Electronic Colloquium on Computational Complexity, Report No. 65 (2020)

Sharp Threshold Results for Computational Complexity[*]

Lijie Chen
MIT
lijieche@mit.edu

Ce Jin
Tsinghua University
jinc16@mails.tsinghua.edu.cn

R. Ryan Williams
MIT
rrw@mit.edu

Abstract

We establish several "sharp threshold" results for computational complexity. For certain tasks, we can prove a resource lower bound of n^c for $c \geq 1$ (or obtain an efficient circuit-analysis algorithm for n^c size), there is strong intuition that a similar result can be proved for larger functions of n, yet we can also prove that replacing "n^c" with "$n^{c+\varepsilon}$" in our results, for any $\varepsilon > 0$, would imply a breakthrough $n^{\omega(1)}$ lower bound.

We first establish such a result for **Hardness Magnification**. We prove (among other results) that for some c, the Minimum Circuit Size Problem for $(\log n)^c$-size circuits on length-n truth tables ($\mathsf{MCSP}[(\log n)^c]$) does not have $n^{2-o(1)}$-size probabilistic formulas. We also prove that an $n^{2+\varepsilon}$ lower bound for $\mathsf{MCSP}[(\log n)^c]$ (for any $\varepsilon > 0$ and $c \geq 1$) would imply major lower bound results, such as NP does not have n^k-size formulas for all k, and $\#\mathsf{SAT}$ does not have log-depth circuits. Similar results hold for time-bounded Kolmogorov complexity. Note that *cubic* size lower bounds are known for probabilistic De Morgan formulas (for other functions).

Next we show a sharp threshold for **Quantified Derandomization** (QD) of probabilistic formulas.

1. For all $\alpha, \varepsilon > 0$, there is a deterministic polynomial-time algorithm that finds satisfying assignments to every probabilistic formula of $n^{2-2\alpha-\varepsilon}$ size with at most 2^{n^α} falsifying assignments.
2. If for some $\alpha, \varepsilon > 0$, there is such an algorithm for probabilistic formulas of $n^{2-\alpha+\varepsilon}$-size and 2^{n^α} unsatisfying assignments, then a *full derandomization of* NC^1 follows: a deterministic poly-time algorithm additively approximating the acceptance probability of *any polynomial-size formula*. Consequently, NP does not have n^k-size formulas, for all k.

Finally we show a sharp threshold result for **Explicit Obstructions**, inspired by Mulmuley's notion of explicit obstructions from GCT. An *explicit obstruction against $S(n)$-size formulas* is a poly-time algorithm A such that $A(1^n)$ outputs a list $\{(x_i, f(x_i))\}_{i \in [\mathrm{poly}(n)]} \subseteq \{0,1\}^n \times \{0,1\}$, and every $S(n)$-size formula F is inconsistent with the (partially defined) function f. We prove that for all $\varepsilon > 0$, there is an explicit obstruction against $n^{2-\varepsilon}$-size formulas, and prove that there is an explicit obstruction against $n^{2+\varepsilon}$-size formulas for some $\varepsilon > 0$ if and only if there is an explicit obstruction against *all polynomial-size* formulas. This in turn is equivalent to the statement that E does not have $2^{o(n)}$-size formulas, which would be a breakthrough in circuit complexity.

[*] L.C. and R.W. are supported by NSF CCF-1741615 and a Google Faculty Research Award. Portions of this work were completed while L.C. and R.W. were visiting the Simons Institute at UC Berkeley.

ISSN 1433-8092

1 Introduction

How far are we from proving major lower bounds such as $\mathsf{P} \neq \mathsf{NP}$, polynomial-size formula lower bounds for NP, or $\mathsf{P}^{\#\mathsf{P}} \neq \mathsf{NC}^1$? Well-known complexity barriers [BGS75, RR97, AW09] have shown that, in the absence of radically new ideas, we appear quite far from resolving many fundamental lower bound problems, given the current proof methods.

Suppose we accept that prognosis, and resign ourselves to being far (for now). Can we quantify "how far" we are from various open lower bounds? Is there a *fine-grained theory* of complexity barriers?[1] Such a theory would ideally help us order lower bound problems by their "distance" from known lower bounds, and gather better intuition overall for the relative difficulty of proving lower bounds in the complexity landscape.

