THEORY OF COMPUTING, Volume 17 (11), 2021, pp. 1–38
www.theoryofcomputing.org

SPECIAL ISSUE: CCC 2019

Hardness Magnification Near State-of-the-Art Lower Bounds

Igor C. Oliveira   Ján Pich   Rahul Santhanam

Received July 23, 2019; Revised February 27, 2021; Published November 30, 2021

Abstract. This article continues the development of hardness magnification, an emerging area that proposes a new strategy for showing strong complexity lower bounds by reducing them to a refined analysis of weaker models, where combinatorial techniques might be successful.

We consider gap versions of the meta-computational problems MKtP and MCSP, where one needs to distinguish instances (strings or truth-tables) of complexity $\leq s_1(N)$ from instances of complexity $\geq s_2(N)$, and $N = 2^n$ denotes the input length. In MCSP, complexity is measured by circuit size, while in MKtP one considers Levin's notion of time-bounded Kolmogorov complexity. (In our results, the parameters $s_1(N)$ and $s_2(N)$ are asymptotically quite close, and the problems almost coincide with their standard formulations without a gap.) We establish that for Gap-MKtP$[s_1, s_2]$ and Gap-MCSP$[s_1, s_2]$, a *marginal improvement* over the state of the art in unconditional lower bounds in a variety of computational models would imply explicit *superpolynomial* lower bounds, including $\mathsf{P} \neq \mathsf{NP}$.

Theorem. There exists a universal constant $c \geq 1$ for which the following hold. If there exists $\varepsilon > 0$ such that for every small enough $\beta > 0$

A conference version of this paper appeared in the Proceedings of the 34th Computational Complexity Conference (CCC'19) [45]. A preprint of this article appeared in the Electronic Colloquium on Computational Complexity (ECCC) as Tech Report TR 18-158.

ACM Classification: F.1.3

AMS Classification: 68Q17

Key words and phrases: circuit complexity, minimum circuit size problem, Kolmogorov complexity

DOI: 10.4086/toc.2021.v017a011

(1) $\mathsf{Gap\text{-}MCSP}[2^{\beta n}/cn, 2^{\beta n}] \notin \mathsf{Circuit}[N^{1+\varepsilon}]$, then $\mathsf{NP} \not\subseteq \mathsf{Circuit}[\mathsf{poly}]$.

(2) $\mathsf{Gap\text{-}MKtP}[2^{\beta n}, 2^{\beta n} + cn] \notin B_2\text{-}\mathsf{Formula}[N^{2+\varepsilon}]$, then $\mathsf{EXP} \not\subseteq \mathsf{Formula}[\mathsf{poly}]$.

(3) $\mathsf{Gap\text{-}MKtP}[2^{\beta n}, 2^{\beta n} + cn] \notin U_2\text{-}\mathsf{Formula}[N^{3+\varepsilon}]$, then $\mathsf{EXP} \not\subseteq \mathsf{Formula}[\mathsf{poly}]$.

(4) $\mathsf{Gap\text{-}MKtP}[2^{\beta n}, 2^{\beta n} + cn] \notin \mathsf{BP}[N^{2+\varepsilon}]$, then $\mathsf{EXP} \not\subseteq \mathsf{BP}[\mathsf{poly}]$.

These results are complemented by lower bounds for Gap-MCSP and Gap-MKtP against different models. For instance, the lower bound assumed in (1) holds for U_2-formulas of near-quadratic size, and lower bounds similar to (2) – (4) hold for various regimes of parameters.

We also identify a natural computational model under which the hardness magnification threshold for Gap-MKtP lies *below* existing lower bounds: U_2-formulas that can compute parity functions at the leaves (instead of just literals). As a consequence, if one managed to adapt the existing lower bound techniques against such formulas to work with Gap-MKtP, then $\mathsf{EXP} \not\subseteq \mathsf{NC}^1$ would follow via hardness magnification.

1 Introduction

1.1 Context

Establishing limits on the efficiency of computations is widely considered to be one of the most important open problems in computer science and mathematics. Unconditional lower bounds are known in many restricted computational settings (see, e. g., [8, 27]), but progress in understanding the limitations of more expressive devices has been slow and incremental (see [1] for a recent survey and references). Table 1 summarizes the current landscape of unconditional lower bounds with respect to general circuits, formulas, branching programs, bounded-depth threshold circuits, and bounded-depth circuits with modular gates. These constitute some of the most widely investigated models extending the weak computational settings for which we already have explicit superpolynomial lower bounds.

A conditional explanation has been proposed to address the difficulty of establishing strong lower bounds in most of these computational settings. The theory of natural proofs [52] shows that if a computational device can compute pseudorandom functions, then sufficiently constructive techniques (such as those that have been successful against weaker models) cannot show lower bounds of the form N^k if k is sufficiently large. This connection has been quite influential, and subsequent articles (see, e. g., [41, 7]) have further investigated the limitations of lower bound techniques from this perspective.

The Razborov–Rudich framework suggests the significance of meta-computational problems of a particular form: *those referring to the computational complexity of strings or truth-tables*. Our results describe a striking phenomenon associated to such problems. They show that in several scenarios, if we could establish slightly stronger lower bounds for them, i. e., lower bounds that marginally improve the size bounds described in Table 1, then *superpolynomial* lower bounds for explicit problems would follow. More specifically, this phenomenon concerns computational problems where the complexity of strings is measured according to circuit complexity (often referred to as MCSP; see [29]) or Levin's time-bounded Kolmogorov complexity [36] (a problem known as MKtP; see [5]).

Computational Model	**Unconditional Lower Bounds**	**Reference(s)**
Boolean Circuits; w.r.t. different forms of explicitness	$\mathsf{P} \not\subseteq \mathsf{Circuit}[cN]$, $\mathsf{MA}/1 \not\subseteq \mathsf{Circuit}[N^k]$ $\mathsf{MA_{EXP}} \not\subseteq \mathsf{Circuit}[\mathsf{poly}]$ $\Sigma_2^P \not\subseteq \mathsf{Circuit}[N^k]$	[26, 18] [10, 53] [31]
Formulas over B_2	$\mathsf{P} \not\subseteq B_2\text{-}\mathsf{Formula}[N^{2-o(1)}]$	[44]
Formulas over U_2	$\mathsf{P} \not\subseteq U_2\text{-}\mathsf{Formula}[N^{3-o(1)}]$	[20, 58, 17]
Branching programs	$\mathsf{P} \not\subseteq \mathsf{BP}[N^{2-o(1)}]$	[44]
Low-depth threshold circuits	$\mathsf{P} \not\subseteq \mathsf{MAJ} \circ \mathsf{THR} \circ \mathsf{THR}[N^{3/2-o(1)}]$	[30]
Depth-d threshold circuits	$\mathsf{P} \not\subseteq \mathsf{TC}^0_d[N^{1+\exp(-d)}]$ (wires)	[25]
Depth-d circuits with mod gates	quasi-$\mathsf{NP} \not\subseteq \mathsf{ACC}^0_d[\mathsf{poly}]$	[43]

Table 1: A summary of several state-of-the-art lower bounds in circuit complexity theory. In our notation, N denotes input length, and $\mathfrak{C}[s]$ refers to $\mathfrak{C}$-circuits of size $\leq s$. Establishing stronger lower bounds in these different models is open (or non-trivial lower bounds for a function in $\mathsf{E} = \mathsf{DTIME}[2^{O(N)}]$ in the case of ACC^0_d).

MCSP and MKtP are important meta-computational problems with connections to areas such as learning theory, cryptography, proof complexity, pseudorandomness and circuit complexity. The fundamental question of whether MCSP is NP-complete has been the subject of intensive research. We refer to [4] for a recent survey of work on these problems.

The new results in this paper are part of an emerging theory of *hardness magnification* showing that weak lower bounds for some problems imply much stronger lower bounds. Several results of this form have been obtained in different contexts [56, 6, 37, 42, 46], and we refer to [46] and [15] for further discussion. In particular, [15] contains the most recent overview of the developements around hardness magnification. Other forms of hardness magnification are known in settings such as communication complexity and arithmetic circuit complexity. A recent example phrased in a way that is closer to our results appears in [13] (see also [22]).

Our hardness magnification results for MCSP and MKtP relate to the Razborov–Rudich framework of natural proofs in two ways. First, the meta-computational problems for which we show magnification are closely related to the notion of natural proofs—indeed MCSP is easy on average if and only if natural proofs exist [21]. Second, our results suggest a possible way to bypass the natural proofs barrier. The reason is that the natural proofs barrier only applies to lower bound techniques that work for *random* functions; our magnification results, on the other hand, exploit specific properties of MCSP and MKtP that do not hold for random functions. Similar observations about bypassing the natural proofs barrier using magnification-type results were made in [6, 46].

Our main contributions can be informally described as follows:

(*i*) We employ new techniques to obtain the first magnification theorem for the worst-case formulation of the MCSP problem.[1]

(*ii*) Extending [46, Theorem 3], our results establish hardness magnification for a natural meta-computational problem (MKtP) near the lower bound frontiers in several standard circuit models. In addition, we identify a computational model where hardness magnification for MKtP lies below existing lower bounds.

(*iii*) Crucially, our hardness magnification theorems hold for problems for which it is possible to establish a variety of non-trivial lower bounds.

We believe these results further highlight the relevance of meta-computational problems in connection to the main open problems in algorithms and complexity theory (see, e. g., [64, 12] for recent breakthroughs), and strongly indicate that the investigation of weak lower bounds for MKtP and MCSP is a fundamental research direction.

1.2 Results

In this section, we formally state our results. We also briefly discuss some of our techniques, which are explained in more detail in the main body of the paper. We defer a more elaborate discussion of some results to Section 1.3.

Notation. We consider formulas over the bases U_2 (fan-in two ANDs and ORs), B_2 (all boolean functions over two input bits), and extended U_2-formulas where the input leaves are labelled by literals, constants, or parity functions over the input bits of arbitrary arity. The corresponding classes of formulas of size at most s (measured by the number of leaves) will be denoted by U_2-$\mathsf{Formula}[s]$, B_2-$\mathsf{Formula}[s]$, and U_2-$\mathsf{Formula}$-$\oplus[s]$, respectively. If we do not specify the type of formulas, we are referring to De Morgan formulas (i. e., formulas over U_2). We also consider bounded-depth majority circuits, where each internal gate computes a boolean-valued majority function (MAJ) of the form $\sum_{i \in S} y_i \geq^? t$ (the circuit has access to input literals $x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_n}$). We measure the size of such circuits by the number of *wires* in the circuit. Depth-d majority circuits of size s will de denoted by $\mathsf{MAJ}^0_d\{s\}$, where $d \geq 1$ is fixed. The choice of braces $\{\}$ is supposed to emphasize that the size is measured by the number of wires. We also consider threshold circuits whose internal gates compute threshold functions (THR) of the form $\sum_{i \in S} w_i \cdot y_i \geq^? t$, for $w_i, t \in \mathbb{R}$. We count gates in this case, and let $\mathsf{TC}^0_d[s]$ denote the corresponding class of circuits. $\mathsf{Circuit}[s]$ denotes fan-in two boolean circuits of size s and of unbounded depth (gate types do not matter in our results: our results hold for every fixed choice of basis for the gates, which can be assumed throughout the paper). More generally, for a circuit class $\mathfrak{C}$, we use $\mathfrak{C}[s]$ to denote $\mathfrak{C}$-circuits of size $\leq s$, where size is measured by number of gates. Finally, $\mathsf{BP}[s]$ denotes deterministic branching programs of size at most s. We refer to standard textbooks (e. g., [27]) for more information about these boolean devices.

Gap-MKtP and lower bounds for EXP. We use N to denote the input length of an instance of Gap-MKtP$[s_1, s_2]$ (see Definition 2.2 below), where we need to distinguish strings of Kt complexity [36] (a

[1]Independently, Dylan McKay, Cody Murray, and Ryan Williams [40] established a magnification theorem for a worst-case formulation of MCSP with a completely different proof. We refer to Section 1.2 for more details.

certain time-bounded variant of Kolmogorov complexity) at most $s_1(N)$ from strings of Kt complexity at least $s_2(N)$. It is not hard to see that for constructive bounds $s_1 < s_2$, $\mathsf{Gap\text{-}MKtP}[s_1, s_2] \in \mathsf{EXP}$. Results from [5] show that $\mathsf{Gap\text{-}MKtP}[N^{\varepsilon}, N^{\varepsilon} + 5\log n]$ is hard for EXP under efficient non-uniform reductions, for every $0 < \varepsilon < 1$.

We establish a hardness magnification theorem for Gap-MKtP. Let $n = \log N$.

Theorem 1.1 (Hardness magnification for MKtP)**.** *There is a universal constant $c \geq 1$ for which the following hold. If there exists $\varepsilon > 0$ such that for every small enough $\beta > 0$*

1. $\mathsf{Gap\text{-}MKtP}[2^{\beta n}, 2^{\beta n} + cn] \notin \mathsf{Circuit}[N^{1+\varepsilon}]$, *then* $\mathsf{EXP} \not\subseteq \mathsf{Circuit}[\mathsf{poly}]$.
2. $\mathsf{Gap\text{-}MKtP}[2^{\beta n}, 2^{\beta n} + cn] \notin U_2\text{-}\mathsf{Formula}\text{-}\oplus[N^{1+\varepsilon}]$, *then* $\mathsf{EXP} \not\subseteq \mathsf{Formula}[\mathsf{poly}]$.
3. $\mathsf{Gap\text{-}MKtP}[2^{\beta n}, 2^{\beta n} + cn] \notin \mathsf{AND\text{-}THR\text{-}THR\text{-}XOR}[N^{1+\varepsilon}]$, *then* $\mathsf{EXP} \not\subseteq \mathsf{TC}^0_2[\mathsf{poly}]$.
4. $\mathsf{Gap\text{-}MKtP}[2^{\beta n}, 2^{\beta n} + cn] \notin \mathsf{MAJ}^0_{2d'+d+1}\{N^{1+(2/d')+\varepsilon}\}$, *then* $\mathsf{EXP} \not\subseteq \mathsf{MAJ}^0_d\{\mathsf{poly}\}$.
5. $\mathsf{Gap\text{-}MKtP}[2^{\beta n}, 2^{\beta n} + cn] \notin B_2\text{-}\mathsf{Formula}[N^{2+\varepsilon}]$, *then* $\mathsf{EXP} \not\subseteq \mathsf{Formula}[\mathsf{poly}]$.
6. $\mathsf{Gap\text{-}MKtP}[2^{\beta n}, 2^{\beta n} + cn] \notin U_2\text{-}\mathsf{Formula}[N^{3+\varepsilon}]$, *then* $\mathsf{EXP} \not\subseteq \mathsf{Formula}[\mathsf{poly}]$.
7. $\mathsf{Gap\text{-}MKtP}[2^{\beta n}, 2^{\beta n} + cn] \notin \mathsf{BP}[N^{2+\varepsilon}]$, *then* $\mathsf{EXP} \not\subseteq \mathsf{BP}[\mathsf{poly}]$.
8. $\mathsf{Gap\text{-}MKtP}[2^{\beta n}, 2^{\beta n} + cn] \notin (\mathsf{AC}^0[6])[N^{1+\varepsilon}]$, *then* $\mathsf{EXP} \not\subseteq \mathsf{AC}^0[6]$.

Interestingly, this result shows the existence of a single meta-computational problem that is connected to several frontiers in complexity theory.

The proof of Theorem 1.1 relies on a refinement of some ideas from [46, Section 3.2]. In fact, item 1 of Theorem 1.1 is a restatement of [46, Theorem 3] and the remaining items follow from an observation that the original argument scales to weaker circuit classes. For a sketch of the argument and its underlying techniques, we refer to the discussion in Section 3. We mention that crucial in the proof is the use of error-correcting codes, and that the complexity of computing these objects using different boolean devices gives rise to the distinct magnification thresholds observed in Theorem 1.1. The formal proof of Theorem 1.1 appears in Sections 3.1 and 3.2.

In contrast, we observe the following unconditional lower bounds.

Theorem 1.2 (Strong lower bounds for large parameters)**.** *For every $\varepsilon > 0$ there exists $\delta > 0$ for which the following results hold:*

1. $\mathsf{Gap\text{-}MKtP}[2^{(1-\delta)n}, 2^{n-1}] \notin U_2\text{-}\mathsf{Formula}[N^{3-\varepsilon}]$.
2. $\mathsf{Gap\text{-}MKtP}[2^{(1-\delta)n}, 2^{n-1}] \notin B_2\text{-}\mathsf{Formula}[N^{2-\varepsilon}]$.
3. $\mathsf{Gap\text{-}MKtP}[2^{(1-\delta)n}, 2^{n-1}] \notin \mathsf{BP}[N^{2-\varepsilon}]$.

The proof of Theorem 1.2 is a simple adaptation of the existing lower bound methods. It relies on the existence of pseudorandom generators against small formulas and small branching programs by Impagliazzo, Meka and Zuckerman [24], together with an observation of Allender [3]. The argument appears in Appendix 5.2.

Note the different regime of parameters for $\mathsf{Gap\text{-}MKtP}[s_1, s_2]$ in Theorems 1.1 and 1.2. In order to magnify a weak lower bound using Theorem 1.1, we need that it holds for $s_1 = 2^{o(n)} = N^{o(1)}$. The next result shows that non-trivial unconditional lower bounds can be obtained in this regime.

Theorem 1.3 (A near-quadratic formula lower bound)**.** *For every constant $0 < \alpha < 2$ there exists $C > 1$ such that* $\mathsf{Gap\text{-}MKtP}[Cn^2, 2^{(\alpha/2)n-2}] \notin U_2\text{-}\mathsf{Formula}[N^{2-\alpha}]$.[2]

The proof of Theorem 1.3 is a simple adaptation of a lower bound of Hirahara and Santhanam [21, Section 4] (see also the exposition in [46, Appendix C.1]) employed in the context of MCSP for larger parameters. A sketch of the argument followed by a proof can be found in Appendix 5.1.

