Simple general magnification of circuit lower bounds

Albert Atserias* Moritz Müller†

June 21, 2025

Abstract

We introduce a technically and conceptually simple approach to magnification of circuit and formula lower bounds. Central to the method are so-called *distinguishers*, sparse matrices that retain some of the key properties of error-correcting codes. As applications, we generalize and strengthen known *general* (not problem specific) magnification results and in particular achieve magnification thresholds below known lower bounds. For example, we show that fixed-polynomial formula-size lower bounds for NP are implied by slightly superlinear formula-size lower bounds for approximating any sufficiently sparse problem in NP. We also show that the thresholds achieved are sharp. Additionally, our approach yields a *uniform* magnification result for the Minimum Circuit Size Problem (MCSP). This seems to sidestep the localization barrier.

1 Introduction

We use standard terminology and notation: by a *circuit* (resp. *formula*) we mean one with inner gates labeled $\neg, \wedge, \vee$ of fan-in at most 2 (and, resp., of fan-out at most 1); its *size* is the number of gates. A *promise problem* is given by disjoint sets YES and NO of binary strings. It is contained in $\mathsf{SIZE}[s]$ (resp. $\mathsf{FML}[s]$), where $s : \mathbb{N} \to \mathbb{N}$, if for all sufficiently large $n \in \mathbb{N}$ there is a circuit (resp. formula) C_n of size $\leqslant s(n)$ that accepts all n-bit strings in YES and rejects all n-bit strings in NO. The class P-*uniform*-$\mathsf{SIZE}[s]$ additionally requires the map $n \mapsto C_n$ to be computable in time polynomial in n.

Note $\mathsf{P/poly} = \mathsf{SIZE}[n^{O(1)}]$ and $\mathsf{NC}_1 = \mathsf{FML}[n^{O(1)}]$.

*Universitat Politècnica de Catalunya, and Centre de Recerca Matemàtica
†Universität Passau

1.1 Why are circuit lower bounds difficult?

The non-uniform version of $\mathsf{P} \neq \mathsf{NP}$ asks for a problem in NP that is not decided by circuits of polynomial size, i.e., to show $\mathsf{NP} \not\subseteq \mathsf{P/poly}$. This seems to be out of reach of current techniques. Today not even fixed-polynomial size lower bounds are known, not even for NE and formulas, i.e., whether $\mathsf{NE} \not\subseteq \mathsf{FML}[n^c]$ for all $c \in \mathbb{N}$, equivalently $\mathsf{NEXP} \not\subseteq \mathsf{NC}_1$.

Why are circuit lower bounds so difficult? The natural proof barrier [33] points out some limitation of current techniques – the notion of a natural proof is, however, informal. Razborov's program [31] asks for a formally precise barrier in form of a natural theory that proves known weak lower bounds (cf. [31, 25]) and does not prove strong conjectured ones. The unprovability is, ironically, again dubbed to be out of reach of current techniques – of proof complexity (cf. [32], [20, Chapter 27-30]).

A more direct approach to somehow explain the apparent hardness of circuit lower bounds is to try to establish that the *Minimum Circuit Size Problem* is computationally hard in some sense. Namely, for $\sigma : \mathbb{N} \to \mathbb{N}$,

$\mathsf{MCSP}[\sigma]$	
Input:	$x \in \{0,1\}^n$ with $n = 2^\ell$ for some $n, \ell \in \mathbb{N}$.
Problem:	is x computable by a circuit of size $\leqslant \sigma(\ell)$?

Here, that x is computed by a circuit C means that C has ℓ inputs and outputs the i-th bit of x when given the lexicographically i-th ℓ-bit string as input.

Research into the complexity of $\mathsf{MCSP}[\sigma]$ dates back to the 1950s [36] and intensified in recent years, starting with [19]. By the idea that its computational hardness reflects the difficulty of proving circuit lower bounds, $\mathsf{MCSP}[\sigma]$ is expected to become harder for larger σ but this is not known. Concerning its envisioned hardness we refer to [1] for a survey and mention here only an obstacle: $\mathsf{MCSP}[2^{o(\ell)}]$ is $2^{n^{o(1)}}$-sparse and Buhrmann and Hitchcock [4] showed that such problems are not NP-hard unless PH collapses. Recall, for $q : \mathbb{N} \to \mathbb{N}$, a problem Q is *q-sparse* if $|Q \cap \{0,1\}^n| \leqslant q(n)$ for all $n \in \mathbb{N}$.

1.2 Magnification

Thus, not only seem strong circuit lower bounds to be out of reach of current techniques, there is also a lack of understanding of what this very statement means. This is strikingly emphasized by what Oliveira and Santhanam [27] called *magnification* results: a strong lower bound that appear(ed) out of reach of current techniques is implied by a lower bound that is "almost known" – meaning that there are known lower bounds that are only marginally weaker or concern slight variations of the problem or the computational model.

Such results are interesting, first, because they unveil inconsistencies in our intuitions for what is or not within reach of current techniques. Second, they provide an approach to strong lower bounds that sidesteps the natural proof barrier – see [9] for an insightful discussion. Both perspectives focus attention on the *magnification threshold* – the "almost known" lower bound.

The initial magnification result from [27] concerns a promise problem relaxing $\mathsf{MCSP}[\sigma]$, namely its approximation version: for $\epsilon : \mathbb{N} \to \mathbb{R}_{>0}$

ϵ-MCSP$[\sigma]$	
Input:	$x \in \{0,1\}^n$ with $n = 2^\ell$ for some $n, \ell \in \mathbb{N}$.
YES:	x is computable by a circuit of size $\leqslant \sigma(\ell)$.
NO:	$d_H(x,y) \geqslant \epsilon(n) \cdot n$ for all $y \in \{0,1\}^n$ computable by a circuit of size $\leqslant \sigma(\ell)$.

Here, $d_H(x,y)$ is the Hamming distance of x, y. In other words, NO instances are truth tables of functions that are not $(1-\epsilon(n))$-approximable by circuits of size $\leqslant \sigma(\ell)$.

Observe, the smaller ϵ, the more NO instances, the harder the problem, and ϵ-MCSP$[\sigma]$ is the same as MCSP$[\sigma]$ if $\epsilon(n) \leqslant 1/n$ for all $n > 0$. Under cryptographic assumptions, it is known that 1/3-MCSP$[2^{\sqrt{\ell}}] \notin$ P/poly (cf. [27]).

Theorem 1 ([27])**.** NP $\not\subseteq$ NC$_1$ *if there exists a real* $\delta > 0$ *such that*

$$n^{-o(1)}\text{-MCSP}[2^{\sqrt{\ell}}] \notin \text{FML}[n^{1+\delta}].$$

This magnification threshold can be considered to be "almost known":

Theorem 2 ([15])**.** MCSP$[2^{\sqrt{\ell}}] \notin$ FML$[n^{2-\delta}]$ *for all reals* $\delta > 0$.

Following [27] a wide variety of magnification results have been discovered, with magnification thresholds concerning almost formulas [9], probabilistic formulas [7], zero-error heuristics [27, 6], sublinear randomized algorithms [27], streaming algorithms [23], refuters [27, 8] or obstruction sets [7]. A few magnification results have also been found in proof complexity [25, 3]. We do not attempt a survey and refer to [9] for a systematic overview.

A central role is played by MCSP$[\sigma]$ and its variants. Besides the mentioned approximation relaxation, prominently its gap relaxation (e.g. [27, 28, 9]), its search version [23] and their siblings for time-bounded Kolmogoroff complexity instead circuit complexity. Many proofs of magnification results build on deep results in complexity theory, and many are somewhat ad hoc, exploiting special properties of the MCSP$[\sigma]$ variants.

1.3 General magnification

As for a more principled approach, a significant step forward was taken by Chen et al. [6]. Only the sparsity matters: "analogous weak circuit lower bounds [...] for **any** equally-sparse NP language also imply major separations in circuit complexity."

We refer to such results as *general* magnification. An example predating [6] is

Theorem 3 ([10])**.** NEXP $\not\subseteq$ P/poly *if there exists an* $2^{n^{o(1)}}$*-sparse problem* $Q \in$ NP *such that*

$$Q \notin \text{SIZE}[n^{1+o(1)}].$$

In [6] the conclusion is strengthened to NP $\not\subseteq$ SIZE$[n^c]$ for all $c \in \mathbb{N}$ and a version for formulas is proved. A version for probabilistic formulas appears in [7] – we use the following ad hoc notation: a promise problem belongs to PFML$[s]$, where $s : \mathbb{N} \to \mathbb{N}$, if for all large enough $n \in \mathbb{N}$ there exists a probabilistic formula $\boldsymbol{F}$ of size $\leqslant s(n)$ (i.e., a random variable whose values are formulas of size $\leqslant s(n)$) such that $\Pr[\boldsymbol{F}(x) = 1] = 1$ for all YES instances x of length n and $\Pr[\boldsymbol{F}(x) = 1] \leqslant 1/4$ for all NO instances x of length n.

Theorem 4 ([6, 7]). $\mathsf{NP} \notin \mathsf{FML}[n^c]$ *for all* $c \in \mathbb{N}$ *if there exists an* $2^{n^{o(1)}}$*-sparse problem* $Q \in \mathsf{NP}$ *such that (a) or (b):*

(a) $Q \notin \mathsf{FML}[n^{3+o(1)}]$.

(b) $Q \notin \mathsf{PFML}[n^{2+o(1)}]$.

The proofs of these results are based on deep results in complexity theory: Theorem 3 relies on the hardness versus pseudorandomness trade-off [37] and the Easy Witness Lemma [16]. The technical core of the proof of Theorem 4 is the construction [6] of a highly efficient hash family based on good error correcting codes and expander-walk sampling.

The main strength of these results is their generality, as it enhances the prospects of magnification to eventually trigger breakthrough lower bounds. Their main weakness is that it is not clear whether their magnification thresholds should be considered "almost known". No explicit problem is known outside $\mathsf{SIZE}[5n]$ or $\mathsf{FML}[n^3]$, and no explicit $2^{n^{o(1)}}$-sparse problem is known outside $\mathsf{FML}[n^2]$. For non-sparse problems, however, even subcubic lower bounds are known for probabilistic formulas, in particular for $\mathsf{MCSP}[2^\ell/\ell^4]$ as shown in [9, Theorem 47] (based on [11]).

For sparse variants, known lower bounds sit sharply below the threshold:

Theorem 5 ([7]). $\mathsf{MCSP}[2^{\sqrt{\ell}}] \notin \mathsf{PFML}[n^{2-\delta}]$ *for all reals* $\delta > 0$.

The proof of this result in [7] is based on a sophisticated construction of a new pseudorandom restriction generator that deserves independent interest.

Remark 6. Both Theorems 4 (b) and 5 are proved for probabilistic formulas with *two-sided error*. For the magnification result the variant cited here is slightly stronger and implied by Theorem 9 below. We only consider one-sided error in this work.

1.4 Distinguishers

The proof idea for a magnification result is as follows. Assume NP has small circuits and construct a tiny circuit for a given problem $Q \in \mathsf{NP}$. The crucial step is an efficient hash of YES instances of Q to very short fingerprints, a set in NP. The tiny circuit is then obtained by computing the hash and running a small circuit on the fingerprint. Typically the hash is randomized, so the tiny circuit is probabilistic. Finally, "derandomize this construction in an elementary but careful way" [27].

