO}). A *solution* to a promise problem is any set A such that $\Pi_{YES} \subseteq A$ and $\Pi_{NO} \subseteq \overline{A}$. A *don't-care instance* of Π is any string that is not in $\Pi_{YES} \cup \Pi_{NO}$. A *language* A can be viewed as a promise problem that has no don't-care instances.

Given any class $\mathcal{C}$ of functions, there is an associated notion of *m-reducibility* or *many-one reducibility*: For two languages A and B, we say that $A \leq_{\mathrm{m}}^{\mathcal{C}} B$ if there is a function f in $\mathcal{C}$ such that $x \in A$ iff $f(x) \in B$. This notion of reducibility extends naturally to promise problems, mapping yes-instances to yes-instances, and no-instances to no-instances. The most familiar notion of m-reducibility is Karp reducibility: $\leq_{\mathrm{m}}^{\mathsf{P}}$; NP-completeness is most commonly defined in terms of Karp reducibility. However, in this paper, we will frequently be reducing problems that are not known to reside in NP to MKTP, which does lie in NP. Thus it is clear that a more powerful notion of reducibility is required. Some of our results are most conveniently stated in terms of $\leq_{\mathrm{m}}^{\mathsf{P/poly}}$ reductions (i.e., reductions computed by nonuniform polynomial-size circuits). We also consider restrictions of $\leq_{\mathrm{m}}^{\mathsf{P/poly}}$ reductions, computed by nonuniform AC^0 and NC^0 circuits: $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ and $\leq_{\mathrm{m}}^{\mathsf{NC}^0}$. Finally we also consider *projections* ($\leq_{\mathrm{m}}^{\mathsf{proj}}$), which are functions computed by NC^0 circuits that have only NOT gates. That is, in a projection, each output bit is either a constant 0 or 1, or is connected by a wire to an input bit or its negation.

We will also make reference to various types of *Turing reducibility*, which are defined in terms of oracle Turing machines, or in terms of circuit families that are augmented with "oracle gates". For instance, we say that $A \leq_{\mathrm{T}}^{\mathsf{BPP}} B$ if there is a probabilistic polynomial time oracle Turing machine M with oracle B that accepts every $x \in A$ with probability $\frac{2}{3}$ and rejects every $x \in \overline{A}$ with probability $\frac{2}{3}$. Note that the computation tree of such a BPP-Turing reduction can contain an exponential number of queries to different elements of B. Just as $\mathsf{BPP} \subseteq \mathsf{P/poly}$, it also holds that $A \leq_{\mathrm{T}}^{\mathsf{BPP}} B$ implies $A \leq_{\mathrm{T}}^{\mathsf{P/poly}} B$. Thus, on any input x, the circuit computing the P/poly-Turing reduction queries only a polynomial number of elements of B. It was shown in [5] that every problem in SZK (that is, every problem with a statistical zero knowledge proof system) is $\leq_{\mathrm{T}}^{\mathsf{BPP}}$-reducible (and hence $\leq_{\mathrm{T}}^{\mathsf{P/poly}}$-reducible) to MCSP and to MKTP. The question of interest to us here is: Is it necessary to ask so many queries? What can we do if we ask only one query? What can be reduced to MKTP via a $\leq_{\mathrm{m}}^{\mathsf{P/poly}}$ reduction?

The complexity class with which we are primarily concerned in this paper is the class of problems that have non-interactive statistical zero knowledge proof systems: NISZK. NISZK was originally defined and studied by Blum et al. [13]. The definition below (in terms of promise problems) is due to Goldreich et al. [18].

▶ **Definition 1.** *A* non-interactive statistical zero-knowledge proof system *for a promise problem* Π *is defined by a triple of probabilistic machines* P, V, *and* S, *where* V *and* S *are polynomial-time and* P *is computationally unbounded, and a polynomial* $r(n)$ *(which will give the size of the random reference string* σ*), such that:*

1. *(Completeness) For all* $x \in \Pi_{YES}$, *the probability that* $V(x, \sigma, P(x, \sigma))$ *accepts is at least* $1 - 2^{-|x|}$.
2. *(Soundness) For all* $x \in \Pi_{NO}$, *the probability that* $V(x, \sigma, P(x, \sigma))$ *accepts is at most* $2^{-|x|}$.
3. *(Zero Knowledge) For all* $x \in \Pi_{YES}$, *the statistical distance between the following two distributions bounded by* $1/\beta(|x|)$
   a. *Choose* σ *uniformly from* $\{0,1\}^{r(|x|)}$, *sample* p *from* $P(x, \sigma)$, *and output* (p, σ).
   b. *S(x) (where the coins for S are chosen uniformly at random.)*

where $\beta(n)$ *is superpolynomial, and the probabilities in Conditions 1 and 2 are taken over the random coins of* V *and* P, *and the choice of* σ *uniformly from* $\{0,1\}^{r(n)}$.

NISZK *is the class of promise problems for which there is a non-interactive statistical zero knowledge proof system.*

NISZK is not known to be closed under complementation; co-NISZK is defined as the class of promise problems $\Pi = (\Pi_{YES}, \Pi_{NO})$ such that (Π_{NO}, Π_{YES}) is in NISZK. It is known that SZK = NISZK iff NISZK = co-NISZK, and that every promise problem in SZK efficiently (and non-adaptively) Turing-reduces to a problem in NISZK [18]. Thus NISZK contains intractable problems if and only if SZK does.

A subclass of SZK, which we will denote by $\mathsf{SZK_L}$, in which the verifier V and simulator S are restricted to being logspace machines, was defined and studied by Dvir et al. [15]. Among other things, they showed that many of the important natural problems in SZK lie in $\mathsf{SZK_L}$, including Graph Isomorphism, Quadratic Residuosity, Discrete Log, and Decisional Diffie-Helman. The non-interactive version of $\mathsf{SZK_L}$, which we denote by $\mathsf{NISZK_L}$, has not been studied previously, but it figures prominently in our results.