Gap-MCSP and lower bounds for NP. We use $N = 2^n$ to denote the input length of an instance of $\mathsf{Gap\text{-}MCSP}[s_1, s_2]$ (see Definition 2.4 below), where one needs to distinguish functions of circuit complexity at most s_1 from functions of circuit complexity at least s_2. It is not hard to see that for constructive bounds $s_1 < s_2$, $\mathsf{Gap\text{-}MCSP}[s_1, s_2] \in \mathsf{NP}$. It is not known if $\mathsf{Gap\text{-}MCSP}[s_1, s_2]$ is NP-complete. Recently, [23] established NP-hardness of several related versions of the MCSP problem. However, note that for $s_1 = N^{o(1)}$, $\mathsf{Gap\text{-}MCSP}[s_1, s_2]$ is computable in time $2^{N^{o(1)}}$, which means that with these parameters the problem is not NP-hard unless NP is contained in subexponential time.

We establish the following magnification theorem for Gap-MCSP.

Theorem 1.4 (Hardness magnification for MCSP)**.** *There is a universal constant $c \geq 1$ for which the following holds. If there exists $\varepsilon > 0$ such that for every small enough $\beta > 0$,* $\mathsf{Gap\text{-}MCSP}[2^{\beta n}/cn, 2^{\beta n}] \notin \mathsf{Circuit}[N^{1+\varepsilon}]$*, then* $\mathsf{NP} \not\subseteq \mathsf{Circuit}[\mathsf{poly}]$.

MCSP and MKtP are quite different problems. In our results, an important distinction is that applying a polynomial-time function to an input of MKtP does not substantially increase its Kt complexity (see Proposition 2.3), but this is not necessarily true in the context of circuit complexity, where the input string represents an entire truth-table. For this reason, the proof of Theorem 1.4 is completely different from the proof of Theorem 1.1.

A magnification theorem for a version of MCSP has been obtained already by Oliveira and Santhanam [46]. For example, they show that superlinear formula lower bounds for the variant of MCSP denoted $(1, 1-\delta)\text{-}\mathsf{MCSP}[s]$ imply $\mathsf{NP} \not\subseteq \mathsf{NC}^1$. $(1, 1-\delta)\text{-}\mathsf{MCSP}[s]$ is different from the version of MCSP discussed in the present paper ($\mathsf{Gap\text{-}MCSP}[2^{\beta n}/cn, 2^{\beta n}]$), and refers to the *average-case* circuit complexity of the input truth table: it asks for distinguishing truth-tables of Boolean functions of circuit complexity at most s from functions which cannot be $(1-\delta)$-approximated by circuits of size s, for $\delta > 0$ and $s = N^{\varepsilon}$ where ε is sufficiently small. As we discuss below, the existing lower bound methods are more suitable for proving lower bounds for $\mathsf{Gap\text{-}MCSP}[2^{\beta n}/cn, 2^{\beta n}]$ than for $(1, 1-\delta)\text{-}\mathsf{MCSP}[s]$.

[2] The constant C has an exponential dependence on $1/\alpha$.

Theorem 1.4 is our main technical contribution. The argument relies on the notion of *anti-checkers*. Roughly speaking, an anti-checker is a bounded collection $\mathcal{S}$ of inputs associated with a hard function f such that any small circuit C differs from f on some input in $\mathcal{S}$. More precisely, it was established in [38] that any function $f\colon \{0,1\}^n \to \{0,1\}$ that requires circuits of size s admits a collection $\mathcal{S}_f$ containing $O(s)$ strings that is an anti-checker against circuits of size roughly s/n. Our argument makes crucial use of anti-checkers, and en route to Theorem 1.4 we give a more constructive proof of their existence. (While the proof in [38] uses min-max theory, our proof is combinatorial and self-contained.)

We remark that anti-checkers were first employed for hardness magnification in the context of *proof complexity* [42]. However, while the existential result from [38] was sufficient in that context, this is not the case in *circuit complexity*, and our argument needs to be more sophisticated. For the reader interested in learning more about hardness magnification in proof complexity, how it relates to meta-computational problems such as MCSP, and how the new results compare with previous work, we refer to Appendix 6.

The proof of Theorem 1.4 is not difficult given a certain lemma about the construction of anti-checkers (see Section 4.1). The crucial Anti-Checker Lemma (see Lemma 4.1) says that $\mathsf{NP} \subseteq \mathsf{Circuit}[\mathsf{poly}]$ implies the existence of circuits of *almost linear size* which given the truth table of a Boolean function f print a corresponding set $\mathcal{S}_f$. The circuits provided by the Anti-Checker Lemma simulate the alternate proof of the existence of anti-checkers, but make the involved argument constructive by using approximate counting and the assumption $\mathsf{NP} \subseteq \mathsf{Circuit}[\mathsf{poly}]$. The strategy for proving the Anti-Checker Lemma is similar to the proof of $\mathsf{S}_2^p \subseteq \mathsf{ZPP}^{\mathsf{NP}}$ [11] and to the classical $\mathsf{ZPP}^{\mathsf{NP}}$ learning algorithm from [9]. A high-level exposition and the complete proof are described in Section 4.[3]

A remark on kernelization. Implicit in our proof of Theorem 1.4 is a Turing kernelization for the parameterized version of Gap-MCSP which might be of independent interest – there are nearly-linear sized circuits which solve any instance of Gap-MCSP with parameter s using oracle access to $\mathsf{poly}(s)$-sized instances of a fixed language in the Polynomial Hierarchy.

We observe that a simple adaptation of the lower bound of Hirahara and Santhanam [21, Section 4] yields the following related unconditional lower bound against *formulas*.

Theorem 1.5. *For each $0 < \alpha < 2$ there exists $d > 1$ such that* $\mathsf{Gap}\text{-}\mathsf{MCSP}[n^d, 2^{(\alpha/2-o(1))n}] \notin U_2\text{-}\mathsf{Formula}[N^{2-\alpha}]$.

Consequently, if one could establish an analogue of Theorem 1.4 for sub-quadratic formulas, then $\mathsf{NP} \not\subseteq \mathsf{Formula}[\mathsf{poly}]$. We explain why the argument behind the proof of Theorem 1.4 fails in the case of formulas in Section 4.2.[4] The proof of Theorem 1.5 is similar to the proof of Theorem 1.3, and we sketch the necessary modifications in Appendix 5.3.

Finally, in Section 4.3 we discuss a certain combinatorial hypothesis ("The Anti-Checker Hypothesis") connected to the techniques behind the proof of Theorem 1.4. If this hypothesis holds, then

[3] We stress that the assumption that $\mathsf{NP} \subseteq \mathsf{Circuit}[\mathsf{poly}]$ allows several computations to be performed in circuit size $O(N^c)$, where N is the input length. Note however that our requirement is much more stringent: we need to construct anti-checkers using circuits of size $O(N^{1+\varepsilon})$ instead of $O(N^c)$ for some $c \in \mathbb{N}$.

[4] Note that Theorem 1.4 implies lower bounds for a problem in NP. Theorem 1.1 only gives lower bounds in EXP, but its proof extends to several low-complexity settings.

$\mathsf{NP} \not\subseteq \mathsf{Formula}[\mathsf{poly}]$. We observe that the hypothesis does hold in the average-case.

Independent and subsequent work. Independently of the conference version of our paper [45], Dylan McKay, Cody Murray, and Ryan Williams [40] established a version of Theorem 1.4 that refers to the standard formulation of MCSP without a gap between positive and negative instances. Their result is obtained using different techniques, and also extends to circuit classes of limited depth and to various versions of MKtP. While Theorem 1.4 offers a weaker statement compared to the results from [40], a more careful analysis of our proof reveals that it survives in the context of weaker computational models, e. g., in order to show that $\mathsf{NP} \not\subseteq \mathsf{NC}^1$ it is enough to establish lower bounds against almost superlinear size circuits with a certain structure, a circuit class for which it is possible to prove non-trivial lower bounds. This is explored in a subsequent paper by Chen, Hirahara, Oliveira, Pich, Rajgopal and Santhanam [15].

More recently, Lijie Chen, Dylan McKay, Cody Murray and Ryan Williams [16] proved an incomparable magnification theorem, showing that weak lower bounds against sufficiently sparse languages imply superpolynomial lower bounds for languages computable in nondeterministic time. This result applies to a large class of problems, but the lower bound implication is weaker than Theorem 1.4. A significant advantage of Theorem 1.1 compared to these results is that it applies to many circuit classes.

In the most recent work on the topic (already mentioned above), [15] addresses several further questions raised in the present paper. In particular, [15] disproves "The Anti-Checker Hypothesis" and explores the potential and limitations of the hardness magnification program by showing that some magnification theorems are non-naturalizable in the sense that they not only imply strong lower bounds such as $\mathsf{NP} \neq \mathsf{P}$ but even the non-existence of natural properties, and by identifying the so called "locality barrier" which prevents us from proving strong circuit lower bounds by combining the existing weak lower bounds methods with hardness magnification theorems.

1.3 Discussion

This work is a sequel to an earlier paper of two of the authors [46], in which hardness magnification was first explored in a systematic way. The results in [46] are for a variety of problems (including SAT, Vertex Cover and variants of MKtP and MCSP) and models (including formulas, circuits and sublinear-time algorithms). For each (problem, model) pair considered in [46], it is shown that *non-trivial* lower bounds for the problem against the model imply *superpolynomial* lower bounds for some other explicit problem.

As discussed in [46], there are two natural interpretations of magnification results. The first, more optimistic, interpretation is that magnification constitutes a new approach to proving strong lower bounds. If we are able to replicate the non-trivial circuit lower bounds we can prove against models such as constant-depth circuits (in the worst case) or formulas (in the average case) for the problems witnessing the magnification phenomenon, then this would lead to new and powerful lower bounds. There are no well-understood obstacles to the success of such an approach. In particular, hardness magnification seems to avoid the natural proofs barrier of Razborov and Rudich [52]. This has been, in fact, formalized in a subsequent paper [15].

The other, more pessimistic, interpretation of magnification results is that they indicate that circuit lower bounds might be *even harder* to achieve than previously thought. Earlier, superpolynomial lower bounds seemed to be out of reach, but there was no strong reason to believe that small *fixed polynomial* lower bounds or at least *barely non-trivial* lower bounds are hard to show. Given the belief that superpolynomial circuit lower bounds for explicit hard problems are hard to show, the magnification phenomenon suggests that for several natural problems of interest, even non-trivial lower bounds are hard to show.

The results of [46] have drawbacks which the present work addresses.

For the optimistic interpretation, it would be good to have examples of natural problems for which some magnification phenomenon holds, and for which we have techniques giving non-trivial lower bounds. For example, although we have a non trivial lower bound of Hirahara and Santhanam for a version of MCSP (Theorem 1.5), we do not know how to adapt their method to get a similar lower bound for $(1,1-\delta)$-$\mathsf{MCSP}[N^{\varepsilon}]$. In this paper, we give magnification results for the Gap-MKtP and Gap-MCSP problems, for both of which we show that there are non-trivial lower bounds in the model of Boolean formulas. Thus there is *some* lower bound technique which works to give a non-trivial result – the question is “merely” whether it can be strengthened to derive a lower bound beyond the magnification threshold.

While the pessimistic interpretation might not lead to new lower bounds, it does have the potential of leading to a better understanding of barriers. From this point of view, [46] is not particularly sensitive to the specific model being considered. It is clear that some models are easier to prove lower bounds for than others – indeed we have near-cubic lower bounds in the De Morgan formula model, near-quadratic lower bounds in the branching program model, and only trivial lower bounds in the Boolean circuit model. Can magnification be used to give a new perspective on these differences between models?

We provide a positive answer to this question, by giving different magnification thresholds for different models. What remains mysterious is why known lower bound techniques fall short of proving lower bounds required to apply magnification. This suggest that there might be limitations of the known techniques above and beyond those captured by natural proofs. A subsequent paper [15] addresses this question by identifying such a limitation more formally, the so-called “locality barrier.”

It is worth emphasizing that there are natural problems for which showing lower bounds that are *weaker* than the current state-of-the-art size bounds would also imply superpolynomial lower bounds [46]. A representative example presented in [46] concerns the above-mentioned average-case version of MCSP, where the problem refers to the average-case circuit complexity of the input function. The reason that result does not imply superpolynomial lower bounds via magnification is that the corresponding unconditional lower bounds and magnification theorems hold for a different regime of the average-case complexity parameter.[5]

Our results and techniques were motivated in part by the desire to address this gap. On the one hand, it seems to be easier to analyse problems that refer to the worst-case complexity of the input. But on the other hand, our new results indicate that the shift from average-case to worst-case complexity (in the description of the problem) often increases the magnification threshold to size bounds that are beyond existing techniques. As a concrete example, if the formula magnification theorem for the average-case MCSP problem investigated in [46] could be established for the worst-case variant investigated here, $\mathsf{NP} \not\subseteq \mathsf{NC}^1$ would follow via Theorem 1.5. Another glimpse of the subtle transition between worst-case and average-case complexity and its role in magnification appears in the discussion of the Anti-Checker

[5] In particular, the lower bounds and magnification theorems from [46] do not hold for the same problems.

Hypothesis in Section 4.3.

Complementing these results, we identify a computational model that has not received much attention in the literature, and for which the magnification threshold for Gap-MKtP lies below existing lower bounds. This corresponds to Theorem 1.4 Item 2, i. e., U_2-formulas augmented with parities in the leaves (our exposition in Section 3 focuses on this model). Note that, by a straightforward simulation, before breaking the cubic barrier for U_2-formulas or the quadratic barrier for B_2-formulas, one needs to show superlinear lower bounds against U_2-Formula-$\oplus$. But a recent result of Tal [59] implies exactly that: the inner product function over N input bits is not in U_2-Formula-$\oplus[N^{1.99}]$.

This makes this computational model particularly attractive in connection to hardness magnification and lower bounds. Indeed, it seems "obvious" that $\mathsf{Gap\text{-}MKtP}[2^{\delta n}, 2^{\delta n}+cn] \notin U_2$-Formula-$\oplus[N^{1.01}]$, given that such formulas cannot compute the much simpler inner product function, and that standard formulas require at least near-quadratic size (Theorem 1.3). Our work shows that if this is the case, then $\mathsf{EXP} \nsubseteq \mathsf{NC}^1$.

2 Preliminaries

For $\ell \in \mathbb{N}$, we use $[\ell]$ to denote the set $\{1,\ldots,\ell\}$. The length of a string w will be denoted by $|w|$. Our logarithms are in base 2, and we use $\exp(x)$ to denote e^x. We use boldface symbols such as $\boldsymbol{i}$ and $\boldsymbol{\rho}$ to denote random variables, and $\boldsymbol{x} \in_R S$ to denote that $\boldsymbol{x}$ is a uniformly random element from a set S. We often identify n with $\log N$ or N with 2^n, depending on the context.

For concreteness, we employ a random-access model to formalize uniform algorithms. The details of the model are not crucial in our results, and only mildly affect the gap parameters s_1 and s_2. We fix an efficient universal machine U, and use $\langle M \rangle$ to denote the string encoding the algorithm M (with respect to U). We assume for convenience the following property of this encoding: if an algorithm C is obtained via the composition of the computations of algorithms A and B, then $|\langle C \rangle| \leq |\langle A \rangle| + |\langle B \rangle| + O(1)$. Roughly speaking, concatenating two codes gives a new code.[6]

We introduce next the notion of Kt complexity.

Definition 2.1 (Kt Complexity ([36]; see also [3]))**.** For a string $x \in \{0,1\}^*$, $\mathsf{Kt}(x)$ denotes the minimum of $|\langle M \rangle| + |a| + \lceil \log t_M(a) \rceil$ over pairs (M,a) such that the machine M outputs x when it is given the input string a. We say that a pair (M,a) witnesses an inequality $\mathsf{Kt}(x) \leq y$ if $|\langle M \rangle| + |a| + \lceil \log t_M(a) \rceil \leq y$.

Definition 2.2 (The Gap-MKtP Problem)**.** We consider the promise problem $\mathsf{Gap\text{-}MKtP}[s_1,s_2]$, where $s_1, s_2 \colon \mathbb{N} \to \mathbb{N}$ and $s_1(N) < s_2(N)$ for all $N \in \mathbb{N}$. For each $N \geq 1$, $\mathsf{Gap\text{-}MKtP}[s_1,s_2]$ is defined by the following sets of instances:

$$\mathcal{YES}_N \stackrel{\text{def}}{=} \{x \in \{0,1\}^N \mid \mathsf{Kt}(x) \leq s_1(N)\}, \text{ and}$$
$$\mathcal{NO}_N \stackrel{\text{def}}{=} \{x \in \{0,1\}^N \mid \mathsf{Kt}(x) > s_2(N)\}.$$

We will need the following simple result.

[6] This holds for instance in assembly code with relative jump instructions (i. e., goto instructions where the new line is encoded relative to the number of the current line).