Thus, the core of magnification is compression. It is parameterized complexity theory that developed a deep theory of compression, namely kernelization theory (cf. [13]).[1] In this context the second authors' PhD Thesis [24] gave a non-uniform hash family for arbitrary sparse problems in order to derive, via some kernelization theory, Burhman and Hitchcock's result [4] mentioned earlier. For this, non-uniformity did not harm but magnification does require uniformity. This leads to our main conceptual contribution:

[1] We use the informal "compression" instead of the formal "kernelization" because our problems are not parameterized. A suitable parameterization would be the logarithm of the sparsity, so only depend on the input length; this is, however, against the spirit of parameterized complexity theory.

Definition 7. Let $n, m \in \mathbb{N}$ and $0 < \epsilon < \delta \leqslant 1$ be real. View $x, y \in \{0,1\}^n$ as row vectors in $\mathbb{F}_2^n$. An (n, m, ϵ, δ)-*distinguisher* is a binary $n \times m$ matrix D such that

$$d_H(x, y) \geqslant \epsilon \cdot n \implies d_H(xD, yD) \geqslant \delta \cdot m$$

for all $x, y \in \{0,1\}^n$. The *weight* of D is the maximum Hamming weight of a column of D.

Observe that a $(n, m, \delta, 1/n)$-distinguisher D is the same as the generator matrix of a linear code with relative distance δ. We trade low weight for larger $\epsilon > 1/n$:

Theorem 8. *Let $0 < \epsilon \leqslant 1$. There is an algorithm that given $n \in \mathbb{N}$ computes in time polynomial in n for some $m \leqslant n^7$ an $(n, m, n^{-\epsilon}, 1/8)$-distinguisher of weight $\leqslant \lceil 2n^\epsilon \rceil$.*

Given an $2^{n^{o(1)}}$-sparse Q, roughly, the fingerprint of an n-bit string y are r random positions in yD. Low weight ensures efficiency of the hash: each bit of the fingerprint is the XOR of $\leqslant \lceil 2n^\epsilon \rceil$ many input bits, so computed by a formula of size $O(n^{2\epsilon})$. Assume $d_H(y, x) \geqslant n^{-\epsilon} \cdot n$ for all $x \in Q$. Then the fingerprint equals that of some $x \in Q$ with probability $\leqslant 2^{n^{o(1)}} \cdot (7/8)^r$ – this is small already for $r \leqslant n^{o(1)}$. This way, a small formula for the set of fingerprints of $x \in Q$ gives a tiny formula distinguishing such y from all $x \in Q$, i.e., decide $n^{-\epsilon}$-Q. This notation generalizes ϵ-$\mathsf{MCSP}[\sigma]$ to arbitrary Q: for $\epsilon : \mathbb{N} \to \mathbb{R}_{>0}$,

ϵ-Q	
Input:	$x \in \{0,1\}^n$ for some $n \in \mathbb{N}$.
YES:	$x \in Q$.
NO:	$d_H(x, y) \geqslant \epsilon(n) \cdot n$ for all $y \in Q$.

1.5 This work

The contribution of the present work is a conceptually modular and technically simple approach to general magnification. It is based on distinguishers and developed from scratch.

To illustrate how magnification is derived from compression, Section 2 gives a highly simplified proof of Theorem 3. Section 3 introduces distinguishers and proves Theorem 8. Section 4 proves our main result:

Theorem 9. $\mathsf{NP} \notin \mathsf{FML}[n^c]$ *for all $c \in \mathbb{N}$ if there exists a real $\epsilon > 0$ and an $2^{n^{o(1)}}$-sparse problem $Q \in \mathsf{NP}$ such that (a) or (b):*

(a) $n^{-\epsilon}$-$Q \notin \mathsf{FML}[n^{1+2\epsilon+o(1)}]$.

(b) $n^{-\epsilon}$-$Q \notin \mathsf{PFML}[n^{2\epsilon+o(1)}]$.

This generalizes Theorem 4: since $1/n$-Q equals Q, Theorem 4 is the special case setting $\epsilon := 1$. For small ϵ, the magnification thresholds are well below known lower bounds, and in (b) even sublinear. Note already tiny improvements of magnification thresholds is what magnification is all about. The proof is based on the hash sketched in the previous subsection which considerably simplifies the hash from [6]; it does, however, not yield the mentioned strengthening of Theorem 3 in [6].

We find it remarkable that the magnification threshold Theorem 9 (b) obtained by our generic method turns out to be sharp. Section 6 generalizes Theorem 5 with a much simpler proof (but only for one-sided error):

Theorem 10. $n^{-\epsilon}$-$\mathsf{MCSP}[2^{\sqrt{\ell}}] \notin \mathsf{PFML}[n^{2\epsilon-\delta}]$ *for all reals* $0 < \epsilon, \delta \leqslant 1$.

An advantage of our simplified approach to magnification is that it works in a uniform setting. In Section 5 we prove:

Theorem 11. $\mathsf{P} \neq \mathsf{NP}^{\oplus\mathsf{P}}$ *if there exist a real* $\epsilon > 0$ *and a function* $\sigma(\ell) \leqslant 2^{o(\ell)}$ *such that*

$$n^{-\epsilon}\text{-}\mathsf{MCSP}[\sigma] \notin \mathsf{P}\text{-}\textit{uniform}\text{-}\mathsf{SIZE}[n^{1+\epsilon+o(1)}].$$

This magnification threshold can be judged "almost known" because Santhanam and Williams [30] showed $\mathsf{P} \notin \mathsf{P}$-*uniform*-$\mathsf{SIZE}[n^c]$ for all $c \in \mathbb{N}$.

Theorem 11 is interesting in that is seems to sidestep the *localization barrier*: [9] shows that many lower bound techniques *localize* in the sense that they still work when the circuits under consideration are enhanced with oracle gates of small fan-in. Such techniques cannot verify the magnification thresholds in Theorem 9 because every $2^{n^{o(1)}}$-sparse problem can be decided by size $\leqslant n^{2+o(1)}$ probabilistic formulas with certain oracle gates of fan-in $n^{o(1)}$ (Corollary 23). But Santhanam and Williams' proof does not seem to localize.

This motivates two questions: the first is for a more constructive proof of the lower bound in [30], one that actually exhibits an explicit hard problem in P; this is also relevant for applications in proof complexity – see [21, 5]. Second, can we find a *general* uniform magnification threshold? Say, plug any $2^{n^{o(1)}}$-sparse problem in place of $\mathsf{MCSP}[\sigma]$ above?

2 An easy example

We showcase how magnification is derived from compression by giving a very simple proof of a strengthening of Theorem 3.

Theorem 12 ([10])**.** *For all* $c \in \mathbb{N}$*, all reals* $\delta > 0$ *and* $\gamma < \delta/(c+1)$*, and all* 2^{n^γ}*-sparse problems* $Q \in \mathsf{NTIME}[2^{n^\gamma}]$*, if* $Q \notin \mathsf{SIZE}[n^{1+\delta}]$*, then* $\mathsf{NE} \notin \mathsf{SIZE}[n^c]$.

Proof. Let δ, c, γ accord the statement and assume $\mathsf{NE} \subseteq \mathsf{SIZE}[n^c]$. Let $Q \in \mathsf{NTIME}[2^{n^\gamma}]$ be 2^{n^γ}-sparse. We show $Q \in \mathsf{SIZE}[n^{1+\delta}]$.

Let K contain the tuples $\langle t, n, 1^{\lceil n^\gamma \rceil}, i\rangle$ such that t, n, i are natural numbers in binary, $1 \leqslant i \leqslant tn$, and there are $x_1, \ldots, x_t \in Q_n$ with $x_1 <_{lex} \cdots <_{lex} x_t$ such that the i-th bit of the concatenation $x_1 \cdots x_t \in \{0,1\}^{tn}$ is 1. Here, $<_{lex}$ denotes (strict) lexicographic order, and we write i with exactly $\lceil \log(tn+1) \rceil$ many bits. Clearly, $K \in \mathsf{NE}$.

We describe small circuits for Q. Fix $n \in \mathbb{N}$. We consider *interesting* inputs to K, namely, $\langle t, n, 1^{\lceil n^\gamma \rceil}, i\rangle$ where $t := |Q \cap \{0,1\}^n|$ and $1 \leqslant i \leqslant tn$. Their length $m = m(n) \leqslant O(n^\gamma)$ depends only on n. By assumption there is a size $\leqslant m^c$ circuit $C(i)$ deciding whether $\langle t, n, 1^{\lceil n^\gamma \rceil}, i\rangle \in Q$. Let $x^* \in \{0,1\}^{tn}$ be the concatenated list of the t strings in $Q \cap \{0,1\}^n$ in

$<_{lex}$ order. For every interesting input that is in K, the $x_1, \dots, x_t$ part of its witness must be x^*. Thus, $C(i) = 1$ if and only if the i-th bit of x^* is 1.

To decide whether an input $x \in \{0,1\}^n$ is in Q we check whether x appears in the list x^*. We employ binary search: this involves $\lceil \log t \rceil$ comparisons of n-bit strings, so $\lceil \log t \rceil \cdot n$ calls to C. In total, this gives a circuit for Q of size $O(\lceil \log t \rceil \cdot n \cdot m^c) \leqslant O(n^{\gamma+1+\gamma c})$. Since $\gamma < \delta/(c+1)$ this is $\leqslant n^{1+\delta}$ for large enough n. □

A strength of this magnification result is that it is general, a weakness is that its magnification threshold is maybe not "almost known". This motivates to study *almost formulas*, circuits where only few gates are allowed fan-out > 1 – for such formulas even subcubic lower bounds are known (cf. [9, Theorem 30]). [9, Theorem 29] magnifies slightly superlinear lower bounds for almost formulas and a gap-version of $\mathsf{MCSP}[\sigma]$, improving an earlier result [28, Theorem 1.4] for circuits. The proof is quite involved, building on a constructive version of Lipton and Young's anticheckers [22]. Compression via anticheckers also underlies the already mentioned proof complexity magnification result in [25].

3 Distinguishers

After some preliminaries, Section 3.3 proves Theorem 8 which is used to prove our main result Theorem 9. Theorem 11 requires *strongly explicit* distinguishers and we give a stand-alone construction in Section 3.4.

3.1 Preliminaries

For $n \in \mathbb{N}$ we write $[n] := \{1, \dots, n\}$ understanding $[0] = \varnothing$. Let $k \leqslant n$ be naturals and F a finite set. A random variable $\boldsymbol{x}$ (with values) in F^n is *k-uniform* if its n projections are k-wise independent and uniform, i.e., for every $I \subseteq [n]$ of size at most k, $\boldsymbol{x}_I$ is uniform in $F^{|I|}$; here, for $I = \{i_1 < \cdots < i_{|I|}\}$ and $x \in F^n$

$$x_I := x_{i_1} \cdots x_{i_{|I|}} \in F^{|I|}.$$

An $n \times m$ matrix X over F is *k-uniform* if for $\boldsymbol{j}$ uniform in $[m]$, $X^{\boldsymbol{j}}$ is k-uniform; here, X^j for $j \in [m]$ denotes the j-th column of X.