▶ **Definition 2.** *The formal definition of* $\mathsf{NISZK_L}$ *is obtained by replacing each occurrence of "polynomial-time" in Definition 1 with "logspace". (It is important to note that, in this model, the logspace-bounded verifier* V *and simulator* S *are allowed two-way access to the reference string* σ *and to their polynomially-long sequences of probabilistic coin flips.)*

The reduction presented in [18] carries over directly to the logspace setting, showing that $\mathsf{NISZK_L}$ contains intractable problems if and only if $\mathsf{SZK_L}$ does. In particular, we have:

▶ **Proposition 3.** *Every promise problem in* $\mathsf{SZK_L}$ *is non-adaptively* AC^0*-Turing-reducible to a problem in* $\mathsf{NISZK_L}$*.*

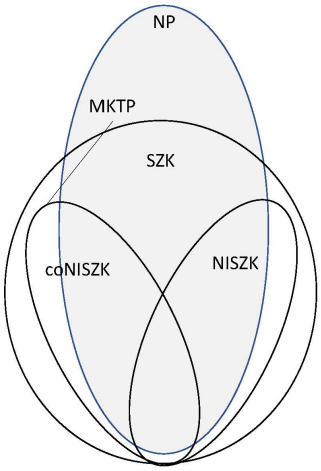

Figure 1 Diagram showing the classes NISZK, co-NISZK, and SZK. The shaded oval represents NP. Every problem in co-NISZK is $\leq_{\mathrm{m}}^{\mathsf{P/poly}}$-reducible to MKTP.

2.2 KT Complexity

The measure KT was defined in [4]. We provide a reproduction of that definition below.

▶ **Definition 4** (KT). *Let* U *be a universal Turing machine. For each string* x*, define* $\mathrm{KT}_U(x)$ *to be*

$$\min\{|d| + T : (\forall \sigma \in \{0,1,*\})\ (\forall i \leq |x|+1)\ U^d(i,\sigma)\ \textit{accepts in}\ T\ \textit{steps iff}\ x_i = \sigma\}$$

We define $x_i = *$ *if* $i > |x|$*; thus, for* $i = |x|+1$ *the machine accepts iff* $\sigma = *$*. The notation* U^d *indicates that the machine* U *has random access to the description* d*.*

To understand the motivation for this definition, see [4]. Briefly: KT is a version of time-bounded Kolmogorov complexity that (in contrast to other notions of resource-bounded Kolmogorov complexity that have been considered) is polynomially-related to circuit complexity. The minimum KT problem, henceforth MKTP, is defined below.

▶ **Definition 5** (MKTP). *Suppose* $y \in \{0,1\}^n$ *and* $\theta \in \mathbb{N} \setminus \{0\}$*, then*

$$\mathsf{MKTP} = \{(y, \theta) \mid \mathrm{KT}(y) \leq \theta\}.$$

In this paper when we view MKTP *as a promise problem, yes-instances will be considered those that are in the language, and no-instances those that are not in the language.*

3 $\overline{\mathsf{MKTP}}$ is Hard For NISZK

In this section, we prove our first hardness result for MKTP; MKTP is hard for co-NISZK under $\leq_{\mathrm{m}}^{\mathsf{P/poly}}$ reductions. In order to prove hardness, it suffices to provide a reduction from the *entropy approximation* problem: EA, which is known to be complete for NISZK under $\leq_{\mathrm{m}}^{\mathsf{P}}$ reductions [18].

▶ **Definition 6** (Promise-EA). *Let a circuit* $C : \{0,1\}^m \to \{0,1\}^n$ *represent a probability distribution* X *on* $\{0,1\}^n$ *induced by the uniform distribution on* $\{0,1\}^m$*. We define Promise-*EA *to be the promise problem*

$$\begin{aligned} \mathsf{EA}_{YES} &= \{(C,k) \mid H(X) > k+1\} \\ \mathsf{EA}_{NO} &= \{(C,k) \mid H(X) < k-1\} \end{aligned}$$

where $H(X)$ *denotes the entropy of* X*.*

We will make use of some machinery that was developed in [7], in order to relate the entropy of a distribution to the KT complexity of samples taken from the distribution. However, these tools are only useful when applied to distributions that are sufficiently "flat". The next subsection provides the necessary tools to "flatten" a distribution.

3.1 Flat Distributions

A distribution is considered *flat* if it is uniform on its support. Goldreich et al. [18] formalized a relaxed notion of flatness, termed Δ-flatness, which relies on the concept of Δ-typical elements. The definitions of both concepts follow:

▶ **Definition 7** (Δ-typical elements). *Suppose* X *is a distribution with element* x *in its support. We say that* x *is* Δ*-typical if,*

$$2^{-\Delta} \cdot 2^{-H(X)} < \Pr[X = x] < 2^{\Delta} \cdot 2^{-H(X)}.$$

▶ **Definition 8** (Δ-flatness). *Suppose* X *is a distribution. We say that* X *is* Δ*-flat if for every* $t > 0$ *the probability that an element of the support,* x*, is* $t \cdot \Delta$*-typical is at least* $1 - 2^{-t^2+1}$*.*

▶ **Lemma 9** (Flattening Lemma, [18]). *Suppose* X *is a distribution such that for all* x *in its support* $\Pr[X = x] \geq 2^{-m}$*. Then* X^k *is* $(\sqrt{k} \cdot m)$*-flat.*

Observe that if X is a distribution represented by a circuit $C : \{0,1\}^m \to \{0,1\}^n$, then the hypothesis of the Flattening Lemma holds for m. Note also that, for any distribution X, $H(X^k) = k \cdot H(X)$. Thus the entropy of the distribution X^k grows linearly with respect to k, while the deviation from flatness diminishes much more rapidly with respect to k.

3.2 Encoding and Blocking

The *Encoding Lemma* is the primary tool that was developed in [7] to give short descriptions of samples from a given distribution. Below, we give a precise statement of the version of the Encoding Lemma that is stated informally as Remark 4.3 of [7]. (Although the statement there is informal, the proof of the Encoding Lemma that is given there does yield our Lemma 11.) First, we need to define Λ-encodings.