Proposition 2.3 (Kt complexity and composition)**.** *Let B be an algorithm that runs in time at most $T_B(N)$ over inputs of length N. Then, for every input $w \in \{0,1\}^N$, as N grows we have*

$$\mathsf{Kt}(B(w)) \;\leq\; \mathsf{Kt}(w) + \log(T_B(N)) + O(1).$$

Proof. Let A be a machine and a be a string such that the pair (A,a) witnesses the value $\mathsf{Kt}(w)$. Let C be the composition of machines A and B, i. e., $C(y) = B(A(y))$. We claim that the pair (C,a) witnesses the inequality in the conclusion of the proposition. Indeed, since $C(a) = B(A(a)) = B(w)$, we get

$$\begin{aligned}
\mathsf{Kt}(B(w)) &\leq |\langle C\rangle| + |a| + \lceil \log t_C(a)\rceil \\
&\leq |\langle A\rangle| + |\langle B\rangle| + O(1) + |a| + \log(t_A(a) + t_B(w)) \\
&\leq |\langle A\rangle| + |a| + \log(t_A(a)) + \log(t_B(w)) + |\langle B\rangle| + O(1) \\
&\leq \mathsf{Kt}(w) + \log(T_B(N)) + O(1),
\end{aligned}$$

where we have used that $|\langle B\rangle|$ is constant as N grows. □

We also consider a natural formulation of the gap version of the Minimum Circuit Size Problem (MCSP). The circuit complexity of a boolean function $f\colon \{0,1\}^n \to \{0,1\}$ is denoted by $\mathsf{Size}(f)$, i. e., $\mathsf{Size}(f)$ is the minimal s such that f is computable by a circuit from $\mathsf{Circuit}[s]$. We use the same notation to represent the circuit complexity of the function encoded by a string $x \in \{0,1\}^{2^n}$.

Definition 2.4 (The Gap-MCSP Problem)**.** We consider the promise problem $\mathsf{Gap\text{-}MCSP}[s_1,s_2]$, where $s_1, s_2 \colon \mathbb{N} \to \mathbb{N}$ and $s_1(n) \leq s_2(n)$ for all $n \in \mathbb{N}$. For each $n \geq 1$, $\mathsf{Gap\text{-}MCSP}[s_1(n), s_2(n)]$ is defined by the following sets of instances:

$$\begin{aligned}
\mathcal{YES}_n &\stackrel{\text{def}}{=} \{x \in \{0,1\}^{2^n} \mid \mathsf{Size}(x) \leq s_1(n)\}, \text{ and} \\
\mathcal{NO}_n &\stackrel{\text{def}}{=} \{x \in \{0,1\}^{2^n} \mid \mathsf{Size}(x) > s_2(n)\}.
\end{aligned}$$

3 Hardness magnification via error-correcting codes

In this section, we prove Theorem 1.1. First, we provide a high-level exposition of the argument.

Proof idea. The result is established in the contrapositive. The idea, which goes back to [46, Theorem 3], is to reduce $\mathsf{Gap\text{-}MKtP}[s_1,s_2]$ to a problem in EXP over instances of size $\mathsf{poly}(s_1,s_2) \ll N$, and to invoke the assumed complexity collapse to solve $\mathsf{Gap\text{-}MKtP}$ using very efficient circuits (or other boolean devices). First, we apply an error-correcting code (ECC) to the input string $w \in \{0,1\}^N$. Since this can be done by a uniform polynomial time computation, we are able to show that $\mathsf{ECC}(w) \in \{0,1\}^{O(N)}$ is a string of Kt complexity $\ell < s_2$ if w has Kt complexity $\leq s_1$. On the other hand, using an efficient decoder for the ECC, we can show that if w has Kt complexity $\geq s_2$, then any string of Kt complexity $> \ell$ differs from $\mathsf{ECC}(w)$ on a constant fraction of coordinates. Let $z = \mathsf{ECC}(w)$. Given the gap in the input instances of $\mathsf{Gap\text{-}MKtP}$, our task now is to distinguish strings z that have Kt complexity at most ℓ from strings that cannot be approximated by strings of Kt complexity at most ℓ, where $s_1 < \ell < s_2$.

We achieve this by using a random projection of the input z to a string y of size roughly $\ell \ll N$. The intuition is that if z has Kt complexity at most ℓ, then every projection of z also agrees with some string (i. e., z) of Kt complexity at most ℓ. However, using random sampling and union bounds it is possible to argue that if z cannot be approximated by a string of Kt complexity at most ℓ, then with high probability no string of Kt complexity at most ℓ agrees with the randomly projected coordinates of z. Checking which case holds when we are given the string y can be done by an exponential time algorithm. Under the assumption that EXP admits small circuits, we are able to solve this problem in complexity $\mathsf{poly}(\ell) \ll N$. Chosing ℓ small enough gives us the 'shrinking' phenomenon that is so crucial in our proof.

The reduction sketched above requires (1) the computation of an appropriate ECC, and (2) is randomized. A careful derandomization and the computation of the ECC in different models of computation provide the size bounds corresponding to the magnification thresholds appearing in the statement of Theorem 1.1. For example, in case of Item (2) the derandomized reduction involves N copies of U_2-$\mathsf{Formula}$-$\oplus$ circuits (with $\oplus$ gates at the bottom computing the ECC and formulas of size $\mathsf{poly}(\ell)$ above them) of size N^{ε}, which yields the total size $N^{1+\varepsilon}$. The remaining items are obtained by analysing the specific computational models in question. We leave the details of this case analysis to the actual proof.

The proof idea above comes from [46]. We, however, start with a detailed proof of Item (2), which covers the more interesting scenario of formulas with parity leaves and includes the observation that the argument from [46] scales to weaker circuit classes. Intuitively, this is because error-correcting codes can be computed efficiently by weak computational models. We then discuss how a simple modification of the argument together with known results imply the other cases.

3.1 Proof of Theorem 1.1 Case 2 (magnification for formulas with parities)

We will need the following explicit construction.

Theorem 3.1 (Explicit linear error-correcting codes (see [28, 54])). *There exists a sequence $\{E_N\}_{N\in\mathbb{N}}$ of error-correcting codes $E_N\colon \{0,1\}^N \to \{0,1\}^{M(N)}$ with the following properties:*

- *$E_N(x)$ can be computed by a uniform deterministic algorithm running in time $\mathsf{poly}(N)$.*
- *$M(N) = b \cdot N$ for a fixed $b \geq 1$.*
- *There exists a constant $\delta > 0$ such that any codeword $E_N(x) \in \{0,1\}^{M(N)}$ that is corrupted on at most a δ-fraction of coordinates can be uniquely decoded to x by a uniform deterministic algorithm D running in time $\mathsf{poly}(M(N))$.*
- *Each output bit is computed by a parity function: for each input length $N \geq 1$ and for each coordinate $i \in [M(N)]$, there exists a set $S_{N,i} \subseteq [N]$ such that for every $x \in \{0,1\}^N$,*

$$E_N(x)_i = \bigoplus_{j \in S_{N,i}} x_j.$$

We proceed with the proof of Theorem 1.1 Part (2). We establish the contrapositive. Assume that $\mathsf{EXP} \subseteq \mathsf{Formula}[\mathrm{poly}]$, and recall that $N = 2^n$. For any $\varepsilon > 0$, we prove that $\mathsf{Gap}\text{-}\mathsf{MKtP}[2^{\beta n}, 2^{\beta n} + cn] \in$

U_2-Formula-$\oplus[N^{1+\varepsilon}]$ for a sufficiently small $\beta > 0$ and a universal choice of the constant c. The value of c will be specified later in the proof (see Claim 3.3 below).

Let $E_N \colon \{0,1\}^N \to \{0,1\}^M$ be the error-correcting code granted by Theorem 3.1, where $M(N) = bN$. Given an instance $w \in \{0,1\}^N$ of Gap-MKtP$[2^{\beta n}, 2^{\beta n} + cn]$, in order to contruct a formula for Gap-MKtP$[2^{\beta n}, 2^{\beta n} + cn]$ we first apply E_N to $w \in \{0,1\}^N$ to get $z = E_N(w) \in \{0,1\}^M$.

Claim 3.2. *There exists $c_0 \geq 1$ such that for every large enough N the following holds. If $\mathsf{Kt}(w) \leq 2^{\beta n}$, then $\mathsf{Kt}(z) \leq 2^{\beta n} + c_0 n$.*

Proof. The claim follows immediately from the upper bound on $\mathsf{Kt}(w)$, the definition of $z = E_N(w)$, the running time of E_N, and Proposition 2.3. □

Claim 3.3. *For each $c_1 > c_0$ there exists $c > c_1$ such that for every large enough N the following holds. If $\mathsf{Kt}(w) > 2^{\beta n} + cn$, then $\mathsf{Kt}(z') > 2^{\beta n} + c_1 n$ for any $z' \in \{0,1\}^M$ that disagrees with z on at most a δ-fraction of coordinates.*

Proof. Suppose that a string $z' \in \{0,1\}^M$ disagrees with z on at most a δ-fraction of coordinates, and that $\mathsf{Kt}(z') \leq 2^{\beta n} + c_1 n$ for some $c_1 > c_0$. We give an upper bound on the Kt complexity of w by combining a description of z' with the decoder D provided by Theorem 3.1. In more detail, assume the pair (F, a) witnesses $\mathsf{Kt}(z')$. Let B be the machine that first applies the machine F to a (producing z'), then D to z'. It follows from Theorem 3.1 that $B(a) = D(F(a)) = D(z') = w$. Similarly to the proof of Proposition 2.3, we also get

$$\begin{aligned}
\mathsf{Kt}(w) &\leq |\langle B \rangle| + |a| + \lceil \log t_B(a) \rceil \\
&\leq |\langle F \rangle| + |\langle D \rangle| + O(1) + |a| + \log(t_F(a) + t_D(z')) \\
&\leq \mathsf{Kt}(z') + \log(t_D(z')) + O(1) \\
&\leq (2^{\beta n} + c_1 n) + O(n) + O(1) \\
&\leq 2^{\beta n} + cn,
\end{aligned}$$

if n is large enough and we choose c sufficiently large. □

Next we define an auxiliary language $L \in \mathsf{EXP}$, efficiently reduce Gap-MKtP to L, and use the assumption that EXP has polynomial size formulas to obtain almost-linear size formulas with parities at the bottom. Roughly speaking, we are able to obtain a formula of non-trivial size for Gap-MKtP because our reduction maps input instances of length N to instances of L of length $N^{o(1)}$ (the $o(1)$ term is captured by the parameter β using $n = \log N$). As we will see shortly, the reduction is randomized. In order to get the final U_2-formula-$\oplus$ computing Gap-MKtP, the argument is derandomized in a straightforward but careful way. More details follow.

An input string y encoding a tuple $(a, 1^b, (i_1, \alpha_1), \ldots, (i_r, \alpha_r))$ belongs to L (where a and b are positive integers, a is encoded in binary, and $\alpha_j \in \{0,1\}$) if each i_j (for $1 \leq j \leq r$) is a string of length $\lceil \log a \rceil$ and there is a string z of length a such that $\mathsf{Kt}(z) \leq b$ and for each index j we have $z_{i_j} = \alpha_j$.

Claim 3.4. $L \in \mathsf{EXP}$.

Proof. L is decidable in exponential time as we can exhaustively search all strings of Kt complexity at most b and length exactly a and check if there is one which has the specified values at the corresponding bit positions. Indeed, using the definition of Kt complexity and an efficient universal machine, a list containing all such strings can be generated in time $\mathsf{poly}(2^b, a)$. In turn, checking that a string of length a satisfies the requirement takes time at most exponential in the total input length, since each index i_j is a string of length $\lceil \log a \rceil$. □

Since $\mathsf{EXP} \subseteq \mathsf{Formula}[\mathsf{poly}]$ by assumption, L has polynomial-size formulas. Assume without loss of generality that L has formulas of size $O(\ell^k)$ for some constant k, where ℓ is its total input length. We choose $\beta = \varepsilon/100k$.

We are ready to describe a low-complexity reduction from $\mathsf{Gap\text{-}MKtP}[2^{\beta n}, 2^{\beta n} + cn]$ to L (which will form a part of the final formula for Gap-MKtP). First, we use the error-correcting code to compute z from w, as described before Claim 3.2. Then we apply the following sampling procedure. We sample uniformly and independently $r = 2^{2\beta n}$ indices $\boldsymbol{i_1}, \ldots, \boldsymbol{i_r} \in_R [M]$, where $M = bN$. We then form the string y encoding the tuple

$$(M, 1^{2^{\beta n} + c_1 n}, (\boldsymbol{i_1}, z_{\boldsymbol{i_1}}), \ldots, (\boldsymbol{i_r}, z_{\boldsymbol{i_r}})),$$

where $c_1 > c_0 \geq 1$ is provided by Claim 3.3. Note that this is a string of length $\ell(N) \leq N^{\varepsilon/10k}$.

Claim 3.5. *The following implications hold:*

(a) *If $w \in \{0,1\}^N$ is a positive instance of* $\mathsf{Gap\text{-}MKtP}[2^{\beta n}, 2^{\beta n} + cn]$*, then $y \in L$ with probability* 1.

(b) *If $w \in \{0,1\}^N$ is a negative instance of* $\mathsf{Gap\text{-}MKtP}[2^{\beta n}, 2^{\beta n} + cn]$*, then $y \notin L$ with probability $> 1/2$.*

Proof. If w is a YES instance, we have by Claim 3.2 that $\mathsf{Kt}(z) \leq 2^{\beta n} + c_0 n \leq 2^{\beta n} + c_1 n$. In this case, z is a string of length M that has the specified values at the specified bit positions, regardless of the random positions that are sampled by the reduction. Consequently, $y \in L$ with probability 1.

For the claim about NO instances, as previously established in Claim 3.3, we have that $\mathsf{Kt}(z') > 2^{\beta n} + c_1 n$ for any z' such that $|z'| = |z| = M$ and $\Pr_{\boldsymbol{i} \in_R [M]}[z'_{\boldsymbol{i}} \neq z_{\boldsymbol{i}}] \leq \delta$. Now consider any string z'' of length M such that $\mathsf{Kt}(z'') \leq 2^{\beta n} + c_1 n$. For such a string z'', for each $j \in [r]$, the probability that the random projection satisfies $z''_{\boldsymbol{i_j}} = z_{\boldsymbol{i_j}}$ (where $\boldsymbol{i_j} \in_R [M]$) is at most $1 - \delta$. Hence the probability that z'' agrees with z at all the specified bit positions is at most $(1-\delta)^r \leq \exp(-\delta r) \leq \exp(-\delta 2^{2\beta n})$. By a union bound over all strings z'' with $\mathsf{Kt}(z'') \leq 2^{\beta n} + c_1 n$, the probability that there exists a string z'' with Kt complexity at most $2^{\beta n} + c_1 n$ which is consistent with the values at the specified bit positions is exponentially small in n. Hence with high probability $y \notin L$. □

To sum up, there is a randomized reduction from $\mathsf{Gap\text{-}MKtP}[2^{\beta n}, 2^{\beta n} + cn]$ over inputs of length N to instances of L of length $\ell(N) \leq N^{\varepsilon/10k}$. Now let $\{F_{\ell(N)}\}_{N \geq 1}$ be a sequence of U_2-formulas of size $O(\ell^k)$ for L. Our randomized formulas $\boldsymbol{G}(\cdot)$ for Gap-MKtP compute as follows.

1. $\boldsymbol{G}(w) = \bigwedge_{j=1}^{N} \boldsymbol{G}^{(j)}(w)$, where each $\boldsymbol{G}^{(j)}$ is an independent copy.

2. Each $\boldsymbol{G}^{(j)}(w)$ is a randomized formula of the form $G^{(j)}(w, \boldsymbol{i_1}, \ldots, \boldsymbol{i_r})$ that first computes z from w, then computes y from z using the (random) input indices $\boldsymbol{i_1}, \ldots, \boldsymbol{i_r} \in \{0,1\}^{\log M}$, and finally applies F_ℓ to y.

It follows from Claim 3.5 using the independence of each $\boldsymbol{G}^{(j)}$ that

$$\Pr[\boldsymbol{G}(w) \text{ is incorrect}] < 2^{-N},$$

where the probability is taken over the choice of the random input of G. Consequently, by a union bound there is a fixed choice $\gamma \in \{0,1\}^*$ of the randomness of G (corresponding to the positions of the different random projections) such that the *deterministic* formula G_γ obtained from G and γ is correct on *every* input string w.

Claim 3.6. *Each deterministic sub-formula $G^{(j)}_\gamma(w)$ can be computed by a U_2-formula extended with parities at the leaves of size at most $O(\ell(N)^k) \leq N^{\varepsilon/2}$.*

Proof. Note that each bit of z can be computed from the input string w using an appropriate parity function (as described in Theorem 3.1). We argue that the leaves of $G^{(j)}_\gamma$ are precisely the leaves of the U_2-formula F_ℓ replaced by appropriate literals, constants, or parities. Recall that $G^{(j)}_\gamma$ applies F_ℓ to the string y obtained from z. However, since γ is fixed, the positions of z that are projected in order to compute y are also fixed, and so are the substrings of y describing the corresponding positions. Consequently, the size (i. e., number of leaves) of each $G^{(j)}_\gamma$ is at most the size of F_ℓ, which proves the claim. □

It follows from this claim that $G_\gamma(w)$ can be computed by a formula containing at most $N^{1+\varepsilon}$ leaves, and hence $\mathsf{Gap\text{-}MKtP}[2^{\beta n}, 2^{\beta n} + cn] \in U_2\text{-}\mathsf{Formula}\text{-}\oplus[N^{1+\varepsilon}]$. (Observe that we have used in a crucial way that the derandomized sub-formulas do not need to compute address functions to generate y from z.) This completes the proof of Theorem 1.1 Part (2).

3.2 Completing the proof of Theorem 1.1

In this section, we discuss how the argument presented in Section 3.1 can be adapted to establish the remaining items of Theorem 1.1.

First, note that Items (5) and (6) immediately follow from Item (2). This is because a parity gate over at most N input variables can be computed by B_2-formulas of size $O(N)$ and by U_2-formulas of size $O(N^2)$[7]. Consequently, using that formula size is measured with respect to the number of leaves, we immediately get $U_2\text{-}\mathsf{Formula}\text{-}\oplus[s(N)] \subseteq B_2\text{-}\mathsf{Formula}[s(N) \cdot O(N)]$ and $U_2\text{-}\mathsf{Formula}\text{-}\oplus[s(N)] \subseteq U_2\text{-}\mathsf{Formula}[s(N) \cdot O(N^2)]$.

