

Cryptographic Hardness under Projections for Time-Bounded Kolmogorov Complexity

Eric Allender 

Rutgers University, NJ, USA

<http://www.cs.rutgers.edu/~allender>

allender@cs.rutgers.edu

John Gouwar

Northeastern University, Boston, USA

gouwar.j@northeastern.edu

Shuichi Hirahara

National Institute of Informatics, Japan

<https://researchmap.jp/shuichi.hirahara/?lang=english>

s_hirahara@nii.ac.jp

Caleb Robelle

MIT, Boston, USA

robelle@mit.edu

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{MKTP} is hard for the (apparently larger) class $NISZK_L$ under not only $\leq_m^{NC^0}$ reductions but even under projections. Also \overline{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 Complexity Classes; Problems, reductions and completeness; 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.CVIT.2016.23

Funding *Eric Allender*: Supported in part by NSF Grants CCF-1909216 and CCF-1909683.



© Eric Allender, John Gouwar, Shuichi Hirahara, and Caleb Robelle; licensed under Creative Commons License CC-BY

42nd Conference on Very Important Topics (CVIT 2016).

Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:26



Leibniz International Proceedings in Informatics

LIPIcs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

37 **1 Introduction**

38 The study of time-bounded Kolmogorov complexity is tightly connected to the study of
 39 circuit complexity. Indeed, the measure that we study most closely in this paper, denoted
 40 KT , was initially defined in order to capitalize on the framework of Kolmogorov complexity in
 41 investigations of the Minimum Circuit Size Problem (MCSP) [4]. If f is a bit string of length
 42 2^k representing the truth-table of a k -ary Boolean function, then $\text{KT}(f)$ is polynomially
 43 related to the size of the smallest circuit computing f . Thus the problem of computing KT
 44 complexity (denoted MKTP) was initially viewed as a more-or-less equivalent encoding of
 45 MCSP, and it is still the case that all theorems that have been proved about the complexity
 46 of MCSP hold also for MKTP (such as those in [5, 8, 9, 16, 20–23, 29, 30, 32, 34]).

47 In recent years, however, a few hardness results were proved for MKTP that are not yet
 48 known to hold for MCSP [6, 7]. We believe that these results can be taken as an indication
 49 of what is likely to be true also for MCSP. The present work gives significantly improved
 50 hardness results for MKTP .

51 Reducibility and completeness are the most effective tools in the arsenal of complexity
 52 theory for giving evidence of intractability. However, it is not clear whether MCSP or MKTP
 53 is NP -complete; neither can be shown to be NP -complete — or even hard for ZPP — under
 54 the usual \leq_m^{P} reductions without first showing that $\text{EXP} \neq \text{ZPP}$, a long-standing open
 55 problem [16, 30].

56 The strongest hardness results that have been proved thus far for MCSP and MKTP are
 57 that both are hard for SZK under BPP -Turing reductions [5]. SZK is the class of problems
 58 that have Statistical Zero Knowledge Interactive Proofs, and contains many problems of
 59 interest to cryptographers. Indeed, if MCSP (or MKTP) is in P/poly , then there are no
 60 cryptographically-secure one-way functions [25].

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

65 SZK is not known to be contained in NP ; until such a containment can be established,
 66 there is no hope of improving the BPP -Turing reduction of [5] to a \leq_m^{P} reduction. But
 67 we come close in this paper. NISZK is the “non-interactive” subclass of SZK ; it contains
 68 intractable problems if and only if SZK does [17]. We show that $\overline{\text{MKTP}}$ is hard for NISZK
 69 under $\leq_m^{\text{P/poly}}$ reductions. (Thus, instead of asking many queries, as in [5], a single query
 70 suffices.¹) Our proof also shows that MKTP is hard for NISZK under BPP reductions that
 71 ask only one query. Combined with [17], this shows that MKTP is hard for SZK under
 72 *non-adaptive* BPP reductions, yielding a modest improvement over [5]; this has implications
 73 regarding the study of worst-case to average-case reductions. (See Section 1.1.)

74 But $\leq_m^{\text{P/poly}}$ reductions are still quite powerful. There is great interest currently in
 75 proving lower bounds for MCSP, MKTP , and related problems such as MKtP (the problem
 76 of computing a different kind of time-bounded Kolmogorov complexity, due to Levin [27]) on
 77 very limited classes of circuits and formulae, as part of the “hardness magnification” program.
 78 For instance, if modest lower bounds can be shown on the size required to compute MKtP
 79 on de Morgan formulae augmented with PARITY gates at the leaves, then EXP is not

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_m^{P} reductions. We do not. In fact, some of us doubt that hardness under uniform deterministic reductions holds.

80 contained in non-uniform NC^1 [31]. Also, there is great interest in finding lower bounds
 81 against a variety of other models, such as depth-three threshold gates, or circuits consisting
 82 of polynomial threshold gates [26]. If a lower bound is known against one of these limited
 83 classes of circuits for some problem A that is reducible to, say, MKTP or MKtP under $\leq_m^{\text{P/poly}}$
 84 reductions, it implies nothing about the complexity of MKTP or MKtP, since the circuitry
 85 involved in computing the reduction is much more powerful than the circuitry in the class of
 86 circuits for which the lower bound is known.

87 Thus there is a great deal of interest in considering reductions that are much less powerful
 88 than $\leq_m^{\text{P/poly}}$ reductions. For extremely weak (uniform) notions of reducibility (such as
 89 log-time reductions), it is known that MCSP and MKTP are *not* hard for any complexity
 90 class that contains the PARITY function [30]. However, this non-hardness result relies
 91 on uniformity; it was later shown that MKTP is hard for the complexity class DET under
 92 *nonuniform* $\leq_m^{\text{NC}^0}$ reductions [7].

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

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

101 Prior to our work, the result of [7] showing that MKTP is hard for DET under $\leq_m^{\text{NC}^0}$
 102 reductions was improved, to show that MKTP is hard for DET even under projections [3].
 103 Since DET is a subclass of P, this provides little ammunition when one is seeking to prove
 104 that MKTP is intractable. One of our main contributions is to show that $\overline{\text{MKTP}}$ is hard for
 105 NISZK_L under projections. As a corollary, we obtain that MKTP cannot be computed by
 106 $\text{THRESHOLD} \circ \text{MAJORITY}$ circuits of size $2^{n^{o(1)}}$. This lower bound relies on the fact that
 107 MKTP is hard under projections.

108 The reader will not be familiar with NISZK_L ; this complexity class makes its first ap-
 109 pearance in the literature here. It is the “non-interactive” counterpart to the complexity
 110 class SZK_L that was studied previously by Dvir et al. [14], and was shown there to contain
 111 several important natural problems of interest to cryptographers (such as Discrete Log and
 112 Decisional Diffie-Hellman). NISZK_L contains intractable problems if and only if SZK_L does
 113 (see Section 2). Thus, for the first time, we show that MKTP is hard under projections for
 114 a complexity class that is widely believed to contain intractable problems. Our hardness
 115 results carry over immediately to MKtP and to similar problems defined in terms of general
 116 Kolmogorov complexity; no hardness results under projections had been known previously
 117 for those problems. We present some complete problems for NISZK_L and establish some
 118 other basic facts about this class in Section 4.

119 1.1 Average-Case Complexity

120 Building on the techniques introduced in [19], we are able to establish new insights regarding
 121 the relationship between worst-case and average-case complexity. In Theorem 48, capitalizing
 122 on the fact that essentially every circuit complexity class \mathcal{C} is closed under projections, we
 123 show that if NISZK_L does not lie in $\text{OR} \circ \mathcal{C}$, then there are problems A in NP that cannot
 124 be solved *in the average case* by errorless heuristics in \mathcal{C} . For instance, if one were able
 125 to show that there is *any* problem NISZK_L (including, but not limited to, some of the
 126 candidate one-way functions believed to reside there) that cannot be solved *in the worst*

127 *case* by depth-four ACC^0 circuits, it would follow that there are problems in NP that are
 128 hard-on-average for depth-three ACC^0 circuits. Such conclusions would *not* follow if our
 129 reductions to MKTP had merely been computable in AC^0 or NC^0 .

130 We are also able to shed more light on worst-case to average-case reductions, in the form
 131 that they were studied by Bogdanov and Trevisan [13]. Bogdanov and Trevisan showed that
 132 there were severe limits on the complexity of problems whose worst-case complexity could
 133 be reduced to the average-case complexity of problems in NP via *non-adaptive* reductions;
 134 all such problems lie in $\text{NP/poly} \cap \text{coNP/poly}$. But it was not known how large this class of
 135 problems could be. Hirahara showed that every problem in SZK has an *adaptive* worst-case to
 136 average-case reduction to a problem in NP [19], but the upper bound of $\text{NP/poly} \cap \text{coNP/poly}$
 137 proved by Bogdanov and Trevisan does not apply for adaptive reductions. As a consequence
 138 of our Corollary 19, showing that MKTP is hard for SZK under nonadaptive BPP reductions,
 139 we are able to show (in Corollary 51) that the class identified by Bogdanov and Trevisan lies
 140 in the narrow range between SZK and $\text{NP/poly} \cap \text{coNP/poly}$.

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

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

154 **2 Preliminaries**

155 **2.1 Complexity Classes and Reducibilities**

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

164 *Branching programs* are a circuit-like model of computation that can be used to charac-
 165 terize logspace computation. A *branching program* is a directed acyclic graph with a single
 166 source and two sinks labeled 1 and 0, respectively. Each non-sink node in the graph is labeled
 167 with a variable in $\{x_1, \dots, x_n\}$ and has two edges leading out of it: one labeled 1 and one
 168 labeled 0. A branching program computes a Boolean function f on input $x = x_1 \dots x_n$ by
 169 first placing a pebble on the source node. At any time when the pebble is on a node v labeled
 170 x_i , the pebble is moved to the (unique) vertex u that is reached by the edge labeled 1 if $x_i = 1$
 171 (or by the edge labeled 0 if $x_i = 0$). If the pebble eventually reaches the sink labeled b , then
 172 $f(x) = b$. Branching programs can also be used to compute functions $f : \{0, 1\}^m \rightarrow \{0, 1\}^n$,

173 by concatenating n branching programs p_1, \dots, p_n , where p_i computes the function $f_i(x) =$
 174 the i -th bit of $f(x)$. For more information on the definitions, backgrounds, and nuances of
 175 these complexity classes, circuits, and branching programs, see the text by Vollmer [35].

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

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

194 We will also make reference to various types of *Turing reducibility*, which are defined in
 195 terms of oracle Turing machines, or in terms of circuit families that are augmented with
 196 “oracle gates”. For instance, we say that $A \leq_T^{BPP} B$ if there is a probabilistic polynomial time
 197 oracle Turing machine M with oracle B that accepts every $x \in A$ with probability $\frac{2}{3}$ and
 198 rejects every $x \in \bar{A}$ with probability $\frac{2}{3}$. Note that the computation tree of such a BPP-Turing
 199 reduction can contain an exponential number of queries to different elements of B . Just as
 200 $BPP \subseteq P/poly$, it also holds that $A \leq_T^{BPP} B$ implies $A \leq_T^{P/poly} B$. Thus, on any input x , the
 201 circuit computing the P/poly-Turing reduction queries only a polynomial number of elements
 202 of B . It was shown in [5] that every problem in SZK (that is, every problem with a statistical
 203 zero knowledge proof system) is \leq_T^{BPP} -reducible (and hence $\leq_T^{P/poly}$ -reducible) to MCSP and
 204 to MKTP. The question of interest to us here is: Is it necessary to ask so many queries?
 205 What can we do if we ask only one query? What can be reduced to MKTP via a $\leq_m^{P/poly}$
 206 reduction?