▶ **Definition 10** (Λ-encodings). *Let $R : S \to T$ be a random variable that induces a distribution X. The* Λ-heavy *elements of T are those elements λ such that $\Pr[X = \lambda] > 1/2^{\Lambda}$. A Λ-encoding of R is given by a mapping $D : [N] \to S$ such that for every Λ-heavy element λ, there exists $i \in [N]$ such that $R(D(i)) = \lambda$. We refer to $\lceil \log(N) \rceil$ as the length of the encoding. The function D is also called the* decoder *for the encoding.*

▶ **Lemma 11** (Encoding Lemma). *[7, Lemma 4.1] Consider an ensemble $\{R_x\}$ of random variables that sample distributions on strings of some length $\mathrm{poly}_1(|x|)$, where there are circuits C_x of size $\mathrm{poly}_2(|x|)$ representing each R_x. Then there is a polynomial poly_3 such that, for every integer Λ, each R_x has a Λ-encoding of length $\Lambda + \log(\Lambda) + O(1)$ that is decodable by circuits of size $\mathrm{poly}_3(|x|)$.*

By itself, the Encoding Lemma says nothing about KT complexity. The other important ingredient in the toolbox developed in [7] is the *Blocking Lemma*, which refers to the process of chopping a string into blocks. Let y be a string of length tn, which we think of as being the concatenation of t samples y_i of a distribution X on strings of length n. Thus $y = y_1 \dots y_t$. Let $r = \lceil t/b \rceil$. Equivalently, we consider y to be equal to $z_1 \dots z_r$ where each z_i is a string of length bn sampled according to X^b. (In the case when $|y|$ is not a multiple of b, z_r is shorter; this does not affect the analysis. We call the strings z_i the *blocks* of y.)

▶ **Lemma 12** (Blocking Lemma). *[7, Lemma 3.3] Let $\{T_x\}$ be an ensemble of sets of strings such that all strings in T_x have the same length $\mathrm{poly}(|x|)$. Suppose that for each $x \in \{0,1\}^*$ and for each $b \in \mathbb{N}$ there is an integer Λ_b and a random variable $R_{x,b}$ whose image contains $(T_x)^b$, and such that $R_{x,b}$ is computable by a circuit of size $\mathrm{poly}(|x|, b)$ and has a Λ_b-encoding of length $s'(x, b)$ decodable by a circuit of size $\mathrm{poly}(|x|, b)$. Then there are constants c_1 and c_2 so that, for every constant $\alpha > 0$, every $t \in \mathbb{N}$, every sufficiently large x, and every $\lceil t^\alpha \rceil$-suitable $y \in (T_x)^t$,*

$$\mathrm{KT}(y) \leq t^{1-\alpha} \cdot s'(x, \lceil t^\alpha \rceil) + t^{\alpha \cdot c_1} \cdot |x|^{c_2}.$$

Here, we say that $y \in (T_x)^t$ is b-suitable if each block of y (of length bn) is Λ_b-heavy.

With the Encoding and Blocking Lemmas in hand, we can now show how to give upper and lower bounds on the KT complexity of concatenated samples from a distribution. The following lemma gives the upper bound.

▶ **Lemma 13.** *Suppose X is a distribution sampled by a circuit $C_x : \{0,1\}^m \to \{0,1\}^n$ of size polynomial in $|x|$. For every polynomial $w = w(|x|)$ with $|x| \leq w$, there exist constants c_0, c_2, and α_0 such that for every sufficiently large polynomial t and for all large x, if y is the concatenation of t samples from X, then with probability at least $(1 - 1/2^{2|x|})$,*

$$\mathrm{KT}(y) \leq tH(X) + wm(t^{1-\alpha_0/2}) + t^{1-\alpha_0}|x|^{c_0+c_2}$$

We now turn to a lower bound on $\mathsf{KT}(y)$.

▶ **Lemma 14.** *Let* $\mathrm{poly}(|x|)$ *denote some fixed polynomial in* $|x|$*, and let* α_0 *be such that* $0 < \alpha_0 < 1/2$*. For all large* x*, if* X *is a distribution sampled by a circuit* $C_x : \{0,1\}^m \to \{0,1\}^n$ *of polynomial size, then it holds that for every* w *and every* $t > w^4$*, if* y *is sampled from* X^t*, then with probability at least* $1 - 2^{-w^2}$*,*

$$\mathrm{KT}(y) \geq tH(X) - wm\sqrt{t} - t^{1-\alpha_0}\mathrm{poly}(|x|)$$

3.3 Reducing co-NISZK to MKTP

▶ **Theorem 15.** MKTP *is hard for* co-NISZK *under* P/poly *many-one reductions.*

Proof. We prove the claim by reduction from the NISZK-complete problem EA. Let $x = (C_x, k)$ be an arbitrary instance of Promise-EA, where $C_x : \{0,1\}^m \to \{0,1\}^n$ is a circuit that represents distribution X. Let $w = 2|x|$, and let α_0, c_0, and c_2 be the constants from Lemma 13. Let $\lambda = wmt^{1-\alpha_0/2}$. Pick the polynomial t so that $t(|x|) > 2(\lambda + t^{1-\alpha_0}|x|^{c_0+c_2})$ and $w^4 < t$ (and note that all large polynomials have this property). Construct y as t samples from X. Let $\theta = tk + \lambda + t^{1-\alpha_0}|x|^{c_0+c_2}$. We claim that, with probability at least $1 - \frac{1}{2^{2|x|}}$, if $(X,k) \in \mathsf{EA}_{YES}$, then $(y,\theta) \in \mathsf{MKTP}_{NO}$ and if $(X,k) \in \mathsf{EA}_{NO}$, then $(y,\theta) \in \mathsf{MKTP}_{YES}$.

If $(X,k) \in \mathsf{EA}_{NO}$, then $H(X) < k$. Then by Lemma 13, we have that, with high probability,

$$\begin{aligned}\mathrm{KT}(y) &\leq tH(X) + \lambda + t^{1-\alpha_0}|x|^{c_0+c_2} \\ &< tk + \lambda + t^{1-\alpha_0}|x|^{c_0+c_2} \\ &= \theta\end{aligned}$$

thus $\mathrm{KT}(y) \leq \theta$, and thus $(y,\theta) \in \mathsf{MKTP}_{YES}$.

If $(X,k) \in \mathsf{EA}_{YES}$, then $H(X) > k+1$. Then by Lemma 14, with probability at least $1 - 2^{-w^2} > 1 - 2^{2|x|}$, we have that

$$\begin{aligned}\mathrm{KT}(y) &\geq tH(X) - wm\sqrt{t} - t^{1-\alpha_0}|x|^{c_0+c_2}, \\ &> tH(X) - \lambda - t^{1-\alpha_0}|x|^{c_0+c_2} && \text{(since } \alpha_0 < 1/2\text{)} \\ &> t(k+1) - \lambda - t^{1-\alpha_0}|x|^{c_0+c_2} \\ &> tk + \lambda + t^{1-\alpha_0}|x|^{c_0+c_2} && \text{(since } t > 2(\lambda + t^{1-\alpha_0}|x|^{c_0+c_2})\text{)} \\ &= \theta\end{aligned}$$

thus $\mathrm{KT}(y) > \theta$, and thus $(y,\theta) \in \mathsf{MKTP}_{NO}$.