Example 13. Joffe [18] (see also [12, 2]) observed that for a finite field $\mathbb{F}$ the $n \times \mathbb{F}^k$ matrix X with entry $\sum_{i \in [k]} x_i \cdot \nu^{i-1}$ at row ν and column $(x_1, \dots, x_k)$ is k-uniform; here, we assume $[n] \subseteq \mathbb{F}$. If $\tilde{\mathbb{F}}$ is a subfield of $\mathbb{F}$, one gets a k-uniform $\tilde{X}$ over $\tilde{\mathbb{F}}$ by applying a suitable surjection from $\mathbb{F}$ onto $\tilde{\mathbb{F}}$ on each entry.

In particular, we represent the field $\mathbb{F}_{2^\ell}$ by $\{0,1\}^\ell$; it is well known that multiplication, addition and multiplicative inverses can be computed in polynomial time in such a representation [34]. Given $\ell' < \ell$ and a k-uniform X over $\mathbb{F}_{2^\ell}$ one obtaines a k-uniform matrix over $\mathbb{F}_{2^{\ell'}}$ by chopping off the last $\ell - \ell'$ many bits of each X_{ij}.

Recall, a *(generator matrix of a linear binary)* (n', n, δ)*-code* is a matrix $C \in \mathbb{F}_2^{n \times n'}$ such that xC, yC have Hamming distance $\geqslant \delta n'$ for all distinct $x, y \in \mathbb{F}_2^n$. We only need a very basic construction of codes because we can allow any rate $n' \leqslant n^{O(1)}$. We describe such a construction, it just composes a Reed-Solomon code with a Hadamard code – having concrete parameters eases the presentation later.

Lemma 14. *There is an algorithm that, given $n \in \mathbb{N}$, outputs an $(n', n, 1/4)$-code for some $n' \leqslant n^4$ in time polynomial in n.*

Proof. Choose a natural $\ell > 0$ such that $n \leqslant 2^\ell(\ell+1) \leqslant n^2$. Given an n-bit string x pad it with 0's to length $2^\ell(\ell+1)$. Then x determines $(x_1, \ldots, x_k) \in (\mathbb{F}_{2^{\ell+1}})^k$ for $k := 2^\ell$. Let p_x be the polynomial $\sum_{i<k} x_i X^i$ over $\mathbb{F}_{2^{\ell+1}}$. For $y \in \mathbb{F}_{2^{\ell+1}}$ let $H(y)$ have length $2^{\ell+1}$ and j-th bit $ay^\top$ where $a \in \{0,1\}^{\ell+1}$ is the lexicographically j-th string (recall we view $(\ell+1)$-bit strings as row vectors in $\mathbb{F}_2^{\ell+1}$). If $y \neq 0^{\ell+1}$, then $H(y)$ has Hamming weight 2^ℓ. Code x by the $n' := 2^{2\ell+2}$-bit string $x' := H(p_x(y_1)) \cdots H(p_x(y_{2^{\ell+1}}))$ where $y_1, \ldots, y_{2^{\ell+1}}$ lists $\mathbb{F}_{2^{\ell+1}}$. If x is non-zero, then $< k = 2^\ell$ many $p_x(y_i)$ are 0 in $\mathbb{F}_{2^{\ell+1}}$. Then x' has Hamming weight $> 2^\ell \cdot 2^{\ell+1}/2 = 2^{2\ell} = n'/4$. Note $n' \leqslant (2^\ell(\ell+1))^2 \leqslant n^4$ for large n.

Note the code is linear. Its matrix is computable in time $n^{O(1)}$ because the code of x can be computed in polynomial time. □

3.2 Existence

Recall, an (n, m, δ, ϵ)-distinguisher for $\epsilon \leqslant 1/n$ is an (m, n, δ)-code and we intend to trade larger ϵ for small weight. We start observing that such a trade-off is possible:

Proposition 15. *There exist $c, d \in \mathbb{N}$ such that for every sufficiently large $n \in \mathbb{N}$ and for every real $\epsilon \leqslant 1/(c \log n)$ there exists an $(n, dn, 1/5, \epsilon)$-distinguisher of weight at most $1/\epsilon$.*

Proof. Let $Hw(y)$ denote the Hamming weight of $y \in \{0,1\}^n$. The following is a biased random subset principle:

Claim. Let $0 \leqslant p \leqslant 1$ and $y \in \{0,1\}^n$. Let $\boldsymbol{x}$ be the random string in $\{0,1\}^n$ obtained by independently, for each $i \in [n]$, setting the i-th bit to 1 with probability p and to 0 with probability $(1-p)$. Then

$$\Pr[y\boldsymbol{x}^\top = 1] = 1/2 - (1-2p)^{Hw(y)}/2. \tag{1}$$

Write $w := Hw(y)$ and p_w for the r.h.s. of (1). The claim is trivial for $w = 0$. For $w > 0$ the claim follows by induction on w:

$$\Pr[y\boldsymbol{x}^\top = 1] = p \cdot (1 - p_{w-1}) + (1-p) \cdot p_{w-1} = (1-2p)p_{w-1} + p = p_w.$$

We choose the constants c, d in the course of proof. Fix $\epsilon \leqslant 1/(c \log n)$. Set $p := 1/(2\epsilon n)$ and let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{dn}$ be independent and distributed as $\boldsymbol{x}$ above. Then $\mathbb{E}[Hw(\boldsymbol{x}_j)] = 1/(2\epsilon)$ for every $j \in [dn]$. By Chernoff, $Hw(\boldsymbol{x}_j) > 1/\epsilon$ with probability $< 2^{-1/(d_1\epsilon)}$ for some constant d_1 (independent of d, c). Then, with probability $\geqslant 1 - dn2^{-1/(d_1\epsilon)}$, all $\boldsymbol{x}_j$s have

Hamming weight $\leqslant 1/\epsilon$. Call this event E_0. If we choose $c > d_1$, then $\epsilon \leqslant 1/(c \log n)$ implies $\Pr[E_0] > 1/2$ for large enough n (and all d).

For every $j \in [dn]$ and $y \in \{0,1\}^n$ of Hamming weight $w \geqslant \epsilon n$, we have by (1)

$$\Pr[y\boldsymbol{x}_j^\top = 1] \geqslant 1/2 - e^{-w/(\epsilon n)}/2 > 0.3.$$

Hence, $\mathbb{E}[\sum_{j\in[dn]} y\boldsymbol{x}_j^\top] > 0.3 \cdot dn$. By the Chernoff Bound, $\sum_{j\in[dn]} y^\top \boldsymbol{x}^j \geqslant 0.2 \cdot dn$ with probability at least $1-2^{-dn/d_2}$ for some constant d_2 (independent of c, d, d_1). Choosing $d > d_2$, this is $\geqslant 1 - 1/2 \cdot 2^{-n}$. By the union bound, this holds for all y as above simultaneously with probability $\geqslant 1/2$. Call this event E_1. For large enough n, the event $E_0 \cap E_1$ has positive probability. Let the dn columns of D consist of corresponding realizations of the $\boldsymbol{x}_j$s. □

3.3 Construction

For magnification we need explicit distinguishers. Theorem 8 follows from the following.

Theorem 16. *For all sufficiently large naturals w, n and all reals $\epsilon, \delta > 0$ satisfying the assumption $\delta \leqslant (1 - 1/(\epsilon w))/4$, there exists an (m, n, ϵ, δ)-distinguisher D of weight at most w, where $m \leqslant n^7$.*

Moreover, there is an algorithm that given sufficiently large $w < n$ outputs in time polynomial in $n + w$ a binary matrix D that is such a distinguisher simultaneously for all $\delta, \epsilon > 0$ satisfying $(1 - 1/(\epsilon w))/4$.

Proof. We can assume $w < n$: otherwise $D := C$ from Lemma 14 is a (m, n, δ, ϵ)-distinguisher for all $\epsilon > 0$ and $\delta \leqslant 1/4$. Note $m \leqslant n^4$ and, trivially, D has weight $\leqslant n \leqslant w$.

We first assume n is a power of 2 and later remove this assumption. Let $w < n$ be sufficiently large so that Lemma 14 applies and gives a $(w', w, 1/4)$-code C with $w' \leqslant w^4$. Let $\epsilon, \delta > 0$ be reals satisfying the assumption $(1 - 1/(\epsilon w))/4$.

The idea is to define a randomized map on n-bit strings x as follows: sample w many positions in the string; this determines a w-bit string y; apply the code above and output a random bit of the result. This outputs 1 with probability $\geqslant 1/4$ in the event that $y \neq 0^w$; and this is likely if w is large enough compared to the Hamming weight of x. We shall show that the map is implemented by a matrix with a column for each of the random choices. To bound the number of these columns we sample the w many positions not uniformly but only with pairwise independence. Details follow.

Let X be a 2-uniform $w \times n^2$ matrix over $\mathbb{F}_n$ (Example 13). View $\mathbb{F}_n$ with universe $[n]$. Write the j-th column X^j of X as $X^j(1) \cdots X^j(w)$. Our randomized map is defined as follows given $x = x(1) \cdots x(n)$:

1. sample $j \in [n^2]$ u.a.r.
2. set $y := y(1) \cdots y(w)$ where $y(i) := x(X^j(i))$ for $i = 1, \ldots, w$
3. sample $j' \in [w']$ u.a.r.
4. output the j'-th bit of yC.

Note the output for choices j in line 1 and j' in line 3 equals

$$\sum_{i\in[w]} C_{ij'}y(i) = \sum_{i\in[w]} C_{ij'}x(X^j(i)) = \sum_{p\in[n]} \sum_{\substack{i\in[w]\\ X^j(i)=p}} C_{ij'}x(p)$$

where the arithmetic is in $\mathbb{F}_2$. Define a binary $n \times n^2w'$-matrix D as follows. We index rows by numbers $p \in [n]$ and columns by pairs $(j, j') \in [n^2] \times [w']$. Define

$$D_{(p,(j,j'))} := \sum_{i\in[w]:X^j(i)=p} C_{ij'}. \tag{2}$$

Thus our randomized map outputs a random bit of xD. For fixed (j, j'), the sum (2) is empty and hence $D_{(p,(j,j'))} = 0$ for all but $\leqslant w$ many $p \in [n]$. Thus, D has weight $\leqslant w$.

Note $m := n^2 \cdot w' \leqslant n^6$. To show D is an (m, n, δ, ϵ)-distinguisher we show that if x has Hamming weight $\geqslant \epsilon n$, then the randomized map outputs 1 with probability

$$\geqslant (1 - 1/(\epsilon w))/4 \geqslant \delta.$$

Let the random variable $\boldsymbol{h}$ be the Hamming weight of the intermediate w-bit string y. It is the sum of w many pairwise independent indicator variables each with expectation $\geqslant \epsilon$. By Chebychev's inequality

$$\Pr[\boldsymbol{h} = 0] \leqslant \mathrm{Var}[\boldsymbol{h}]/\mathbb{E}[\boldsymbol{h}]^2 \leqslant 1/\mathbb{E}[\boldsymbol{h}] \leqslant 1/(\epsilon w)$$

since $\mathrm{Var}[\mathbf{h}] \leqslant \mathbb{E}[\mathbf{h}]$ by pairwise independence. Thus, $y \neq 0^w$ with probability $\geqslant 1 - 1/(\epsilon w)$. In this case yC has Hamming weight $\geqslant w'/4$.