207 The complexity class with which we are primarily concerned in this paper is the class of
 208 problems that have non-interactive statistical zero knowledge proof systems: NISZK. NISZK
 209 was originally defined and studied by Blum et al. [12]. The definition below (in terms of
 210 promise problems) is due to Goldreich et al. [17].

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

- 215 1. (Completeness:) For all $x \in \Pi_{YES}$, the probability that $V(x, \sigma, P(x, \sigma))$ accepts is at least
 216 $1 - 2^{-|x|}$.
- 217 2. (Soundness:) For all $x \in \Pi_{NO}$, the probability that $V(x, \sigma, P(x, \sigma))$ accepts is at most
 218 $2^{-|x|}$.
- 219 3. (Zero Knowledge:) For all $x \in \Pi_{YES}$, the statistical distance between the following two
 220 distributions bounded by $1/\beta(|x|)$

- 221 (A) Choose σ uniformly from $\{0,1\}^{r(|x|)}$, sample p from $P(x, \sigma)$, and output (p, σ) .
 222 (B) $S(x)$ (where the coins for S are chosen uniformly at random.)

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

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

227 NISZK is not known to be closed under complementation; co-NISZK is defined as the
 228 class of promise problems $\Pi = (\Pi_{YES}, \Pi_{NO})$ such that (Π_{NO}, Π_{YES}) is in NISZK. It is
 229 known that $SZK = NISZK$ iff $NISZK = \text{co-NISZK}$, and that every promise problem in SZK
 230 efficiently (and non-adaptively) Turing-reduces to a problem in NISZK [17]. Thus NISZK
 231 contains intractable problems if and only if SZK does.

232 A subclass of SZK, which we will denote by SZK_L , in which the verifier V and simulator
 233 S are restricted to being logspace machines, was defined and studied by Dvir et al. [14].
 234 Among other things, they showed that many of the important natural problems in SZK lie
 235 in SZK_L , including Graph Isomorphism, Quadratic Residuosity, Discrete Log, and Decisional
 236 Diffie-Helman. The non-interactive version of SZK_L , which we denote by $NISZK_L$, has not
 237 been studied previously, but it figures prominently in our results.

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

242 The reduction presented in [17] carries over directly to the logspace setting, showing that
 243 $NISZK_L$ contains intractable problems if and only if SZK_L does. In particular, we have:

244 **► Proposition 3.** *Every promise problem in SZK_L is non-adaptively AC^0 -Turing-reducible to*
 245 *a problem in $NISZK_L$.*

246 2.2 KT Complexity

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

248 **► Definition 4 (KT).** *Let U be a universal Turing machine. For each string x , define $KT_U(x)$*
 249 *to be*

$$250 \min\{|d| + T : (\forall \sigma \in \{0, 1, *\}) (\forall i \leq |x| + 1) U^d(i, \sigma) \text{ accepts in } T \text{ steps iff } x_i = \sigma\}$$

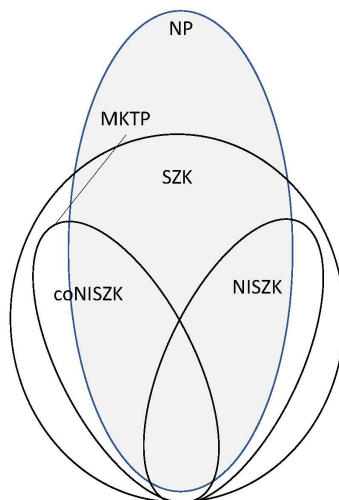
251 *We define $x_i = *$ if $i > |x|$; thus, for $i = |x| + 1$ the machine accepts iff $\sigma = *$. The notation*
 252 *U^d indicates that the machine U has random access to the description d .*

253 To understand the motivation for this definition, see [4]. Briefly: KT is a version of time-
 254 bounded Kolmogorov complexity that (in contrast to other notions of resource-bounded
 255 Kolmogorov complexity that have been considered) is polynomially-related to circuit com-
 256 plexity. 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*

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

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



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

2.3 Discrete Probability and Entropy

► Definition 6. Discrete Random Variables and Distributions

- 261 ■ A random variable $R : S \rightarrow T$ is a function where S is a finite set with a probability
- 262 distribution on its elements. We will refer to S as the sample space. R with a uniform
- 263 distribution on S will induce a distribution p on T .
- 264 ■ The support of a distribution is the set of elements in the distribution with positive
- 265 probability. Alternatively, the support of a random variable R can be understood as the
- 266 set $\text{Im}(R)$.
- 267 ■ In an abuse of notation, often given a distribution X , we will refer to X as both the
- 268 random variable that induces the distribution, and the distribution itself.
- 269 ■ Given a distribution X , we will use the notation X^k to denote the k -fold direct product
- 270 of X . Alternatively, this can be understood as the concatenation of k independent copies
- 271 of X .

272 Given a function $f : \{0, 1\}^m \rightarrow \{0, 1\}^n$ we write U_m to denote the uniform distribution

273 on m bits, and $f(U_m)$ for the output distribution of f when evaluated on a uniformly

274 chosen element of $\{0, 1\}^m$. Throughout this paper, our random variables, and in turn the

275 distributions they induce, will be of the form $C(U_m)$, where C is a multi-output Boolean

276 circuit $C : \{0, 1\}^m \rightarrow \{0, 1\}^n$.

277 The entropy of a distribution can be understood informally as measuring how much

278 “randomness” is present in the distribution.

- 279 ► **Definition 7.** Suppose X is a distribution. The Shannon entropy of X (denoted $H(X)$) is
- 280 the expected value of $\log(1/\Pr[X = x])$.

281 **3** $\overline{\text{MKTP}}$ is Hard For NISZK

282 In this section, we prove our first hardness result for MKTP; MKTP is hard for co-NISZK
 283 under $\leq_m^{\text{P/poly}}$ reductions. In order to prove hardness, it suffices to provide a reduction from
 284 the *entropy approximation* problem: EA, which is known to be complete for NISZK under
 285 \leq_m^{P} reductions [17].

► **Definition 8** (Promise-EA). *Let a circuit $C : \{0, 1\}^m \rightarrow \{0, 1\}^n$ 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*

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

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

287 We will make use of some machinery that was developed in [6], in order to relate the
 288 entropy of a distribution to the KT complexity of samples taken from the distribution.
 289 However, these tools are only useful when applied to distributions that are sufficiently “flat”.
 290 The next subsection provides the necessary tools to “flatten” a distribution.

291 **3.1 Flat Distributions**

292 A distribution is considered *flat* if it is uniform on its support. Goldreich et al. [17] formalized
 293 a relaxed notion of flatness, termed Δ -flatness, which relies on the concept of Δ -typical
 294 elements. The definitions of both concepts follow:

295 ► **Definition 9** (Δ -typical elements). *Suppose X is a distribution with element x in its support.*
 296 *We say that x is Δ -typical if,*

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

298 ► **Definition 10** (Δ -flatness). *Suppose X is a distribution. We say that X is Δ -flat if for*
 299 *every $t > 0$ the probability that an element of the support, x , is $t \cdot \Delta$ -typical is at least*
 300 $1 - 2^{-t^2+1}$.

301 ► **Lemma 11** (Flattening Lemma). *[17] Suppose X is a distribution such that for all x in*
 302 *its support $\Pr[X = x] \geq 2^{-m}$. Then X^k is $(\sqrt{k} \cdot m)$ -flat.*

303 Observe that if X is a distribution represented by a circuit $C : \{0, 1\}^m \rightarrow \{0, 1\}^n$, then the
 304 hypothesis of the Flattening Lemma holds for m . Note also that, for any distribution X ,
 305 $H(X^k) = k \cdot H(X)$. Thus the entropy of the distribution X^k grows linearly with respect to
 306 k , while the deviation from flatness diminishes much more rapidly with respect to k .

307 **3.2 Encoding and Blocking**

308 The *Encoding Lemma* is the primary tool that was developed in [6] to give short descriptions
 309 of samples from a given distribution. Below, we give a precise statement of the version
 310 of the Encoding Lemma that is stated informally as Remark 4.3 of [6]. (Although the
 311 statement there is informal, the proof of the Encoding Lemma that is given there does yield
 312 our Lemma 13.) First, we need to define Λ -encodings.

313 ► **Definition 12** (Λ -encodings). *Let $R : S \rightarrow T$ be a random variable that induces a distribution*
 314 *X . The Λ -heavy elements of T are those elements λ such that $\Pr[X = \lambda] > 1/2^\Lambda$. A Λ -*
 315 *encoding of R is given by a mapping $D : [N] \rightarrow S$ such that for every Λ -heavy element*
 316 *λ , there exists $i \in [N]$ such that $R(D(i)) = \lambda$. We refer to $\lceil \log(N) \rceil$ as the length of the*
 317 *encoding. The function D is also called the decoder for the encoding.*

318 ► **Lemma 13** (Encoding Lemma). *[6, Lemma 4.1] Consider an ensemble $\{R_x\}$ of random*
 319 *variables that sample distributions on strings of some length $\text{poly}_1(|x|)$, where there are*
 320 *circuits C_x of size $\text{poly}_2(|x|)$ representing each R_x . Then there is a polynomial poly_3 such*
 321 *that, for every integer Λ , each R_x has a Λ -encoding of length $\Lambda + \log(\Lambda) + O(1)$ that is*
 322 *decodable by circuits of size $\text{poly}_3(|x|)$.*

323 By itself, the Encoding Lemma says nothing about KT complexity. The other important
 324 ingredient in the toolbox developed in [6] is the *Blocking Lemma*, which refers to the process
 325 of chopping a string into blocks. Let y be a string of length tn , which we think of as being the
 326 concatenation of t samples y_i of a distribution X on strings of length n . Thus $y = y_1 \dots y_t$.
 327 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
 328 length bn sampled according to X^b . (In the case when $|y|$ is not a multiple of b , z_r is shorter;
 329 this does not affect the analysis. We call the strings z_i the *blocks* of y .)