We have shown that there is a polynomial-time-computable function f, such that, if $x \in \mathsf{EA}_{YES}$, then with high probability (for random r) $f(x,r) = (y,\theta)$ is in MKTP_{NO}, and if $x \in \mathsf{EA}_{NO}$, then with high probability $f(x,r) = (y,\theta)$ is in MKTP_{YES}. By a standard counting argument (similar to the proof that $\mathsf{BPP} \subseteq \mathsf{P/poly}$), since the probability of success for either bound is greater than $(1 - 1/2^{2n})$, we can fix a sequence of random bits to hardwire in to this reduction and obtain a family of circuits computing a $\leq_m^{\mathsf{P/poly}}$ reduction from any problem in NISZK to $\overline{\mathsf{MKTP}}$. ◀

▶ **Corollary 16.** MKTP *is hard for* NISZK *under* BPP *reductions that make at most one query along any path of the* BPP *machine.*

Proof. This follows from the proof of Theorem 15. Namely, on input $x = (C_x, k)$, construct the string y consisting of t random samples from C_x and query the oracle on (y,θ). On Yes-instances, y will have KT complexity greater than θ (with high probability), and on No-instances, y will have KT complexity less than θ (with high probability). ◀

▶ **Corollary 17.** MKTP *is hard for* SZK *under non-adaptive* BPP*-Turing reductions.*

Proof. Recall from [18] that SZK reduces to Promise-EA via non-adaptive (deterministic) reductions. The result is now immediate, from Corollary 16. ◀

4 A Complete Problem for $\mathsf{NISZK_L}$

Having established a hardness result for MKTP under $\leq_m^{\mathsf{P/poly}}$ reductions, we now establish an analogous hardness result under the much more restrictive $\leq_m^{\mathsf{proj}}$ reductions. For this, we first need to present a complete problem for $\mathsf{NISZK_L}$.

Recall that the NISZK-complete problem EA deals with the question of approximating the entropy of a distribution represented by a circuit. In order to talk about $\mathsf{NISZK_L}$, we shall need to consider probability distributions that are represented using restricted class of circuits. In particular, we shall focus on a problem that we denote $\mathsf{EA_{NC^0}}$.

▶ **Definition 18** (Promise-$\mathsf{EA_{NC^0}}$)**.** *Promise-*$\mathsf{EA_{NC^0}}$ *is the promise problem obtained from Promise-*EA*, by considering only instances* (C, k) *such that* C *is a circuit of fan-in two gates, where no output gate depends on more than* four *input gates.*

It is not surprising that $\mathsf{EA_{NC^0}}$ is complete for $\mathsf{NISZK_L}$. The completeness proof that we present owes much to the proof presented by Dvir et al. [15] (showing that an NC^0-variant of the SZK-complete problem EntropyDifference is complete for $\mathsf{SZK_L}$) and to the proof presented by Goldreich et al. [18] showing that EA is complete for NISZK. We will need to make use of various detailed aspects of the constructions presented in this prior work, and thus we will present the full details here.

First, we show membership in $\mathsf{NISZK_L}$.

4.1 Membership in $\mathsf{NISZK_L}$

▶ **Theorem 19.** *Promise-*$\mathsf{EA_{NC^0}} \in \mathsf{NISZK_L}$

The following corollary is a direct analog to [18, Proposition 1].

▶ **Corollary 20.** *If* Π *is any promise problem that is* $\leq_m^{\mathsf{L}}$ *reducible to* $\mathsf{EA_{NC^0}}$*, then* $\Pi \in \mathsf{NISZK_L}$.

We close this section by presenting an example of a well-studied natural problem in $\mathsf{NISZK_L}$. (A graph is said to be *rigid* if it has no nontrivial automorphism.)

▶ **Corollary 21.** *The Non-Isomorphism Problem for Rigid Graphs lies in* $\mathsf{NISZK_L}$

Proof. First note that the proof of Theorem 19 carries over to show that a problem that we may call $\mathsf{EA_{BP}}$ (defined just as $\mathsf{EA_{NC^0}}$ but where the distribution is represented as a branching program instead of as an NC^0 circuit) also lies in $\mathsf{NISZK_L}$. Now observe that the reduction given in Section 3.1 of [7] shows how to take as input two rigid graphs on n vertices (G_0, G_1) and build a branching program that takes as input a bitstring w of length t and t permutations $\pi_1, \ldots, \pi_t$ and output the sequence of t permuted graphs $\pi_i(G_{w_i})$. It is observed in [7] that this distribution has entropy $t(1+\log n!)$ if the graphs are non-isomorphic, and has entropy at most $t \log n!$ otherwise. ◀

4.2 Hardness for $\mathsf{NISZK_L}$

In order to re-use the tools developed in [18], we will follow the structure of the proof given there, showing that EA is hard for NISZK. Namely, we introduce the problem SDU (Statistical Distance from Uniform) and its NC^0 variant, and prove hardness for $\mathsf{SDU}_{\mathsf{NC}^0}$.

▶ **Definition 22** (SDU and $\mathsf{SDU}_{\mathsf{NC}^0}$). *Consider Boolean circuits* $C_X : \{0,1\}^m \to \{0,1\}^n$ *representing distributions* X. *The promise problem*

$$\mathsf{SDU} = (\mathsf{SDU}_{YES}, \mathsf{SDU}_{NO})$$

is given by

$$\mathsf{SDU}_{YES} \stackrel{def}{=} \{C_X : \Delta(X, U_n) < 1/n\}$$
$$\mathsf{SDU}_{NO} \stackrel{def}{=} \{C_X : \Delta(X, U_n) > 1 - 1/n\}$$

where $\Delta(X,Y) = \Sigma_\alpha |\Pr[X=\alpha] - \Pr[Y=\alpha]|/2$.