In order to get Item (1), it is sufficient to compute an error-correcting code as in Theorem 3.1 using circuits of (almost) linear size. In other words, we need the entire codeword (and not just each output bit) to be computable from the input message using a circuit of size $O(N)$. The existence of such codes is well-known [54, 55]. The rest of the reduction produces an *additive* overhead in circuit size of at most $N^{1+\varepsilon}$ gates. For more details see [46].

[7]To see that, note that $\mathsf{PARITY}(x_1, \ldots, x_n) = (\mathsf{PARITY}(x_1, \ldots, x_{\lfloor \frac{n}{2} \rfloor}) \wedge \neg\mathsf{PARITY}(x_{\lfloor \frac{n}{2} \rfloor + 1}, \ldots, x_n)) \vee (\neg\mathsf{PARITY}(x_1, \ldots, x_{\lfloor \frac{n}{2} \rfloor}) \wedge \mathsf{PARITY}(x_{\lfloor \frac{n}{2} \rfloor + 1}, \ldots, x_n))$. If we recursively apply this formula, we obtain a formula of depth $2\log n + O(1)$. Consequently, the resulting formula for $\mathsf{PARITY}(x_1, \ldots, x_n)$ has size $O(n^2)$.

To establish Item (4), we use the following construction from [61].

Theorem 3.7 (Computing ECCs in parallel using majorities and few wires [61])**.** *For every depth $d' \geq 1$ there are constants $\delta(d') > 0$ and $b(d') \geq 1$ and a sequence $\{E_N\}_{N \in \mathbb{N}}$ of error-correcting codes $E_N \colon \{0,1\}^N \to \{0,1\}^M$ with the following properties:*

- *$E_N(x)$ can be computed by a uniform deterministic algorithm running in time $\mathsf{poly}(N)$.*
- *$M(N) = b \cdot N$.*
- *Any codeword $E_N(x) \in \{0,1\}^M$ that is corrupted on at most a δ-fraction of coordinates can be uniquely decoded to x by a uniform deterministic algorithm D running in time $\mathsf{poly}(M)$.*[8]
- *$E_N(x) \in \{0,1\}^M$ can be computed by a multi-output circuit from $\mathsf{MAJ}^0_{2d'}\{O(N^{1+(2/d')})\}$, where circuit size is measured by number of wires.*

Following the steps of the reduction described in Section 3.1, under the assumption that $\mathsf{EXP} \subseteq \mathsf{MAJ}^0_d\{\mathsf{poly}\}$ the final depth of the circuit solving $\mathsf{Gap}\text{-}\mathsf{MKtP}$ is $2d' + d + 1$, where the terms in this sum correspond respectively to the computation of the error-correcting code (for a choice of $d' \geq 1$), each (circuit) $G^{(j)}_\alpha$, and the topmost AND gate in G_α (constant bits can be produced in depth 1 from input literals). Similarly, the overall size (number of wires) of the circuit is $O(N^{1+(2/d')}) + O(N^{1+\varepsilon}) + O(N) \leq N^{1+(2/d')+\varepsilon}$.

Item (3) is established analogously to Item (2): Assuming $\mathsf{EXP} \subseteq \mathsf{TC}^0_2[\mathsf{poly}]$, we conclude that $G^{(j)}_\gamma(w)$ can be computed by a $\mathsf{THR}\text{-}\mathsf{THR}\text{-}\mathsf{XOR}$ circuit of size at most $N^{\varepsilon/2}$ (counted as a number of gates) and $G_\gamma(w)$ by an $\mathsf{AND}\text{-}\mathsf{THR}\text{-}\mathsf{THR}\text{-}\mathsf{XOR}$ circuit of size at most $N^{1+\varepsilon}$. Item (8) uses that parity gates can be simulated using $O(1)$ mod 6 gates. That is, assuming $\mathsf{EXP} \subseteq \mathsf{AC}^0[6]$, we conclude that $G^{(j)}_\gamma(w)$ can be computed by an $\mathsf{AC}^0[6]$ circuit of size at most $N^{\varepsilon/2}$ and $G_\gamma(w)$ by an $\mathsf{AC}^0[6]$ circuit of size at most $N^{1+\varepsilon}$.

Finally, we deal with case (7), which refers to branching program complexity. First, note that the parity of n bits can be computed by a branching program of size $O(n)$. In addition, if $f(x) = g(h_1(x), \ldots, h_k(x))$, each h_i has a branching program of size s, and g has a branching program of size t, then f has a branching program of size $\ell = O(t \cdot s)$. Finally, a conjunction of N branching programs of size ℓ has branching program size at most $O(N \cdot \ell)$. Combining these facts in the natural way yields case (7). This completes the proof of all cases in Theorem 1.1.

[8] This claim does not appear explicitly in [61], but it is an immediate consequence of their construction via well-known results on the decodability of direct product codes (see, e. g., [39] for more information about such codes). In more detail, the code provided by [60, Proposition 6.5] (full version of [61]) is obtained by constantly many compositions of the following product operation on codes, which decreases the distance but improves the parallel complexity of computing the code. Given a code $E \colon \{0,1\}^n \to \{0,1\}^\ell$ of relative distance δ, we define a new code $E' \colon \{0,1\}^{n^2} \to \{0,1\}^{\ell^2}$ of relative distance $\delta' = \delta^2$ as follows. We view a message $M \in \{0,1\}^{n \times n}$ as a square matrix, apply E to each row of M to obtain a new matrix $M_1 \in \{0,1\}^{n \times \ell}$, then apply E again to each column of M_1 to obtain a codeword $M_2 \in \{0,1\}^{\ell \times \ell}$. Note that, given a corrupted codeword $\widetilde{M}$, to recover the (i,j)-bit of the original message M it is sufficient to recover a $(1-\delta)$-fraction of the entries of the i-th row of the intermediate matrix. In turn, this can be achieved if at least a $(1-\delta)$-fraction of the columns of $\widetilde{M}$ are corrupted on less than a δ-fraction of entries (which must happen because we assumed a smaller distance $\delta' = \delta^2$). Using the obvious 2-step decoding procedure for E', it is not hard to see that E' is efficiently decodable if E is efficiently decodable.

4 Hardness magnification via anti-checkers

4.1 Proof of Theorem 1.4 (magnification for MCSP)

In this section, we derive Theorem 1.4 from Lemma 4.1, whose proof appears in Section 4.2. Informally, an anti-checker (see [38]) for a function f is a multi-set of input strings such that any circuit of bounded size that does not compute f is incorrect on at least one of these strings.[9]

Lemma 4.1 (Anti-Checker Lemma). *If* $\mathsf{NP} \subseteq \mathsf{Circuit}[\mathsf{poly}]$ *there is a constant* $k \in \mathbb{N}$ *for which the following hold. For every sufficiently small* $\beta > 0$*, there is a circuit* C *of size* $\leq 2^{n+k\beta n}$ *that when given as input a truth-table* $\mathsf{tt}(f) \in \{0,1\}^N$*, where* $f \colon \{0,1\}^n \to \{0,1\}$*, outputs* $t = 2^{10\beta n}$ *strings* $y_1, \ldots, y_t \in \{0,1\}^n$ *such that if* $f \notin \mathsf{Circuit}[2^{\beta n}]$ *then every circuit of size* $\leq s$ *where* $s = 2^{\beta n}/10n$ *fails to compute* f *on at least one of these strings.*

The Anti-Checker Lemma is a powerful tool that might be of independent interest. It says that anti-checkers of bounded size for functions requiring circuits of size $2^{o(n)}$ can be produced in time that is almost-linear in the size of the function (viewed as a string), under the assumption that circuit lower bounds do not hold.[10]

Proof of Theorem 1.4. Assume that $\mathsf{NP} \subseteq \mathsf{Circuit}[\mathsf{poly}]$. We prove that for every given $\varepsilon > 0$ there exists a small enough $\beta > 0$ such that $\mathsf{Gap\text{-}MCSP}[2^{\beta n}/10n, 2^{\beta n}] \in \mathsf{Circuit}[N^{1+\varepsilon}]$.

We consider the problem $\mathsf{Succinct\text{-}MCSP}$. Its input instances are of the form

$$\langle 1^n, 1^s, 1^t, (x_1, b_1), \ldots, (x_t, b_t) \rangle,$$

where $x_i \in \{0,1\}^n$ and $b_i \in \{0,1\}$, $i \in [t]$. Note that each instance can be encoded by a string of length exactly $m = n+1+s+1+t+1+t \cdot (n+1)$. An input string is a positive instance if and only if it is in the appropriate format and there exists a circuit D over n input variables and of size at most s such that $D(x_i) = b_i$ for all $i \in [t]$. Note that the problem is in NP as a function of its total input length m. Under the assumption that NP is easy for non-uniform circuits, there exists $\ell \in \mathbb{N}$ such that $\mathsf{Succinct\text{-}MCSP}$ can be solved by circuits E_m of size m^ℓ on every large enough input length m.

Take $\beta = \varepsilon/(100 \cdot \ell \cdot k)$, where k is the constant from Lemma 4.1. In order to construct a circuit for $\mathsf{Gap\text{-}MCSP}$, first we reduce this problem to an instance of $\mathsf{Succinct\text{-}MCSP}$ of length m using Lemma 4.1, then we invoke the m^l-sized circuit for this problem. More precisely, on an input $f \colon \{0,1\}^n \to \{0,1\}$, we use the circuit C (as in Lemma 4.1) to produce a list of strings $y_1, \ldots, y_t \in \{0,1\}^n$, generate from this list and f the input instance $z = \langle 1^n, 1^s, 1^t, ((y_1, f(y_1)), \ldots, (y_t, f(y_t))\rangle$, for parameters $s = 2^{\beta n}/10n$, $t = 2^{10\beta n}$, $m = \mathsf{poly}(n) \cdot 2^{10\beta n}$, and output $E_m(z)$.

Correctness follows immediately from Lemma 4.1 and our choice of parameters. Indeed, if $f \in \mathsf{Circuit}[2^{\beta n}/10n]$ then no matter the choice of $y_1, \ldots, y_t$ the circuit E_m accepts z thanks to our choice of $s = 2^{\beta n}/10n$. On the other hand, when $f \notin \mathsf{Circuit}[2^{\beta n}]$ then by Lemma 4.1 every circuit of size s fails on some string from the list, and consequently $E_m(z) = 0$.

[9] Lipton and Young [38] discuss several versions of the notion of 'anti-checkers'. The main version of anti-checkers they work with is defined as a multiset of inputs such that each small circuit fails on a significant fraction of inputs. In contrast, our anti-checkers are weaker—it suffices that each small circuit fails on at least one input from the set.

[10] We have made no attempt to optimize the constants in Lemma 4.1.

We give an upper bound on the total circuit size using the choice of β. Circuit C has size at most $2^{n+k\beta n} \leq N^{1+\varepsilon}/3$. In addition, producing the input z can be done from f and $y_1, \dots, y_t$ by a circuit of size at most $O(t \cdot N) \leq N^{1+\varepsilon}/3$, since each address function can be computed in linear size $O(N)$ (see, e. g., [63]). Finally, E_m has size at most $m^\ell \leq N^{1+\varepsilon}/3$. Overall, it follows that $\mathsf{Gap\text{-}MCSP}[2^{\beta n}/10n, 2^{\beta n}]$ is computable by circuits of size $N^{1+\varepsilon}$. □

4.2 Proof of Lemma 4.1 (Anti-Checker Lemma)

This section is dedicated to the proof of Lemma 4.1. This completes the proof of Theorem 1.4. We start with a high-level exposition of the argument.

Proof idea. We take $\beta \to 0$, for simplicity of the exposition. In principle, the challenge is to *construct* the list of strings from the description of f using a circuit of size $N^{1+o(1)}$, given that the *existence* of such strings is guaranteed by [38]. But it is not clear how to use this existential result and the assumption that NP has polynomial size circuits to construct *almost-linear* size circuits for this task. In order to achieve this, we use a *self-contained* argument that produces the strings one by one until very few circuits of bounded size are consistent with the values of f on the partial list of strings. We then find polynomially many (in n) *additional* strings that eliminate the remaining circuits, completing the list of strings.[11]

To produce the i-th string $y_i \in \{0,1\}^n$ given $y_1, \dots, y_{i-1} \in \{0,1\}^n$ and f, we estimate the number of circuits of size $\leq 2^{\beta n}/10n$ that agree with f over all strings in $\{y_1, \dots, y_i\}$. We show that *some string* y_i will reduce the number of consistent circuits from the previous round by a factor of (roughly) $1 - 1/n$ as long as there are at least (roughly) n^2 surviving circuits (this is a combinatorial existential proof that relies on the lower bound on the circuit complexity of f). In fact, as discussed below, we will be able to find such string y_i efficiently. This means that after $2^{O(\beta n)} = N^{o(1)}$ rounds we will reduce the fraction of consistent circuits of size $\leq 2^{\beta n}/10n$ to $(1-1/n)^{N^{o(1)}} \leq 1/\mathrm{e}^{N^{o(1)}/n}$. That is, we will show that at most $2^{O(\beta n)} = N^{o(1)}$ rounds suffice to produce the required set of strings (modulo handling the few surviving circuits). The existence of a *good* string y_i is at the heart of our argument, and we defer the exposition of this result to the formal proof.

In order to find y_i efficiently, in each round, we exhaustively check each of the N candidate strings y_i. As we will explain soon, estimating the number of surviving circuits after picking a new candidate string y_i can be done by a circuit of size $N^{o(1)}$ given access to $y_1, \dots, y_i$ and to the corresponding bits $f(y_1), \dots, f(y_i)$.[12] In summary, there are $N^{o(1)}$ rounds, and in each one of them we can find a good string y_i using a circuit of size $N^{1+o(1)}$. We remark that it will also be possible to produce the additional strings in circuit complexity $N^{o(1)}$, so that the complete list $y_1, \dots, y_t$ can be computed from f by a circuit of size $N^{1+o(1)}$.

It remains to explain how to fix a good string in each round which could be, in principle, an infeasible task. We simply pick the most promising string, using that we can give an upper bound on the complexity of estimating the number of surviving circuits. The latter relies on the assumed inclusion $\mathsf{NP} \subseteq \mathsf{Circuit}[\mathsf{poly}]$. Indeed, from this assumption it follows that the polynomial hierarchy

[11] In particular, our argument implies the worst-case version of the anti-checker result from [38] with slightly different parameters. Lipton and Young [38] establish the existence of a stronger version of anticheckers defined as a multiset of inputs such that every small circuit fails on a big fraction of inputs from the multiset.

[12] In each round we will generate not only y_i but also $f(y_i)$ by a circuit of size $N^{1+o(1)}$.

PH $\subseteq$ Circuit[poly], and it is known that *relative approximate counting* can be done in the polynomial hierarchy. In our formal proof, we take a slightly more direct route to compute the relative approximations. Crucially, as described in the paragraph above, the input length of each sub-problem that we need to solve (in order to estimate the number of surviving circuits) is $\leq N^{o(1)}$ (using that i is at most $N^{o(1)}$), so a polynomial overhead will not be an issue when solving a sub-task of input length $N^{o(1)}$. This completes the sketch of the proof.

Comparison to Bshouty et al. [9]. A reviewer pointed out to us that the algorithm from our proof is similar to the classical $\mathsf{ZPP}^{\mathsf{NP}}$ learning algorithm from [9] with an extra property that the queries to the NP oracle have small fan-in. In more detail, Bshouty et al. show that it is possible to learn efficiently circuits of size s with the equivalence queries. Their algorithm proceeds in rounds by searching for strings $y_1,\dots,y_i$ so that in each round i it finds a new string y_i significantly reducing the number of functions computable by circuits of size s and consistent with the target function restricted to $y_1,\dots,y_{i-1}$. The string y_i is obtained as a response to the equivalence query on a properly designed concept corresponding to the restriction of the target function to $y_1,\dots,y_{i-1}$. Our algorithm proceeds in the same way, except that we cannot use equivalence queries. (If the target function was, e. g., SAT it would be possible to simulate equivalence queries with NP oracles but this is not our case.) Instead of using equivalence queries, we go through all N possible candidate strings y and select the right string using approximate counting. The main point is that the oracle queries needed to perform the approximate counting represent predicates with inputs of size $N^{o(1)}$. This allows us to get in the end a very efficient algorithm. Our proof differs slightly from Bshouty et al. also in the proof of the existence of a suitable string y_i, which reduces the number of consistent functions, see Lemma 4.7.

We proceed with a formal proof of Lemma 4.1. Let R be a polynomial-time relation, where R is a subset of $\bigcup_m \{0,1\}^m \times \{0,1\}^{q(m)}$ for some polynomial q. For every x, we use $R_\#(x)$ to denote the set $|\{y \in \{0,1\}^{q(|x|)} : (x,y) \in R\}|$. A randomized algorithm $\mathbf{\Pi}$ is called an (ε,δ)-*approximator* for R if for every input x it holds that

$$\Pr\big[\,\big|\mathbf{\Pi}(x) - R_\#(x)\big| \;\geq\; \varepsilon(|x|)\cdot R_\#(x)\,\big] \;\leq\; \delta(|x|).$$

Theorem 4.2 (Relative approximate counting in $\mathsf{BPP}^{\mathsf{NP}}$ ([57]; see, e. g., [19, Section 6.2.2]))**.** *For every polynomial-time relation R and every polynomial p, there exists a probabilistic polynomial-time algorithm $\boldsymbol{A}$ with access to a* SAT *oracle that is an $(1/p(m), 2^{-p(m)})$-approximator for R over inputs x of length m.*

Corollary 4.3. *Assume* NP $\subseteq$ Circuit[poly]. *For every polynomial-time relation R and for each $m \geq 1$, there is a multi-output circuit $C_R : \{0,1\}^m \to \{0,1\}^{\mathsf{poly}(m)}$ of polynomial size such that on every input $x \in \{0,1\}^m$,*

$$(1 - 1/m^2)\cdot R_\#(x) \;\leq\; C_R(x) \;\leq\; (1 + 1/m^2)\cdot R_\#(x).$$

Proof. This follows from Theorem 4.2 (using $p(m) = m^2$) by non-uniformly fixing the randomness of the algorithm, replacing the SAT oracle using the assumption that NP has small circuits, and translating the resulting deterministic algorithm into a boolean circuit. □

We define a relation Q. The first input x is of the form $\langle 1^n, 1^s, 1^i, 1^{t-i}, (z_1,b_1),\dots,(z_i,b_i), 1^{(t-i)(n+1)}\rangle$, where $z_j \in \{0,1\}^n$ and $b_j \in \{0,1\}$ for $1 \leq j \leq i \leq t = 2^{10\beta n}$ (t is used here to pad the input appropriately).