330 ► **Lemma 14** (Blocking Lemma). *[6, Lemma 3.3] Let $\{T_x\}$ be an ensemble of sets of strings*
 331 *such that all strings in T_x have the same length $\text{poly}(|x|)$. Suppose that for each $x \in \{0, 1\}^*$*
 332 *and for each $b \in \mathbb{N}$ there is an integer Λ_b and a random variable $R_{x,b}$ whose image contains*
 333 *$(T_x)^b$, and such that $R_{x,b}$ is computable by a circuit of size $\text{poly}(|x|, b)$ and has a Λ_b -encoding*
 334 *of length $s'(x, b)$ decodable by a circuit of size $\text{poly}(|x|, b)$. Then there are constants c_1 and*
 335 *c_2 so that, for every constant $\alpha > 0$, every $t \in \mathbb{N}$, every sufficiently large x , and every*
 336 *$\lceil t^\alpha \rceil$ -suitable $y \in (T_x)^t$,*

$$337 \quad \text{KT}(y) \leq t^{1-\alpha} \cdot s'(x, \lceil t^\alpha \rceil) + t^{\alpha c_1} \cdot |x|^{c_2}.$$

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

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

342 ► **Lemma 15.** *Suppose X is a distribution sampled by a circuit $C_x : \{0, 1\}^m \rightarrow \{0, 1\}^n$ of*
 343 *size polynomial in $|x|$. For every polynomial $w = w(|x|)$ with $|x| \leq w$, there exist constants*
 344 *c_0, c_2 , and α_0 such that for every sufficiently large polynomial t and for all large x , if y is*
 345 *the concatenation of t samples from X , then with probability at least $(1 - 1/2^{2^{|x|}})$,*

$$346 \quad \text{KT}(y) \leq tH(X) + wm(t^{1-\alpha_0/2}) + t^{1-\alpha_0}|x|^{c_0+c_2}$$

347 **Proof.** Pick c_0 so that $|x|^{c_0} > m + wm + |x|$, and observe that for all large x we have
 348 $|x|^{c_0} > H(X) + wm + O(\log(|x|))$. Let $t = t(|x|)$ be any polynomial. Let $b \in \mathbb{N}$ with $b < t$,
 349 and let $\Lambda_b = bH(X) + wm\sqrt{b}$. Then, by the Encoding Lemma $X^b = \otimes^b X$ has a Λ_b -encoding
 350 of length $\Lambda_b + \log(\Lambda_b) + O(1)$ that is decodable by circuits of size $\text{poly}(b|x|)$. Let $r = \lceil t/b \rceil$.
 351 Recall that $y = y_1 \dots y_t$ where each y_i is a string of length n sampled according to the
 352 distribution X . Equivalently, we can consider y to be equal to $z_1 \dots z_r$ where each z_i is
 353 a string of length bn sampled according to X^b ; the strings z_i are the blocks of y . By the
 354 Flattening Lemma, the probability that any given z_b is not Λ_b -heavy is at most 2^{-w^2+1} .
 355 Thus, by the union bound, the probability that y is not b -suitable (i.e., the probability that

356 there is at least one block that is not Λ_b -heavy) is at most $r \cdot 2^{-w^2+1} < t \cdot 2^{-w^2}$. Since
 357 $w \geq |x|$ and t is polynomial in $|x|$, it follows that for all large x , with probability at least
 358 $(1 - 1/2^{2|x|})$, each of the r blocks is Λ_b -heavy and hence, by the Encoding Lemma, each block
 359 has an encoding of length $s'(n, b) = \Lambda_b + \log(\Lambda_b) + O(1)$. Thus, by the Blocking Lemma, for
 360 certain constants c_1 and c_2 (which do not depend on t), for any constant $\alpha > 0$ (for all large
 361 enough y),

$$\begin{aligned}
 362 \quad \text{KT}(y) &\leq t^{1-\alpha} \cdot s'(x, \lceil t^\alpha \rceil) + t^{\alpha \cdot c_1} \cdot |x|^{c_2} \\
 363 \quad &= t^{1-\alpha} \cdot (\Lambda_{\lceil t^\alpha \rceil} + \log(\Lambda_{\lceil t^\alpha \rceil}) + O(1)) + t^{\alpha \cdot c_1} \cdot |x|^{c_2} \\
 364 \quad &= t^{1-\alpha} \cdot (\lceil t^\alpha \rceil H(X) + wm\sqrt{\lceil t^\alpha \rceil} + \log(\Lambda_{\lceil t^\alpha \rceil}) + O(1)) + t^{\alpha \cdot c_1} \cdot |x|^{c_2} \\
 365 \quad &\leq t^{1-\alpha} \cdot (t^\alpha H(X) + |x|^{c_0} + wm\sqrt{t^\alpha}) + t^{\alpha \cdot c_1} \cdot |x|^{c_2}
 \end{aligned}$$

366

367

Recall that the inequality above holds for *all* $\alpha > 0$. If we now pick $\alpha_0 \leq 1/(1 + c_1)$, we
 obtain the claimed inequality

$$\text{KT}(y) \leq tH(x) + wmt^{1-\alpha_0/2} + t^{1-\alpha_0}(|x|^{c_0+c_2}).$$

368

◀

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

370 ▶ **Lemma 16.** *Let $\text{poly}(|x|)$ denote some fixed polynomial in $|x|$, and let α_0 be such that $0 <$
 371 $\alpha_0 < 1/2$. For all large x , if X is a distribution sampled by a circuit $C_x : \{0, 1\}^m \rightarrow \{0, 1\}^n$
 372 of polynomial size, then it holds that for every w and every $t > w^4$, if y is sampled from X^t ,
 373 then with probability at least $1 - 2^{-w^2}$,*

$$374 \quad \text{KT}(y) \geq tH(X) - wm\sqrt{t} - t^{1-\alpha_0} \text{poly}(|x|)$$

375 **Proof.** Consider the distribution $X^t = \otimes^t X$ and sample y from it. Recall that $H(X^t) =$
 376 $tH(x)$. By the Flattening Lemma, X^t is $\sqrt{t} \cdot m$ -flat. Therefore, the probability that y is
 377 $wm\sqrt{t}$ -typical is at least $1 - 2^{-w^2+1}$. We would like to bound the probability that $\text{KT}(y) <$
 378 $tH(X) - wm\sqrt{t} - t^{1-\alpha_0} \cdot \text{poly}(|x|)$. To bound this probability, note that $\Pr[\text{KT}(y) < k]$ is
 379 equal to

$$\begin{aligned}
 380 \quad &\Pr[\text{KT}(y) < k \wedge y \text{ is typical}] + \Pr[\text{KT}(y) < k \wedge y \text{ is atypical}] \\
 381 \quad &\leq \Pr[\text{KT}(y) < k \wedge y \text{ is typical}] + \Pr[y \text{ is atypical}]
 \end{aligned}$$

383 where we are interested in $k = tH(x) - wm\sqrt{t} - t^{1-\alpha_0} \cdot \text{poly}(|x|)$ and “ y is typical” means
 384 “ y is $wm\sqrt{t}$ -typical.” We have already observed above that the second term is bounded by
 385 2^{-w^2+1} . For the first term, we have

$$\begin{aligned}
 386 \quad \Pr[\text{KT}(y) < k \wedge y \text{ is typical}] &= \sum_{\{y: \text{KT}(y) < k \wedge y \text{ is typical}\}} \Pr(y) \\
 387 \quad &\leq \sum_{\{y: \text{KT}(y) < k \wedge y \text{ is typical}\}} 2^{wm\sqrt{t}} \cdot 2^{-H(X^t)} \\
 388 \quad &\leq 2^k \cdot 2^{wm\sqrt{t}} \cdot 2^{-H(X^t)} \\
 389 \quad &= 2^{tH(x) - wm\sqrt{t} - t^{1-\alpha_0} \cdot \text{poly}(|x|)} \cdot 2^{wm\sqrt{t}} \cdot 2^{-tH(X)} \\
 390 \quad &= 2^{-t^{1-\alpha_0} \cdot \text{poly}(|x|)}
 \end{aligned}$$

391

392

393 where the first inequality follows from the definition of typicality, and the second inequality
 394 follows since there are only $\sum_{i=0}^{k-1} 2^i < 2^k$ descriptions of strings with complexity less than k .

395 Summarizing, we conclude that the probability that $\text{KT}(y) < tH(x) - wm\sqrt{t} - t^{1-\alpha_0} \cdot$
 396 $\text{poly}(|x|)$ is at most

$$397 \quad 2^{-t^{1-\alpha_0} \cdot \text{poly}(|x|)} + 2^{-w^2+1}.$$

398 To show that the above probability is less than $1/2^{w^2}$ is equivalent to showing that

$$399 \quad 2^{-t^{1-\alpha_0} \cdot \text{poly}(|x|)} < 2^{-w^2+1}.$$

400 Thus we must show that $w^2 - 1 < t^{1-\alpha_0} \cdot \text{poly}(|x|)$. This holds, since

$$\begin{aligned} 401 \quad w^2 - 1 &< w^2 \\ 402 \quad &< (t^{1/4})^2 \\ 403 \quad &= \sqrt{t} \\ 404 \quad &\leq t^{1-\alpha_0} \\ 405 \quad &\leq t^{1-\alpha_0} \cdot \text{poly}(|x|). \end{aligned}$$

407

408 3.3 Reducing co-NISZK to MKTP

409 ► **Theorem 17.** *MKTP is hard for co-NISZK under P/poly many-one reductions.*

410 **Proof.** We prove the claim by reduction from the NISZK-complete problem EA. Let
 411 $x = (C_x, k)$ be an arbitrary instance of Promise-EA, where $C_x : \{0, 1\}^m \rightarrow \{0, 1\}^n$ is a circuit
 412 that represents distribution X . Let $w = 2|x|$, and let α_0, c_0 , and c_2 be the constants from
 413 Lemma 15. Let $\lambda = wmt^{1-\alpha_0}/2$. Pick the polynomial t so that $t(|x|) > 2(\lambda + t^{1-\alpha_0}|x|^{c_0+c_2})$
 414 and $w^4 < t$ (and note that all large polynomials have this property). Construct y as t samples
 415 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
 416 $(X, k) \in \text{EA}_{YES}$, then $(y, \theta) \in \text{MKTP}_{NO}$ and if $(X, k) \in \text{EA}_{NO}$, then $(y, \theta) \in \text{MKTP}_{YES}$.

417

418 If $(X, k) \in \text{EA}_{NO}$, then $H(X) < k$. Then by Lemma 15, we have that, with high
 419 probability,

$$\begin{aligned} 420 \quad \text{KT}(y) &\leq tH(X) + \lambda + t^{1-\alpha_0}|x|^{c_0+c_2} \\ 421 \quad &< tk + \lambda + t^{1-\alpha_0}|x|^{c_0+c_2} \\ 422 \quad &= \theta \end{aligned}$$

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

425 If $(X, k) \in \text{EA}_{YES}$, then $H(X) > k + 1$. Then by Lemma 16, with probability at least
 426 $1 - 2^{-w^2} > 1 - 2^{-2^{|x|}}$, we have that

$$\begin{aligned} 427 \quad \text{KT}(y) &\geq tH(X) - wm\sqrt{t} - t^{1-\alpha_0}|x|^{c_0+c_2}, \\ 428 \quad &> tH(X) - \lambda - t^{1-\alpha_0}|x|^{c_0+c_2} && \text{(since } \alpha_0 < 1/2) \\ 429 \quad &> t(k+1) - \lambda - t^{1-\alpha_0}|x|^{c_0+c_2} \\ 430 \quad &> tk + \lambda + t^{1-\alpha_0}|x|^{c_0+c_2} && \text{(since } t > 2(\lambda + t^{1-\alpha_0}|x|^{c_0+c_2})) \\ 431 \quad &= \theta \end{aligned}$$

432

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

434 We have shown that there is a polynomial-time-computable function f , such that, if
 435 $x \in \text{EA}_{YES}$, then with high probability (for random r) $f(x, r) = (y, \theta)$ is in MKTP_{NO} , and
 436 if $x \in \text{EA}_{NO}$, then with high probability $f(x, r) = (y, \theta)$ is in MKTP_{YES} . By a standard
 437 counting argument (similar to the proof that $\text{BPP} \subseteq \text{P/poly}$), since the probability of success
 438 for either bound is greater than $(1 - 1/2^{2^n})$, we can fix a sequence of random bits to hardwire
 439 in to this reduction and obtain a family of circuits computing a $\leq_m^{\text{P/poly}}$ reduction from any
 440 problem in NISZK to $\overline{\text{MKTP}}$. ◀

441 ▶ **Corollary 18.** *MKTP is hard for NISZK under BPP reductions that make at most one
 442 query along any path of the BPP machine.*

443 **Proof.** This follows from the proof of Theorem 17. Namely, on input $x = (C_x, k)$, construct
 444 the string y consisting of t random samples from C_x and query the oracle on (y, θ) . On
 445 Yes-instances, y will have KT complexity greater than θ (with high probability), and on
 446 No-instances, y will have KT complexity less than θ (with high probability). ◀

447 ▶ **Corollary 19.** *MKTP is hard for SZK under non-adaptive BPP-Turing reductions.*

448 **Proof.** Recall from [17] that SZK reduces to Promise-EA via non-adaptive (deterministic)
 449 reductions. The result is now immediate, from Corollary 18. ◀

450 4 A Complete Problem for NISZK_L

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

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

458 ▶ **Definition 20** ($\text{Promise-EA}_{\text{NC}^0}$). *Promise- EA_{NC^0} is the promise problem obtained from
 459 Promise-EA, by considering only instances (C, k) such that C is a circuit of fan-in two gates,
 460 where no output gate depends on more than four input gates.*

461 It is not surprising that EA_{NC^0} is complete for NISZK_L . The completeness proof that we
 462 present owes much to the proof presented by Dvir et al. [14] (showing that an NC^0 -variant of
 463 the SZK-complete problem ENTROPYDIFFERENCE is complete for SZK_L) and to the proof
 464 presented by Goldreich et al. [17] showing that EA is complete for NISZK . We will need to
 465 make use of various detailed aspects of the constructions presented in this prior work, and
 466 thus we will present the full details here.