$\mathsf{SDU}_{\mathsf{NC}^0}$ *is the analogous problem, where the distributions* X *are represented by* NC^0 *circuits where no output bit depends on more than* four *input bits.*

It is shown in [18, Lemma 4.1] that C_X is in SDU if and only if $(C_X, n-3)$ is in EA. This yields the following corollary:

▶ **Corollary 23.** $\mathsf{SDU}_{\mathsf{NC}^0} \leq_{\mathrm{m}}^{\mathsf{proj}} \mathsf{EA}_{\mathsf{NC}^0}$.

Proof. This is trivial if we assume an encoding of $\mathsf{SDU}_{\mathsf{NC}^0}$ instances, such that the NC^0 circuits $C_X : \{0,1\}^m \mapsto \{0,1\}^n$ are encoded by strings of length given by the standard pairing function $\frac{m^2+3m+2mn+n+n^2}{2}$, so that the length of an instance of $\mathsf{SDU}_{\mathsf{NC}^0}$ completely determines n. (An NC^0 circuit with m inputs and n outputs has a description of size $O(n \log m)$, and thus it is easy to devise a padded encoding of this much larger length.) Thus, in the projection circuit computing the reduction $C_X \mapsto (C_X, n-3)$, the output bits encoding $n-3$ are hardwired to constants, and the input bits encoding C_X are copied directly to the output. ◀

▶ **Theorem 24.** *Promise-*$\mathsf{EA}_{\mathsf{NC}^0}$ *and Promise-*$\mathsf{SDU}_{\mathsf{NC}^0}$ *are hard for* $\mathsf{NISZK_L}$ *under* $\leq_{\mathrm{m}}^{\mathsf{proj}}$ *reductions.*

Proof. By Corollary 23, it suffices to show hardness for $\mathsf{SDU}_{\mathsf{NC}^0}$. In order to establish hardness, we need to develop the machinery of *perfect randomized encodings*, which were developed by Applebaum et al. [12] and then were applied in the setting of $\mathsf{SZK_L}$ by Dvir et al. [15]. Due to space limitations, we refer the reader to [6] for the discussion of perfect randomized encodings.

4.2.1 $\mathsf{SDU}_{\mathsf{NC}^0}$ is Complete for $\mathsf{NISZK_L}$

We now have all of the tools required to complete the proof of Theorem 24.

Let $\prod$ be an arbitrary promise problem in $\mathsf{NISZK_L}$ with proof system (P,V) and simulator S and let x be an instance of $\prod$. Recall that the job of the simulator S is to take a string x and some uniformly-generated random bits as input, and produce as output a transcript of the form (σ, p), such that the probability that any transcript (σ, p) is output by S is very close to the probability that, on input x with shared randomness σ, the prover P sends message p to the verifier V. Let $M_x(s)$ denote a routine that simulates $S(x)$ with randomness s to obtain a transcript (σ, p); if $V(x, \sigma, p)$ accepts, then $M_x(s)$ outputs σ, otherwise it outputs $0^{|\sigma|}$. (We can assume without loss of generality that $|\sigma| = |x|^k$.) It is shown in [18, Lemma 4.2] that the map $x \mapsto M_x$ is a reduction of Π to SDU:

▷ Claim 25. If $x \in \prod_{YES}$, then $\Delta(M_x, U_{|x|^k}) < 1/|x|^k$, and $x \in \prod_{NO}$ implies $\Delta(M_x, U_{|x|^k})) > 1 - 1/|x|^k$.

The proof of the preceding claim in [18, Lemma 4.2] actually shows a stronger result. It shows that, if the statistical difference between the output distribution of the simulator and the distribution of true transcripts is at most $1/e(n)$, then the statistical difference of M_x and the uniform distribution is at most $1/e(n) + 2^{-n}$ on inputs of length n. Thus, using Definition 1 (which is equivalent to the definition of NISZK given in [18]), the simulator produces a distribution that differs from the uniform distribution by only $1/n^{\omega(1)}$. Thus we have the following claim:

▷ Claim 26. Let $c \in \mathbb{N}$. Then for all large x, If $x \in \prod_{YES}$, then $\Delta(M_x, U_{|x|^k}) < 1/|x|^{kc}$, and $x \in \prod_{NO}$ implies $\Delta(M_x, U_{|x|^k})) > 1 - 1/|x|^{kc}$.

Furthermore, it is also shown in [18, Lemma 3.1] that EA has a NISZK protocol in which the completeness error, soundness error, and simulator deviation are all at most 2^{-m} on inputs of length m. Furthermore, that proof carries over to show that $\mathsf{EA}_{\mathsf{BP}} \in \mathsf{NISZK}_{\mathsf{L}}$ with these same parameters. Thus we obtain the following fact, which we will use later in Section 6.

▷ Claim 27. Let $c \in \mathbb{N}$. Then for all large x, If x is a Yes-instance of $\mathsf{EA}_{\mathsf{BP}}$, then $\Delta(M_x, U_{|x|^k}) < 1/2^{|x|-1}$, and if x is a No-instance of $\mathsf{EA}_{\mathsf{BP}}$, then $\Delta(M_x, U_{|x|^k})) > 1-1/2^{|x|-1}$.

Since S runs in logspace, each bit of $M_x(s)$ can be simulated with a branching program Q_x. Furthermore, it is straightforward to see that there is an AC^0-computable function that takes x as input and produces an encoding of Q_x as output, and it can even be seen that this function can be a *projection*. (To see this, note that in the standard construction of a Turing machine from a logspace-bounded Turing machine S (with input (x, s)) each node of the branching program has a name that encodes a configuration of the machine, a time step, and the position of the input head. This branching program can be constructed in AC^0, given only the *length* of x. In order to construct Q_x, it suffices merely to hardwire in the transitions from any node that is "scanning" some bit position x_i. That is, if the transition out of node v goes to node v_0 if $x_i = 0$ and to node v_1 if $x_i = 1$, then in the adjacency matrix for Q_x, entry $(v, v_1) = x_i$ and entry (v, v_0) is $\neg x_i$. This is clearly a projection.)

Now apply the projection of [6, Lemma 37] to (each output bit of) the branching program Q_x of size ℓ, to obtain an NC^0 circuit C_x computing a perfect randomized encoding with blowup $b = 2^{|x|^k(\binom{\ell}{2}-1)(2(\ell-1)^2-1)}$. ($C_x$ has $\log b + |x|^k$ output bits.)