Our idea for addressing these questions starts with asking: *what would a lower bound we are "close to" look like?* We will use the following natural heuristic: If we can prove a resource lower bound of n^c, and we have no evidence against (possibly also evidence for) proving an $n^{c+\varepsilon}$ lower bound for arbitrarily small $\varepsilon > 0$, it seems reasonable to say we are "close to" an $n^{c+\varepsilon}$ lower bound. Viewed from this perspective, we would be "close to" a lower bound L if there is a task Π such that:

- an n^c lower bound can be proved for Π,
- we *know* how to prove $n^{c+\varepsilon}$ lower bounds for other natural tasks,
- yet an $n^{c+\varepsilon}$ lower bound for Π, for *any* $\varepsilon > 0$, would imply lower bound L.

(Again, this notion only makes sense if we believe the current lower bound against Π is not optimal.) Note, saying we are "close to" a lower bound obviously does not mean that it will be easy to prove! It just means that what is necessary does not look *quantitatively* far, in our relative estimation of lower bound difficulty.[2]

In this paper, we establish several counterintuitive "sharp threshold" phenomena for some major open problems in lower bounds and derandomization. For certain tasks Π, we are able to prove a lower bound of n^c resources and/or provide an efficient derandomization for n^c-size objects, we have good reasons to believe that a stronger result is provable, yet improving our result to $n^{c+\varepsilon}$ for some $\varepsilon > 0$ would unexpectedly imply a superpolynomially-strong lower bound. Whether or not this means we are "close" to a major breakthrough remains to be seen. Our key message is that our collective intuition about what is a "weak" lower bound (and what is a "weak" derandomization) is still surprisingly poor and incomplete.

1.1 Sharp Thresholds for MCSP Lower Bounds Against Probabilistic Formulas

A major object of study in this paper is the computational model of probabilistic formulas. This model is a natural choice because it is an interesting expressive model for which we know how to prove nearly cubic lower bounds. For simplicity, throughout this paper, "formulas" means "De Morgan formulas" by default.

Definition 1.1. A *probabilistic formula* is a distribution $\mathcal{F}$ over De Morgan formulas. We say $\mathcal{F}$ computes a function f, if for all x, $\Pr_{F \sim \mathcal{F}}[F(x) = f(x)] \geq 2/3$.

Remark 1.2. *By a simple "Chernoff bound, and union bound" argument, we may assume the distribution is a uniform distribution over $O(n)$ De Morgan formulas. Thus a probabilistic formula can always be written as an* "Approximate-Majority" *of $O(n)$ De Morgan formulas. This is helpful when discussing algorithms that take probabilistic formulas as inputs, as in that case we need a succinct way to describe a (potentially exponential-sized) distribution over De Morgan formulas.*

[1] See Section 1.4.1 for a brief comparison with the methodology of fine-grained complexity.

[2] For example, it is reasonable to believe that it will be easier to prove SAT does not have an $O(n)$-time algorithm in a random access model, than it will be to prove SAT does not have an $O(n^2)$-time algorithm. That does not mean it will be easy to prove SAT isn't in linear time! Note that this example does not quite fit our mold, in that, modulo time hierarchies, we do not really know how to prove super-linear time lower bounds.

We present a sharp hardness magnification threshold for probabilistic formulas. We first show that slightly super-quadratic probabilistic formula lower bounds for computing basic compression problems would imply breakthrough complexity separations. The Minimum Circuit Size Problem for size parameter $s(n)$ ($\mathsf{MCSP}[s(n)]$) asks whether a given length-n truth-table has an $s(n)$-size circuit on $m = \log n$ input bits. This a basic problem that has seen a flurry of research activity in recent years (e.g. [Tra84, KC00, HP15, AHK17, HW16, HS17, MW17, Hir18, HOS18, AH19]). The Minimum Kt-complexity Problem ($\mathsf{MKtP}[s(n)]$) asks whether a given length-n input string has Kt-complexity at most $s(n)$. See Section 2.2 for formal definitions.