467 First, we show membership in NISZK_L .

468 4.1 Membership in NISZK_L

469 ▶ **Theorem 21.** *Promise- $\text{EA}_{\text{NC}^0} \in \text{NISZK}_L$*

470 **Proof.** In order to show membership, we must show the existence of a non-interactive proof
 471 system where the verifier and simulator are both in logspace. To do this, we adapt the
 472 protocol that is used in [17] to show that EA is in NISZK . Their protocol works by first
 473 transforming an instance (C, k) of EA, of length s , (where C represents a distribution X)
 474 into a representation of a distribution Z on ℓ bits. The transformation consists of four steps:

- 475 1. Take $\text{poly}(s)$ samples from X and concatenate them. Call this distribution X' and let
476 $C_{X'}$ be the circuit representing X' .
- 477 2. Hash the output of X' with a hash function h chosen at random from a 2-universal family
478 of hash functions. (The parameters of the hash function depend on the value k of the EA
479 instance.) Let this distribution be Y , represented by C_Y .
- 480 3. Take $\text{poly}(s)$ copies of Y and concatenate their output. Call this distribution Y' , repre-
481 sented by $C_{Y'}$.
- 482 4. Hash a sample of Y' with a hash function h' chosen at random from a 2-universal family
483 of hash functions. Let this distribution be Z , represented by C_Z .

484 Section 2 and Appendix C of [17] give a careful proof of the fact that, with Z defined as
485 above from the EA instance (C, k) , a NISZK protocol for EA is given by:

- 486 1. With reference string σ , the prover selects a string r uniformly at random from the set
487 $\{r' : Z(r') = \sigma\}$.
- 488 2. The verifier accepts if $C_Z(r) = \sigma$ and rejects otherwise.

489 They also show that the following simulator satisfies the required zero-knowledge proper-
490 ties:

- 491 1. Select an input r to Z uniformly at random and let $\sigma = C_Z(r)$.
- 492 2. return (σ, r) .

493 It suffices for us to show that, if (C, k) is an instance of EA_{NC^0} , then the tasks of the
494 verifier and the simulator in the protocol above can be implemented in logspace.

495 First, we observe that, given (C, k) , a branching program P_Z realizing the distribution
496 Z can be constructed in logspace. Indeed, it is trivial to construct a branching program
497 P_X that realizes X (since each output bit of the NC^0 circuit Z has an easy-to-compute
498 branching program of constant size). Then a branching program $P_{X'}$ realizing X' consists
499 of several copies of P_X concatenated together (where each copy uses independent random
500 input bits). The hash functions h considered in [17] are represented by Boolean matrices
501 M_h , where computing $h(w)$ is simply multiplying M_h by the vector w . Since Boolean matrix
502 multiplication is easy to compute in $\text{NC}^1 \subseteq \text{L}$, and since the composition of two logspace-
503 computable functions is also logspace-computable, it is easy to build a branching program P_Y
504 representing the distribution Y (That is, given a branching program for computing $M_h \cdot w$,
505 for any node v that queries a bit of w , replace the pair of edges leaving v by a branching
506 program that computes that bit of w (as a sample from X')). In the same way, branching
507 programs for Y' and Z are easy to construct, given P_Y .

508 Hence a logspace verifier, with access to r, σ , and an EA_{NC^0} instance (C, k) , can construct
509 the branching program P_Z and compute $P_Z(r)$ and check if the output is equal to σ . It
510 is equally easy to see that the simulator can be implemented in logspace. This establishes
511 membership in NISZK_{L} . ◀

512 The following corollary is a direct analog to [17, Proposition 1].

513 ▶ **Corollary 22.** *If Π is any promise problem that is \leq_m^{L} reducible to EA_{NC^0} , then $\Pi \in \text{NISZK}_{\text{L}}$.*

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

516 ▶ **Corollary 23.** *The Non-Isomorphism Problem for Rigid Graphs lies in NISZK_{L}*

517 **Proof.** First note that the proof of Theorem 21 carries over to show that a problem that
 518 we may call EA_{BP} (defined just as EA_{NC^0} but where the distribution is represented as a
 519 branching program instead of as an NC^0 circuit) also lies in NISZK_L . Now observe that
 520 the reduction given in Section 3.1 of [6] shows how to take as input two rigid graphs on n
 521 vertices (G_0, G_1) and build a branching program that takes as input a bitstring w of length t
 522 and t permutations π_1, \dots, π_t and output the sequence of t permuted graphs $\pi_i(G_{w_i})$. It is
 523 observed in [6] that this distribution has entropy $t(1 + \log n!)$ if the graphs are non-isomorphic,
 524 and has entropy at most $t \log n!$ otherwise. ◀

525 4.2 Hardness for NISZK_L

526 In order to re-use the tools developed in [17], we will follow the structure of the proof
 527 given there, showing that EA is hard for NISZK . Namely, we introduce the problem SDU
 528 ($\text{STATISTICAL DISTANCE FROM UNIFORM}$) and its NC^0 variant, and prove hardness for
 529 SDU_{NC^0} .

▶ **Definition 24** (SDU and SDU_{NC^0}). Consider Boolean circuits $C_X : \{0, 1\}^m \rightarrow \{0, 1\}^n$ representing distributions X . The promise problem

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

530 is given by

$$\begin{aligned} 531 \quad \text{SDU}_{\text{YES}} &\stackrel{\text{def}}{=} \{C_X : \Delta(X, U_n) < 1/n\} \\ 532 \quad \text{SDU}_{\text{NO}} &\stackrel{\text{def}}{=} \{C_X : \Delta(X, U_n) > 1 - 1/n\} \end{aligned}$$

533 where $\Delta(X, Y) = \sum_{\alpha} |\Pr[X = \alpha] - \Pr[Y = \alpha]|/2$.

534 SDU_{NC^0} is the analogous problem, where the distributions X are represented by NC^0
 535 circuits where no output bit depends on more than four input bits.

536 It is shown in [17, Lemma 4.1] that C_X is in SDU if and only if $(C_X, n - 3)$ is in EA. This
 537 yields the following corollary:

538 ▶ **Corollary 25.** $\text{SDU}_{\text{NC}^0} \leq_m^{\text{proj}} \text{EA}_{\text{NC}^0}$.

539 **Proof.** This is trivial if we assume an encoding of SDU_{NC^0} instances, such that the NC^0
 540 circuits $C_X : \{0, 1\}^m \mapsto \{0, 1\}^n$ are encoded by strings of length given by the standard
 541 pairing function $\frac{m^2 + 3m + 2mn + n + n^2}{2}$, so that the length of an instance of SDU_{NC^0} completely
 542 determines n . (An NC^0 circuit with m inputs and n outputs has a description of size
 543 $O(n \log m)$, and thus it is easy to devise a padded encoding of this much larger length.)
 544 Thus, in the projection circuit computing the reduction $C_X \mapsto (C_X, n - 3)$, the output bits
 545 encoding $n - 3$ are hardwired to constants, and the input bits encoding C_X are copied directly
 546 to the output. ◀

547 ▶ **Theorem 26.** *Promise- EA_{NC^0} and Promise- SDU_{NC^0} are hard for NISZK_L under \leq_m^{proj}*
 548 *reductions.*

549 **Proof.** By Corollary 25, it suffices to show hardness for SDU_{NC^0} . In order to establish
 550 hardness, we need to develop the machinery of *perfect randomized encodings*, which were
 551 developed by Applebaum et al. [11] and then were applied in the setting of SZK_L by Dvir et
 552 al. [14].

4.2.1 Perfect Randomized Encodings

553

554 ► **Definition 27.** Let $f : \{0, 1\}^n \rightarrow \{0, 1\}^\ell$ be a function. We say that $\hat{f} : \{0, 1\}^n \times \{0, 1\}^m \rightarrow$
 555 $\{0, 1\}^s$ is a perfect randomized encoding of f with blowup b if it is:

- 556 ■ **Input independent:** for every $x, x' \in \{0, 1\}^n$ such that $f(x) = f(x')$, the random
 557 variables $\hat{f}(x, U_m)$ and $\hat{f}(x', U_m)$ are identically distributed.
- 558 ■ **Output Disjoint:** for every $x, x' \in \{0, 1\}^n$ such that $f(x) \neq f(x')$, $\text{Supp}(\hat{f}(x, U_m)) \cap$
 559 $\text{Supp}(\hat{f}(x', U_m)) = \emptyset$.
- 560 ■ **Uniform:** for every $x \in \{0, 1\}^n$ the random variable $\hat{f}(x, U_m)$ is uniform over $\text{Supp}(\hat{f}(x, U_m))$.
- 561 ■ **Balanced:** for every $x, x' \in \{0, 1\}^n$ $|\text{Supp}(\hat{f}(x, U_m))| = |\text{Supp}(\hat{f}(x', U_m))| = b$

562 The following property of perfect randomized encodings is established in [14].

563 ► **Lemma 28 (entropy).** Let $f : \{0, 1\}^n \rightarrow \{0, 1\}^\ell$ be a function and let $\hat{f} : \{0, 1\}^n \times$
 564 $\{0, 1\}^m \rightarrow \{0, 1\}^s$ be a perfect randomized encoding of f with blowup b . Then $H(\hat{f}(U_n, U_m)) =$
 565 $H(f(U_n)) + \log b$

566 The following two properties are given in Applebaum et al. [11].

567 ► **Lemma 29 (concatenation).** For $i = 1, \dots, \ell$ let $f_i : \{0, 1\}^n \rightarrow \{0, 1\}$ be the Boolean function
 568 computing the i -th bit of $f : \{0, 1\}^n \rightarrow \{0, 1\}^\ell$. If $\hat{f}_i : \{0, 1\}^n \times \{0, 1\}^{m_i} \rightarrow \{0, 1\}^{s_i}$ is a perfect
 569 randomized encoding of f_i , then the function $\hat{f} : \{0, 1\}^n \times \{0, 1\}^{m_1 + \dots + m_\ell} \rightarrow \{0, 1\}^{s_1 + \dots + s_\ell}$
 570 defined by $\hat{f}(x, (r_1, \dots, r_\ell)) \stackrel{\text{def}}{=} (\hat{f}_1(x, r_1), \dots, \hat{f}_\ell(x, r_\ell))$ is a perfect randomized encoding of
 571 f .

572 ► **Lemma 30 (composition).** Let $g(x, r_g)$ be a perfect randomized encoding of $f(x)$ and
 573 let $h((x, r_g), r_h)$ be a perfect randomized encoding of $g(x, r_g)$ (viewed as a single argument
 574 function). Then, the function $\hat{f}((x, r_g), r_h) \stackrel{\text{def}}{=} h((x, r_g), r_h)$ is a perfect randomized encoding
 575 of f .