Finally, assume n is not a power of two. Let k be such that $2^{k-1} < n < 2^k$. Consider the distinguisher D' just shown to exist with parameters n, ϵ, δ, w reset to $n' = 2^k$, $\delta' = \delta$, $\epsilon' = \epsilon/2, w' = 2w$ – note $\delta' \leqslant (1 - 1/(\epsilon' w'))/4$ by assumption $(1 - 1/(\epsilon w))/4$. D' is a $(n', m', \delta', \epsilon')$-distinguisher with $m' \leqslant (n')^6 \leqslant n^7$ for $n \geqslant 2^6$. Let D be D' with its last $n' - n$ rows removed. If $x \in \{0,1\}^n$ has Hamming weight $\geqslant \epsilon n$, then $x0^{n'-n}$ has Hamming weight $\geqslant \epsilon' n'$. Then the Hamming weight of xD equals that of $(x0^{n'-n})D'$ which is $\geqslant \delta' n' \geqslant \delta n$.

For the moreover-part, note that the definition of D does not depend on δ, ϵ. The polynomial time computability is clear by (2), Example 13, and Lemma 14. □

We do not know whether one can replace n^7 by $O(n)$ in Theorem 8. This is a matter of no concern for our applications.

3.4 Strongly explicit distinguishers

It is possible to modify our distinguisher construction to make it strongly explicit. Instead, we give an alternative construction from scratch by elaborating Naor and Naor's seminal work [26]. The wording "distinguishing" is from this work.

Theorem 17. *For every large enough n and every rational $0 < \epsilon < 1$ there exists $m \leqslant n^{11}$ and an $(n, m, 1/16, \epsilon)$-distinguisher D of weight $\leqslant 8\log(2n)/\epsilon$.*

Moreover, D is strongly explicit: *there is a polynomial time algorithm that given n, ϵ as above, and numbers $i \in [n], j \in [m]$ rejects if $i \notin [n]$ or $j \notin [m]$, and otherwise outputs D_{ij}.*

Proof. Assume first that n is a power of two. We show later how to get rid of this assumption. Using Example 13, let U be a 7-uniform matrix over $\mathbb{F}_2$ with n rows, and let V be a 2-uniform matrix over $\mathbb{F}_n$ with $2n$ rows. More concretely, U is obtained from a 7-uniform $n \times \mathbb{F}_n^7$ matrix over $\mathbb{F}_n$ by chopping off all but the first bit of each entry; V is obtained from a 2-uniform $2n \times \mathbb{F}_{2n}^2$ matrix X over $\mathbb{F}_{2n}$ with $X_{i,(a,b)} := \hat{\imath} \cdot a + b$ where $\hat{\imath}$ is the i-th element of $\mathbb{F}_{2n}$ in lexicographic order; recall, we represent $\mathbb{F}_{2n}$ by $\{0,1\}^{\log(2n)}$; then V over $\mathbb{F}_n$ is obtained letting $V_{i,(a,b)}$ be $X_{i,(a,b)}$ with the last bit chopped off.

Let $\boldsymbol{u}, \boldsymbol{v}$ be independent and uniformly distributed columns of U, V, respectively. Let K be the set of powers of 2 below n. For $k \in K$ define a random string $\boldsymbol{v}^k$ as the characteristic n-bit string of the subset of $[n]$ given by the first $2n/k$ components of $\boldsymbol{v}$. More formally, for $i \in [n]$, let $\hat{\imath}$ denote the i-th element of $\mathbb{F}_n$, and define the i-th bit of $\boldsymbol{v}^k$ by

$$\boldsymbol{v}^k_i := \begin{cases} 1 & \text{if there is } p \in [2n/k] : \boldsymbol{v}_p = \hat{\imath} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Note that $\boldsymbol{v}^k$ has Hamming weight $Hw(\boldsymbol{v}^k) \leqslant 2n/k$ with probability 1. For $x, y \in \{0,1\}^n$ let $x \wedge y \in \{0,1\}^n$ be the obtained by applying $\wedge$ bitwise, i.e., the i-th bit is 1 if and only if both x and y have i-th bit 1.

Claim 1: $\Pr[1 \leqslant Hw(y \wedge \boldsymbol{v}^k) \leqslant 7] \geqslant 1/4$ for all $y \in \{0,1\}^n$ with $k \leqslant Hw(y) \leqslant 2k$.

Proof of Claim 1: Write $w := Hw(y)$. Define $\boldsymbol{h}^k_j := y_{\boldsymbol{v}_j}$ for $j \in [2n/k]$ and set $\boldsymbol{h}_k := \sum_{j \in [2n/k]} \boldsymbol{h}^k_j$. Then $\mathbb{E}[\boldsymbol{h}^k_j] = w/n$, so $\mathbb{E}[\boldsymbol{h}_k] = 2w/k$, and $\mathrm{Var}[\boldsymbol{h}_k] \leqslant \mathbb{E}[\boldsymbol{h}_k]$ by pairwise independence. The event $Hw(y \wedge \boldsymbol{v}^k) = 0$ implies $\boldsymbol{h}_k = 0$. By Chebychev, the latter has probability $\leqslant \mathrm{Var}[\boldsymbol{h}_k]/\mathbb{E}[\boldsymbol{h}_k]^2 \leqslant 1/\mathbb{E}[\boldsymbol{h}_k] \leqslant k/(2w) \leqslant 1/2$. Since $\boldsymbol{h}_k \geqslant Hw(y \wedge \boldsymbol{v}^k)$ with probability 1 and $2w/k \leqslant 4$ the event $Hw(y \wedge \boldsymbol{v}^k) \geqslant 8$ implies $\boldsymbol{h}_k \geqslant 8$ and hence $|\boldsymbol{h}_k - 2w/k| \geqslant 4$. By Chebychev, this has probability $\leqslant \mathrm{Var}[\boldsymbol{h}_k]/16 \leqslant \mathbb{E}[\boldsymbol{h}_k]/16 \leqslant 2w/(16k) \leqslant 1/4$. ⊣

Define $\boldsymbol{r}^k := \boldsymbol{v}^k \wedge \boldsymbol{u}$.

Claim 2: $\Pr[y(\boldsymbol{r}^k)^\top = 1] \geqslant 1/8$ for all $y \in \{0,1\}^n$ with $k \leqslant Hw(y) \leqslant 2k$.

Proof of Claim 2: If $z \in \{0,1\}^n$ and $1 \leqslant Hw(z) \leqslant 7$, then $\Pr[z\boldsymbol{u}^\top = 1] = 1/2$. Indeed, let $I := \{i \in [n] \mid z_i = 1\}$; then $\Pr[z\boldsymbol{u}^\top = 1] = \Pr[Hw(\boldsymbol{u}_I) \text{ is odd}]$ and this equals $1/2$ because $\boldsymbol{u}_I$ is uniformly distributed in $\{0,1\}^{|I|}$.

Since $\Pr[y(\boldsymbol{r}^k)^\top = 1] = \Pr[(y \wedge \boldsymbol{v}^k)\boldsymbol{u}^\top = 1]$ and $\boldsymbol{v}^k$ and $\boldsymbol{u}$ are independent, Claim 2 follows from Claim 1. ⊣

We can assume that $\epsilon \geqslant 1/n$. Let k_0 be the maximal element in K that is $\leqslant \epsilon n$, and let $K_0 := \{k \in K \mid k \geqslant k_0\}$. Note $k_0 \geqslant \epsilon n/2$. Hence, for every $k \in K_0$ we have $2/\epsilon \geqslant n/k$ and $Hw(\boldsymbol{r}^k) \leqslant Hw(\boldsymbol{v}^k) \leqslant 2n/k \leqslant 4/\epsilon$ with probability 1.

Let $\boldsymbol{R}$ be the random $[n] \times K_0$ matrix whose column with index $k \in K_0$ is $\boldsymbol{r}^k$. Let $y \in \{0,1\}^n$ with $Hw(y) \geqslant \epsilon n$. Choose $k \in K_0$ such that $k \leqslant Hw(y) \leqslant 2k$. Thus $y(\boldsymbol{r}^k)^\top = 1$ and hence $y\boldsymbol{R} \neq 0$ with probability $\geqslant 1/8$ by Claim 2. For $\boldsymbol{z}$ uniformly distributed in $\{0,1\}^{K_0}$ (binary vectors indexed by K_0) and independent of $\boldsymbol{u}, \boldsymbol{v}$ and hence from $\boldsymbol{r}^k$, we thus have $(y\boldsymbol{R})\boldsymbol{z}^\top = y(\boldsymbol{R}\boldsymbol{z}^\top) = 1$ with probability $\geqslant 1/2 \cdot 1/8 = 1/16$. Further note that, with probability 1,

$$Hw(\boldsymbol{R}\boldsymbol{z}^\top) \leqslant |K_0| \cdot 4/\epsilon \leqslant \log n \cdot 4/\epsilon. \tag{4}$$

Observe that $\boldsymbol{R}\boldsymbol{z}^\top = g(\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{z})$ for some function g. Let the matrix D have a column $g(u, v, z)$ for every column u of U, column v of V, and $z \in \{0,1\}^{K_0}$. These are

$$m := |U| \cdot |V| \cdot 2^{|K_0|} = n^7 \cdot (2n)^2 \cdot n \tag{5}$$

many columns. For $x \in \{0,1\}^n$ with $Hw(x) \geqslant \epsilon n$ we have $Hw(xD) \geqslant m/16$. Thus, D is a $(n, m, \epsilon, 1/16)$-distinguisher of weight $\leqslant \log n \cdot 4/\epsilon$.

We check that D is strongly explicit. Given a row index i and a column index j we want to compute D_{ij}. The column index j determines the indices of columns u, v of U, V and a string $z \in \{0,1\}^{K_0}$. The column index of V is (a, b) for some $a, b \in \mathbb{F}_{2n}$. We have D_{ij} is the i-th entry in $\sum_{k \in K_0 : z_k = 1} v^k \wedge u$ (sum in $\mathbb{F}_2^n$). Test whether $u_i = 0$ in polynomial time given i and the column index of u (by evaluating a degree 7 polynomial in $\mathbb{F}_n$). If this is the case, then $D_{ij} = 0$. Otherwise, $D_{ij} = \sum_{k \in K_0 : z_k = 1} v_i^k \mod 2$. To determine the bits v_i^k recall (3): check whether there is $p \in [2n/k]$ such that $v_p = \hat{\imath}$; recall $\hat{\imath} \in \{0,1\}^{\log n}$. For this, compute the 2 solutions $p \in \mathbb{F}_{2n}$ to the 2 equations $p \cdot a + b = \hat{\imath} c$ where c is a bit.