Theorem 1.3 (From "Weak" Lower Bounds to Super-Polynomial Lower Bounds, adapting [CJW19])**.**

1. *If there is an $\varepsilon > 0$ such that for all sufficiently small $\alpha > 0$, $\mathsf{MCSP}[n^{\alpha}]$ does not have $n^{2+\varepsilon}$-size probabilistic formulas, then $\oplus\mathsf{P} \not\subset \mathsf{NC}^1$.*

2. *If there is an $\varepsilon > 0$ such that for all sufficiently small $\alpha > 0$, $\mathsf{MKtP}[n^{\alpha}]$ does not have $n^{2+\varepsilon}$-size probabilistic formulas, then $\mathsf{EXP} \not\subset \mathsf{NC}^1$.*

3. *If there is an $\varepsilon > 0$ and a family of languages $\{L_{\beta}\}$ (indexed over $\beta \in (0,1)$) such that L_{β} is a $2^{n^{\beta}}$-sparse NP language and for all β L_{β} does not have $n^{2+\varepsilon}$-size probabilistic formulas, then NP does not have n^k-size formulas, for all k.*

In fact, lower bounds which are "barely" above quadratic size would still imply a breakthrough.

Theorem 1.4. *For any unbounded function $f(n)$, the following hold.*

1. *If there is a $d > 0$ such that $\mathsf{MCSP}[(\log n)^d]$ does not have $n^2(\log n)^{f(n)}$-size probabilistic formulas, then $\oplus\mathsf{P} \not\subset \mathsf{NC}^1$.*

2. *If there is a $d > 0$ such that, $\mathsf{MKtP}[(\log n)^d]$ does not have $n^2(\log n)^{f(n)}$-size probabilistic formulas, then $\mathsf{EXP} \not\subset \mathsf{NC}^1$.*

Note it is widely believed that $\mathsf{MCSP}[s(n)]$ and $\mathsf{MKtP}[s(n)]$ are not in $\mathsf{P}_{/\mathsf{poly}}$, for any $s(n)$ which is a sufficiently large $\mathrm{poly}(\log n)$ [KC00, ABK$^+$02]. We show how to complement the hardness magnification theorem above in a sharp way, proving lower bounds for MCSP and MKtP that are quantitatively very close to the hypotheses in Theorem 1.3 and Theorem 1.4.

Theorem 1.5 (Sub-Quadratic MKtP and MCSP Lower Bounds)**.** *There are $c > 0$, $d > 0$, $K > 0$ such that for all $t(n)$ satisfying $c \log n < t(n) \leq n/20$, and $s(n)$ satisfying $(\log n)^d < s(n) \leq n/200 \log n$,*

1. *$\mathsf{MKtP}[t(n)]$ does not have probabilistic formulas of $n^{2-K/\log\log n}/t(n)$ size.*

2. *$\mathsf{MCSP}[s(n)]$ does not have probabilistic formulas of $n^{2-K/\log\log n}/s(n)$ size.*

More generally, the lower bound of Theorem 1.5 holds for "gap" versions of MCSP and MKtP (see Section 5.2 for details). The MKtP lower bound is particularly interesting because $\mathsf{MKtP}[c\log n]$ is a *polynomially-sparse* language computable in P.[3] If the lower bound of Theorem 1.5 can be improved to $n^{2+\varepsilon}$ for a subexponentially-sparse problem computable in $2^{n^{o(1)}}$ time, then $\mathsf{EXP} \not\subset \mathsf{NC}^1$.

When the circuit size parameter is close to maximal, Theorem 1.5 can be improved. We adapt recent MCSP lower bounds for larger circuit sizes [CKLM19] to achieve super-quadratic formula lower bounds.

[3] To decide whether $\mathsf{Kt}(x) \leq c\log n$, it suffices to enumerate all Turing machines of $c\log n$-bit description length and run them for n^c steps, which takes $\mathrm{poly}(n)$ time (see Definition 2.2).