576 The following claim is not formally stated in [11] but can be found in their discussion of
 577 perfect randomized encodings in section 4.1 of that paper.

578 ▷ **Claim 31.** Let $f : \{0, 1\}^n \rightarrow \{0, 1\}^\ell$ be a function. If $\hat{f} : \{0, 1\}^n \times \{0, 1\}^m \rightarrow \{0, 1\}^s$ is a
 579 perfect randomized encoding of f , then \hat{f} has blowup 2^m .

580 The following is apparent from Lemma 29, Lemma 30, and Claim 31.

581 ▷ **Claim 32.** The blowup of a perfect randomized encoding \hat{f} created by composing or
 582 concatenating perfect randomized encodings $\hat{f}_1, \dots, \hat{f}_\ell$ is $\prod_{i=1}^\ell b_i$.

4.2.2 Constructing an NC^0 perfect randomized encoding

583

584 The first step in showing completeness of EANC^0 is to use the following construction of perfect
 585 randomized encodings of functions computed by branching programs, from [11].

586 ► **Definition 33.** Let Q be a branching program of size ℓ computing a Boolean function
 587 $f : \{0, 1\}^n \rightarrow \{0, 1\}$. Fix some topological ordering of the vertices of Q where the source
 588 vertex is labelled 1 and the terminal vertex is labelled ℓ . Let $A(x)$ be the $\ell \times \ell$ adjacency
 589 matrix of G_x where entry (i, j) is a degree-1 polynomial $q_{i,j} \in \{x_k, (1 - x_k), 1, 0\}$, such that
 590 the transition from node i to node j queries variable x_k and proceeds if $q_{i,j}(x_k) = 1$. Define
 591 $L(x)$ as the submatrix of $A(x) - I$ obtained by deleting the first column and last row.

$$\begin{pmatrix} * & * & * & * & * \\ -1 & * & * & * & * \\ 0 & -1 & * & * & * \\ 0 & 0 & -1 & * & * \\ 0 & 0 & 0 & -1 & * \end{pmatrix}$$

Let $r^{(1)}$, and $r^{(2)}$ be vectors over $GF(2)$ of length $\binom{\ell-1}{2}$ and $\ell-2$ respectively. Let $R_1(r^{(1)})$ be an $\ell \times \ell$ matrix with 1's on the main diagonal, 0's below it and the elements of $r^{(1)}$ in the remaining $\binom{\ell-1}{2}$ entries above the main diagonal. Let $R_2(r^{(2)})$ be an $\ell \times \ell$ matrix with 1's on the main diagonal, 0's below it, and the elements of $r^{(2)}$ in the last column.

$$\begin{pmatrix} 1 & r_1^{(1)} & r_2^{(1)} & \cdot & r_{\ell-1}^{(1)} \\ 0 & 1 & \cdot & \cdot & \cdot \\ 0 & 0 & 1 & \cdot & \cdot \\ 0 & 0 & 0 & 1 & r_{\binom{\ell-1}{2}}^{(1)} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & r_1^{(2)} \\ 0 & 1 & 0 & 0 & \cdot \\ 0 & 0 & 1 & 0 & \cdot \\ 0 & 0 & 0 & 1 & r_{\ell-2}^{(2)} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The following lemma appears as [11, Lemma 4.15].

► **Lemma 34.** *Let Q be a branching program of size ℓ computing a Boolean function $f : \{0, 1\}^n \rightarrow \{0, 1\}$. Let the function $\hat{f}(x, (r^{(1)}, r^{(2)}))$ produce as output the $\binom{\ell}{2}$ entries on or above the main diagonal of the matrix*

$$R_1(r^{(1)})L(x)R_2(r^{(2)}).$$

Then \hat{f} is a perfect randomized encoding of f .

► **Lemma 35.** *There is a function computable in AC^0 (in fact, it can be a projection) that takes as input a branching program Q of size ℓ computing a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, and produces as output a list $(q_1, \dots, q_{\binom{\ell}{2}})$ of degree-three polynomials over $GF(2)$, where $q_i(x, (r^{(1)}, r^{(2)}))$ produces the i -th output bit of $\hat{f}(x, (r^{(1)}, r^{(2)}))$. The blowup of the encoding \hat{f} is $2^{\binom{\ell}{2}-1}$.*

Proof. Claim 31 establishes the claim regarding blowup. Constructing the three matrices $L(x)$, R_1 and R_2 can clearly be done in AC^0 . Their product cannot be computed in AC^0 (since this involves computing PARITY), but it is easy to compute an *encoding* of the expression for entry (i, m) of the product, which is given by the degree-three polynomial $\sum_{j,k} R_{1(i,k)} L_{(k,j)} R_{2(j,m)}$. To see that this can be a projection, note that the entries of the matrices R_1 and R_2 are entirely determined by the *size* ℓ (and thus they depend only on the length of the encoding of the branching program). The entries of $L(x)$ are essentially the entries of the adjacency matrix encoding the branching program Q , and thus they can be copied directly via a projection. Then, given the encodings of the matrices, the encodings of the terms of each polynomial q_i are simply copied from the encodings of the matrices, and thus this can be done via a projection also. ◀

Note that each polynomial q_i in the statement of the preceding lemma is most conveniently expressed as a sum of terms. We now show how to replace each q_i with NC^0 circuitry, using the following lemma from [11, Lemma 4.17].

619 ► **Lemma 36.** *Let $f(x) = T_1(x) + \dots + T_k(x)$ where $f, T_1, \dots, T_k : GF(2)^n \rightarrow GF(2)$, and*
 620 *summation is over $GF(2)$, and each term T_i has degree at most 3. Let the local encoding $\hat{f} :$*
 621 *$GF(2)^{n+(2k-1)} \rightarrow GF(2)^{2k}$ be such that $\hat{f}(x, (r_1, \dots, r_k, r'_1, \dots, r'_{k-1}))$ is equal to*

$$\begin{aligned} & (T_1(x) - r_1, T_2(x) - r_2, \dots, T_k(x) - r_k, \\ & r_1 - r'_1, r'_1 + r_2 - r'_2, \dots, r'_{k-2} + r_{k-1} - r'_{k-1}, r'_{k-1} + r_k) \end{aligned}$$

623 *Then \hat{f} is a perfect randomized encoding of f where each bit of the output depends on at most*
 624 *4 bits of $(x, (r_1, \dots, r_k, r'_1, \dots, r'_{k-1}))$.*

625 ► **Lemma 37.** *There is a function computable in AC^0 (in fact, it can be a projection) that*
 626 *takes as input a branching program Q of size ℓ computing a function $f : \{0, 1\}^n \rightarrow \{0, 1\}$,*
 627 *and produces as output a list p_i of NC^0 circuits, where p_i computes the i -th bit of a function*
 628 *\hat{f} that is a perfect randomized encoding of f that has blowup $2^{\binom{\ell}{2}-1(2(\ell-1)^2-1)}$. Each p_i*
 629 *depends on at most four input bits from (x, r) (where r is the sequence of random bits in the*
 630 *randomized encoding).*

631 **Proof.** This follows immediately by applying the construction of Lemma 36 to the degree-
 632 three polynomials for each entry in the product matrix given by AC^0 -computable function
 633 given by Lemma 35. Each of those polynomials has $(\ell - 1)^2$ terms, and it is apparent from
 634 Lemma 36 that each such entry gives rise to $2(\ell - 1)^2 - 1$ new random bits in the randomized
 635 encoding. The assertion regarding blowup now follows from Claim 31. The assertions
 636 regarding the bits upon which each p_i depends follows from inspection. The construction of
 637 Lemma 36 can clearly be accomplished via a projection, and composing that projection with
 638 the projection from Lemma 35 again yields a projection. ◀

639 4.2.3 SDU_{NC^0} is Complete for $NISZK_L$

640 We now have all of the tools required to complete the proof of Theorem 26.

641 Let Π be an arbitrary promise problem in $NISZK_L$ with proof system (P, V) and simulator
 642 S and let x be an instance of Π . Recall that the job of the simulator S is to take a string x
 643 and some uniformly-generated random bits as input, and produce as output a transcript of
 644 the form (σ, p) , such that the probability that any transcript (σ, p) is output by S is very close
 645 to the probability that, on input x with shared randomness σ , the prover P sends message
 646 p to the verifier V . Let $M_x(s)$ denote a routine that simulates $S(x)$ with randomness s to
 647 obtain a transcript (σ, p) ; if $V(x, \sigma, p)$ accepts, then $M_x(s)$ outputs σ , otherwise it outputs
 648 $0^{|\sigma|}$. (We can assume without loss of generality that $|\sigma| = |x|^k$.) It is shown in [17, Lemma
 649 4.2] that the map $x \mapsto M_x$ is a reduction of Π to SDU :

650 \triangleright **Claim 38.** *If $x \in \Pi_{YES}$, then $\Delta(M_x, U_{|x|^k}) < 1/|x|^k$, and $x \in \Pi_{NO}$ implies $\Delta(M_x, U_{|x|^k}) >$*
 651 *$1 - 1/|x|^k$.*

652 The proof of the preceding claim in [17, Lemma 4.2] actually shows a stronger result. It
 653 shows that, if the statistical difference between the output distribution of the simulator and
 654 the distribution of true transcripts is at most $1/e(n)$, then the statistical difference of M_x
 655 and the uniform distribution is at most $1/e(n) + 2^{-n}$ on inputs of length n . Thus, using
 656 Definition 1 (which is equivalent to the definition of $NISZK$ given in [17]), the simulator
 657 produces a distribution that differs from the uniform distribution by only $1/n^{\omega(1)}$. Thus we
 658 have the following claim:

659 \triangleright **Claim 39.** *Let $c \in \mathbb{N}$. Then for all large x , If $x \in \Pi_{YES}$, then $\Delta(M_x, U_{|x|^k}) < 1/|x|^{kc}$,*
 660 *and $x \in \Pi_{NO}$ implies $\Delta(M_x, U_{|x|^k}) > 1 - 1/|x|^{kc}$.*

661 Furthermore, it is also shown in [17, Lemma 3.1] that EA has a NISZK protocol in which
 662 the completeness error, soundness error, and simulator deviation are all at most 2^{-m} on
 663 inputs of length m . Furthermore, that proof carries over to show that $\text{EA}_{\text{BP}} \in \text{NISZK}_{\text{L}}$ with
 664 these same parameters. Thus we obtain the following fact, which we will use later in Section 6.
 665

666 \triangleright **Claim 40.** Let $c \in \mathbb{N}$. Then for all large x , If x is a Yes-instance of EA_{BP} , then
 667 $\Delta(M_x, U_{|x|^k}) < 1/2^{|x|^{-1}}$, and if x is a No-instance of EA_{BP} , then $\Delta(M_x, U_{|x|^k}) > 1 - 1/2^{|x|^{-1}}$.

668 Since S runs in logspace, each bit of $M_x(s)$ can be simulated with a branching program
 669 Q_x . Furthermore, it is straightforward to see that there is an AC^0 -computable function that
 670 takes x as input and produces an encoding of Q_x as output, and it can even be seen that
 671 this function can be a *projection*. (To see this, note that in the standard construction of a
 672 Turing machine from a logspace-bounded Turing machine S (with input (x, s)) each node
 673 of the branching program has a name that encodes a configuration of the machine, a time
 674 step, and the position of the input head. This branching program can be constructed in AC^0 ,
 675 given only the *length* of x . In order to construct Q_x , it suffices merely to hardwire in the
 676 transitions from any node that is “scanning” some bit position x_i . That is, if the transition
 677 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
 678 for Q_x , entry $(v, v_1) = x_i$ and entry (v, v_0) is $\neg x_i$. This is clearly a projection.)