This proves the theorem in the case n is a power of 2. For arbitrary n let $n \leqslant n' < 2n$ be a power of two, and D' be a distinguisher for n' as constructed above but for $\epsilon' := \epsilon/2$. Let D consist of the first n rows of $\tilde{D}$. If $x \in \{0,1\}^n$ has Hamming weight $\geqslant \epsilon n$, then $Hw(x0^{n'-n}) \geqslant \epsilon' n'$, so $Hw(xD) = Hw((x0^{n'-n})D') \geqslant n'/16 \geqslant n/16$. Hence, D is a $(n, m, 1/16, \epsilon)$-distinguisher. By (4), its weight is $4 \log n'/\epsilon' \leqslant 8 \log(2n)/\epsilon$. By (5) is has $\leqslant n^{11}$ columns for large enough n. □

4 General magnification

In this section we fix

- $m, w, r : \mathbb{N} \to \mathbb{N}$, $\epsilon, \delta : \mathbb{N} \to [0,1]$, $q : \mathbb{N} \to \mathbb{R}$; assume $m(n) \geqslant n$ for all $n \in \mathbb{N}$.
- D is a function that maps every sufficiently large natural n to an $(n, m(n), \delta(n), \epsilon(n))$-distinguisher $D(n)$ of weight $w(n)$;
- $Q \subseteq \{0,1\}^*$ is $2^{q(n)}$-sparse.

When n is clear from context we shall often write $m = m(n), w = w(n)$ etc.

4.1 The kernel

Definition 18. For $x \in \{0,1\}^n$ and $u = (u_1, \dots, u_{r(n)}) \in [m(n)]^{r(n)}$ define

$$k(x,u) := \langle n, u, b_1, \dots, b_{r(n)} \rangle$$

where $b_j = (xD)_{u_j}$ for all $j \in [r(n)]$. Define

$$K := K(Q,D,r) := \left\{ k(x,u) \mid x \in Q, u \in [m(|x|)]^{r(|x|)} \right\}.$$

Note that, using some straightforward encoding, $k(x,u)$ is a binary string of length independent of x and at most

$$10 r(n) \log m(n).$$

Lemma 19. *Let $n \in \mathbb{N}, x \in \{0,1\}^n$ and $\boldsymbol{u}$ be uniform in $[m]^r$. Then*

(a) if $x \in Q$, then $\Pr[k(x,\boldsymbol{u}) \in K] = 1$;

(b) if x is a NO instance of ϵ-Q, then $\Pr[k(x,\boldsymbol{u}) \in K] \leqslant 2^q (1-\delta)^r$.

Proof. (a) is obvious. For (b), note $k(x,u) = k(y,v)$ implies $u = v$ and $|y| = n$ and $(xD)_{u_j} = (yD)_{u_j}$ for all $j \in [r]$. For $y \in Q$, $d_H(x,y) \geqslant \epsilon n$, so $d_H(xD, yD) \geqslant \delta m$. Hence, the event that $k(x,\boldsymbol{u}) = k(y,v)$ for some v has probability at most $(1-\delta)^r$. A union bound over all $\leqslant 2^q$ many $y \in Q \cap \{0,1\}^n$ gives the claim. □

Lemma 20. *Assume r, D are computable in time polynomial in n, and $r(n) \geqslant n^{\Omega(1)}$. If $Q \in \mathsf{NP}$, then $K \in \mathsf{NP}$.*

Proof. Choose $c \in \mathbb{N}$ such that $r(n) \geqslant n^{1/c}$ for sufficiently large n. Given an input y check it has the form $\langle n, u, b_1, \dots, b_s \rangle$ for some $s \in \mathbb{N}$ with $s \geqslant n^{1/c}$. If the check fails, reject. Otherwise $n \leqslant |y|^c$, so $r := r(n), D := D(n), m := m(n)$ can be computed in time polynomial in $|y|$. Check $s = r$ and $u \in [m]^r$. If the check fails, reject. Otherwise guess $x \in \{0,1\}^n$ and verify $x \in Q$ in nondeterministic time polynomial in n, hence in $|y|$. Verify $b_j = (xD)_{u_j}$ for all $j \in [r]$. □

4.2 Small formulas with local oracles

The *locality barrier* from [9] states that many known circuit lower bounds methods stay true when the circuits are allowed *local* oracles, i.e., of small fan-in. "The fact that existing magnification theorems produce such circuits is a consequence of the algorithmic nature of the underlying proofs" [9, Appendix A.2]. To stress this aspect we isolate the construction of formulas with local oracles as an own lemma.

We consider formulas of the form

$$\mathsf{AND}_a \circ K_b \circ \mathsf{XOR}_c,$$

where $a, b, c \in \mathbb{R}_{\geqslant 0}$. Such a formula has a top AND-gate which receives $\leqslant a$ inputs from K-oracle gates which receive $\leqslant b$ inputs from XOR-gates which receive $\leqslant c$ inputs from the input gates and the constant 1; each inner gate has fan-out $\leqslant 1$.

That a promise problem Q is *decided by a probabilistic formula of the form* $K_b \circ \mathsf{XOR}_c$ *on input length* n means that there is a random variable $\boldsymbol{F}$ whose values are formulas of the form $K_b \circ \mathsf{XOR}_c$ and such that $\Pr[\boldsymbol{F}(x) = 1] = 1$ for all YES instances x of length n, and $\Pr[\boldsymbol{F}(x) = 1] \leqslant 1/4$ for all NO instances x of length n.

Lemma 21. *Let* $K := K(Q, D, r)$.

(a) If n *is sufficiently large and* $b : \mathbb{N} \to \mathbb{N}$ *satisfies*

$$b(n)r(n)\delta(n) - b(n)q(n) \geqslant n, \tag{6}$$

then ϵ*-*Q *on input length* n *is decided by a formula of the form*

$$\mathsf{AND}_{b(n)} \circ K_{10r(n)\log m(n)} \circ \mathsf{XOR}_{w(n)}.$$

(b) If n *is sufficiently large and*

$$r(n)\delta(n) - q(n) \geqslant 2, \tag{7}$$

then ϵ*-*Q *on input length* n *is decided by a probabilistic formula of the form*

$$K_{10r(n)\log m(n)} \circ \mathsf{XOR}_{w(n)}.$$

Proof. Observe that each bit b_j in $k(x, u)$ is the XOR of the input bits with index i such that $D_{iu_j} = 1$ and there are at most w many such i's. Hence, for fixed $u \in [m]^r$ every bit of $k(x, u)$ is computed by an XOR-gate of fan-in at most w. Applying a K-oracle gate on top of these XOR-gates gives a $K_{10r\log m} \circ \mathsf{XOR}_w$ formula F_u.

For $\boldsymbol{u}$ uniform in $[m]^r$ we get a random formula $F_{\boldsymbol{u}}$ such that the event $k(x, \boldsymbol{u}) \in K$ equals the event $F_{\boldsymbol{u}}(x) = 1$. By Lemma 19,

(Fa) if $x \in Q$, then $\Pr[F_{\boldsymbol{u}}(x) = 1] = 1$;

(Fb) if x is a NO instance of ϵ-Q, then $\Pr[F_{\boldsymbol{u}}(x) = 1] \leqslant 2^q(1-\delta)^r$.

Note $2^q(1-\delta)^r \leqslant 2^{q-r\delta} \leqslant 1/4$ by (7), so $F_{\boldsymbol{u}}$ witnesses (b). To prove (a), let $\bar{\boldsymbol{u}} = (\boldsymbol{u}^1, \ldots, \boldsymbol{u}^b)$ be a tuple of b independent random variables, each uniform in $[m]^r$, and set $F_{\bar{\boldsymbol{u}}} := \bigwedge_{i\in[b]} F_{\boldsymbol{u}^i}$. If $x \in Q$, then $\Pr[F_{\bar{\boldsymbol{u}}}(x) = 1] = 1$. If x is a NO instance of ϵ-Q, then by (6)

$$\Pr[F_{\bar{\boldsymbol{u}}}(x) = 1] \leqslant (2^q(1-\delta)^r)^b < 2^{bq-br\delta} \leqslant 2^{-n}.$$

Fix values of $\bar{\boldsymbol{u}}$ so that the resulting formula rejects all NO instances of length n. Clearly, this formula accepts all $x \in Q \cap \{0,1\}^n$. It has the required form. □

Remark 22. The derandomization argument above shows $\mathsf{PFML}[s(n)] \subseteq \mathsf{FML}[O(ns(n))]$. Hence, (a) implies (b) in Theorems 4, 9.

Corollary 23. *For all $0 < \epsilon \leqslant 1$ and all $2^{n^{o(1)}}$-sparse problems Q there is $K \subseteq \{0,1\}^*$ such that on sufficiently large input length n, n^ϵ-Q is decided by a probabilistic formula of size $\leqslant n^{2\epsilon+o(1)}$ with a single K-oracle gate of fan-in $\leqslant n^{o(1)}$.*

Proof. Assume Q is $q(n)$-sparse for $q(n) \leqslant 2^{n^{o(1)}}$. For large enough n, let D be an $(n, m, 1/5, n^{-\epsilon})$-distinguisher of weight $\leqslant n^\epsilon$ with $m \leqslant dn$ for some constant d (Proposition 15). Let $K := K(Q, D, r)$ for $r(n) := 5q(n) + 10$. Then (b) above gives a probabilistic formula of the form $K_{10r(n)\log m} \circ \mathsf{XOR}_{n^{-\epsilon}}$. Replace the XOR-gates by formulas of size $O(n^{2\epsilon})$. The K-oracle gate has fan-in $10r(n)\log m \leqslant n^{o(1)}$. □

4.3 General magnification

The following implies our main result Theorem 9. It generalizes [7, Theorem 1.3(3)] and [7, Theorem 1.1(4)] which essentially state the special case for $\epsilon = 1$.

Theorem 24. *For all $c \in \mathbb{N}$, all reals $\delta, \epsilon > 0$ and $\gamma < \delta/c$ and all 2^{n^γ}-sparse problems $Q \in \mathsf{NP}$, if (a) $n^{-\epsilon}$-$Q \notin \mathsf{FML}[n^{1+2\epsilon+\delta}]$ or (b) $n^{-\epsilon}$-$Q \notin \mathsf{PFML}[n^{2\epsilon+\delta}]$, then $\mathsf{NP} \not\subseteq \mathsf{FML}[n^c]$.*

Proof. Let $c, \delta, \gamma, \epsilon$ accord the assumption. We can assume $\epsilon \leqslant 1$ because the statement for $\epsilon > 1$ is implied by the one for $\epsilon = 1$. Set

$$q(n) := n^\gamma,\ \delta(n) := 1/8,\ w(n) := \lceil 2n^\epsilon \rceil,\ r(n) := 17\lceil n^\gamma \rceil.$$

If n is sufficiently large, Theorem 8 gives an $(m, n, \delta(n), n^\epsilon)$ distinguisher of weight $w(n)$ with $m \leqslant n^7$. Let $Q \in \mathsf{NP}$ be $2^{q(n)}$-sparse. Let $K := K(Q, D, r)$ be the kernel from Definition 18. Then $K \in \mathsf{NP}$ by Lemma 20. Assume

$$K \in \mathsf{FML}[n^c]. \tag{8}$$

(a): for $b(n) := \lceil n^{1-\gamma} \rceil$ we have $br\delta - bq \geqslant n$ for sufficiently large n, so (6) of Lemma 21 (a) is satisfied. We get a

$$\mathsf{AND}_{\lceil n^{1-\gamma} \rceil} \circ K_{170\lceil n^\gamma \rceil \log m} \circ \mathsf{XOR}_{\lceil 2n^\epsilon \rceil}$$

formula F deciding $n^{-\epsilon}$-Q on instances of sufficiently large length n. By (8), the oracle gates in F can be replaced by formulas of size $O((n^\gamma \log n)^c)$. The XOR-gates can be replaced by quadratic size formulas, i.e., size $O(n^{2\epsilon})$. The resulting formula is equivalent to F and has size (assuming n is sufficiently large)

$$O(n^{1-\gamma} \cdot n^{\gamma c} \cdot (\log n)^c \cdot n^{2\epsilon}) \leqslant n^{1+\delta+2\epsilon}.$$

(b): $r\delta - q \geqslant n^\gamma \geqslant 2$ for sufficiently large n, so (7) of Lemma 21 (b) is satisfied. We get a probabilistic formula of the form $K_{cn^\gamma \log n} \circ \mathsf{XOR}_{\lceil 2n^\epsilon \rceil}$ for some $c \in \mathbb{N}$. Replacing the oracle gates by formulas of size $O((n^\gamma \log n)^c) \leqslant o(n^\delta)$ and the XOR-gates by formulas of size $O(n^{2\epsilon})$ yields size $\leqslant n^{2\epsilon+\delta}$. □

5 Uniform magnification

Section 5.1 generalizes a magnification result for $\mathsf{MCSP}[\sigma]$ from [7]. This illustrates a use of strongly explicit distinguishers and a certain flexibility of our method. Section 5.2 then infers Theorem 11 by a modification of the proof.