Theorem 1.6 (Super-Quadratic MCSP Lower Bounds)**.** *For every $\alpha \in (0,1)$ and $\varepsilon > 0$, $\mathsf{MCSP}[n^{\alpha}]$ doesn't have probabilistic $n^{2+\alpha-\varepsilon}$-size formulas.*

Thus, MCSP gets "harder" as the size parameter increases. Theorem 1.6 should be contrasted with Item (1) of Theorem 1.3. If we could prove a similar theorem for $\mathsf{MCSP}[n^{o(1)}]$ (or any sparse-enough NP problem, by Item (3) of Theorem 1.3), then we could apply Theorem 1.3 and conclude a breakthrough complexity separation.

Comparison With Prior Work. Several recent papers [OS18, OPS19, CT19, Oli19, CMMW19, CJW19] also establish hardness magnification results for problems such as MCSP and MKtP. Most recently, the reference [CJW19] proves Theorem 1.3 for deterministic formula lower bounds of size $n^{3+\varepsilon}$; we generalize their results for probabilistic formula lower bounds of size $n^{2+\varepsilon}$. Our Theorem 1.5 generalizes $n^{2-\varepsilon}$-size formula lower bounds for MCSP [HS17, OS18, OPS19] on larger circuit sizes. In [OS18, OPS19, CJW19] it is shown that an $n^{3+\varepsilon}$-size formula lower bound for $\mathsf{MKtP}[n^{o(1)}]$ implies $\mathsf{EXP} \not\subset \mathsf{NC}^1$, and [HS17, OPS19] showed that $\mathsf{MKtP}[\mathrm{polylog}(n)]$ does not have $n^{2-\varepsilon}$-size formulas. That is, the "gap" between the magnification threshold and known lower bounds was $n^{3+\varepsilon}$ versus $n^{2-\varepsilon}$. Theorems 1.3 and 1.5 improve both of these results, and show an *arbitrarily small* gap between the magnification threshold and known lower bounds for probabilistic formulas.

Similar sharp threshold results were known before, but apparently only in cases where lower bound techniques are few. For example, [MMW19] show $\mathsf{MCSP}[n^{o(1)}] \notin \mathsf{SIZE}[n^{1+\varepsilon}]$ implies $\mathsf{NP} \not\subset \mathsf{P}_{/\mathsf{poly}}$, and it is not hard to establish an $\Omega(n)$-size lower bound for $\mathsf{MCSP}[n^{o(1)}]$. However, in the case of general circuits, researchers have been stuck for decades on proving even a $6n$-size lower bound for De Morgan circuits,[4] so an $n^{1+\varepsilon}$-size lower bound on MCSP seems *far*, relatively speaking. For another example, Allender and Koucky [AK10] (later refined by Chen and Tell [CT19]) show that if certain NC^1-complete problems do not have $n^{1+\varepsilon}$ size TC^0-size circuits, then NC^1 does not have polynomial-size TC^0 circuits. However, there is still a gap between these results and the best known lower bounds for TC^0 [IPS97, CSS16].

To compare with our case, note that any average-case lower bound for (deterministic) formulas immediately implies worst-case lower bounds for probabilistic formulas. Thus there are already multiple known methods for proving $n^{3-\varepsilon}$-size lower bounds against probabilistic formulas computing certain functions (see e.g., [KRT17, CKLM19]).

Additional Related Works on Hardness Magnification. Several other similar hardness magnification phenomena have been established: for $n^{1-\varepsilon}$ approximation to CLIQUE [Sri03], sublinear-depth circuit lower bounds for P [LW13], proof complexity [MP20], and lower bounds for non-commutative arithmetic circuits [CILM18].

In [CHO$^+$20], a barrier for proving strong circuit lower bounds via the hardness magnification techniques (termed as "The Locality Barrier") was formulated, saying that:

(a) almost all current lower bound proofs (random restrictions, polynomial approximation, etc.) satisfy a certain property, and,

(b) lower bound proofs with such a property cannot be used to prove the required lower bounds for MCSP or MKtP to achieve hardness magnification.

It is discussed in [CHO$^+$20] (HM Frontier D) that our magnification result for probabilistic formulas is also subject to this barrier.