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

682 Now consider $|H(C_x) - H(U_{\log b + |x|^k})|$. By Lemma 28 this is equal to $|H(Q_x) + \log b -$
 683 $H(U_{\log b + |x|^k})|$. Since $H(Q_x) = H(M_x)$ and $H(U_{\log b + |x|^k}) = \log b + H(U_{|x|^k})$, we have that
 684 $|H(C_x) - H(U_{\log b + |x|^k})| = |H(M_x) - H(U_{|x|^k})|$. The proof of Theorem 26 is now complete,
 685 by appeal to Claim 39. \blacktriangleleft

686 5 Hardness of MKTP under Projections

687 \blacktriangleright **Theorem 41.** MKTP is hard for $\text{co-NISZK}_{\text{L}}$ under nonuniform $\leq_{\text{m}}^{\text{AC}^0}$ reductions.

688 **Proof.** We build on the proof of Theorem 17, and present a reduction from the NISZK_{L} -
 689 complete problem EA_{NC^0} . Let $x = (C_x, k)$ be an arbitrary instance of Promise- EA_{NC^0} , where
 690 $C_x : \{0, 1\}^m \rightarrow \{0, 1\}^n$ is an NC^0 circuit that represents distribution X . Let $|x| < w < \sqrt[4]{t}$,
 691 and let α_0, c_0 , and c_2 be the constants from Lemma 15. Let $\lambda = wmt^{1-\alpha_0/2}$ and construct y
 692 as t samples from X . Let $\theta = tk + \lambda + t^{1-\alpha_0} |x|^{c_0+c_2}$.

693 As in the proof of Theorem 17, we have that, with probability at least $1 - \frac{1}{2^{|x|}}$, if (X, k)
 694 is a Yes-instance of EA_{NC^0} , then $(y, \theta) \in \text{MKTP}_{\text{NO}}$ and if (X, k) is a No-instance of EA_{NC^0} ,
 695 then $(y, \theta) \in \text{MKTP}_{\text{YES}}$.

696 Thus we have shown that there is a uniform AC^0 -computable function f , such that, if
 697 $x \in \text{EA}_{\text{YES}}$, then with high probability (for random r) $f(x, r) = (y, \theta)$ is in MKTP_{NO} , and
 698 if $x \in \text{EA}_{\text{NO}}$, then with high probability $f(x, r) = (y, \theta)$ is in MKTP_{YES} . (Namely, the AC^0
 699 function takes $x = (C_x, k)$ and r as input, computes θ from k and $|x|$, and computes y by
 700 feeding t segments of r into the NC^0 circuit C_x .)

701 As in the proof of Theorem 17, we can fix a sequence of random bits to hardwire in to
 702 this reduction and obtain a (nonuniform) $\leq_{\text{m}}^{\text{AC}^0}$ reduction from EA_{NC^0} to $\overline{\text{MKTP}}$.
 703 \blacktriangleleft

704 An immediate corollary (making use of the “Gap Theorem” of [1]) is that MKTP is hard
 705 for co-NISZK_L under $\leq_m^{\text{NC}^0}$ reductions. Below, we improve this, showing hardness under
 706 projections.

707 ► **Corollary 42.** *MKTP is hard for co-NISZK_L under nonuniform $\leq_m^{\text{NC}^0}$ reductions.*

708 **Proof.** Corollary 22, combined with the NISZK_L -completeness of EA_{NC^0} , implies that co-NISZK_L
 709 is closed under \leq_m^L reductions. It is shown in the “Gap Theorem” of [1] that, for any class \mathcal{C}
 710 closed under \leq_m^L reductions, any set that is hard for \mathcal{C} under $\leq_m^{\text{AC}^0}$ reductions is also hard
 711 under $\leq_m^{\text{NC}^0}$ reductions. Thus from Theorem 41, we have that MKTP is hard for co-NISZK_L
 712 under $\leq_m^{\text{NC}^0}$ reductions. ◀

713 ► **Corollary 43.** *MKTP is hard for co-NISZK_L under nonuniform \leq_m^{proj} reductions.*

714 **Proof.** We now need to claim that the AC^0 -computable reduction of Theorem 41 can be
 715 replaced by a projection. Note that, since SDU_{NC^0} is complete for NISZK_L under projections,
 716 and since the reduction from SDU_{NC^0} to EA_{NC^0} given in Corollary 25 always uses the same
 717 entropy bound $n - 3$, we have that it suffices to consider EA_{NC^0} instances $x = (C_x, k)$ where
 718 the bound k depends only on the *length* of x . Thus the bound θ produced by our AC^0
 719 reduction also depends only on the length of x , and hence can be hardwired in.

720 We now need only consider the string y . The $\leq_m^{\text{AC}^0}$ reduction presented in the proof of
 721 Theorem 41 takes as input C_x and r and produces the bits of y by feeding bits of r into C_x .
 722 Let us recall where the NC^0 circuitry producing the i -th bit of y comes from.

723 Lemma 35 shows how to take an arbitrary branching program and encode the problem
 724 of whether the program accepts as a question about the entropy of a distribution repre-
 725 sented as a matrix of degree-three polynomials. Each term in this matrix is of the form
 726 $\sum_{j,k} R_1(i,k) L(k,j) R_2(j,m)$, where the matrices R_1 and R_2 are the same for all inputs of of the
 727 same length. Thus we need only concern ourselves with the matrix L .

728 In Section 4.2.3, it is observed that, given an instance x of a promise problem in NISZK_L ,
 729 the branching program Q_x that is used, in order to build the matrix L , can be constructed
 730 from x by means of a projection. The “input” to this branching program Q_x is a sequence
 731 of random bits (part of the random sequence r that is hardwired in, in order to create the
 732 nonuniform AC^0 reduction in the proof of Theorem 41). Thus, the only entries of the matrix
 733 L that depend on x are entries of the form (u, v) where u and v are configurations of a
 734 logspace machine, where the machine is scanning x_i in configuration u , and it is possible
 735 to move to configuration v . Lemma 37 then shows how to construct NC^0 circuitry for each
 736 term in the degree-three polynomial constructed from Q_x in the proof of Lemma 35. The
 737 important thing to notice here is that each output bit in the NC^0 circuit depends on at most
 738 one term of one of the degree-three polynomials, and hence it depends on at most one entry
 739 of the matrix L – which means that it depends on at most one bit of the string x .

740 Thus, consider any bit y_i of the string y produced by the nonuniform AC^0 reduction from
 741 Theorem 41. Either y_i does not depend on any bit of x , or it depends on exactly one bit x_j of
 742 x . In the latter case, either $y_i = x_j$ or $y_i = \neg x_j$. This defines the projection, as required. ◀

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

744 ► **Corollary 44.** *MKTP does not have $\text{THRESHOLD} \circ \text{MAJORITY}$ circuits of size $2^{n^{o(1)}}$.*

745 **Proof.** This is immediate from the lower bound on the Inner Product mod 2 function that
 746 is presented in [15]. (See also [10] for a slightly stronger lower bound.) ◀

747 It should be noted that it remains unknown whether MCSP has $\text{THRESHOLD} \circ \text{MAJORITY}$
 748 circuits of *polynomial size*.

749 **6 An Application: Average-Case Complexity**

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

752 ► **Definition 45.** *Let A be any language. An errorless heuristic for A is an algorithm (or
 753 oracle) H such that, for every x , $H(x) \in \{\text{YES}, \text{NO}, ?\}$, and*

754 ■ $H(x) = \text{YES}$ implies $x \in A$.

755 ■ $H(x) = \text{NO}$ implies $x \notin A$.

756 ► **Definition 46.** *A language A has no average-case errorless heuristics in \mathcal{C} if, for every
 757 polynomial p , and every errorless heuristic $H \in \mathcal{C}$ for A , there exist infinitely many n such
 758 where $\Pr_{x \in U_n}[H(x) = ?] > 1 - 1/p(n)$.*

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

761 ► **Definition 47.** *Let \mathcal{C} be any complexity class. (Usually, we will think of \mathcal{C} being a class
 762 defined in terms of circuits, and the definition is thus phrased in terms of circuits, although it
 763 can be adapted for other complexity classes as well.) The class $\text{OR} \circ \mathcal{C}$ is the class of problems
 764 that can be solved by a family of circuits whose output gate is an unbounded fan-in OR gate,
 765 connected to the outputs of circuits in the class \mathcal{C} .*

766 If problems in NISZK_L are hard in the worst case, then there are problems in NP that are
 767 hard on average:

768 ► **Theorem 48.** *Let \mathcal{C} be any complexity class that is closed under \leq_m^{proj} reductions. If
 769 $\text{NISZK}_L \not\subseteq \text{OR} \circ \mathcal{C}$, then there is a set A in NP that has no average-case errorless heuristics
 770 in \mathcal{C} .*

771 **Proof.** Consider the reduction from $\overline{\text{EA}}_{\text{NC}^0}$ to MKTP given in the proof of Corollary 43. This
 772 reduction takes as input a pair $(C, n - 3)$ where C is an NC^0 circuit that produces output
 773 of length n . The reduction produces as output a string of length tn where $t = t(n)$ is a
 774 polynomial in n . The proof of Corollary 43 shows that, if $(C, n - 3)$ is a No-instance (a
 775 low-entropy instance) of EA_{NC^0} , then concatenating t samples from $C(r)$ (for independent
 776 uniformly random samples r) produces output that, with probability exponentially-close to
 777 1, has KT-complexity less than $\theta < (n - 2)t(n)$ for all large n . Let f be a function computed
 778 as follows: On input d of length m' , compute the smallest n such that $m' < (n - 2)t(n)$,
 779 and then simulate the universal Turing machine U on d for $t(n)^2$ steps, and produce as
 780 output the first $nt(n)$ bits of output that $U(d)$ produces in this amount of time. Let
 781 $A = \{y : \exists d f(d) = y\}$ be the range of f . Note that A contains all strings y of length $nt(n)$
 782 such that $\text{KT}(y) \leq (n - 2)t(n)$. Clearly, $A \in \text{NP}$. We will show that if A has an average-case
 783 errorless heuristic in \mathcal{C} , then $\text{NISZK}_L \in \text{OR} \circ \mathcal{C}$.²

784 If A has an average-case errorless heuristic in \mathcal{C} , then there is a family $\{C_m : m \in \mathbb{N}\}$ of
 785 \mathcal{C} circuits (or other algorithms, if \mathcal{C} is not a circuit family) with the property that, for all
 786 large n , for all strings x of length n , $C_n(x) \in \{\text{YES}, \text{NO}, ?\}$, where

787 ■ $C_n(x) = \text{YES}$ implies $x \in A$.

² In fact, A can be taken to be any set in NP that contains all strings of KT complexity below a certain threshold, while still containing only a small fraction of the strings of any length n .

- 788 ■ $C_n(x) = \text{NO}$ implies $x \notin A$.
- 789 ■ $\Pr_x[C_n(x) = ?] < 1 - \frac{1}{p_1(n)}$

790 for some polynomial p_1 .

791 Since there are three possible outputs, there must be two output bits, which we can call a
 792 and b . The encoding of YES, NO and ? below is chosen in order to simplify the statement of
 793 our results. If a different encoding is chosen, then the form of the circuits for NISZK_L might
 794 be slightly different.

a	b	
1	0	YES
0	1	NO
0	0	?
1	1	Illegal

796 Now consider the family $\{C'_m : m \in \mathbb{N}\}$, where C'_m is just like C_m but has only output
 797 bit b .

798 For any $m = nt(n)$,

$$\begin{aligned}
 799 \Pr_x[C'_m(x) = 1] &= 1 - \Pr[C_m(x) = \text{YES}] - \Pr[C_m(x) = ?] \\
 800 &\geq 1 - \frac{|A \cap \{0, 1\}^m|}{2^m} - \left(1 - \frac{1}{p_1(m)}\right) \\
 801 &\geq 1 - \frac{2^{(n-2)t(n)}}{2^{nt(n)}} - \left(1 - \frac{1}{p_1(m)}\right) \\
 802 &= \frac{1}{p_1(nt(n))} - \frac{1}{2^{2t(n)}} \\
 803 &> \frac{1}{p_2(n)}
 \end{aligned}$$

804
805

806 for some polynomial p_2 .