Now consider $|H(C_x) - H(U_{\log b+|x|^k})|$. By [6, Lemma 28] this is equal to $|H(Q_x) + \log b - H(U_{\log b+|x|^k})|$. Since $H(Q_x) = H(M_x)$ and $H(U_{\log b+|x|^k}) = \log b + H(U_{|x|^k})$, we have that $|H(C_x) - H(U_{\log b+|x|^k})| = |H(M_x) - H(U_{|x^k|})|$. The proof of Theorem 24 is now complete, by appeal to Claim 26. ◀

5 Hardness of MKTP under Projections

▶ **Theorem 28.** MKTP *is hard for* co-$\mathsf{NISZK}_{\mathsf{L}}$ *under nonuniform* $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ *reductions.*

An immediate corollary (making use of the "Gap Theorem" of [1]) is that MKTP is hard for co-$\mathsf{NISZK}_{\mathsf{L}}$ under $\leq_{\mathrm{m}}^{\mathsf{NC}^0}$ reductions. Below, we improve this, showing hardness under projections.

▶ **Corollary 29.** MKTP *is hard for* co-$\mathsf{NISZK}_{\mathsf{L}}$ *under nonuniform* $\leq_{\mathrm{m}}^{\mathsf{NC}^0}$ *reductions.*

▶ **Corollary 30.** MKTP *is hard for* co-$\mathsf{NISZK_L}$ *under nonuniform* $\leq_m^{\mathsf{proj}}$ *reductions.*

Proof. We now need to claim that the AC^0-computable reduction of Theorem 28 can be replaced by a projection. Note that, since $\mathsf{SDU}_{\mathsf{NC}^0}$ is complete for $\mathsf{NISZK_L}$ under projections, and since the reduction from $\mathsf{SDU}_{\mathsf{NC}^0}$ to $\mathsf{EA}_{\mathsf{NC}^0}$ given in Corollary 23 always uses the same entropy bound $n-3$, we have that it suffices to consider $\mathsf{EA}_{\mathsf{NC}^0}$ instances $x = (C_x, k)$ where the bound k depends only on the *length* of x. Thus the bound θ produced by our AC^0 reduction also depends only on the length of x, and hence can be hardwired in.

We now need only consider the string y. The $\leq_m^{\mathsf{AC}^0}$ reduction presented in the proof of Theorem 28 takes as input C_x and r and produces the bits of y by feeding bits of r into C_x. Let us recall where the NC^0 circuitry producing the i-th bit of y comes from.

[6, Lemma35] shows how to take an arbitrary branching program and encode the problem of whether the program accepts as a question about the entropy of a distribution represented as a matrix of degree-three polynomials. Each term in this matrix is of the form $\sum_{j,k} R_{1\,(i,k)} L_{(k,j)} R_{2\,(j,m)}$, where the matrices R_1 and R_2 are the same for all inputs of of the same length. Thus we need only concern ourselves with the matrix L.

In Section 4.2.1, it is observed that, given an instance x of a promise problem in $\mathsf{NISZK_L}$, the branching program Q_x that is used, in order to build the matrix L, can be constructed from x by means of a projection. The "input" to this branching program Q_x is a sequence of random bits (part of the random sequence r that is hardwired in, in order to create the nonuniform AC^0 reduction in the proof of Theorem 28). Thus, the only entries of the matrix L that depend on x are entries of the form (u, v) where u and v are configurations of a logspace machine, where the machine is scanning x_i in configuration u, and it is possible to move to configuration v. [6, Lemma 37] then shows how to construct NC^0 circuitry for each term in the degree-three polynomial constructed from Q_x in the proof of [6, Lemma 35]. The important thing to notice here is that each output bit in the NC^0 circuit depends on at most one term of one of the degree-three polynomials, and hence it depends on at most one entry of the matrix L – which means that it depends on at most one bit of the string x.

Thus, consider any bit y_i of the string y produced by the nonuniform AC^0 reduction from Theorem 28. Either y_i does not depend on any bit of x, or it depends on exactly one bit x_j of x. In the latter case, either $y_i = x_j$ or $y_i = \neg x_j$. This defines the projection, as required. ◀

The following corollary was pointed out to us by Rahul Santhanam.

▶ **Corollary 31.** MKTP *does not have* THRESHOLD∘MAJORITY *circuits of size* $2^{n^{o(1)}}$.

Proof. This is immediate from the lower bound on the Inner Product mod 2 function that is presented in [16]. (See also [11] for a slightly stronger lower bound.) ◀

It should be noted that it remains unknown whether MCSP has THRESHOLD∘MAJORITY circuits of *polynomial size*.

6 An Application: Average-Case Complexity

The efficient reductions that we have presented have some immediate applications regarding worst-case to average-case reductions. First, we recall the definition of errorless heuristics:

▶ **Definition 32.** *Let A be any language. An* errorless heuristic *for A is an algorithm (or oracle) H such that, for every x, $H(x) \in \{$Yes, No, ?$\}$, and*

- $H(x) =$ Yes *implies* $x \in A$.
- $H(x) =$ No *implies* $x \notin A$.

▶ **Definition 33.** *A language A* has no average-case errorless heuristics in $\mathcal{C}$ *if, for every polynomial p, and every errorless heuristic $H \in \mathcal{C}$ for A, there exist infinitely many n such where* $\Pr_{x \in U_n}[H(x) =?] > 1 - 1/p(n)$.

In order to state our first theorem relating to average-case complexity, we need the following circuit-based definition:

▶ **Definition 34.** *Let $\mathcal{C}$ be any complexity class. (Usually, we will think of $\mathcal{C}$ being a class defined in terms of circuits, and the definition is thus phrased in terms of circuits, although it can be adapted for other complexity classes as well.) The class* $\mathsf{OR} \circ \mathcal{C}$ *is the class of problems that can be solved by a family of circuits whose output gate is an unbounded fan-in* OR *gate, connected to the outputs of circuits in the class $\mathcal{C}$.*

If problems in $\mathsf{NISZK_L}$ are hard in the worst case, then there are problems in NP that are hard on average:

▶ **Theorem 35.** *Let $\mathcal{C}$ be any complexity class that is closed under $\leq_m^{\mathsf{proj}}$ reductions. If* $\mathsf{NISZK_L} \not\subseteq \mathsf{OR} \circ \mathcal{C}$, *then there is a set A in* NP *that has no average-case errorless heuristics in $\mathcal{C}$.*

The following definition is implicit in the work of Bogdanov and Trevisan [14].