[4] The state-of-the-art lower bound against De Morgan circuits for an explicit function is $5 \cdot n$ [LR01, IM02], while the state-of-the-art lower bound against B_2-circuits is the very recent $(3 + 1/86) \cdot n$ by [FGHK16].

1.2 Sharp Threshold for Quantified Derandomization

Next we discuss sharp thresholds for quantified derandomization of probabilistic formulas. We first provide the relevant definitions.

Definition 1.7 (Quantified Derandomization (QD) Problem, [GW14])**.** For a Boolean circuit class $\mathcal{C}$ and a function $B : \mathbb{N} \to \mathbb{N}$, the QD problem for $\mathcal{C}$ with B exceptional inputs is the following: *Given an n-input circuit $C \in \mathcal{C}$ that evaluates to 0 on at most $B(n)$ inputs, deterministically output an n-bit string on which C evaluates to 1.* [5]

Note the standard derandomization problem has $B(n) = 2^n/3$, but studying the case of $B(n) = 2^{n^\alpha}$ for $\alpha < 1$ turns out to also be a very interesting and subtle problem [GW14, Tel18, Tel19, CT19], with implications for general derandomization.

One way to design QD algorithms is by deterministically constructing hitting sets. A hitting set generator outputs a list L_n of inputs (independent of the given circuit), such that for all possible input circuits C with at most $B(n)$ exceptional inputs, at least one $x \in L_n$ makes C accept. A hitting set generator immediately implies a black-box QD algorithm, which only needs oracle access to C, rather than an explicit description of C. Conversely, from any black-box QD algorithm we obtain a hitting set generator: the hitting set consists of all input strings queried by the black-box algorithm on the all-zeroes circuit.

Generalized Probabilistic Formulas. By definition, a probabilistic formula has probability at least $2/3$ of accepting every YES instance, and at most $1/3$ of accepting every NO instance. We can also consider a generalization of probabilistic formulas which outputs **?** if the input does not satisfy either of the two conditions. This allows us to consider arbitrary distributions of formulas. Specifically, a *generalized probabilistic formula* $\mathcal{F}$ defined to be (simply) a distribution over formulas that take n inputs with a different output criterion. On input $x \in \{0,1\}^n$, define $p_x := \Pr_{F\sim\mathcal{F}}[F(x) = 1]$. We say the *value* of $\mathcal{F}$ on x is **1** if $p_x \geq 2/3$, **0** if $p_x \leq 1/3$, and **?** otherwise. We say $\mathcal{F}$ *has at most $B(n)$ exceptional inputs*, if the value of $\mathcal{F}$ is in $\{\mathbf{0}, \mathbf{?}\}$ for at most $B(n)$ inputs. We can adapt the definition of hitting set accordingly.

Definition 1.8 (Hitting Sets for Generalized Probabilistic Formulas)**.** $\mathcal{H} \subseteq \{0,1\}^n$ is a *hitting set for generalized probabilistic formulas with at most $B(n)$ exceptions* if, for every generalized $\mathcal{F}$ with at most $B(n)$ exceptional inputs, there is an $x \in \mathcal{H}$ such that the value of $\mathcal{F}$ on x is **1** or **?**.[6]

Sharp Threshold for QD of Generalized Probabilistic Formulas. We present a sharp threshold for quantified derandomization of generalized probabilistic formulas. First, we give a polynomial-time construction of hitting sets for QD of subquadratic-size generalized probabilistic formulas with subexponentially many exceptional inputs.

Theorem 1.9 (Quantified Derandomization of Generalized Probabilistic Formulas)**.** *Let $1 \leq s(n) \leq n/20$. There is a poly-time computable hitting set for generalized probabilistic formulas of size $n^{2-K/\log\log n}/s(n)^2$ with at most $B(n) = 2^{s(n)}$ exceptions, where $K > 0$ is a universal constant.*

Complementing Theorem 1.9, we prove that if QD for generalized formulas of any slightly larger size is possible, then we obtain a *full derandomization* of all polynomial-size formulas, in polynomial time!

[5] In this paper we consider the *one-sided* QD problem, as defined above. By convention, the majority output value is 1 for $B(n) < 2^n/2$. One can also study a stronger two-sided version, where the majority output value $b \in \{0,1\}$ is unknown and needs to be decided deterministically.