The second input is a string w of length $m^{1/5}$ (for $m = |x|$) that is interpreted as a boolean circuit C_w over n input variables and of size at most s. We let $(x,w) \in Q$ if and only if $C_w(z_j) = b_j$ for all $j \in [i]$. Note that Q is a polynomial-time relation.

We employ circuits obtained from Corollary 4.3 using parameters $s = 2^{\beta n}/10n$ and $1 \leq i \leq t$, where $t = 2^{10\beta n}$. The following result is immediate from Corollary 4.3 given that for our choice of parameters $m = \mathsf{poly}(2^{\beta n})$.

Proposition 4.4 (Circuits for approximate counting)**.** *There is a constant $k_1 \in \mathbb{N}$ for which the following holds. For every $n \geq 1$, let $s = 2^{\beta n}/10n$, $t = 2^{10\beta n}$, $1 \leq i \leq t$. Then there is a multi-output circuit $C_{n,i}$ of size $\leq 2^{k_1 \beta n}$ that outputs $\leq 2^{k_1 \beta n}$ bits such that on every input $a = ((z_1,b_1),\dots,(z_i,b_i)) \in \{0,1\}^{i\cdot(n+1)}$,*

$$(1 - 1/n^{10}) \cdot Q_{\#}(x) \;\leq\; C_{n,i}(a) \;\leq\; (1 + 1/n^{10}) \cdot Q_{\#}(x),$$

where $x = x(a) = \langle 1^n, 1^s, 1^i, 1^{t-i}, (z_1,b_1),\dots,(z_i,b_i), 1^{(t-i)(n+1)} \rangle$.

The next step is to guarantee that once just a few circuits remain consistent with f over our partial list of strings (as described in the proof sketch above), we can efficiently find a small number of strings to eliminate all of them.

Lemma 4.5 (Listing the remaining circuits)**.** *Assume $\mathsf{NP} \subseteq \mathsf{Circuit}[\mathsf{poly}]$. There exists a constant $k_2 \in \mathbb{N}$ such that for each sufficiently big n the following holds. Let $a = ((z_1,b_1),\dots,(z_{t'},b_{t'}))$, where $t' \leq t = 2^{10\beta n}$, and $x = x(a)$ be the corresponding input of Q. There is a circuit $D_{n,t'}$ of size $\leq 2^{k_2 \beta n}$ such that if $Q_{\#}(x) \leq n^3$, then $D_{n,t'}(a)$ outputs a string describing all circuits of size $s = 2^{\beta n}/10n$ consistent with the partial list a.*

Proof. It follows from $\mathsf{NP} \subseteq \mathsf{Circuit}[\mathsf{poly}]$ using a standard argument that $\mathsf{PH} \subseteq \mathsf{Circuit}[\mathsf{poly}]$. In addition, it is not hard to define a relation in PH (using a padded input containing the string 1^t) that checks if a given input a satisfies $Q_{\#}(x(a)) \leq n^3$. Consequently, checking if a string λ describes a list of circuits of size s consistent with a can be done by a circuit of size at most $\mathsf{poly}(t)$. Using again that $\mathsf{NP} \subseteq \mathsf{Circuit}[\mathsf{poly}]$ and a self-reduction, we obtain circuits $D_{n,t'}$ as in the statement of the lemma. More precisely, $D_{n,t'}$ keeps generating new circuits of size s consistent with a until it generates all of them. Since generating each bit of each new circuit can be done by solving a task in PH and checking if we have generated already all such circuits is in PH as well, this proves the lemma. □

Lemma 4.6 (Completing the list of strings)**.** *There is a constant $k_3 \in \mathbb{N}$ for which the following holds. For every $n \geq 1$ there is a circuit E_n of size $\leq 2^{n+k_3\beta n}$ that given as an input a truth-table $f \in \{0,1\}^{2^n}$ and a string $w \in \{0,1\}^{2^{\beta n}}$ describing a circuit C_w of size $s \leq 2^{\beta n}/10n$ that does not compute f, $E_n(f,w)$ outputs a string y such that $C(y) \neq f(y)$.*

Proof. First, E_n evaluates C_w on every string $z \in \{0,1\}^n$. This can be easily done by a circuit of size $2^n \cdot \mathsf{poly}(|w|)$ under a reasonable encoding of the circuit C_w. Then E_n inspects one-by-one each tuple $C_w(z), f_z$ and outputs the first string where C_w and f differ. Note that a circuit of size $\leq 2^n \cdot \mathsf{poly}(|w|)$ can print this string from the truth-table of f and C_w. It follows that the overall complexity of E_n is $2^{n+k_3\beta n}$ for some constant k_3. □

The previously established results will allow us to find in each round a string y_i that significantly reduces the number of remaining circuits (while at least one such string exists), and then to complete the list so that no circuit of bounded size is consistent with all strings in the final list. Each y_i will be found by going over all candidates and using Proposition 4.4 or Lemma 4.5 and Lemma 4.6 to pick the right one. We show next that if f is hard and a reasonable number of circuits of bounded size are consistent with the current list of strings, then a good string y_i exists.

For convenience, we introduce a function to capture the fraction of strings encoding circuits that are consistent with a set of inputs and their corresponding labels. Given $a = ((z_1,b_1),\dots,(z_i,b_i))$, let $x = x(a)$ be the corresponding input to Q under our choice of parameters. Furthermore, let $m = |x|$, and recall that $Q \subseteq \bigcup_{m\geq 1}\{0,1\}^m \times \{0,1\}^{m^{1/5}}$. We assume without loss of generality (using appropriate padding) that the encoding of x has a fixed length $m = m(n)$ and note that m will be actually of size $\leq 2^{11\beta n}$, which will be useful when giving upper bounds on the number of necessary rounds. We let $\phi(a) \in [0,1]$ denote the ratio $Q_\#(x(a))/2^{m^{1/5}}$. (Thus in our formal argument we count circuits using their descriptions as binary strings.)

Lemma 4.7 (Existence of a good string y_i)**.** *For every integer $i \geq 1$ and for every $z_1,\dots,z_{i-1} \in \{0,1\}^n$, let $a = ((z_1, f(z_1)),\dots,(z_{i-1}, f(z_{i-1})))$. If*

$$f \notin \mathsf{Circuit}[2^{\beta n}] \quad \textit{and} \quad Q_\#(x(a)) \geq 4n^2,$$

then there is some string $y_i \in \{0,1\}^n$ such that if a' denotes the sequence a augmented with $(y_i, f(y_i))$, then

$$\phi(a') \;\leq\; \phi(a)\cdot(1-1/2n).$$

Proof. The argument is inspired by a combinatorial principle discussed in [33]. An alternative approach can be found in [9].[13]

Consider the tuple a and the string $x = x(a)$ as in the statement of the lemma. Moreover, let $Q(x) = \{w \in \{0,1\}^{m^{1/5}} : (x,w) \in Q\}$. For convenience, let $r = |Q(x)| = Q_\#(x) \geq 4n^2$, using our assumption. Define an auxiliary undirected bipartite graph $G = (L,R,E)$ as follows. Set $L = \{0,1\}^n$, $R = \binom{Q(x)}{n}$, and $(y,\{w^1,\dots,w^n\}) \in E(G)$ if and only if for $\leq n/2$ of the circuits C_{w^i} we have $f(y) = C_{w^i}(y)$.

Note that for any right vertex $v = (w^1,\dots,w^n) \in R$ there is a left vertex $y \in L$ such that $(y,v) \in E$. If not, then $D = \mathsf{Majority}_n(C_{w^1}(x),\dots,C_{w^n}(x))$ is a circuit that computes f on every input string y. The size of D is at most $n\cdot(2^{\beta n}/10n) + 5n \leq 2^{\beta n}$, using the definition of Q and that the majority function can be computed (with room to spare) by a circuit of size at most $5n$ [63]. This contradicts the hardness of f.

By an averaging argument, there is a left vertex y^* that is connected to at least $|R|/|L| = \binom{r}{n}/2^n$ vertices in R. We show below (Claim 4.8) that for at least $r/2n$ strings $w \in Q(x)$, the corresponding circuit C_w satisfies $C_w(y^*) \neq f(y^*)$. This implies that by taking y^* as the string y_i described in the statement of the lemma, we get $Q_\#(x(a')) \leq r - r/2n = r(1-1/2n)$, and consequently

$$\phi(a') \;=\; \frac{Q_\#(x(a'))}{2^{m^{1/5}}} \;\leq\; \frac{r(1-1/2n)}{2^{m^{1/5}}} \;=\; \frac{Q_\#(x(a))\cdot(1-1/2n)}{2^{m^{1/5}}} \;=\; \phi(a)\cdot(1-1/2n).$$

[13] The proof from [9] was pointed out to us by a reviewer. It is conceptually simpler and suffices to prove essentially the same result. Informally, the proof proceeds in the following way: randomly choose $O(n)$ circuits of size $s = 2^{\beta n}/O(n)$, and let C be their majority, so C is of size $2^{\beta n}$. With high probability, for every input x at least $1/3$ of the consistent circuits agree with C. But because f is hard for the size of C there is a string y_i on which C fails and hence $1/3$ of the consistent circuits fail. This shrinks the number of the remaining consistent circuits by a constant fraction (instead of $1-1/2n$).

Claim 4.8. *Let $y^* \in L$ be a left-vertex connected to at least $\binom{r}{n} \cdot 2^{-n}$ right-vertices in R, where $r \geq 4n^2$ and n is sufficiently large. Then, for at least $r/2n$ distinct strings $w \in Q(x)$, we have $C_w(y^*) \neq f(y^*)$.*

Proof. The claim follows using a standard counting argument. If the conclusion were false, the vertex y^* would be adjacent to *strictly* fewer than

$$\sum_{j=0}^{n/2} \binom{r/2n}{\frac{n}{2}+j} \cdot \binom{r}{\frac{n}{2}-j} \leq \binom{r}{n} \cdot 2^{-n} \quad \text{(see upper bound below)}$$

vertices in R, which is a contradiction. (For simplicity we made the assumption that n is even and $r/2n$ is an integer.) It remains to verify this inequality, which can be done using some careful estimates. First, note that

$$\begin{aligned}
\sum_{j=0}^{n/2} \binom{r/2n}{\frac{n}{2}+j}\binom{r}{\frac{n}{2}-j} &\leq \sum_{j=0,\dots,\frac{n}{2}-1} \frac{r^n/(2n)^{\frac{n}{2}+j}}{(\frac{n}{2}+j)!(\frac{n}{2}-j)!} + \frac{r^n}{n!(2n)^n} && (\text{using } \binom{n}{k} \leq \frac{n^k}{k!}) \\
&\leq \sum_{j=0,\dots,\frac{n}{2}-1} \frac{\mathrm{e}^n r^n/(2n)^{\frac{n}{2}+j}}{\mathrm{e}^2(\frac{n}{2}+j)^{\frac{n}{2}+j}(\frac{n}{2}-j)^{\frac{n}{2}-j}} + \frac{\mathrm{e}^n r^n}{\mathrm{e}n^n(2n)^n} && (\text{since } \mathrm{e}\left(\frac{n}{\mathrm{e}}\right)^n \leq n!) \\
&\leq \sum_{j=0,\dots,\frac{n}{2}-1} \frac{\mathrm{e}^n r^n/(2n)^{\frac{n}{2}}}{\mathrm{e}^2(\frac{n}{2})^j(\frac{n}{2}+j)^{\frac{n}{2}}(\frac{n}{2}-j)^{\frac{n}{2}}} + \frac{\mathrm{e}^n r^n}{\mathrm{e}n^n(2n)^n} && (*)
\end{aligned}$$

By considering the cases $j < \frac{n}{4}$ and $\frac{n}{2} > j \geq \frac{n}{4}$, we get $(\frac{n}{2})^j((\frac{n}{2})^2 - j^2)^{\frac{n}{2}} \geq (n/8)^{3n/4}$, so

$$\begin{aligned}
(*) &\leq \sum_{j=0,\dots,\frac{n}{2}-1} \frac{\mathrm{e}^n r^n}{\mathrm{e}^2(n/8)^{3n/4}(2n)^{n/2}} + \frac{\mathrm{e}^n r^n}{\mathrm{e}n^n(2n)^n} \\
&\leq \frac{n\mathrm{e}^n r^n}{\mathrm{e}^2(n/8)^{3n/4}(2n)^{n/2}} \leq \frac{\sqrt{2\pi}r^n}{\mathrm{e}^2 n^{1/2}(2n)^n} \leq \frac{\sqrt{2\pi}r^r r^{1/2}}{\mathrm{e}^2(r-n)^{r-n+1/2}n^{n+1/2}} \cdot \frac{1}{2^n} \\
&\leq \binom{r}{n}/2^n,
\end{aligned}$$

where n is assumed to be sufficiently large, $r > n$, and the last inequality makes use of Stirling's approximation $\sqrt{2\pi}(\frac{n}{\mathrm{e}})^n n^{1/2} \leq n! \leq \mathrm{e}(\frac{n}{\mathrm{e}})^n n^{1/2}$. This completes the proof of Claim 4.8. □

This completes the proof of Lemma 4.7. □

We are ready to combine these results and define a circuit C of size $\leq 2^{n+k\beta n}$ with the property stated in Lemma 4.1. This circuit on an input $f \in \{0,1\}^N$ where $N = 2^n$ computes as follows.

1. C sequentially computes the string $a^{(i)} = (y_1, f(y_1)), \dots, (y_i, f(y_i))$ for $1 \leq i \leq t'$ and $t' = 2^{10\beta n} - n^3$.

During stage i, C inspects all strings $y \in \{0,1\}^n$, using the circuit $C_{n,i}$ (Proposition 4.4) to fix y_i as the string that minimizes $C_{n,i}(a^{(i)})$.

2. C uses the circuit $D_{n,t'}$ (Lemma 4.5) to print the descriptions of n^3 circuits of size at most $s = 2^{\beta n}/10n$.

3. Finally, C invokes n^3 copies of the circuit E_n (Lemma 4.6) to complete the list $y_1, \ldots, y_t$ of strings, where $t = t' + n^3 = 2^{10\beta n}$.

Correctness of the construction follows from the properties of the circuits $C_{n,i}$, $D_{n,t'}$, and E_n in combination with Lemma 4.7. More precisely, if $f \notin \mathsf{Circuit}[2^{\beta n}]$, then for every $1 \leq i \leq t'$, either $\phi(a^{(i)}) \leq (1-1/4n)^i$ or $Q_\#(x(a^{(i-1)})) < 4n^2$. To see this, note that if the latter condition does not hold, then for some string y^* as in Lemma 4.7 we get with respect to the corresponding extension $a^{(i)}$ that $\phi(a^{(i)}) \leq \phi(a^{i-1}) \cdot (1-1/2n)$. Since C tries all strings during its computation in step **1** when in stage i, and the relative approximation given by circuit $C_{n,i}$ is sufficiently precise, we are guaranteed in this case (using an inductive argument) to fix a string y_i such that $\phi(a^{(i)}) \leq \phi(a^{i-1}) \cdot (1-1/4n) \leq (1-1/4n)^i$. On the other hand, if the condition $Q_\#(x(a^{(i-1)})) < 4n^2$ holds for some $i \leq t'$, then by monotonicity it is maintained until we reach $i = t'$. Consequently, using that initially $\phi(\varepsilon) = 1$, $t' = 2^{10\beta n} - n^3$, $m(n) \leq 2^{11\beta n}$, and recalling that the second input of the relation Q has length $m^{1/5}$ and that this parameter is related to the definition of ϕ, when C reaches $i = t'$ at the end of step **1** we have

$$\begin{aligned} Q_\#(x(a^{(t')})) &\leq \max\{4n^2, (1-1/4n)^{t'} \cdot 2^{m^{1/5}}\} \\ &\leq n^3. \end{aligned}$$

This implies using Lemmas 4.5 and 4.6 and the description of C that if $f \notin \mathsf{Circuit}[2^{\beta n}]$ then every circuit of size at most $s = 2^{\beta n}/10n$ disagrees with f on some input string among $y_1, \ldots, y_t$.