807 We now present efficient circuits for promise problems in NISZK_L .

808 Since the NISZK_L -complete problem EA_{NC^0} is a special case of EA_{BP} , we know that EA_{BP}
 809 is also complete for NISZK_L (say, under \leq_m^L reductions). Thus it follows from Claim 40
 810 that, for any problem $\Pi \in \text{NISZK}_L$, and for any instance $x \in \Pi_{\text{YES}}$, the distribution M_x
 811 introduced in Section 4.2.3 can actually be assumed to have statistical difference at most
 812 $1/2^{|x|^\epsilon}$ from the uniform distribution, for some $\epsilon > 0$. This in turn implies that the NC^0
 813 circuit C_x (which is constructed in the paragraphs right after Claim 40) also has statistical
 814 difference at most $1/2^{|x|^\epsilon}$ from the uniform distribution (again, if $x \in \Pi_{\text{YES}}$). We highlight
 815 this fact, so that we can refer to it more easily later:

816 \triangleright **Claim 49.** If $x \in \Pi_{\text{YES}}$, then the NC^0 circuit C_x has statistical difference at most $1/2^{|x|^\epsilon}$
 817 from the uniform distribution.

818 Now consider the circuit family $\{D_{n_0} : n_0 \in \mathbb{N}\}$ that has the following form: The input is
 819 a string x of length n_0 . Recall that the NC^0 circuit C_x from Section 4.2.3 takes “random”
 820 inputs r of length polynomial in $|x|$ and produces output of length n which is also polynomial
 821 in $|x|$. Let $\{E_n : n \in \mathbb{N}\}$ be a circuit family that takes (x, r) as input and computes $C_x(r)$.
 822 (The family E_n can in fact be chosen to be very efficient, but we do not need that; it will
 823 turn out later that E_n can be replaced by a single wire connected to a possibly-negated bit
 824 of x , or by a constant.) The “bottom layer” of D_{n_0} consists of $n_0^2 p_2^2(n) t(n)$ copies of E_n ,

825 using $n_0^2 p_2^2(n) t(n)$ independent random strings $r_1, \dots, r_{n_0^2 p_2^2(n) t(n)}$, and producing a string
 826 of length $n_0^2 p_2^2(n) t(n) n$, which is then fed into $n_0^2 p_2^2(n)$ copies of $C'_{t(n)n}$. Finally, the output
 827 gate of each of the copies of $C'_{t(n)n}$ is fed into an OR gate, which is the output gate of D_{n_0} .

828 If $x \in \prod_{NO}$ then, as in the proof of Theorem 41, with probability (over the random
 829 inputs) exponentially close to 1, the string feeding into the inputs of each of the copies of C'
 830 has low KT complexity, and consequently (by the definition of C') each C' outputs 0, and
 831 thus D_{n_0} outputs 0.

832 If $x \in \prod_{YES}$ then, by Claim 49, the distribution represented by each copy of E_n (using
 833 random inputs r) has statistical difference from the uniform distribution bounded by 2^{-n^ϵ} .
 834 The strings that are fed into each copy of $C'_{nt(n)}$ are generated by $t(n)$ independent copies of
 835 E_n . By [33, Lemma 3.4], we can conclude that the distribution that is fed into each copy of
 836 $C'_{nt(n)}$ has statistical distance from the uniform distribution bounded by $\frac{t(n)}{2^{n^\epsilon}}$. We showed
 837 above that the probability that $C'_{nt(n)}$ accepts a uniformly-random string of length $nt(n)$ is
 838 greater than $\frac{1}{p_2(n)}$. It follows that the probability that $C'_{nt(n)}$ accepts the string produced
 839 by $t(n)$ independent copies of E_n is no less than $\frac{1}{p_2(n)} - \frac{t(n)}{2^{n^\epsilon}} > \frac{1}{np_2(n)}$. Thus the probability
 840 that *none* of the $n_0^2 p_2^2(n)$ independent copies of $C'_{nt(n)}$ accepts is at most $2^{-n_0^2}$.

841 A simple counting argument now shows that there is a sequence of probabilistic bits r
 842 that can be hardwired in to D_{n_0} so that, for all x of length n_0 , $D_{n_0}(x, r) = 1$ if $x \in \prod_{YES}$
 843 and $D_{n_0}(x, r) = 0$ if $x \in \prod_{NO}$. It still remains to simplify D_{n_0} so that it lies in $\text{OR} \circ \mathcal{C}$.

844 As in the proof of Corollary 43, each bit that feeds into any of the copies of $C'_{nt(n)}$ depends
 845 on *at most one* bit of x ; many of the bits may be set to constants after hardwiring in the
 846 choice of r . Thus we build the circuit family F_{n_0} that takes x as input, and projects the bits
 847 of x into the $n_0^2 p_2^2(n)$ copies of $C'_{nt(n)}$, to obtain a $\text{OR} \circ \mathcal{C}$ circuit family for \prod . ◀

848 The following definition is implicit in the work of Bogdanov and Trevisan [13].

849 ▶ **Definition 50.** A worst-case to errorless average-case reduction from a promise problem \prod
 850 to a language A is given by a polynomial p and BPP machine M , such that A is accepted by
 851 M^H for every oracle errorless heuristic H for A such that $\Pr_{x \in U_n}[H(x) = ?] < 1 - 1/p(n)$.

852 ▶ **Corollary 51.** There is a problem $A \in \text{NP}$ such that there is a non-adaptive worst-case to
 853 errorless average-case reduction from every problem in SZK to A .

854 **Proof.** We mimic the proof of Theorem 48, and use the same set A . Consider the BPP
 855 reduction from the NISZK complete problem EA to MKTP given in Corollary 18. This
 856 reduction takes as input a pair (C, k) (where C is a circuit that produces output of length
 857 n) and makes a single query along each path, where the query is a string y of length tn
 858 where $t = t(n)$ is a polynomial in n . (Since SDU is complete for NISZK, we can assume
 859 that $k = n - 3$, as in the proof of Theorem 48.) Rather than using MKTP as an oracle,
 860 instead we will use an errorless heuristic H for A where the $\Pr_z[H(z) = ?] < 1 - 1/p(|z|)$,
 861 interpreting any answer where $H(y) = \text{“NO”}$ as meaning “ $\text{KT}(y) > \theta$ ” and any answer where
 862 $H(y) \in \{?, \text{YES}\}$ as meaning “ $\text{KT}(y) < \theta$ ”. (We will actually replace each query to MKTP by
 863 a polynomial number of independent queries to the heuristic H , and if *any* of these queries
 864 returns $H(y) = \text{“NO”}$, we will conclude that $(C, k) \in \text{EA}_{YES}$, and otherwise conclude that
 865 $(C, k) \in \text{EA}_{NO}$.)

866 As in the proof Theorem 48, if the distribution represented by C has low entropy, then
 867 with probability exponentially close to 1, the query y will have low KT complexity, and
 868 thus $H(y)$ will return a value in $\{?, \text{YES}\}$ (and this probability will remain small even if a
 869 polynomial number of independent trials are made). And if C has high entropy, then (as in
 870 the proof of Theorem 48) we can assume that the distribution given by C is exponentially

871 close to the uniform distribution, and thus the distribution on the queries y will have small
 872 statistical difference from the uniform distribution, and hence, with exponentially high
 873 probability, at least one of the queries will return the value NO. Thus every problem in
 874 NISZK has an errorless non-adaptive worst-case to average-case reduction to A .

875 The proof is completed by recalling from [17] that SZK is non-adaptively (deterministically)
 876 polynomial-time reducible to NISZK. ◀

877 **Remark:** It is implicitly shown by Hirahara [19] that Corollary 51 holds under *adaptive*
 878 reductions. The significance of the improvement from adaptive and non-adaptive reductions
 879 lies in the fact that Bogdanov and Trevisan showed that the problems in NP for which there
 880 is a non-adaptive worst-case to errorless average-case reduction to a problem in NP lie in
 881 $\text{NP/poly} \cap \text{coNP/poly}$ [13, Remark (iii) in Section 4]. Thus SZK may be close to the largest
 882 class of problems for which non-adaptive worst-case to errorless average-case reductions to
 883 problems in NP exist.

884 The worst-case to average-case reductions of Definition 50, must work for *every* errorless
 885 heuristic that has a sufficiently small probability of producing “?” as output. If we consider
 886 a less-restrictive notion (allowing a different reduction for different errorless heuristics) then
 887 in some cases we can lower the complexity of the reduction from BPP to AC^0 .

888 ▶ **Definition 52.** Let \mathcal{D} be a complexity class, and let \mathcal{R} be a class of reducibilities. We say that
 889 errorless heuristics for language A are average-case hard for \mathcal{D} under \mathcal{R} reductions if, for every
 890 polynomial p and every errorless heuristic H for A where $\Pr_{x \in U_{|x|}}[H(x) = ?] < 1 - 1/p(|x|)$,
 891 and for every language $B \in \mathcal{D}$, there is a reduction $r \in \mathcal{R}$ reducing B to H .

892 ▶ **Corollary 53.** There is a language $A \in \text{NP}$, such that errorless heuristics for A are
 893 average-case hard for SZK_L under non-adaptive AC^0 -Turing reductions.

894 **Proof.** The proof of Theorem 48 introduces a language $A \in \text{NP}$ that is defined in terms of
 895 the parameters of the reduction from the NISZK $_L$ -complete promise problem EA_{NCO} . We show
 896 that errorless heuristics for this same A are average-case hard for SZK_L under non-adaptive
 897 AC^0 -Turing reductions. By Proposition 3 and Theorem 26, every problem in SZK_L is non-
 898 adaptively AC^0 -Turing-reducible to EA_{NCO} ; let this AC^0 -Turing reduction be computed by the
 899 family $\{D_n : n \in \mathbb{N}\}$. In the proof of Theorem 48, if we take the circuit family $\{C_m : m \in \mathbb{N}\}$
 900 to consist of oracle gates for an errorless heuristic H for A , we obtain that every query that
 901 D_n makes to EA_{NCO} can be replaced by an OR of queries consisting of oracle gates from
 902 $\{C_m : m \in \mathbb{N}\}$. This yields the desired non-adaptive AC^0 -Turing reduction to the errorless
 903 heuristic for A . ◀

904 ▶ **Corollary 54.** Let \mathcal{C} be any class that is closed under non-adaptive AC^0 -Turing reductions.
 905 If $\text{SZK}_L \not\subseteq \mathcal{C}$, then there is a problem in NP that has no average-case errorless heuristic in \mathcal{C} .

906 **Proof.** If $\text{SZK}_L \not\subseteq \mathcal{C}$, then by Proposition 3, NISZK $_L$ is also not contained in \mathcal{C} . The result is
 907 now immediate from Theorem 48. ◀

908 **Remark:** Building on earlier work of Goldwasser et al. [18], average-case hardness results
 909 for some subclasses of P based on reductions computable by constant-depth threshold circuits
 910 were presented in [2]. (Although certain aspects of the reductions presented in [2, 18] are
 911 computable in AC^0 , in order to obtain deterministic worst-case algorithms, MAJORITY gates
 912 are required in those constructions.) We are not aware of any prior work that provides average-
 913 case hardness results based on reductions computable in AC^0 , particularly for classes that
 914 are believed to contain problems whose complexity is suitable for cryptographic applications.