5.1 How to use strongly explicit distinguishers

The following generalizes [7, Theorem 1.3(1)] which states the special case for $\epsilon = 1$ and $\sigma(\ell) = 2^{\gamma\ell}$. The generality with relation to σ is worth to be stated because, as mentioned in the introduction, it is unknown whether $\mathsf{MCSP}[\sigma]$ becomes harder for larger σ.

Theorem 25. *For all reals $\delta, \epsilon > 0$ there is a real $\gamma > 0$ such that for all $\sigma : \mathbb{N} \to \mathbb{R}_{\geqslant 0}$ with $\sigma(\ell) \leqslant 2^{\gamma\ell}$, if $n^{-\epsilon}$-$\mathsf{MCSP}[\sigma] \notin \mathsf{PFML}[n^{2\epsilon+\delta}]$, then $\oplus\mathsf{P} \not\subseteq \mathsf{NC}_1$.*

Proof. Let $\delta, \epsilon > 0$ and assume $\oplus\mathsf{P} \subseteq \mathsf{NC}_1$; we can assume $0 < \epsilon \leqslant 1$. For large enough n, let $D = D(n)$ be a strongly explicit $(n, m(n), 1/16, n^{-\epsilon})$-distinguisher of weight $\leqslant 8\log(2n)n^{\epsilon}$ and $m(n) \leqslant n^{11}$ according to Theorem 17.

Call a binary string *good* if it has the form

$$\langle n, s, u, b_1, \ldots, b_r \rangle$$

where $n, s, r \in \mathbb{N}$, $n = 2^{\ell}$ for some $\ell \in \mathbb{N}$, $s \leqslant r$, $u \in [m(n)]^r$ and the b_j are bits. Define $K \subseteq \{0,1\}^*$ as the set of such strings such that there exists $x \in \{0,1\}^n$ that is computed by a circuit of size $\leqslant s$ and $b_j = (xD)_{u_j}$ for all $j \in [r]$.

Claim: $K \in \mathsf{NC}_1$.

Proof of the claim: We first describe a nondeterministic polynomial time algorithm with a $\oplus\mathsf{P}$ oracle. A given input can be checked to be good in polynomial time: since D is strongly explicit, $u_j \in [m(n)]$ can be checked in polynomial time. We guess a circuit C of size s with ℓ inputs: this takes polynomial time since $s \leqslant r$. For each $j \in [r]$ we have to check $b_j = (xD)_{u_j}$ where $x \in \{0,1\}^n$ is the truth table of C, i.e., for all $i \in [n]$ we have $x_i = C(\hat{\imath})$ where $\hat{\imath}$ is the lexicographically i-th length ℓ string. We thus have to check whether there is an odd number of $i \in [n]$ such that $C(\hat{\imath}) \cdot D_{iu_j} = 1$. But, since D is strongly explicit, this is a $\oplus\mathsf{P}$-property of $\hat{\imath} \in \{0,1\}^{\ell}$, so checked by the oracle.

Thus, $K \in \mathsf{NP}^{\oplus\mathsf{P}}$. Now argue:

$$\mathsf{NP}^{\oplus\mathsf{P}} \subseteq \mathsf{RP}^{\oplus\mathsf{P}} \subseteq \mathsf{P}^{\oplus\mathsf{P}}/\mathsf{poly} \subseteq \oplus\mathsf{P}/\mathsf{poly} \subseteq \mathsf{NC}_1;$$

the 1st inclusion follows from $\mathsf{NP} \subseteq \mathsf{RP}^{\oplus\mathsf{P}}$ by the Valiant-Vazirani Lemma, the 2nd from Adleman's trick, the 3rd from $\oplus\mathsf{P}^{\oplus\mathsf{P}} = \oplus\mathsf{P}$ [29], and the 4th from our assumption. ⊣

Choose $c \in \mathbb{N}$ such that $K \in \mathsf{FML}[n^c]$ and set $\gamma := \delta/(4c)$. Let $\sigma(\ell) \leqslant 2^{\gamma\ell}$ be given. We have to show that $n^{-\epsilon}$-$\mathsf{MCSP}[\sigma] \in \mathsf{PFML}[n^{\delta+2\epsilon}]$.

Set $q(n) := n^{2\gamma}$ and note $\mathsf{MCSP}[\sigma]$ is $2^{q(n)}$-sparse. Let $\tilde{r}(n) := \lceil 16q(n) + 32 \rceil$ and set $\tilde{K} := K(\mathsf{MCSP}[\sigma], D, \tilde{r})$ according to Definition 18. This satisfies (7) of Lemma 21, so $n^{-\epsilon}$-$\mathsf{MCSP}[\sigma]$ has probabilistic formulas of the form

$$\tilde{K}_{O(n^{2\gamma}\log n)} \circ \mathsf{XOR}_{O(n^{\epsilon}\log n)}$$

on inputs of sufficiently large length $n = 2^{\ell}$. We want to replace the $\tilde{K}$-oracle gates by K-oracle gates. To this end, note that, by the proof of Lemma 21, the XOR-gates produce an input to $\tilde{K}$ of the required form and this is a good string (with $r = \tilde{r}(n)$) except that some s is missing. Hence, we just have to add some gates producing $s := \sigma(\ell)$. Note $s \leqslant \tilde{r}(n)$, so the result is good.

Replace the K-oracle and XOR gates by formulas of size $O((q(n)\log n)^c) \leqslant O(n^{\delta/2})$ and $O(n^{2\epsilon}\log^2 n)$. This yields a formula of size $\leqslant n^{2\epsilon+\delta}$ for sufficiently large n. □

Corollary 26. *$\oplus\mathsf{P} \notin \mathsf{NC}_1$ if there exist $\epsilon > 0$ and a function $\sigma(\ell) \leqslant 2^{o(\ell)}$ such that*

$$n^{-\epsilon}\text{-}\mathsf{MCSP}[\sigma] \notin \mathsf{PFML}[n^{2\epsilon+o(1)}].$$

5.2 Adding uniformity

The following implies Theorem 11.

Theorem 27. *For all reals $\delta, \epsilon > 0$ there is a real $\gamma > 0$ such that for all $\sigma : \mathbb{N} \to \mathbb{R}_{\geqslant 0}$ with $\sigma(\ell) \leqslant 2^{\gamma\ell}$, if $n^{-\epsilon}$-$\mathsf{MCSP}[\sigma] \notin \mathsf{P}$-uniform-$\mathsf{SIZE}[n^{1+\epsilon+\delta}]$, then $\mathsf{P} \neq \mathsf{NP}^{\oplus\mathsf{P}}$.*

Proof. Assume $\mathsf{P} = \mathsf{NP}^{\oplus\mathsf{P}}$ and proceed as in the proof of Theorem 25. The proof of the claim now gives $K \in \mathsf{P}$. Choose $c \in \mathbb{N}$ such that $K \in \mathsf{P}$-uniform-$\mathsf{SIZE}[n^c]$ and set $\gamma = \delta/(4c)$, as before. As seen, Lemma 21 gives a probabilistic formula of the form

$$K_{O(n^{2\gamma}\log n)} \circ \mathsf{XOR}_{O(n^{\epsilon}\log n)}$$

deciding $n^{-\epsilon}$-$\mathsf{MCSP}[\sigma]$. Recalling the proof of Lemma 21, the formula equals $F_{\boldsymbol{u}}$ for $\boldsymbol{u}$ uniform in $[m]^{\tilde{r}}$ where $m, \tilde{r}$ are as in the proof of Theorem 25. It is clear that the map $u \mapsto F_u$ is computable in polynomial time.

Let $\bar{\boldsymbol{u}} = (\boldsymbol{u_1}, \ldots, \boldsymbol{u}_n)$ be uniform in $([m]^{\tilde{r}})^n$. Then $F_{\bar{\boldsymbol{u}}} := \bigwedge_{i\in[n]} F_{\boldsymbol{u}_i}$ accepts all YES instances and rejects NO instances with probability $\leqslant 4^{-n} < 2^{-n}$. Hence there is a realization $\bar{u}$ of $\bar{\boldsymbol{u}}$ such that $F_{\bar{u}}$ rejects all NO instances. It is not hard to see that such $\bar{u}$ can be computed from n in time polynomial in n with the help of a PH oracle. But our assumption implies $\mathsf{PH} = \mathsf{P}$, so $F_{\bar{u}}$ can be computed in time polynomial in n.

Compute a circuit to replace the K-oracle gates and a linear size circuit to replace the XOR-gates. The resulting circuit is computable in time polynomial in n and has size

$$O(n \cdot (n^{2\gamma}\log n)^c \cdot n^{\epsilon}\log n) \leqslant n^{1+\delta+\epsilon}.$$ □

Discussion For P-uniformity the distinguishers are not required to be strongly explicit. But the P-uniform version of Theorem 24 is void: its conclusion $\mathsf{NP} \not\subseteq \mathsf{P}$-*uniform*-$\mathsf{FML}[n^c]$ is known [30], as discussed in the introduction. Thus, contrary to our focus in the introduction on the magnification threshold, strengthening the conclusion of Theorem 24 could yield *general* uniform magnification as asked for in the introduction.

6 A sharp lower bound

This section proves Theorem 10. The method of proof follows Hirahara and Santhanam's [15] which in turn builds on [17]. We start recalling what is needed.

6.1 Preliminaries

Let $n \in \mathbb{N}$. A *random restriction (on* $[n]$*)* $\boldsymbol{\rho}$ is a random variable in $\{0,1,*\}^n$, the set of *restrictions (on* $[n]$*)*; we write restrictions ρ as functions $\rho : [n] \to \{0,1,*\}$. For $p \in [0,1]$ and $k \in [n]$, $\boldsymbol{\rho}$ is (p,k)-*regular* if the projections $\boldsymbol{\rho}(1), \dots, \boldsymbol{\rho}(n)$ are k-wise independent and each takes values $*, 0, 1$ with probabilities $p, (1-p)/2, (1-p)/2$.