[6] Our definition is a little non-standard: if there is an $x \in \mathcal{H}$ such that $\mathcal{F}(x)$ outputs **?**, we also consider $\mathcal{F}$ as "hit" by $\mathcal{H}$. The reason is that in our argument, we are only guaranteed there is $x \in \mathcal{H}$ with $p_x \geq 2/3 - \delta$ for some small constant δ, as there are some losses in the acceptance probability in the proof. See Section 4.2 for details.

Theorem 1.10 (From "Weak" QD to Full Derandomization)**.** *Suppose there are $\varepsilon > 0$, $\alpha \in (0,1)$, and a poly-time algorithm for QD of generalized probabilistic formulas of size $n^{2-\alpha+\varepsilon}$ with $B(n) = 2^{n^\alpha}$ exceptional inputs. Then, there is a poly-time algorithm finding a satisfying assignment to any poly-size formula with at most $2^n/3$ falsifying assignments. As a corollary,* NP *does not have n^k-size formulas for all k.*

Moreover, if the hypothesized algorithm is black-box *(only calls the input formula as an oracle), then there is a poly-time algorithm that, for* every polynomial-size formula*, additively approximates its acceptance probability up to any desired constant error.*

Theorem 1.11. *Suppose there are $\varepsilon > 0$, $\alpha \in (0,1)$, and a $2^{n^{o(1)}}$-time algorithm for QD of generalized probabilistic formulas of size $n^{2-\alpha+\varepsilon}$, with $B(n) = 2^{n^\alpha}$ exceptional inputs. Then, there is a $2^{n^{o(1)}}$-time algorithm that finds a satisfying assignment to every polynomial-size formula having at most $2^n/3$ falsifying assignments. As a corollary,* $\mathsf{NE} = \mathsf{NTIME}[2^{O(n)}] \not\subset \mathsf{NC}^1$.

Moreover, if the above algorithm is black-box*, then there is a $2^{n^{o(1)}}$-time algorithm that, for* every polynomial-size formula*, additively approximates its acceptance probability up to any constant error.*

Furthermore, when our probabilistic formulas satisfy the typical BPP-style promise (on every input, it outputs 1 with probability $\geq 2/3$, or 0 with probability $\geq 2/3$), we can obtain a slightly improved hitting set construction than that of Theorem 1.9; see Section Section 4 for details and more results.

Comparison With Prior Work. Several papers [GW14, Tel19, Tel18, CT19] prove obtaining quantified derandomization "thresholds" for circuit classes such as $\mathsf{AC}^0, \mathsf{TC}^0$ and $\mathsf{AC}^0[\oplus]$. Most related to ours is [Tel18, CT19], showing if there is a QD algorithm for n^{1+1/ϕ^d}-wire TC^0_d circuits with $B(n) = 2^{n^{o(1)}}$ exceptional inputs for any $\phi < 1.62$, then a full derandomization of TC^0 follows. They also provide a corresponding QD algorithm for n^{1+1/c^d}-wires TC^0_d circuits for some constant $c \approx 30$ [Tel18]. However, in their case, the gap is between two constants 1.62 and 30 in the exponent; in our case, the gap in the exponent can be made arbitrarily small.

1.3 Sharp Threshold for "Explicit Proofs" of Formula Lower Bounds

We can also prove a sharp threshold for proving lower bounds in a certain "explicit" sense, along the lines of Mulmuley [Mul11] in his pioneering work on GCT. He suggests a concept of *explicit obstruction* where a polynomial-time algorithm produces a proof of a lower bound. It turns out that studying the problem of constructing "explicit proofs" against formulas and branching programs also reveals a sharp threshold phenomenon. We start by defining an *explicit obstruction* against a circuit class $\mathcal{C}$. Our definition is more generic than that of Mulmuley (it has fewer constraints).