Finally, we prove an upper bound on the circuit size of C. For every $i \leq t'$ in step **1** and each string $y \in \{0,1\}^n$, C feeds $C_{n,i}$ with the appropriate bit in the input string f and the previously computed string $a^{(i-1)}$. This produces an estimate $v_y \in \mathbb{N}$ represented as a string of length $2^{O(\beta n)}$ that is stored as a pair (y, v_i). Using Proposition 4.4, all pairs (y, v_y) can be simultaneously computed by a circuit of size at most $2^n \cdot 2^{O(\beta n)}$. By inspecting each such pair in sequence, C can pick the string $y_i \in \{0,1\}^n$ minimizing v_i using a sub-circuit of size $2^n \cdot \mathsf{poly}(2^{O(\beta n)})$. Also note that the bit $f(y_i)$ can be easily computed from y_i and f by a circuit of size $O(N \log N)$. ($f(y_i)$ is the disjunction of N depth-2 formulas and each formula is an AND of the indicator function (over y) of a fixed input and of a bit (matching the fixed input) in f). Therefore, each stage i can be done by a circuit of size at most $2^{n+O(\beta n)}$, and since there are $t' \leq 2^{10\beta n}$ stages, the computation in step **1.** can be done by a circuit of size $2^{n+O(\beta n)}$. Lastly, steps **2** and **3** can be implemented by circuits of size at most $2^{O(\beta n)}$ and $2^{(1+O(\beta))n}$, resp., using the upper bounds on circuit size provided by Lemma 4.5 and Lemma 4.6, resp., and the description of C. It follows that the overall circuit size of C is at most $2^{n+k\beta n}$, where k is a constant that only depends on the circuits provided by the initial assumption that $\mathsf{NP} \subseteq \mathsf{Circuit}[\mathsf{poly}]$.

A remark on formulas vs. circuits. An obstacle to producing the anti-checker using formulas of size $N^{1+o(1)}$ under the assumption that $\mathsf{NP} \subseteq \mathsf{Formula}[\mathsf{poly}]$ comes from the sequential aspect of the construction. A string y_j produced after the j-th round is inspected during each subsequent round of the construction. In the case of formulas, the corresponding bits need to be recomputed each time, and the overall complexity becomes prohibitive.

4.3 The Anti-Checker Hypothesis

The existence of anti-checkers of bounded size witnessing the hardness of Boolean functions is far from obvious. In this section, we explore consequences of a hypothetical phenomenon manifesting on a higher level: the existence of a small collection of anti-checker sets witnessing hardness of all hard functions. We show that a certain formulation of this Anti-Checker Hypothesis (AH) implies unconditional lower bounds. Complementing this result, we prove unconditionally that (AH) holds for functions that are hard in the average case.

For simplicity, we adopt a concrete setting of parameters for the hypothesis and in the results presented in this section. Understanding the validity of (AH) with respect to other non-trivial setting of parameters would also be interesting.

The Anti-Checker Hypothesis (AH). For every $\lambda \in (0,1)$, there are $\varepsilon > 0$ and a collection $\mathcal{Y} = \{Y_1,\dots,Y_\ell\}$ of sets $Y_i \subseteq \{0,1\}^n$, where $\ell = 2^{(2-\varepsilon)n}$ and each $|Y_i| = 2^{n^{1-\varepsilon}}$, for which the following holds.

If $f\colon \{0,1\}^n \to \{0,1\}$ and $f \notin \mathsf{Circuit}[2^{n^\lambda}]$, then some set $Y \in \mathcal{Y}$ forms an anti-checker for f: For each circuit C of size $2^{n^\lambda}/10n$, there is an input $y \in Y$ such that $C(y) \neq f(y)$.

The Anti-Checker Hypothesis can be shown to imply the hardness of a specific meta-computational problem in NP.

Definition 4.9 (Succinct MCSP)**.** Let $s,t\colon \mathbb{N} \to \mathbb{N}$ be functions. The Succinct Minimum Circuit Size Problem with parameters s and t, abbreviated $\mathsf{Succinct\text{-}MCSP}(s,t)$, is the problem of deciding given a list of $t(n)$ pairs (y_i,b_i), where $y_i \in \{0,1\}^n$ and $b_i \in \{0,1\}$, if there exists a circuit C of size $s(n)$ computing the partial function defined by these pairs, i. e., $C(y_i) = b_i$ for every $i \in [t]$.

Note that $\mathsf{Succinct\text{-}MCSP}(s,t) \in \mathsf{NP}$ whenever s and t are constructive functions.

Theorem 4.10. *Assume* (AH) *holds, and let $\varepsilon = \varepsilon(\lambda) > 0$ be the corresponding constant for $\lambda = 1/2$. Then* $\mathsf{Succinct\text{-}MCSP}(2^{n^{1/2}}/10n, 2^{n^{1-\varepsilon}}) \notin \mathsf{Formula}[\mathsf{poly}]$. *In particular,* $\mathsf{NP} \nsubseteq \mathsf{Formula}[\mathsf{poly}]$.

Proof. The proof is by contradiction. Take $\lambda = 1/2$ in the Anti-Checker Hypothesis, and let $\varepsilon = \varepsilon(\lambda) > 0$ be the given constant. Let $F_m\colon \{0,1\}^N \to \{0,1\}$ be a formula for $\mathsf{Succinct\text{-}MCSP}(2^{n^{1/2}}/10n, 2^{n^{1-\varepsilon}})$. Assume F_m has size m^k, where $m \leq \mathsf{poly}(n)\cdot 2^{n^{1-\varepsilon}}$ is the total input length for this problem. We argue below that from these assumptions it follows that $\mathsf{Gap\text{-}MCSP}[2^{n^{1/3}}, 2^{n^{2/3}}] \in \mathsf{Formula}[N^{2-\delta}]$ for some $\delta > 0$. This contradicts Theorem 1.5 if α is taken to be a sufficiently small constant, which completes the proof.

We define a formula $E\colon \{0,1\}^N \to \{0,1\}$ that solves $\mathsf{Gap\text{-}MCSP}[2^{n^{1/3}}, 2^{n^{2/3}}]$. It projects the appropriate bits of the input f to produce $T = 2^{(2-\varepsilon)n}$ instances of the problem $\mathsf{Succinct\text{-}MCSP}(2^{n^{1/2}}/10n, 2^{n^{1-\varepsilon}})$ obtained from f and from the collection $\mathcal{Y}$ in the natural way. The formula E is defined as the conjunction of T independent copies of the formula F_m from above. Note that E has at most $T \cdot m^k \leq N^{2-\delta}$ leaves, where $\delta = \delta(\varepsilon) > 0$. Finally, it is easy to see that it correctly solves $\mathsf{Gap\text{-}MCSP}$ using our choice of parameters and (AH). $\square$

We say that a Boolean function f with n inputs is hard on average for circuits of size s if every circuit of size s fails to compute f on at least $1/s$ fraction of all inputs.

Proposition 4.11 (Average-Case AH). *For every $\lambda \in (0,1)$ there is $\varepsilon > 0$ such that for every large enough $n \in \mathbb{N}$ there is a collection $\mathcal{Y} = \{Y_1, \ldots, Y_\ell\}$ of $\ell = 2^n$ sets $Y_i \subseteq \{0,1\}^n$ of size $2^{n^{1-\varepsilon}}$ for which the following holds. If $f\colon \{0,1\}^n \to \{0,1\}$ is hard on average for circuits of size 2^{n^λ}, then some set $Y \in \mathcal{Y}$ constitutes an anti-checker for f: For each circuit C of size 2^{n^λ} there is a string $y \in Y$ such that $C(y) \neq f(y)$.*

Proof. Let $\mathcal{H}$ be the set of all Boolean functions f over n inputs that are hard on average for circuits of size $s = 2^{n^\lambda}$. Then we can generate anti-checkers for $f \in \mathcal{H}$ by choosing n-bit strings uniformly at random: for each $i \in [2^n]$, we let $\boldsymbol{Y_i}$ be the set obtained by sampling with repetition $2^{n^{1-\varepsilon}}$ random strings in $\{0,1\}^n$, where $1 - \varepsilon > \lambda$. Then, for every large enough n, for each circuit C of size at most 2^{n^λ} and for each $f \in \mathcal{H}$,

$$\Pr[C|_{\boldsymbol{Y_i}} \equiv f|_{\boldsymbol{Y_i}}] \;\leq\; (1 - 1/2^{n^\lambda})^{2^{n^{1-\varepsilon}}} \;\leq\; \exp(-2^{n^{1-\varepsilon}}/2^{n^\lambda}).$$

Now by a union bound over all such circuits, for a fixed $f \in \mathcal{H}$ we get

$$\Pr[\boldsymbol{Y_i} \text{ is not an anti-checker set for } f] \;\leq\; \exp(O(n \cdot 2^{n^\lambda})) \cdot \exp(-2^{n^{1-\varepsilon}}/2^{n^\lambda}) \;<\; 1/4,$$

where the last inequality used our choice of ε. Finally,

$$\Pr[\exists f \in \mathcal{H} \text{ s.t. none of } \boldsymbol{Y_1}, \ldots, \boldsymbol{Y_{2^n}} \text{ is an anti-checker set for } f] \;\leq\; 2^{2^n} \cdot (1/4)^{2^n} \;<\; 1.$$

There is therefore a collection $\mathcal{Y}$ with the desired properties. $\square$

Theorem 4.10 and Proposition 4.11 show a connection between establishing superpolynomial formula size lower bounds for NP and understanding the difference between worst-case and average-case collections of anti-checkers.

Appendix

5 Unconditional lower bounds for Gap-MKtP and Gap-MCSP

5.1 MKtP – A near-quadratic lower bound against U_2-formulas

In this section, we provide the proof of Theorem 1.3.

Proof idea. We employ the technique of random restrictions to show that $\mathsf{Gap\text{-}MKtP}$ requires near-quadratic size formulas. The idea is that, with high probability, a formula F of sub-quadratic size simplifies under a random restriction $\boldsymbol{\rho}\colon [N] \to \{0,1,*\}$. This will allow us to complete a fixed restriction ρ either to a string w^y of Kt complexity $\leq s_1$, or to a string w^n of Kt complexity $\geq s_2$. Because the simplified formula $F\lceil_\rho$ depends on few input variables in $\rho^{-1}(*)$, if we define w^y and w^n appropriately $F\lceil_\rho$ won't be able to distinguish the two instances. Consequently, F does not compute $\mathsf{Gap\text{-}MKtP}[s_1, s_2]$.

In order for this idea to work, we cannot use a truly random restriction. This is because our restrictions will set most of the variables indexed in $[N]$ to simplify a near-quadratic size formula, and a typical random restriction cannot be completed to a string of low Kt complexity. We use instead pseudorandom restrictions, which can be computed from a much smaller number of random bits. Previous work established that such restrictions also simplify sub-quadratic size formulas. As a consequence, we are able to extend any restriction in the support of a pseudorandom distribution of restrictions to either an "easy" or a "hard" string, as explained in the paragraph above. (We remark that in order to improve our parameter s_1 in Gap-MKtP$[s_1, s_2]$, it is useful to compose a sequence of pseudodeterministic restrictions.)

We proceed with the technical details. Let $\rho\colon [N] \to \{0,1,*\}$ be a *restriction*, and $\boldsymbol{\rho}$ be a *random restriction*, i. e., a distribution of restrictions. We say that $\boldsymbol{\rho}$ is *p-regular* if $\Pr[\boldsymbol{\rho}(i) = *] = p$ and $\Pr[\boldsymbol{\rho}(i) = 0] = \Pr[\boldsymbol{\rho}(i) = 1] = (1-p)/2$ for every $i \in [N]$. In addition, $\boldsymbol{\rho}$ is *k-wise independent* if any k coordinates of $\boldsymbol{\rho}$ are independent.

Lemma 5.1 (see [24, 62])**.** *There exist q-regular k-wise independent random restrictions $\boldsymbol{\rho}$ distributed over $\rho\colon [N] \to \{0,1,*\}$ samplable with $O(k\log(N)\log(1/q))$ bits. Furthermore, each output coordinate of the random restriction can be computed in time polynomial in the number of random bits.*

Proof sketch. It is known that a k-wise independent distribution over $\{0,1\}^N$ can be generated with $O(k\log N)$ bits so that each output coordinate of the distribution can be computed in polynomial time given $O(k\log N)$ random bits and $\log N$ bits specifying the address of the coordinate, see [14]. In Lemma 5.1 the distribution is supported over $\{0,1,*\}^N$, and in each coordinate we should have $*$ with probability q and otherwise a uniform 0/1, and every set of k coordinates should be independent. We want to generate such a distribution using only $O(k\log N\log(1/q))$ bits.

Let $N' = N\log(1/q)$ and $k' = k\log(1/q)$, and assume that $q \geq 2^{-N}$.

Let D' be a k'-wise independent distribution supported over $\{0,1\}^{N'}$. Note that it can be sampled with $O(k'\log(N')) = O(k\log N\log(1/q))$ bits. Now interpret any string y in the support of D' as N blocks of $\log(1/q)$ bits, and convert each block into a value in $\{0,1,*\}$ where $*$ appears with bias q under this conversion. Let D be the resulting distribution, which is supported over $\{0,1,*\}^N$. Since k coordinates of D depend on $k' = k\log(1/q)$ coordinates of D', and D' is k'-wise independent, D is k-wise independent and in each coordinate $*$ is seen with bias q. ☐

As a consequence, we get p-regular k-wise independent random restrictions where each restriction in the support has bounded Kt complexity. In order to define the Kt complexity of a restriction $\rho\colon [N] \to \{0,1,*\}$, we view it as a $2N$-bit string $\mathsf{encoding}(\rho)$ where each symbol in $\{0,1,*\}$ is encoded by an element in $\{0,1\}^2$. We abuse notation and write $\mathsf{Kt}(\rho)$ to denote $\mathsf{Kt}(\mathsf{encoding}(\rho))$.

Proposition 5.2. *There is a distribution $\mathcal{D}_{q,k}$ of q-regular k-wise independent restrictions such that each restriction $\rho\colon [N] \to \{0,1,*\}$ in the support of $\mathcal{D}_{q,k}$ satisfies $\mathsf{Kt}(\rho) = O(k\log(N)\log(1/q))$. Furthermore, this is witnessed by a pair (M, w_ρ) where the machine M does not depend on ρ.*

Proof. By Lemma 5.1, each output coordinate of ρ can be computed in time $\mathsf{poly}(\ell)$ from a seed w_ρ of length $\ell = O(k\log(N)\log(1/q))$. Therefore, the binary string describing ρ can be computed in time $O(N\cdot\mathsf{poly}(\ell))$ from a string w_ρ with $\mathsf{Kt}(w_\rho) = O(k\log(N)\log(1/q))$. It follows from Proposition 2.3 that $\mathsf{Kt}(\rho) = O(k\log(N)\log(1/q))$. The furthermore part follows from the fact that the machine M is obtained from the generator provided by Lemma 5.1, i. e., in order to produce different restrictions one only needs to modify the input seeds, which are encoded in w_ρ. ☐

Let $N = 2^n$. Given a function $F\colon \{0,1\}^N \to \{0,1\}$ and a restriction $\rho\colon [N] \to \{0,1,*\}$, we let $F\restriction_\rho$ be the function in $\{0,1\}^{\rho^{-1}(*)} \to \{0,1\}$ obtained in the natural way from F and ρ. In this section, we use $L(F)$ to denote the size (number of leaves) of the smallest U_2-formula that computes a function F.

The next result allows us to shrink the size of a formula using a pseudorandom restriction. This restriction can be obtained by a composition of restrictions. This reduces the amount of randomness and the corresponding complexity of the restriction.

Lemma 5.3 (Shrinkage from pseudorandom restrictions ([21, Theorem 28]; see [24, 32]))**.** *Let $F\colon \{0,1\}^N \to \{0,1\}$, $q = p^{1/r}$ for an integer $r \geq 1$, and $L(F)\cdot p^2 \geq 1$. Moreover, let $\mathcal{R}^r_{p,k}$ be a distribution obtained by the composition of r independent q-regular k-wise independent random restrictions supported over $[N] \to \{0,1,*\}$, where $k = q^{-2}$. Finally, assume that $q \leq 10^{-3}$. Then,* [14]

$$\mathbb{E}_{\boldsymbol{\rho}\in_R \mathcal{R}^r_{p,k}}[L(F\restriction_{\boldsymbol{\rho}})] \;\leq\; c^r p^2 L(F),$$

where $c \geq 1$ is an absolute constant.

Proposition 5.4. *There is a (p-regular k-wise independent) distribution $\mathcal{R}^r_{p,k}$ obtained by the composition of r independent q-regular k-wise independent random restrictions supported over $[N] \to \{0,1,*\}$, where $k = q^{-2}$ and $q = p^{1/r}$, such that each restriction $\rho\colon [N] \to \{0,1,*\}$ in the support of $\mathcal{R}^r_{p,k}$ satisfies $\mathsf{Kt}(\rho) = O(rk\log(N)\log(1/q))$.*

Proof. We use the distribution $\mathcal{D}_{q,k}$ of restrictions provided by Proposition 5.2. A restriction ρ in the support of $\mathcal{R}^r_{p,k}$ is therefore obtained through the composition of r restrictions $\rho_1,\ldots,\rho_r$ in the support of $\mathcal{D}_{q,k}$. For each $i \in [r]$, $\mathsf{Kt}(\rho_r) = O(k\log(N)\log(1/q))$. Moreover, each Kt upper bound is witnessed by a pair (M, w_i), where M can be taken to be the same machine for all $i \in [r]$. It is not hard to see that for the string $w = 1^{|w_1|}0w_1 1^{|w_2|}0w_2 \ldots 1^{|w_r|}0w_r$ there is a machine M' satisfying $|\langle M'\rangle| \leq |\langle M\rangle| + O(1)$ and running in time $t_{M'}(w) \leq r\cdot \max_i t_M(w_i) + \mathsf{poly}(rN)$ such that the pair (M', w) witnesses that $\mathsf{Kt}(\rho) = O(rk\log(N)\log(1/q))$. □

We will also need the following simple proposition, which holds even with respect to Kolmogorov complexity instead of Kt complexity.