For a Boolean function f with n variables and a restriction ρ on $[n]$, the function $f{\restriction}\rho$ is f with i-th variable fixed to $\rho(i)$ for all $i \in [n]$ with $\rho(i) \neq *$. For another restriction ρ' observe $(f{\restriction}\rho){\restriction}\rho' = f{\restriction}\rho\rho'$ where $\rho\rho'$ maps i to $\rho(i)$ if $\rho(i) \neq *$, and to $\rho'(i)$ otherwise.

Example 28. Let $k, 2^r \leqslant n$ and X be an $n \times \mathbb{F}_{2^r}^k$-matrix over $\mathbb{F}_{2^r}$ according to Example 13. Assume $r = r' + 1$ and set $p := 2^{-r'}$. Choose $f : \mathbb{F}_{2^r} \to \{0,1,*\}$ such that $*$ has $p2^r = 2$ preimages and $0, 1$ both have $(1-p)2^r/2 = 2^{r'} - 1$ preimages. Define $\boldsymbol{\rho}$ to be a uniformly chosen column of X with entries replaced using f. Then $\boldsymbol{\rho}$ is a (p,k)-regular random restriction on $[n]$.

By $L(f)$ we denote the minimal number of leafs of (the formula tree of) a formula computing f. The following is a version of [15, Lemma 27] (there attributed to [17]).

Lemma 29. *There exists a real* $c \geqslant 1$ *such that for every* $n \in \mathbb{N}_{>0}$*, every* $f : \{0,1\}^n \to \{0,1\}$*, every* $p \in [0,1]$*, every natural* $k \geqslant 1/p^2$ *and every* (p,k)*-regular random restriction* $\boldsymbol{\rho}$ *on* $[n]$*:*

$$\mathbb{E}[L(f{\restriction}\boldsymbol{\rho})] \leqslant \max\{cp^2 L(f), c\}.$$

Remark 30. [15, Lemma 27] has $O(p^2 L(f))$ instead $\max\{cp^2 L(f), c\}$ and an additional assumption $L(f) \geqslant p^{-2}$. In the proof, this assumption is used to bound $\mathbb{E}[L(g{\restriction}\boldsymbol{\rho})]$ by $O(p^2 L(g))$ for certain functions g satisfying $L(g) \geqslant p^{-2}/6$ by the additional assumption; indeed, then $O(p^2 L(g)) \geqslant O(p^2 L(g) + p\sqrt{L(g)}) \geqslant \mathbb{E}[L(g{\restriction}\boldsymbol{\rho})]$ as proved in [35] (improving [14]). Our version uses the bound $\max\{cp^2 L(g), c\}$ instead.

A random restriction $\boldsymbol{\rho}$ is a *composition of* t *many* (p,k)*-regular restrictions on* $[n]$ if $\boldsymbol{\rho} = \boldsymbol{\rho}_1 \cdots \boldsymbol{\rho}_t$ for independent (p,k)-regular restrictions $\boldsymbol{\rho}_1, \dots, \boldsymbol{\rho}_t$ on $[n]$. Note that such $\boldsymbol{\rho}$ is (p^t, k)-regular. The following is a variant of [15, Theorem 28] (there attributed to [17]).

Lemma 31. *There exists a real $c \geqslant 1$ such that for all $t, n \in \mathbb{N}_{>0}$, every $f : \{0,1\}^n \to \{0,1\}$, every real p with $0 < p < 1/(2c)^{1/2}$, every natural $k \geqslant 1/p^2$, and every composition $\boldsymbol{\rho}$ of t many (p,k)-regular random restrictions on $[n]$:*

$$\mathbb{E}[L(f{\restriction}\boldsymbol{\rho})] \leqslant c^t p^{2t} L(f) + 2c.$$

Proof. Let c be the constant from Lemma 29. We proceed by induction on t and prove a slightly stronger claim with $2c$ replaced by $c/(1-cp^2)$ (which is $\leqslant 2c$ by $p < 1/(2c)^{1/2}$).

The case $t = 1$ follows from Lemma 29 because $\max\{cp^2 L(f), c\} \leqslant cp^2 L(f) + c/(1-cp^2)$. For the inductive step, let $t > 0$ and recall $\boldsymbol{\rho} = \boldsymbol{\rho}'\boldsymbol{\rho}_t$ for $\boldsymbol{\rho}' := \boldsymbol{\rho}_1 \cdots \boldsymbol{\rho}_{t-1}$ where the $\boldsymbol{\rho}_i$ are independent and (p,k)-regular. Letting ρ' range over realizations of $\boldsymbol{\rho}'$,

$$\begin{aligned}
\mathbb{E}[L(f{\restriction}\boldsymbol{\rho})] &= \textstyle\sum_{\rho'} \Pr[\boldsymbol{\rho}' = \rho'] \cdot \mathbb{E}[L((f{\restriction}\rho'){\restriction}\boldsymbol{\rho}_t)] && \text{as } \boldsymbol{\rho}', \boldsymbol{\rho}_t \text{ are independent} \\
&\leqslant \textstyle\sum_{\rho'} \Pr[\boldsymbol{\rho}' = \rho'] \cdot \big(cp^2 L(f{\restriction}\rho') + c\big) && \text{by Lemma 29} \\
&= cp^2 \mathbb{E}[L(f{\restriction}\boldsymbol{\rho}')] + c && \\
&\leqslant cp^2\big(c^{t-1}p^{2(t-1)} L(f) + c/(1-cp^2)\big) + c && \text{by induction} \\
&= c^t p^{2t} L(f) + c/(1-cp^2). &&
\end{aligned}$$

□

6.2 The lower bound

The following implies Theorem 10.

Theorem 32. *There exists $d \in \mathbb{N}_{>0}$ such that for all reals $0 < \epsilon, \delta \leqslant 1$ and all $\sigma : \mathbb{N} \to \mathbb{R}_{\geqslant 0}$ with $\ell^d \leqslant \sigma(\ell) \leqslant 2^{o(\ell)}$,*

$$n^{-\epsilon}\text{-}\mathsf{MCSP}[\sigma] \notin \mathsf{PFML}[n^{2\epsilon-\delta}].$$

Proof. We choose $d > 0$ in the end of the proof. Let δ, ϵ, σ be as stated. Write $2^\ell = n$ and $\sigma = \sigma(\ell)$. It suffices to rule out probabilistic formulas of size at most

$$s := n^{2\epsilon - 6\delta}.$$

We want to choose p, t, k in Lemma 31 to ensure that $L(F{\restriction}\boldsymbol{\rho})$ is probably constant. Let c be the lemma's constant, and set

$$k := \lceil \sigma^{1/d} \rceil, \quad t := \left\lceil \frac{2d \log s}{\log \sigma} \right\rceil.$$

and for p choose a negative power of 2 such that

$$(c^{1/2} s^{1/(2t)})^{-1}/2 \leqslant p \leqslant (c^{1/2} s^{1/(2t)})^{-1}.$$

This ensures $c^t p^{2t} \leqslant 1/s$. As $p \leqslant 1/(2c)^{1/2}$ and $1/p^2 \leqslant 4cs^{1/t} \leqslant 4c\sigma^{1/(2d)} \leqslant k$ for large enough n, the lemma applies and gives $\mathbb{E}[L(F{\restriction}\boldsymbol{\rho})] \leqslant 1 + 2c$. Thus, for a suitable constant $s_0 \in \mathbb{N}$,

$$\Pr[L(F{\restriction}\boldsymbol{\rho}) \geqslant s_0] < 1/2. \tag{9}$$

Given $\boldsymbol{\rho}$ and an independent $\boldsymbol{z}$ uniform in $\{0,1\}^n$ define the random variable $\boldsymbol{x}$ in $\{0,1\}^n$ to have i-th bit equal to $\boldsymbol{\rho}(i)$ if $\boldsymbol{\rho}(i) \neq *$, and otherwise equal to the i-th bit of $\boldsymbol{z}$.

Let E be the event that $\boldsymbol{x}$ is *not* a NO instance of $n^{-\epsilon}$-MCSP$[\sigma]$, and $\bar{E}$ its complement.

Claim: $\Pr[E] \leqslant o(1)$.

Proof of the claim: Set $\boldsymbol{S} \coloneqq \boldsymbol{\rho}^{-1}(*)$. Since $\boldsymbol{\rho}$ is (p^t, k)-regular, $|\boldsymbol{S}|$ has expectation $p^t n$. Using $(4c)^{t/2} \leqslant s^{o(1)}$,

$$p^t n \geqslant n/((4c)^{t/2} s^{1/2}) \geqslant n \cdot s^{-1/2-o(1)} \geqslant n^{1-\epsilon+3\delta-o(1)} \geqslant 4n^{1-\epsilon+2\delta}. \tag{10}$$

The variance of $|\boldsymbol{S}|$ is $\leqslant \mathbb{E}[|\boldsymbol{S}|]$ by $k \geqslant 2$-wise independence. Let E_0 be the event that $|\boldsymbol{S}| \geqslant 2n^{1-\epsilon+2\delta}$, and $\bar{E}_0$ be its complement. By Chebychev, $\Pr[\bar{E}_0] \leqslant 4/\mathbb{E}[|\boldsymbol{S}|] \leqslant o(1)$. Thus, it suffices to show $\Pr[E \mid E_0] \leqslant o(1)$.

Let S range over realizations of $\boldsymbol{S}$ with $|S| \geqslant 2n^{1-\epsilon+2\delta}$, and let y range over YES instances. Then $E \subseteq \bigcup_y E_y$ where E_y is the event that $d_H(\boldsymbol{x}_{\boldsymbol{S}}, y_{\boldsymbol{S}}) < n^{1-\epsilon}$. Since there are at most $2^{n^{o(1)}}$ many YES instances, $\Pr[E \mid E_0] \leqslant 2^{n^{o(1)}} \cdot \max_y \Pr[E_y \mid E_0]$. Hence, it suffices to show $\Pr[E_y \mid E_0] \leqslant 2^{-n^{1-\epsilon+\delta}}$ for all y. Since $\Pr[E_y \mid E_0] \leqslant \max_S \Pr[E_y \mid \boldsymbol{S} = S]$ it suffices to show

$$\Pr[E_y \mid \boldsymbol{S} = S] \leqslant 2^{-n^{1-\epsilon+\delta}},$$

for all y, S. To see this, note $\Pr[E_y \mid \boldsymbol{S} = S] = \Pr[d_H(\boldsymbol{z}_S, y_S) < n^{1-\epsilon}]$ (by independence of $\boldsymbol{z}$ and $\boldsymbol{S}$) and $d_H(\boldsymbol{z}_S, y_S)$ is the sum of $|S|$ many independent indicators each with expectation $1/2$. By Chernoff, $\Pr[d_H(\boldsymbol{z}_S, y_S) < n^{1-\epsilon}] \leqslant 2^{-\Omega(|S|)} \leqslant 2^{-n^{1-\epsilon+\delta}}$. ⊣

For contradiction, assume $\boldsymbol{F}$ is a size $\leqslant s$ probabilistic formula for $n^{-\epsilon}$-MCSP$[\sigma]$. Then, letting x range over NO instances,

$$\Pr[\boldsymbol{F}(\boldsymbol{x}) = 1] \leqslant \Pr[E] + \textstyle\sum_x \Pr[\boldsymbol{F}(x) = 1, \boldsymbol{x} = x] \leqslant o(1) + 1/4 \Pr[\bar{E}] < 1/2.$$

We thus find realizations F, ρ, z, x of $\boldsymbol{F}, \boldsymbol{\rho}, \boldsymbol{z}, \boldsymbol{x}$ such that $F(x) = 0$ and and $L(F{\restriction}\rho) < s_0$ (by (9)). We get a contradiction by finding a YES instance x' rejected by F.