▶ **Definition 36.** *A* worst-case to errorless average-case reduction *from a promise problem* $\prod$ *to a language A is given by a polynomial p and* BPP *machine M, such that A is accepted by M^H for every oracle errorless heuristic H for A such that* $\Pr_{x \in U_n}[H(x) =?] < 1 - 1/p(n)$.

▶ **Corollary 37.** *There is a problem $A \in$* NP *such that there is a non-adaptive worst-case to errorless average-case reduction from every problem in* SZK *to A.*

▶ Remark. It is implicitly shown by Hirahara [20] that Corollary 37 holds under *adaptive* reductions. The significance of the improvement from adaptive and non-adaptive reductions lies in the fact that Bogdanov and Trevisan showed that the problems in NP for which there is a non-adaptive worst-case to errorless average-case reduction to a problem in NP lie in $\mathsf{NP/poly} \cap \mathsf{coNP/poly}$ [14, Remark (iii) in Section 4]. Thus SZK may be close to the largest class of problems for which non-adaptive worst-case to errorless average-case reductions to problems in NP exist.

The worst-case to average-case reductions of Definition 36, must work for *every* errorless heuristic that has a sufficiently small probability of producing "?" as output. If we consider a less-restrictive notion (allowing a different reduction for different errorless heuristics) then in some cases we can lower the complexity of the reduction from BPP to AC^0.

▶ **Definition 38.** *Let $\mathcal{D}$ be a complexity class, and let $\mathcal{R}$ be a class of reducibilities. We say that* errorless heuristics for language A are average-case hard for $\mathcal{D}$ under $\mathcal{R}$ reductions *if, for every polynomial p and every errorless heuristic H for A where* $\Pr_{x \in U_{|x|}}[H(x) =?] < 1 - 1/p(|x|)$, *and for every language $B \in \mathcal{D}$, there is a reduction $r \in \mathcal{R}$ reducing B to H.*

▶ **Corollary 39.** *There is a language $A \in$* NP, *such that errorless heuristics for A are average-case hard for* $\mathsf{SZK_L}$ *under non-adaptive* AC^0*-Turing reductions.*

▶ **Corollary 40.** *Let $\mathcal{C}$ be any class that is closed under non-adaptive* AC^0*-Turing reductions. If* $\mathsf{SZK_L} \not\subset \mathcal{C}$, *then there is a problem in* NP *that has no average-case errorless heuristic in $\mathcal{C}$.*

Proof. If $\mathsf{SZK_L} \not\subset \mathcal{C}$, then by Proposition 3, $\mathsf{NISZK_L}$ is also not contained in $\mathcal{C}$. The result is now immediate from Theorem 35. ◀

▶ Remark. Building on earlier work of Goldwasser et al. [19], average-case hardness results for some subclasses of P based on reductions computable by constant-depth threshold circuits were presented in [2]. (Although certain aspects of the reductions presented in [2, 19] are computable in AC^0, in order to obtain deterministic worst-case algorithms, MAJORITY gates are required in those constructions.) We are not aware of any prior work that provides average-case hardness results based on reductions computable in AC^0, particularly for classes that are believed to contain problems whose complexity is suitable for cryptographic applications.

7 Conclusion and Open Problems

By focusing on non-uniform versions of $\leq_m^{\mathsf{P}}$ reductions, we have shed additional light on how MKTP relates to subclasses of SZK. Some researchers are of the opinion that MCSP (and MKTP) are likely complete for NP under some type of reducibility, and some recent progress seems to support this [25]. For those who share this opinion, a plausible first step would be to show that MKTP is hard not only for co-NISZK, but also for NISZK, under $\leq_m^{\mathsf{P/poly}}$ reductions. (Work by Lovett and Zhang points out obstacles to providing such a reduction via "black box" techniques [29].) And of course, it will be very interesting to see if our hardness results for MKTP hold also for MCSP.

References

1 Manindra Agrawal, Eric Allender, and Steven Rudich. Reductions in circuit complexity: An isomorphism theorem and a gap theorem. *Journal of Computer and System Sciences*, 57(2):127–143, 1998.

2 Eric Allender, V Arvind, Rahul Santhanam, and Fengming Wang. Uniform derandomization from pathetic lower bounds. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370(1971):3512–3535, 2012. doi:10.1098/rsta.2011.0318.

3 Eric Allender, Azucena Garvia Bosshard, and Amulya Musipatla. A note on hardness under projections for graph isomorphism and time-bounded Kolmogorov complexity. Technical Report TR20-158, Electronic Colloquium on Computational Complexity (ECCC), 2020.

4 Eric Allender, Harry Buhrman, Michal Kouckỳ, Dieter Van Melkebeek, and Detlef Ronneburger. Power from random strings. *SIAM Journal on Computing*, 35(6):1467–1493, 2006. doi:10.1007/978-3-662-03927-4.

5 Eric Allender and Bireswar Das. Zero knowledge and circuit minimization. *Information and Computation*, 256:2–8, 2017. Special issue for MFCS '14. doi:10.1016/j.ic.2017.04.004.

6 Eric Allender, John Gouwar, Shuichi Hirahara, and Caleb Robelle. Cryptographic hardness under projections for time-bounded Kolmogorov complexity. Technical Report TR21-010, Electronic Colloquium on Computational Complexity (ECCC), 2021.

7 Eric Allender, Joshua A Grochow, Dieter Van Melkebeek, Cristopher Moore, and Andrew Morgan. Minimum circuit size, graph isomorphism, and related problems. *SIAM Journal on Computing*, 47(4):1339–1372, 2018. doi:10.1137/17M1157970.

8 Eric Allender and Shuichi Hirahara. New insights on the (non-) hardness of circuit minimization and related problems. *ACM Transactions on Computation Theory*, 11(4):1–27, 2019. doi:10.1145/3349616.

9 Eric Allender, Dhiraj Holden, and Valentine Kabanets. The minimum oracle circuit size problem. *Computational Complexity*, 26(2):469–496, 2017. doi:10.1007/s00037-016-0124-0.