915 **7 Conclusion and Open Problems**

916 By focusing on non-uniform versions of \leq_m^P reductions, we have shed additional light on
 917 how MKTP relates to subclasses of SZK. Some researchers are of the opinion that MCSP
 918 (and MKTP) are likely complete for NP under some type of reducibility, and some recent
 919 progress seems to support this [24]. For those who share this opinion, a plausible first step
 920 would be to show that MKTP is hard not only for co-NISZK, but also for NISZK, under
 921 $\leq_m^{P/\text{poly}}$ reductions. (Work by Lovett and Zhang points out obstacles to providing such a
 922 reduction via “black box” techniques [28].) And of course, it will be very interesting to see if
 923 our hardness results for MKTP hold also for MCSP.

924 **Acknowledgments**

925 EA was supported in part by NSF Grants CCF-1909216 and CCF-1909683. JG and CR
 926 were supported in part by the DIMACS 2020 REU program under NSF grants CCF-1852215
 927 and CCF-1836666. We thank Rahul Santhanam, Oded Goldreich, Salil Vadhan, Harsha
 928 Tirumala, Dieter van Melkebeek and Andrew Morgan for helpful discussions. We also thank
 929 the anonymous reviewers who provided helpful comments.

930 **References**

- 931 **1** Manindra Agrawal, Eric Allender, and Steven Rudich. Reductions in circuit complexity:
 932 An isomorphism theorem and a gap theorem. *Journal of Computer and System Sciences*,
 933 57(2):127–143, 1998.
- 934 **2** Eric Allender, V Arvind, Rahul Santhanam, and Fengming Wang. Uniform derandomization
 935 from pathetic lower bounds. *Philosophical Transactions of the Royal Society A: Mathematical,*
 936 *Physical and Engineering Sciences*, 370(1971):3512–3535, 2012. doi:10.1098/rsta.2011.0318.
- 937 **3** Eric Allender, Azucena Garvia Bosshard, and Amulya Musipatla. A note on hardness under
 938 projections for graph isomorphism and time-bounded Kolmogorov complexity. Technical
 939 Report TR20-158, Electronic Colloquium on Computational Complexity (ECCC), 2020.
- 940 **4** Eric Allender, Harry Buhrman, Michal Koucký, Dieter Van Melkebeek, and Detlef Ronneburger.
 941 Power from random strings. *SIAM Journal on Computing*, 35(6):1467–1493, 2006. doi:
 942 10.1007/978-3-662-03927-4.
- 943 **5** Eric Allender and Bireswar Das. Zero knowledge and circuit minimization. *Information and*
 944 *Computation*, 256:2–8, 2017. Special issue for MFCS ’14. doi:10.1016/j.ic.2017.04.004.
- 945 **6** Eric Allender, Joshua A Grochow, Dieter Van Melkebeek, Christopher Moore, and Andrew
 946 Morgan. Minimum circuit size, graph isomorphism, and related problems. *SIAM Journal on*
 947 *Computing*, 47(4):1339–1372, 2018. doi:10.1137/17M1157970.
- 948 **7** Eric Allender and Shuichi Hirahara. New insights on the (non-) hardness of circuit minimization
 949 and related problems. *ACM Transactions on Computation Theory*, 11(4):1–27, 2019. doi:
 950 10.1145/3349616.
- 951 **8** Eric Allender, Dhiraj Holden, and Valentine Kabanets. The minimum oracle circuit size
 952 problem. *Computational Complexity*, 26(2):469–496, 2017. doi:10.1007/s00037-016-0124-0.
- 953 **9** Eric Allender, Rahul Ilango, and Neekon Vafa. The non-hardness of approximating circuit size.
 954 *Theory of Computing Systems*, 65(3):559–578, 2021. doi:10.1007/s00224-020-10004-x.
- 955 **10** Kazuyuki Amano. On the size of depth-two threshold circuits for the inner product mod
 956 2 function. In Alberto Leporati, Carlos Martín-Vide, Dana Shapira, and Claudio Zandron,
 957 editors, *Language and Automata Theory and Applications - 14th International Conference*
 958 *(LATA)*, volume 12038 of *Lecture Notes in Computer Science*, pages 235–247. Springer, 2020.
 959 doi:10.1007/978-3-030-40608-0_16.

- 960 11 Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0 . *SIAM Journal*
961 *on Computing*, 36(4):845–888, 2006. doi:10.1137/S0097539705446950.
- 962 12 Manuel Blum, Alfredo De Santis, Silvio Micali, and Giuseppe Persiano. Noninteractive
963 zero-knowledge. *SIAM Journal on Computing*, 20(6):1084–1118, 1991. doi:10.1137/0220068.
- 964 13 Andrej Bogdanov and Luca Trevisan. On worst-case to average-case reductions for NP
965 problems. *SIAM J. Comput.*, 36(4):1119–1159, 2006. doi:10.1137/S0097539705446974.
- 966 14 Zeev Dvir, Dan Gutfreund, Guy N Rothblum, and Salil P Vadhan. On approximating the
967 entropy of polynomial mappings. In *Second Symposium on Innovations in Computer Science*,
968 2011.
- 969 15 Jürgen Forster, Matthias Krause, Satyanarayana V. Lokam, Rustam Mubarakzjanov, Niels
970 Schmitt, and Hans Ulrich Simon. Relations between communication complexity, linear
971 arrangements, and computational complexity. In *Proc. 21st Foundations of Software Technology*
972 *and Theoretical Computer Science (FSTTCS)*, volume 2245 of *Lecture Notes in Computer*
973 *Science*, pages 171–182. Springer, 2001. doi:10.1007/3-540-45294-X_15.
- 974 16 Bin Fu. Hardness of sparse sets and minimal circuit size problem. In *Proc. Computing and*
975 *Combinatorics - 26th International Conference (COCOON)*, volume 12273 of *Lecture Notes in*
976 *Computer Science*, pages 484–495. Springer, 2020. doi:10.1007/978-3-030-58150-3_39.
- 977 17 Oded Goldreich, Amit Sahai, and Salil Vadhan. Can statistical zero knowledge be made
978 non-interactive? or On the relationship of SZK and NISZK. In *Annual International Cryptology*
979 *Conference*, pages 467–484. Springer, 1999. doi:10.1007/3-540-48405-1_30.
- 980 18 Shafi Goldwasser, Dan Gutfreund, Alexander Healy, Tali Kaufman, and Guy N. Rothblum.
981 A (de)constructive approach to program checking. In *Proceedings of the 40th Annual ACM*
982 *Symposium on Theory of Computing (STOC)*, pages 143–152. ACM, 2008. doi:10.1145/
983 1374376.1374399.
- 984 19 Shuichi Hirahara. Non-black-box worst-case to average-case reductions within NP. In *59th*
985 *IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 247–258. IEEE
986 Computer Society, 2018. doi:10.1109/FOCS.2018.00032.
- 987 20 Shuichi Hirahara. Non-disjoint promise problems from meta-computational view of pseudoran-
988 dom generator constructions. In *35th Computational Complexity Conference (CCC)*, volume
989 169 of *LIPICs*, pages 20:1–20:47. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.
990 doi:10.4230/LIPICs.CCC.2020.20.
- 991 21 Shuichi Hirahara and Rahul Santhanam. On the average-case complexity of MCSP and its
992 variants. In *32nd Conference on Computational Complexity (CCC)*, volume 79 of *LIPICs*, pages
993 7:1–7:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPICs.
994 CCC.2017.7.
- 995 22 Shuichi Hirahara and Osamu Watanabe. Limits of minimum circuit size problem as oracle. In
996 *31st Conference on Computational Complexity (CCC)*, volume 50 of *LIPICs*, pages 18:1–18:20.
997 Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPICs.CCC.2016.18.
- 998 23 John M. Hitchcock and Aduri Pavan. On the NP-completeness of the minimum circuit
999 size problem. In *35th IARCS Annual Conference on Foundation of Software Technology*
1000 *and Theoretical Computer Science (FSTTCS)*, volume 45 of *LIPICs*, pages 236–245. Schloss
1001 Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. doi:10.4230/LIPICs.FSTTCS.2015.236.
- 1002 24 Rahul Ilango, Bruno Loff, and Igor Carboni Oliveira. NP-hardness of circuit minimization
1003 for multi-output functions. In *35th Computational Complexity Conference (CCC)*, volume
1004 169 of *LIPICs*, pages 22:1–22:36. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.
1005 doi:10.4230/LIPICs.CCC.2020.22.
- 1006 25 Valentine Kabanets and Jin-yi Cai. Circuit minimization problem. In *Proceedings of the*
1007 *Thirty-Second Symposium on Theory of Computing (STOC)*, pages 73–79, 2000. doi:10.1145/
1008 335305.335314.
- 1009 26 Valentine Kabanets, Daniel M. Kane, and Zhenjian Lu. A polynomial restriction lemma with
1010 applications. In *Proceedings of the 49th Annual Symposium on Theory of Computing (STOC)*,
1011 pages 615–628. ACM, 2017. doi:10.1145/3055399.3055470.

- 1012 27 Leonid A. Levin. Randomness conservation inequalities; information and independence in math-
1013 ematical theories. *Information and Control*, 61(1):15–37, 1984. doi:10.1016/S0019-9958(84)
1014 80060-1.
- 1015 28 Shachar Lovett and Jiapeng Zhang. On the impossibility of entropy reversal, and its application
1016 to zero-knowledge proofs. In *Theory of Cryptography - 15th International Conference (TCC)*,
1017 volume 10677 of *Lecture Notes in Computer Science*, pages 31–55. Springer, 2017. doi:
1018 10.1007/978-3-319-70500-2_2.
- 1019 29 Dylan M. McKay, Cody D. Murray, and R. Ryan Williams. Weak lower bounds on resource-
1020 bounded compression imply strong separations of complexity classes. In *Proceedings of the*
1021 *51st Annual Symposium on Theory of Computing (STOC)*, pages 1215–1225, 2019. doi:
1022 10.1145/3313276.3316396.
- 1023 30 Cody Murray and Ryan Williams. On the (non) NP-hardness of computing circuit complexity.
1024 *Theory of Computing*, 13(4):1–22, 2017. doi:10.4086/toc.2017.v013a004.
- 1025 31 Igor Carboni Oliveira, Ján Pich, and Rahul Santhanam. Hardness magnification near state-
1026 of-the-art lower bounds. In *34th Computational Complexity Conference (CCC)*, volume
1027 137 of *LIPICs*, pages 27:1–27:29. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2019.
1028 doi:10.4230/LIPICs.CCC.2019.27.
- 1029 32 Michael Rudow. Discrete logarithm and minimum circuit size. *Information Processing Letters*,
1030 128:1–4, 2017. doi:10.1016/j.ipl.2017.07.005.
- 1031 33 Amit Sahai and Salil P. Vadhan. A complete problem for statistical zero knowledge. *J. ACM*,
1032 50(2):196–249, 2003. doi:10.1145/636865.636868.
- 1033 34 Michael Saks and Rahul Santhanam. Circuit lower bounds from NP-hardness of MCSP
1034 under Turing reductions. In *35th Computational Complexity Conference (CCC)*, volume
1035 169 of *LIPICs*, pages 26:1–26:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.
1036 doi:10.4230/LIPICs.CCC.2020.26.
- 1037 35 Heribert Vollmer. *Introduction to circuit complexity: a uniform approach*. Springer Science &
1038 Business Media, 1999. doi:10.1007/978-3-662-03927-4.