Proposition 5.5. *Let $S \subseteq [N]$ be a set of size at least two. There exists a function $h\colon S \to \{0,1\}$ such that for every string $w \in \{0,1\}^N$, if w agrees with h over S then $\mathsf{Kt}(w) \geq |S| - 5\log|S|$.*

Proof. It is easy to encode a pair (M,a) (as in Definition 2.1) satisfying $|\langle M\rangle| + |a| < |S| - 5\log|S|$ by a binary string of length at most $2\log|S| + 2 + |\langle M\rangle| + |a| < |S|$. Since each pair (M,a) outputs at most one binary string of length N, it follows by a counting argument that for some choice of $h\colon S \to \{0,1\}$, no string w of length N that agrees with h over S has $\mathsf{Kt}(w) < |S| - 5\log|S|$. □

[14] The assumption that $q \leq 10^{-3}$ does not appear in [21, Theorem 28]. The proof sketch appearing there does not seem to address the cases where $p^\Gamma L(\psi) < 1$ in their analyses of formula shrinkage in Lemma 27 and Theorem 28. This can be easily fixed using appropriate expressions of the form $1 + p^2 L(\psi)$. Lemma 27 is only affected by a constant factor. Then, proceeding by induction as in the proof of their Theorem 28 but also addressing this possibility, one gets instead an upper bound of the form $1 + cq^\Gamma(1 + cq^\Gamma(\ldots))$, which translates to $1 + (cq^\Gamma) + (cq^\Gamma)^2 + \ldots + (cq^\Gamma)^{r-1} + (cq^\Gamma)^r L(f)$. This can still be shown to be less than $c^r p^2 L(F)$ (for a different universal constant c as in the statement of Lemma 5.3) using that q is sufficiently small and therefore $cq^\Gamma \leq 1/2$ (note that $\Gamma = 2$ and $c \leq 500$ in [21]).

The next lemma describes the high-level strategy of the lower bound proof.

Lemma 5.6 (Adaptation of Lemma 27 from [21])**.** *There exists a constant $a \geq 1$ such that the following holds. Let $\rho\colon [N] \to \{0,1,*\}$ be a restriction, $V = \rho^{-1}(*)$, and let $F\colon \{0,1\}^N \to \{0,1\}$ be a function such that $L(F\restriction_\rho) \leq M$. If*

$$\mathsf{Kt}(\rho) + a \cdot n \;\leq\; s_1(n) \quad \textit{and} \quad (|V| - M) - 5\log(|V| - M) \;\geq\; s_2(n) \quad \textit{and} \quad |V| \geq M + a,$$

then F does not compute $\mathsf{Gap\text{-}MKtP}[s_1(n), s_2(n)]$, where $n = \log N$.

Proof. Under these assumptions, we define a positive instance $w^y \in \mathcal{YES}_N$ and a negative instance $w^n \in \mathcal{NO}_N$ such that $F(w^y) = F(w^n)$.

– $w^y \in \{0,1\}^N$ is obtained from ρ by additionally setting each $*$-coordinate of this restriction to 0. Note that, given the $2N$-bit binary string encoding ρ, w^y can be computed in time polynomial in N. It follows from Proposition 2.3 that $\mathsf{Kt}(w^y) \leq \mathsf{Kt}(\rho) + a \cdot n$, for some universal constant $a \geq 1$. Since this bound is at most $s_1(n)$, we get that $w^y \in \mathcal{YES}_N$.

– $w^n \in \{0,1\}^N$ is defined as follows. Since $L(F\restriction_\rho) \leq M$, $F\restriction_\rho$ depends on at most M input coordinates (indexed by elements in V). Let $W \subseteq V \subseteq [N]$ be this set of coordinates. Moreover, let $S = V \setminus W$. The string $w^n \in \{0,1\}^N$ is obtained from ρ by additionally setting each $*$-coordinate of this restriction in W to 0, and then setting each remaining $*$-coordinate in S to agree with the function $h\colon S \to \{0,1\}$ provided by Proposition 5.5. Since $|S| \geq |V| - M$ and the real-valued function $\phi(x) = x - 5\log x$ is non-decreasing if $x \geq a$ for a large enough constant a, our assumptions and Proposition 5.5 imply that $\mathsf{Kt}(w^n) \geq s_2(n)$. Consequently, $w^n \in \mathcal{NO}_N$.

Using that F restricted to ρ depends only on variables from $W \subseteq \rho^{-1}(*)$, and that the strings w^y and w^n agree over coordinates in $\rho^{-1}(\{0,1\}) \cup W$, it follows that $F(w^y) = F(w^n)$. Since w^y is a positive instance while w^n is a negative instance, F does not compute $\mathsf{Gap\text{-}MKtP}[s_1(n), s_2(n)]$. □

We are now ready to set parameters in order to complete the proof of Theorem 1.3. For a sufficiently large constant $C' \geq 1$, let

$$n \stackrel{\text{def}}{=} \log N, \quad p \stackrel{\text{def}}{=} N^{-1+\alpha/2}, \quad r \stackrel{\text{def}}{=} n/C', \quad q \stackrel{\text{def}}{=} p^{1/r}, \quad k \stackrel{\text{def}}{=} q^{-2},$$

and assume that N is sufficiently large. Note that, under this choice of parameters, $q = 2^{C'(-1+\alpha/2)} = \Omega(1)$ and $q \leq 10^{-3}$.

Proposition 5.7 (Concentration Bound for $|\boldsymbol{\rho}^{-1}(*)|$)**.** *For $\boldsymbol{\rho} \sim \mathcal{R}^r_{p,k}$ with parameters as above, we have* $\Pr[\,|\boldsymbol{\rho}^{-1}(*)| \geq pN/2\,] \geq 1/2$.

Proof. Note that $\boldsymbol{\rho}$ is p-regular and pairwise independent (i. e., $k \geq 2$ for our choice of parameters). The result then follows from Chebyshev's inequality using mean $\mu = pN$, variance $\sigma^2 = Np(1-p)$, and the value of p. □

Using Proposition 5.4, we can sample a random restriction $\boldsymbol{\rho} \in_R \mathcal{R}^r_{p,k}$ as described in the statement of Lemma 5.3 such that each $\rho\colon [N] \to \{0,1,*\}$ in the support of $\mathcal{R}^r_{p,k}$ satisfies

$$\mathsf{Kt}(\rho) \;=\; O(rk\log(N)\log(1/q)) \;=\; O((n/C')q^{-2}n\log(1/q)) \;\le\; (C/2)n^2,$$

if C is a sufficiently large constant.

Toward a contradiction, let $F\colon \{0,1\}^N \to \{0,1\}$ be a formula of size $L(F) = N^{2-\alpha}$ that supposedly computes $\mathsf{Gap\text{-}MKtP}[Cn^2, 2^{(\alpha/2)n-2}]$, where $p^2L(F) = 1$, and let

$$M \stackrel{\text{def}}{=} 10\cdot c^r p^2 L(F) \;=\; 10\cdot c^r \;\le\; 2^{(\alpha/4)n},$$

for a constant $c \ge 1$ as in Lemma 5.3, and using that $C' = C'(\alpha)$ is large enough in the definition of r.

Invoking Lemma 5.3 and Markov's inequality, Proposition 5.7, and a union bound, there is a fixed restriction $\rho\colon [N] \to \{0,1,*\}$ for which the following holds:

- For $V \stackrel{\text{def}}{=} \rho^{-1}(*)$, we have $|V| \;\ge\; pN/2 \;=\; 2^{(\alpha/2)n}/2$;
- $\mathsf{Kt}(\rho) \;\le\; (C/2)n^2$.
- $L(F\restriction_\rho) \;\le\; M \;\le\; 2^{(\alpha/4)n}$.

Using these parameters in the statement of Lemma 5.6, it is easy to check that its hypotheses are satisfied given our choices of $s_1(n) = Cn^2$ and $s_2(n) = 2^{(\alpha/2)n-2}$. This is a contradiction to our assumption that F computes $\mathsf{Gap\text{-}MKtP}$ for these parameters, which completes the proof.

5.2 MKtP – Stronger lower bounds for large parameters

The goal of this section is to prove Theorem 1.2. First, we need a definition. We say that a generator $G\colon \{0,1\}^r \to \{0,1\}^N$ *δ-fools* a function $f\colon \{0,1\}^N \to \{0,1\}$ if

$$\left| \Pr_{\boldsymbol{x}\in_R\{0,1\}^N}[f(x)=1] - \Pr_{\boldsymbol{y}\in_R\{0,1\}^r}[f(G(y))=1] \right| \;\le\; \delta.$$

Similarly, G δ-fools a class of functions $\mathcal{F}$ if G δ-fools every function $f \in \mathcal{F}$. The parameter r is called the *seed-length* of G. We say that G is *explicit* if there is a uniform algorithm computing G in time $\mathsf{poly}(N, 1/\delta)$ on all input lengths r.

Theorem 5.8 ([24]). *Let $c > 0$ be an arbitrary constant. The following hold:*

1. *There is an explicit generator $G^{U_2}\colon \{0,1\}^r \to \{0,1\}^N$ using a seed of length $r = s^{1/3+o(1)}$ that s^{-c}-fools the class U_2-$\mathsf{Formula}[s(N)]$ of formulas on N input variables.*
2. *There is an explicit generator $G^{B_2}\colon \{0,1\}^r \to \{0,1\}^N$ using a seed of length $r = s^{1/2+o(1)}$ that s^{-c}-fools the class B_2-$\mathsf{Formula}[s(N)]$ of formulas on N input variables.*
2. *There is an explicit generator $G^{BP}\colon \{0,1\}^r \to \{0,1\}^N$ using a seed of length $r = s^{1/2+o(1)}$ that s^{-c}-fools the class $\mathsf{BP}[s(N)]$ of branching programs on N input variables.*

We now prove Theorem 1.2 (Part 1). The other cases are similar. We instantiate G^{U_2} with $s(N) = N^{3-\varepsilon}$ and $c = 1$. Then $G^{U_2}\colon \{0,1\}^{N^{1-\delta'}} \to \{0,1\}^N$ for some $\delta' = \delta'(\varepsilon) > 0$.

Proposition 5.9. *For every string $w \in \{0,1\}^{N^{1-\delta'}}$, let $G^{U_2}(w) \in \{0,1\}^N$ be the N-bit output of G^{U_2} on w. Then*

$$\mathsf{Kt}(G^{U_2}(w)) \;\leq\; 2^{(1-\delta'/2)n}$$

for every large enough $n = \log N$.

Proof. This follows from Proposition 2.3 using that G^{U_2} is explicit and therefore runs in time $\mathsf{poly}(N)$ under our choice of parameters. □

As a consequence of Proposition 5.9, every output of G^{U_2} is always an N-bit string of Kt complexity at most $2^{(1-\delta)n}$, for a fixed $\delta > 0$. On the other hand, it is well-known that a random N-bit string (where $N = 2^n$) has Kolmogorov complexity (and thus Kt complexity) at least 2^{n-1} with high probability. It follows that $\mathsf{Gap\text{-}MKtP}[2^{(1-\delta)n}, 2^{n-1}] \notin U_2\text{-}\mathsf{Formula}[N^{3-\varepsilon}]$, since otherwise this would violate the security of the generator G^{U_2} against formulas of this type and size.

5.3 MCSP – A similar near-quadratic lower bound against U_2-formulas

In this section, we sketch the proof of Theorem 1.5, which is the analogue of Theorem 1.3 in the context of MCSP. More precisely, we explain why the argument carries over when we measure the complexity of a string by circuit size instead of via Kt complexity, modulo small changes to the involved parameters. A more detailed writeup of the result can be found in [15] where the lower bound was, in fact, strengthened so that it works against formulas with certain local oracles.

As explained in Section 5.1, the crucial idea in the proof of Theorem 1.3 is that a pseudorandom restriction simplifies a U_2-formula of bounded size. For technical reasons, we employ a composition of restrictions of small complexity, so that the overall complexity of the combined restriction is bounded. This allows us to trivialize any small formula F using a fixed restriction ρ of bounded complexity, where $|\rho^{-1}(*)|$ is sufficiently large compared to other relevant parameters of the argument. Then, Lemma 5.6 employs a counting argument (via Proposition 5.5) to extend this restriction to a positive instance w^y and to a negative instance w^n such that $F(w^y) = F(w^n)$. This can be used to show that no small formula correctly computes $\mathsf{Gap\text{-}MKtP}$ for our choice of parameters.

In order to establish Theorem 1.5, we make two observations. Firstly, Lemma 5.1 already gives individual restrictions of low *circuit complexity* instead of low Kt complexity. Secondly, the *counting argument* used to extend ρ to a negative instance w^n works for most complexity measures including circuit size, Kolmogorov complexity, etc.

Using these two observations, the proof goes through under minor adjustments of the relevant parameters. We remark that one obtains a lower bound for $\mathsf{Gap\text{-}MCSP}[n^d, 2^{(\alpha/2-o(1))n}]$ instead of $\mathsf{Gap\text{-}MCSP}[Cn^2, 2^{(\alpha/2)n-2}]$ because of a polynomial circuit complexity overhead in the argument, which is not present in the case of Kt complexity since there one takes the logarithmic of the running time when measuring complexity, and because the circuit complexity (measured by number of gates) of a random string can be slightly smaller than its Kt complexity.

6 Hardness magnification and proof complexity

One instance of hardness magnification from [46] says that if an average-case version of MCSP (with inputs being truth tables of Boolean functions) is worst-case hard for formulas of superlinear size, then its succinct version (with inputs being lists of input-output tuples representing partial Boolean functions) is hard for NC^1 (see [46, Theorem 1]).

Hardness magnification for MCSP thus attacks strong circuit lower bounds by 1. employing the natural proofs barrier, which states a conditional hardness of MCSP, as a suggestion to focus on lower bounds against MCSP, and 2. exploiting the relation between feasible (succinct) and infeasible (uncompressed) formulations of a meta-computational problem like MCSP.

This strategy has a history in proof complexity. The work of Razborov [49, 50] and Krajíček [35] formulated the natural proofs barrier as a conditional lower bound in proof complexity, expressing hardness of tautologies encoding circuit lower bounds. This idea was further developed in the theory of proof complexity generators [34, 2]. It has led, in particular, to Razborov's conjecture [51] about hardness of Nisan-Wigderson generators for strong proof systems. Razborov's conjecture is designed to imply hardness of circuit lower bounds formalized in a way so that the whole truth table of the hard function is hardwired into the formula.

The realization that a feasible formulation of circuit lower bounds should be much harder than the infeasible truth table formulas inspired the result about unprovability of circuit lower bounds in theories of bounded arithmetic such as VNC^1, see [48], and the proposal [47, 0.1 Circuit lower bounds and Complexity-Theoretic tautologies] to study exponentially harder lb formulas. Once the definitions are given, it is for example clear that polynomial-size proofs of the lb formulas transform into almost linear-size proofs of the truth table formulas. Another instance of this phenomena says that:

If the truth table formulas encoding a polynomial circuit lower bound require superlinear-size proofs in AC^0-Frege *systems, then* lb *formulas encoding the same polynomial circuit lower bound require* (NC^1)-Frege *proofs of superpolynomial size* (implicit in the proof of [42, Proposition 4.14]).

Since AC^0-Frege lower bounds are known, this suggests a way for attacking Frege lower bounds. ([46] established analogous results in circuit complexity, where it might be easier to prove lower bounds. However, their version of the MCSP problem refers to the average-case complexity of truth-tables, which seems harder to analyse. We refer to [46] for further discussion.)

The lb formulas result from the feasible witnessing of circuit lower bounds. In [42], the witnessing was provided by a theorem of Lipton and Young [38] establishing the existence of anti-checkers, described in Section 1.2. This allows to express the hardness of f without using its whole truth table. The present paper extends the idea of anti-checkers into the context of hardness magnification in circuit complexity for the standard worst-case formulation of MCSP.

Acknowledgements

We thank Avishay Tal for bringing [59] to our attention in connection to the problem of proving lower bounds against U_2-Formula-$\oplus$. We are grateful to Jan Krajíček for discussions related to hardness magnification. We also thank anonymous reviewers for many useful comments. This work was supported in part by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2014)/ERC Grant Agreement no. 615075 and by the Austrian Science Fund (FWF) under project number P 28699. This project has received funding from the Eurpoean Union's Horizon 2020 research and innovation programme under the Marie Skłodovska-Curie grant agreement No 890220. This work also received support from the Royal Society University Research Fellowship URF\R1\191059.