10 Eric Allender, Rahul Ilango, and Neekon Vafa. The non-hardness of approximating circuit size. *Theory of Computing Systems*, 65(3):559–578, 2021. doi:10.1007/s00224-020-10004-x.

11 Kazuyuki Amano. On the size of depth-two threshold circuits for the inner product mod 2 function. In Alberto Leporati, Carlos Martín-Vide, Dana Shapira, and Claudio Zandron, editors, *Language and Automata Theory and Applications – 14th International Conference (LATA)*, volume 12038 of *Lecture Notes in Computer Science*, pages 235–247. Springer, 2020. doi:10.1007/978-3-030-40608-0_16.

12 Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0. *SIAM Journal on Computing*, 36(4):845–888, 2006. doi:10.1137/S0097539705446950.

13 Manuel Blum, Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Noninteractive zero-knowledge. *SIAM Journal on Computing*, 20(6):1084–1118, 1991. doi:10.1137/0220068.

14 Andrej Bogdanov and Luca Trevisan. On worst-case to average-case reductions for NP problems. *SIAM J. Comput.*, 36(4):1119–1159, 2006. doi:10.1137/S0097539705446974.

15 Zeev Dvir, Dan Gutfreund, Guy N Rothblum, and Salil P Vadhan. On approximating the entropy of polynomial mappings. In *Second Symposium on Innovations in Computer Science*, 2011.

16 Jürgen Forster, Matthias Krause, Satyanarayana V. Lokam, Rustam Mubarakzjanov, Niels Schmitt, and Hans Ulrich Simon. Relations between communication complexity, linear arrangements, and computational complexity. In *Proc. 21st Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 2245 of *Lecture Notes in Computer Science*, pages 171–182. Springer, 2001. doi:10.1007/3-540-45294-X_15.

17 Bin Fu. Hardness of sparse sets and minimal circuit size problem. In *Proc. Computing and Combinatorics – 26th International Conference (COCOON)*, volume 12273 of *Lecture Notes in Computer Science*, pages 484–495. Springer, 2020. doi:10.1007/978-3-030-58150-3_39.

18 Oded Goldreich, Amit Sahai, and Salil Vadhan. Can statistical zero knowledge be made non-interactive? or On the relationship of SZK and NISZK. In *Annual International Cryptology Conference*, pages 467–484. Springer, 1999. doi:10.1007/3-540-48405-1_30.

19 Shafi Goldwasser, Dan Gutfreund, Alexander Healy, Tali Kaufman, and Guy N. Rothblum. A (de)constructive approach to program checking. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 143–152. ACM, 2008. doi:10.1145/1374376.1374399.

20 Shuichi Hirahara. Non-black-box worst-case to average-case reductions within NP. In *59th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 247–258. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00032.

21 Shuichi Hirahara. Non-disjoint promise problems from meta-computational view of pseudorandom generator constructions. In *35th Computational Complexity Conference (CCC)*, volume 169 of *LIPIcs*, pages 20:1–20:47. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CCC.2020.20.

22 Shuichi Hirahara and Rahul Santhanam. On the average-case complexity of MCSP and its variants. In *32nd Conference on Computational Complexity (CCC)*, volume 79 of *LIPIcs*, pages 7:1–7:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.CCC.2017.7.

23 Shuichi Hirahara and Osamu Watanabe. Limits of minimum circuit size problem as oracle. In *31st Conference on Computational Complexity (CCC)*, volume 50 of *LIPIcs*, pages 18:1–18:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.CCC.2016.18.

24 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)*, volume 45 of *LIPIcs*, pages 236–245. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.FSTTCS.2015.236.

25 Rahul Ilango, Bruno Loff, and Igor Carboni Oliveira. NP-hardness of circuit minimization for multi-output functions. In *35th Computational Complexity Conference (CCC)*, volume 169 of *LIPIcs*, pages 22:1–22:36. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CCC.2020.22.

26 Valentine Kabanets and Jin-yi Cai. Circuit minimization problem. In *Proceedings of the Thirty-Second Symposium on Theory of Computing (STOC)*, pages 73–79, 2000. doi:10.1145/335305.335314.

27 Valentine Kabanets, Daniel M. Kane, and Zhenjian Lu. A polynomial restriction lemma with applications. In *Proceedings of the 49th Annual Symposium on Theory of Computing (STOC)*, pages 615–628. ACM, 2017. doi:10.1145/3055399.3055470.

28 Leonid A. Levin. Randomness conservation inequalities; information and independence in mathematical theories. *Information and Control*, 61(1):15–37, 1984. doi:10.1016/S0019-9958(84)80060-1.

29 Shachar Lovett and Jiapeng Zhang. On the impossibility of entropy reversal, and its application to zero-knowledge proofs. In *Theory of Cryptography – 15th International Conference (TCC)*, volume 10677 of *Lecture Notes in Computer Science*, pages 31–55. Springer, 2017. doi:10.1007/978-3-319-70500-2_2.

30 Dylan M. McKay, Cody D. Murray, and R. Ryan Williams. Weak lower bounds on resource-bounded compression imply strong separations of complexity classes. In *Proceedings of the 51st Annual Symposium on Theory of Computing (STOC)*, pages 1215–1225, 2019. doi:10.1145/3313276.3316396.

31 Cody Murray and Ryan Williams. On the (non) NP-hardness of computing circuit complexity. *Theory of Computing*, 13(4):1–22, 2017. doi:10.4086/toc.2017.v013a004.

32 Igor Carboni Oliveira, Ján Pich, and Rahul Santhanam. Hardness magnification near state-of-the-art lower bounds. In *34th Computational Complexity Conference (CCC)*, volume 137 of *LIPIcs*, pages 27:1–27:29. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.CCC.2019.27.

33 Michael Rudow. Discrete logarithm and minimum circuit size. *Information Processing Letters*, 128:1–4, 2017. doi:10.1016/j.ipl.2017.07.005.

34 Michael Saks and Rahul Santhanam. Circuit lower bounds from NP-hardness of MCSP under Turing reductions. In *35th Computational Complexity Conference (CCC)*, volume 169 of *LIPIcs*, pages 26:1–26:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CCC.2020.26.

35 Heribert Vollmer. *Introduction to circuit complexity: a uniform approach*. Springer Science & Business Media, 1999.