Let x'' be ρ with $*$ replaced by 0. Then define x' from x'' by flipping 0s to 1s in positions corresponding to variables in $F{\restriction}\rho$ that are 1 in z. Then $F(x') = (F{\restriction}\rho)(x) = F(x) = 0$. Note that $< s_0$ many positions are flipped. We are left to show that x' is computed by a circuit of size $\leqslant \sigma$.

First consider x''. We ask for a small circuit with 2 output bits that computes ρ in the encoding $01, 11, 00$ of $0, 1, *$. Then, taking the conjunction of the output bits gives a small circuit for x''. To get such a circuit we choose $\boldsymbol{\rho}$ concretely as the composition of t many (p,k)-regular $\tilde{\boldsymbol{\rho}}$ according to Example 28 – this can be done because $p = 2^{-r}$ for some $r \in \mathbb{N}$ with $2^{r+1} \leqslant n$; indeed, $2^{r+1} = 2/p \leqslant 4c^{1/2}\sigma^{1/(4d)} \leqslant k \leqslant n$ for large enough n.

We claim there is a circuit of size polynomial in $k\ell$. Since $t < \ell$, it suffices to show each of the corresponding realizations $\tilde{\rho}$ is computed by a small circuit. Each $\tilde{\rho}$ is determined by some $j \in \mathbb{F}_{2^{r+1}}^k$, and $i \mapsto \tilde{\rho}(i)$ is computed (given $k, 2^{r+1} \leqslant k, n = 2^\ell, j$) in time polynomial in $k\ell$. Choose $e \in \mathbb{N}$ such that x'' is computed by a circuit C'' of size $\leqslant (k\ell)^e$.

To get a circuit C' for x' we change $< s_0$ values of C''. By a look-up table, C' has size $\leqslant (k\ell)^e + 10 s_0 \ell$. As $k, \ell \leqslant \lceil \sigma^{1/d} \rceil$, we can choose $d \in \mathbb{N}$ such that C' has size $\leqslant \sigma$. □

Remark 33. Given a real $c > 0$ we can choose d sufficiently large, so that the YES instance x' constructed has circuit complexity $\sigma^{1/c}$. Hence, the above holds for the gap / approximation version of MCSP whose YES instances are strings computed by circuits of size $\leqslant \sigma^{1/c}$ and whose NO instances are strings that cannot be $(1 - n^{-\epsilon})$-approximated by circuits of size $\leqslant \sigma$.

Discussion [7, Theorem 1.5] proves a slight strengthening of the special case $\epsilon = 1$. As mentioned in the introduction, the proof method in [7] is much more complicated and deserves independent interest. It can handle two-sided error and for $\sigma(\ell) = 2^{\gamma\ell}$ achieves even slightly superquadratic lower bounds $n^{2+\gamma/2}$ [7, Theorem 1.6] (based on [11]). This outperforms the simpler method here. We do not know whether it extends to $\epsilon < 1$.

Concerning lower bounds for the approximation version, [27, Theorem 4.7] extended Theorem 2 to $\epsilon(n)$-MCSP$[2^{\sqrt{\ell}}]$ but only for tiny $\epsilon(n) \leqslant n^{-1+\Omega(1/\sqrt{\log n})}$. [28] states that "existing lower bound methods are more suitable for proving lower bounds" [28, p.6] for the gap version as opposed to the approximation version of MCSP$[\sigma]$. We doubt this, given Theorem 32 and its proof. Maybe this is an interesting insight.

References

[1] E. Allender. Vaughan Jones, Kolmogorov Complexity, and the New Complexity Landscape around Circuit Minimization. New Zealand Journal of Mathematics 52: 585-604, 2021.

[2] N. Alon, O. Goldreich, J. Håstad, R. Peralta. Simple constructions of almost k-wise independent random variables. Random Structures and Algorithms 3 (3): 289-304, 1992. Preliminary version in 31st Annual Symposium on Foundations of Computer Science (FOCS90), pp. 544-553, 1990.

[3] A. Atserias, S. Buss, M. Müller. On the consistency of circuit lower bounds for non-deterministic time. Journal of Mathematical Logic, to appear. Preliminary version: 55th Annual ACM Symposium on Theory of Computing (STOC23), pp. 1257-1270, 2023.

[4] H. Buhrman, J. M. Hitchcock. NP-hard sets are exponentially dense unless coNP C NP/poly. 23rd Conference on Computational Complexity (CCC08), pp.1-7, 2008.

[5] J. Bydzovsky, M. Müller. Polynomial time ultrapowers and the consistency of circuit lower bounds. Archive for Mathematical Logic 59 (1): 127-147, 2020.

[6] L. Chen, C. Jin, R. R. Williams. Hardness magnification for all sparse NP languages. 60th Symposium on Foundations of Computer Science (FOCS19), pp. 1240-1255, 2019.

[7] L. Chen, C. Jin, and R. R. Williams. Sharp threshold results for computational complexity. 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC 2020), pp. 1335–1348, 2020.

[8] L. Chen, C. Jin, R. Santhanam, R. Williams. Constructive Separations and Their Consequences. TheoretiCS 3, 2024. Preliminary version: 62nd Annual Symposium on Foundations of Computer Science (FOCS22) pp. 646-657, 2022.

[9] L. Chen, S. Hirahara, I. C. Oliveira, J. Pich, N. Rajgopal, R. Santhanam. Beyond natural proofs: hardness magnification and locality. Journal of the ACM 69 (4): Article 25, 2022. Preliminary version: 11th Innovations in Theoretical Computer Science (ITCS20), LIPIcs 151, pp. 70:1-70:48, 2020.

[10] L. Chen, D. M. McKay, C. D. Murray, R. Ryan Williams. Relations and equivalences between circuit lower bounds and Karp-Lipton theorems. 34th Computational Complexity Conference (CCC19), LIPIcs 137, pp. 30:1–30:21, 2019.

[11] M. Cheraghchi, V. Kabanets, Z. Lu, D. Myrisiotis. Circuit Lower Bounds for MCSP from Local Pseudorandom Generators. ACM Transactions of Computation Theory 12 (3): Article 21, 2020. Preliminary version: 6th International Colloquium on Automata, Languages, and Programming (ICALP19). LIPIcs 132, pp. 39:1-39:14, 2019.

[12] B. Chor, O. Goldreich, On the power of two-point based sampling. Journal of Complexity 5, pp.96-106, 1989.

[13] R. G. Downey, M. R. Fellows. Fundamentals of Parameterized Complexity. Springer, 2013.

[14] J. Håstad. The Shrinkage Exponent of de Morgan Formulas is 2. SIAM Journal on Computing 27 (1): 48-64, 1998.

[15] S. Hirahara, R. Santhanam. On the average-case complexity of MCSP and its variants. 32nd Computational Complexity Conference (CCC17), LIPIcs 79, pp. 7:1-7:20, 2017.

[16] R. Impagliazzo, V. Kabanets and A. Wigderson. In search of an easy witness: Exponential time vs. probabilistic polynomial time. Journal of Computer and Systems Sciences 65 (4): 672-694, 2002.

[17] R. Impagliazzo, R. Meka, D. Zuckerman. Pseudorandomness from Shrinkage. Journal of the ACM 66 (2): Article 11, 2019.

[18] A. Joffe. On a set of almost deterministic k-independent random variables. Annals of Probability 2 (1): 161-162, 1974.

[19] V. Kabanets, J. Cai. Circuit minimization problem. 32nd Symposium on Theory of Computing (STOC00), pp. 73-79, ACM, 2000.

[20] J. Krajíček. Forcing with random variables and proof complexity. London Mathematical Society Lecture Note Series, No. 382, Cambridge University Press, 2011.

[21] J. Krajíček, I. C. Oliveira. Unprovability of circuit upper bounds in Cook's theory PV. Logical Methods in Computer Science 13 (1:4): 1-6, 2017.

[22]

[23] D. M. McKay, C. D. Murray, R. R. Williams. Weak lower bounds on resource-bounded compression imply strong separations of complexity classes. 51st Symposium on Theory of Computing (STOC19), pp.1215-1225, ACM, 2019.

[24] M. Müller. Parameterized Randomization. PhD Thesis, Albert Ludwigs Universität Freiburg i.Br., 2008. http://www.freidok.uni-freiburg.de/volltexte/6401/

[25] M. Müller, J. Pich. Feasibly constructive proofs of succinct weak circuit lower bounds. Annals of Pure and Applied Logic 171 (2): Article 102735, 2020.

[26] J. Naor, M. Naor. Small-bias probability spaces: efficient constructions and applications. SIAM Journal on Computing 22, pp.213-223, 1993.

[27] I. C. Oliveira, R. Santhanam. Hardness magnification for natural problems. 59th Symposium on Foundations of Computer Science (FOCS18), pp. 65-76, 2018.

[28] I. C. Oliveira, J. Pich, R. Santhanam. Hardness magnification near state-of-the-art lower bounds. Theory of Computing 17 (11): 1-38, 2021. Preliminary version in 34th Computational Complexity Conference (CCC19), LIPIcs 137, pp. 27:1-27:29, 2019.

[29] C. H. Papadimitriou, S. Zachos. Two remarks on the power of counting. In 6th GI-Conference on Theoretical Computer Science, pp. 269-276, 1983.

[30] R. Santhanam, R. Williams. On Uniformity and Circuit Lower Bounds. Computational Complexity 23: 177-205, 2014.

[31] A. A. Razborov. Bounded arithmetic and lower bounds in Boolean complexity. Feasible Mathematics II, pp. 344-386, 1995.

[32] A. A. Razborov. Pseudorandom generators hard for k-DNF resolution and polynomial calculus. Annals of Mathematics 181 (2): 415-472, 2015.

[33] A. A. Razborov, S. Rudich. Natural proofs. Journal of Computer and System Sciences 55 (1): 24-35, 1997. Preliminary version in 26th Symposium on Theory of Computing (STOC94), pp. 204-213, ACM, 1994.

[34] V. Shoup. New algorithms for finding irreducible polynomials over finite fields. Mathematics of Computation 54 (189): 435-447, 1990.

[35] A. Tal. Shrinkage of De Morgan Formulae by Spectral Techniques. 55th Annual Symposium on Foundations of Computer Science (FOCS14), pp. 551-560, 2014.

[36] B. A. Trakhtenbrot. A Survey of Russian Approaches to Perebor (Brute-Force Searches) Algorithms. IEEE Annals of the History of Computing 6 (4): 384-400, 1984.

[37] C. Umans. Pseudo-random generators for all hardnesses. Journal of Computer and System Sciences 67 (2): 419-440, 2003.