References

[1] SCOTT AARONSON: $\mathrm{P} \stackrel{?}{=} \mathrm{NP}$. In *Open Problems in Mathematics*, pp. 1–122. Springer, 2016. [ECCC:TR17-004] 2

[2] MICHAEL ALEKHNOVICH, ELI BEN-SASSON, ALEXANDER A. RAZBOROV, AND AVI WIGDERSON: Pseudorandom generators in propositional proof complexity. *SIAM J. Comput.*, 34(1):67–88, 2004. [doi:10.1137/S0097539701389944, ECCC:TR00-023] 31

[3] ERIC ALLENDER: When worlds collide: Derandomization, lower bounds, and Kolmogorov complexity. In *Proc. 21st Found. Softw. Techn. Theoret. Comp. Sci. Conf. (FSTTCS'01)*, pp. 1–15. Springer, 2001. [doi:10.1007/3-540-45294-X_1] 6, 10

[4] ERIC ALLENDER: The new complexity landscape around circuit minimization. In *Proc. 14th Internat. Conf. Lang. Autom. Theory and Appl. (LATA'20)*, pp. 3–16. Springer, 2020. [doi:10.1007/978-3-030-40608-0_1, ECCC:TR20-078] 3

[5] ERIC ALLENDER, HARRY BUHRMAN, MICHAL KOUCKÝ, DIETER VAN MELKEBEEK, AND DETLEF RONNEBURGER: Power from random strings. *SIAM J. Comput.*, 35(6):1467–1493, 2006. [doi:10.1137/050628994] 2, 5

[6] ERIC ALLENDER AND MICHAL KOUCKÝ: Amplifying lower bounds by means of self-reducibility. *J. ACM*, 57(3):14:1–36, 2010. [doi:10.1145/1706591.1706594] 3

[7] ANDREJ BOGDANOV: Small-bias requires large formulas. In *Proc. 45th Internat. Colloq. on Automata, Languages, and Programming (ICALP'18)*, pp. 22:1–12. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018. [doi:10.4230/LIPIcs.ICALP.2018.22, ECCC:TR18-139] 2

[8] RAVI B. BOPPANA AND MICHAEL SIPSER: The complexity of finite functions. In *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity*, pp. 757–804. Elsevier, 1990. [doi:10.1016/B978-0-444-88071-0.50019-9] 2

[9] NADER H. BSHOUTY, RICHARD CLEVE, RICARD GAVALDÀ, SAMPATH KANNAN, AND CHRISTINO TAMON: Oracles and queries that are sufficient for exact learning. *J. Comput. System Sci.*, 52(3):421–433, 1996. [doi:10.1006/jcss.1996.0032] 7, 19, 21

[10] HARRY BUHRMAN, LANCE FORTNOW, AND THOMAS THIERAUF: Nonrelativizing separations. In *Proc. 13th IEEE Conf. on Comput. Complexity (CCC'98)*, pp. 8–12. IEEE Comp. Soc., 1998. [doi:10.1109/CCC.1998.694585] 3

[11] JIN-YI CAI: $\mathsf{S}_2^p \subseteq \mathsf{ZPP}^{\mathsf{NP}}$. *J. Comput. System Sci.*, 73(1):25–35, 2007. [doi:10.1016/j.jcss.2003.07.015] 7

[12] MARCO L. CARMOSINO, RUSSELL IMPAGLIAZZO, VALENTINE KABANETS, AND ANTONINA KOLOKOLOVA: Learning algorithms from natural proofs. In *Proc. 31st Comput. Complexity Conf. (CCC'16)*, pp. 10:1–24. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016. [doi:10.4230/LIPIcs.CCC.2016.10] 4

[13] MARCO L. CARMOSINO, RUSSELL IMPAGLIAZZO, SHACHAR LOVETT, AND IVAN MIHAJLIN: Hardness amplification for non-commutative arithmetic circuits. In *Proc. 33rd Comput. Complexity Conf. (CCC'18)*, pp. 12:1–16. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018. [doi:10.4230/LIPIcs.CCC.2018.12, ECCC:TR18-095] 3

[14] J. LAWRENCE CARTER AND MARK N. WEGMAN: Universal classes of hash functions. *J. Comput. System Sci.*, 18(2):143–154, 1979. Preliminary version in STOC'77. [doi:10.1016/0022-0000(79)90044-8] 26

[15] LIJIE CHEN, SHUICHI HIRAHARA, IGOR C. OLIVEIRA, JÁN PICH, NINAD RAJGOPAL, AND RAHUL SANTHANAM: Beyond natural proofs: Hardness magnification and locality. In *Proc. 11th Innovations in Theoret. Comp. Sci. conf. (ITCS'20)*, volume 151, pp. 70:1–48. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020. [doi:10.4230/LIPIcs.ITCS.2020.70] 3, 8, 9, 30

[16] LIJIE CHEN, DYLAN M. MCKAY, CODY D. MURRAY, AND R. RYAN WILLIAMS: Relations and equivalences between circuit lower bounds and Karp-Lipton theorems. In *Proc. 34th Comput. Complexity Conf. (CCC'19)*, pp. 30:1–21. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019. [doi:10.4230/LIPIcs.CCC.2019.30, ECCC:TR19-075] 8

[17] IRIT DINUR AND OR MEIR: Toward the KRW composition conjecture: Cubic formula lower bounds via communication complexity. *Comput. Complexity*, 27(3):375–462, 2018. Preliminary version in CCC'16. [doi:10.1007/s00037-017-0159-x, ECCC:TR16-035] 3

[18] MAGNUS GAUSDAL FIND, ALEXANDER GOLOVNEV, EDWARD A. HIRSCH, AND ALEXANDER S. KULIKOV: A better-than-$3n$ lower bound for the circuit complexity of an explicit function. In *Proc. 57th FOCS*, pp. 89–98. IEEE Comp. Soc., 2016. [doi:10.1109/FOCS.2016.19, ECCC:TR15-166] 3

[19] ODED GOLDREICH: *Computational Complexity – A Conceptual Perspective*. Cambridge Univ. Press, 2008. [doi:10.1017/CBO9780511804106] 19

[20] JOHAN HÅSTAD: The shrinkage exponent of De Morgan formulas is 2. *SIAM J. Comput.*, 27(1):48–64, 1998. [doi:10.1137/S0097539794261556] 3

[21] SHUICHI HIRAHARA AND RAHUL SANTHANAM: On the average-case complexity of MCSP and its variants. In *Proc. 32nd Comput. Complexity Conf. (CCC'17)*, pp. 7:1–20. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017. [doi:10.4230/LIPIcs.CCC.2017.7] 3, 6, 7, 27, 28

[22] PAVEL HRUBEŠ, AVI WIGDERSON, AND AMIR YEHUDAYOFF: Non-commutative circuits and the Sum-of-Squares problem. *J. AMS*, 24(3):871–898, 2011. Preliminary version in STOC'10. [doi:10.1090/S0894-0347-2011-00694-2, ECCC:TR10-021] 3

[23] RAHUL ILANGO, BRUNO LOFF, AND IGOR C. OLIVEIRA: NP-hardness of circuit minimization for multi-output functions. In *Proc. 35th Comput. Complexity Conf. (CCC'20)*, volume 169, pp. 22:1–36. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020. [doi:10.4230/LIPIcs.CCC.2020.22, ECCC:TR20-021] 6

[24] RUSSELL IMPAGLIAZZO, RAGHU MEKA, AND DAVID ZUCKERMAN: Pseudorandomness from shrinkage. *J. ACM*, 66(2):1–16, 2019. Preliminary version in FOCS'12. [doi:10.1145/3230630, ECCC:TR12-057] 6, 26, 27, 29

[25] RUSSELL IMPAGLIAZZO, RAMAMOHAN PATURI, AND MICHAEL E. SAKS: Size-depth tradeoffs for threshold circuits. *SIAM J. Comput.*, 26(3):693–707, 1997. [doi:10.1137/S0097539792282965] 3

[26] KAZUO IWAMA AND HIROKI MORIZUMI: An explicit lower bound of $5n - o(n)$ for Boolean circuits. In *Proc. Internat. Symp. Math. Foundations of Comp. Sci. (MFCS'02)*, pp. 353–364. Springer, 2002. [doi:10.1007/3-540-45687-2_29] 3

[27] STASYS JUKNA: *Boolean Function Complexity – Advances and Frontiers*. Springer, 2012. [doi:10.1007/978-3-642-24508-4] 2, 4

[28] JØRN JUSTESEN: Class of constructive asymptotically good algebraic codes. *IEEE Trans. Inform. Theory*, 18(5):652–656, 1972. [doi:10.1109/TIT.1972.1054893] 12

[29] VALENTINE KABANETS AND JIN-YI CAI: Circuit minimization problem. In *Proc. 32nd STOC*, pp. 73–79. ACM Press, 2000. [doi:10.1145/335305.335314, ECCC:TR99-045] 2

[30] DANIEL M. KANE AND R. RYAN WILLIAMS: Super-linear gate and super-quadratic wire lower bounds for depth-two and depth-three threshold circuits. In *Proc. 48th STOC*, pp. 633–643. ACM Press, 2016. [doi:10.1145/2897518.2897636, arXiv:1511.07860, ECCC:TR15-188] 3

[31] RAVI KANNAN: Circuit-size lower bounds and non-reducibility to sparse sets. *Inform. Control*, 55(1–3):40–56, 1982. [doi:10.1016/S0019-9958(82)90382-5] 3

[32] ILAN KOMARGODSKI, RAN RAZ, AND AVISHAY TAL: Improved average-case lower bounds for de Morgan formula size: Matching worst-case lower bound. *SIAM J. Comput.*, 46(1):37–57, 2017. [doi:10.1137/15M1048045] 27

[33] JAN KRAJÍČEK: Extensions of models of PV. In *Proc. Logic Colloquium*, ASL Lecture Notes in Logic, pp. 104–114. Springer, 1995. [doi:10.1017/9781316716830.011] 21

[34] JAN KRAJÍČEK: On the Weak Pigeonhole Principle. *Fundamenta Mathematicae*, 170(1):123–140, 2001. [doi:10.4064/fm170-1-8] 31

[35] JAN KRAJÍČEK: Dual Weak Pigeonhole Principle, pseudo-surjective functions, and provability of circuit lower bounds. *J. Symbolic Logic*, 69(1):265–286, 2004. [doi:10.2178/jsl/1080938841] 31

[36] LEONID LEVIN: Randomness conservation inequalities; information and independence in mathematical theories. *Inform. Control*, 61(1):15–37, 1984. [doi:10.1016/S0019-9958(84)80060-1] 2, 4, 10

[37] RICHARD J. LIPTON AND R. RYAN WILLIAMS: Amplifying circuit lower bounds against polynomial time, with applications. *Comput. Complexity*, 22(2):311–343, 2013. [doi:10.1007/s00037-013-0069-5] 3

[38] RICHARD J. LIPTON AND NEAL E. YOUNG: Simple strategies for large zero-sum games with applications to complexity theory. In *Proc. 26th STOC*, pp. 734–740. ACM Press, 1994. [doi:10.1145/195058.195447] 7, 17, 18, 31

[39] FLORENCE JESSIE MACWILLIAMS AND NEIL JAMES ALEXANDER SLOANE: *The Theory of Error-Correcting Codes*. Elsevier, 1977. 16

[40] DYLAN M. MCKAY, CODY D. MURRAY, AND R. RYAN WILLIAMS: Weak lower bounds on resource-bounded compression imply strong separations of complexity classes. In *Proc. 51st STOC*, pp. 1215–1225. ACM Press, 2019. [doi:10.1145/3313276.3316396] 4, 8

[41] ERIC MILES AND EMANUELE VIOLA: Substitution-permutation networks, pseudorandom functions, and natural proofs. *J. ACM*, 62(6):46:1–29, 2015. [doi:10.1145/2792978] 2

[42] MORITZ MÜLLER AND JÁN PICH: Feasibly constructive proofs of succinct weak circuit lower bounds. *Ann. Pure Appl. Logic*, 171(2):102735:1–45, 2020. [doi:10.1016/j.apal.2019.102735, ECCC:TR17-144] 3, 7, 31

[43] CODY D. MURRAY AND R. RYAN WILLIAMS: Circuit lower bounds for nondeterministic quasi-polytime from a new easy witness lemma. *SIAM J. Comput.*, 49(5):300–322, 2019. Preliminary version in STOC'18. [doi:10.1137/18M1195887] 3

[44] ÈDUARD IVANOVIČ NEČIPORUK: On a Boolean function (russian). *Dokl. Akad. Nauk SSSR (Russian)*, 169(4):765–766, 1966. Math-Net.Ru. 3

[45] IGOR C. OLIVEIRA, JÁN PICH, AND RAHUL SANTHANAM: Hardness magnification near state-of-the-art lower bounds. In *Proc. 34th Comput. Complexity Conf. (CCC'19)*, pp. 27:1–29. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019. [doi:10.4230/LIPIcs.CCC.2019.27, ECCC:TR18-158] 1, 8

[46] IGOR C. OLIVEIRA AND RAHUL SANTHANAM: Hardness magnification for natural problems. In *Proc. 59th FOCS*, pp. 65–76. IEEE Comp. Soc., 2018. [doi:10.1109/FOCS.2018.00016, ECCC:TR18-139] 3, 4, 5, 6, 8, 9, 11, 12, 15, 31

[47] JÁN PICH: *Complexity Theory in Feasible Mathematics*. Ph. D. thesis, Charles University, Prague, 2014. Charles U. Digital Rep. 31

[48] JÁN PICH: Circuit lower bounds in bounded arithmetics. *Ann. Pure Appl. Logic*, 166(1):29–45, 2015. [doi:10.1016/j.apal.2014.08.004, ECCC:TR13-077] 31

[49] ALEXANDER A. RAZBOROV: On provably disjoint NP-pairs. *BRICS Report Ser.*, 1(36), 1994. [doi:10.7146/brics.v1i36.21607, ECCC:TR94-006] 31

[50] ALEXANDER A. RAZBOROV: Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic. *Izvestiya Russ. Akad. Sci. Ser. Math. (Russian)*, 59(1):201–224, 1995. [doi:10.1070/IM1995v059n01ABEH000009] 31

[51] ALEXANDER A. RAZBOROV: Pseudorandom generators hard for k-DNF resolution and polynomial calculus. *Ann. Math.*, 182(2):415–472, 2015. [doi:10.4007/annals.2015.181.2.1] 31

[52] ALEXANDER A. RAZBOROV AND STEVEN RUDICH: Natural proofs. *J. Comput. System Sci.*, 55(1):24–35, 1997. [doi:10.1006/jcss.1997.1494] 2, 8

[53] RAHUL SANTHANAM: Circuit lower bounds for Merlin–Arthur classes. *SIAM J. Comput.*, 39(3):1038–1061, 2009. [doi:10.1137/070702680] 3

[54] MICHAEL SIPSER AND DANIEL A. SPIELMAN: Expander codes. *IEEE Trans. Inform. Theory*, 42(6):1710–1722, 1996. [doi:10.1109/18.556667] 12, 15

[55] DANIEL A. SPIELMAN: Linear-time encodable and decodable error-correcting codes. *IEEE Trans. Inform. Theory*, 42(6):1723–1731, 1996. [doi:10.1109/18.556668] 15

[56] ARAVIND SRINIVASAN: On the approximability of clique and related maximization problems. *J. Comput. System Sci.*, 67(3):633–651, 2003. [doi:10.1016/S0022-0000(03)00110-7] 3

[57] LARRY J. STOCKMEYER: The complexity of approximate counting (Preliminary version). In *Proc. 15th STOC*, pp. 118–126. ACM Press, 1983. [doi:10.1145/800061.808740] 19

[58] AVISHAY TAL: Shrinkage of De Morgan formulae by spectral techniques. In *Proc. 55th FOCS*, pp. 551–560. IEEE Comp. Soc., 2014. [doi:10.1109/FOCS.2014.65] 3

[59] AVISHAY TAL: The bipartite formula complexity of inner-product is quadratic. *Electron. Colloq. Comput. Complexity*, TR16-181, 2016. Conf. version FOCS'17. [ECCC] 10, 31

[60] ROEI TELL: Quantified derandomization of linear threshold circuits. *Electron. Colloq. Comput. Complexity*, TR17-145, 2017. Conf. version [61]. [ECCC, arXiv:1709.07635] 16

[61] ROEI TELL: Quantified derandomization of linear threshold circuits. In *Proc. 50th STOC*, pp. 855–865. ACM Press, 2018. [doi:10.1145/3188745.3188822, arXiv:1709.07635, ECCC:TR17-145] 16, 36

[62] SALIL P. VADHAN: Pseudorandomness. *Found. Trends Theor. Comp. Sci.*, 7(1–3):1–336, 2012. [doi:10.1561/0400000010] 26

[63] INGO WEGENER: *The Complexity of Boolean Functions*. Wiley, 1987. ECCC. 18, 21

[64] RYAN WILLIAMS: Nonuniform ACC circuit lower bounds. *J. ACM*, 61(1):2:1–32, 2014. [doi:10.1145/2559903] 4

AUTHORS

Igor C. Oliveira
Royal Society University Research Fellow and
Assistant Professor
Department of Computer Science
University of Warwick
Coventry, UK
igor.oliveira@warwick.ac.uk
https://www.dcs.warwick.ac.uk/~igorcarb/

Ján Pich
Royal Society University Research Fellow and
Marie Skłodowska-Curie Research Fellow
Department of Computer Science
University of Oxford
Oxford, Oxfordshire, UK
jan.pich@cs.ox.ac.uk
http://users.ox.ac.uk/~coml0742/

Rahul Santhanam
Professor
Department of Computer Science
University of Oxford
Oxford, Oxfordshire, UK
rahul.santhanam@cs.ox.ac.uk
https://www.cs.ox.ac.uk/people/Rahul.Santhanam/

ABOUT THE AUTHORS

IGOR C. OLIVEIRA is a Royal Society University Research Fellow and Assistant Professor at the University of Warwick. He received his Ph. D. from Columbia University in 2015, advised by Tal Malkin and Rocco Servedio. Before joining the University of Warwick, he was a postdoctoral researcher in the Algorithms and Complexity Theory Group at the University of Oxford, a research fellow at the Simons Institute for the Theory of Computing (Berkeley), and a postdoctoral fellow at the School of Mathematics at Charles University in Prague. He is interested in computational complexity theory and its connections to algorithms, combinatorics, and mathematical logic.

JÁN PICH is a Royal Society University Research Fellow and a Marie Skłodowska-Curie Research Fellow in the Algorithms and Complexity Theory Group at the University of Oxford. He received his Ph. D. from Charles University in Prague under the supervision of Jan Krajíček. Subsequently, he held postdoctoral positions at the University of Toronto, University of Leeds, University of Vienna and the Czech Academy of Sciences. He is interested in mathematical logic and complexity theory, in particular, in the proof complexity of circuit lower bounds.

RAHUL SANTHANAM is a Professor of Computer Science at the University of Oxford, and a Tutorial Fellow at Magdalen College. He obtained his Ph. D. from the University of Chicago in 2005, working under the supervision of Lance Fortnow and Janos Simon. After postdoctoral stints at Simon Fraser University and the University of Toronto, he joined the University of Edinburgh as a Lecturer in Computer Science in 2008, and moved to the University of Oxford in 2016. His primary interests are in computational complexity theory and experimental poetry.