Definition 1.12. An *explicit $\mathcal{C}$-obstruction* (or *explicit obstruction against $\mathcal{C}$*) is a (deterministic) polynomial-time algorithm A such that for all large enough n, $A(1^n)$ outputs a list $L_n = \{(x_i, y_i)\}$ such that $x_i \neq x_j$ for all $i \neq j$, and for all n-input $C \in \mathcal{C}$, there is an $(x_i, y_i) \in L_n$ such that $C(x_i) \neq y_i$.

First let us compare this obstruction notion with a similar one, that of *anti-checkers* [LY94, OPS19]. An *anti-checker for a function f against $\mathcal{C}$* is a list $\{(x_i, f(x_i))\}$ such that every circuit in $\mathcal{C}$ fails to compute f on at least one x_i in the list. Observe that every anti-checker is also an obstruction, but an anti-checker is more constrained than an explicit obstruction (the input-output pairs must be from f). Mulmuley [Mul11] argues philosophically that, in order to prove results such as $\mathsf{NP} \not\subset \mathsf{P}_{/\mathsf{poly}}$, we should strive to construct *explicit obstructions* against small circuits. His Geometric Complexity Theory program is rooted in the idea that this task may be more feasible to achieve for arithmetic circuits. It is clear that explicit obstructions against $\mathcal{C}$ imply lower bounds against $\mathcal{C}$. For example, the following is easy to prove:

Proposition 1.13. *If there is an explicit obstruction against $\mathcal{C}$, then there is a function in* P *that is not in $\mathcal{C}$.*

However, it still unclear what the extra advantage of this "explicit" approach might be. How could thinking about explicit obstructions be helpful, compared to other approaches to proving lower bounds? To understand this question, we ask: what *known* lower bound techniques can be used to achieve explicit obstructions in the Boolean world? First, we apply our technical results to construct an explicit obstruction against sub-quadratic size (De Morgan) formulas.

Theorem 1.14 (Explicit Obstructions Against Sub-Quadratic De Morgan Formulas)**.** *For a universal constant $K > 0$, there is an explicit obstruction against formulas of size $n^{2-K/\log\log n}$.*

Therefore certain lower bound techniques (based on random restrictions) can be made explicit. However, it is well-known that the best De Morgan formula lower bounds are not quadratic size; they are nearly cubic [Hås98, Tal14]. So one would expect an improvement on Theorem 1.14.

We show that if any super-quadratic formula lower bound can be used to build an explicit obstruction, we would in fact prove *exponential-size* formula lower bounds for exponential time! In particular, we prove an *equivalence* between explicit obstructions for super-quadratic formulas, exponential-size formula lower bounds, explicit obstructions for *all* polynomial-size formulas, and the existence of optimal PRGs for polynomial-size formulas.

Theorem 1.15 (Better Explicit Obstructions Are Equivalent to Super-Polynomial Obstructions)**.** *The following are equivalent:*

1. *There is an $\alpha > 0$ and an explicit obstruction against formulas of size $n^{2+\alpha}$.*
2. *For all k, there is an explicit obstruction against formulas of size n^k.*
3. *There is an $\varepsilon > 0$ and a function in $\mathsf{E} = \mathsf{TIME}[2^{O(n)}]$ that does not have $2^{\varepsilon n}$-size formulas, even infinitely often.*

As a corollary, any poly-time computable anti-checker for any f that does not have $n^{2+\varepsilon}$-size De Morgan formulas would also imply strong formula lower bounds sufficient for a full derandomization of $\mathrm{poly}(n)$-size De Morgan formulas (any anti-checker is also an obstruction). In fact, extending our explicit obstruction even *slightly* beyond quadratic size would imply $n^{\omega(1)}$ size formula lower bounds:

Theorem 1.16. *If there is an unbounded function $f(n)$ and an explicit obstruction against formulas of size $n^2 \cdot (\log n)^{f(n)}$, then $\mathsf{E} \not\subset \mathsf{NC}^1$.*

Explicit Obstructions for Other Models. One can obviously extend the notion of explicit obstruction to other well-studied models of computation. However, for two natural models just slightly more powerful than De Morgan formulas, branching programs and B_2-formulas (formulas over AND, OR, and XOR), we observe that a "sharp threshold" already kicks in for explicit obstructions against super-linear size. This is very interesting, because $n^{2-o(1)}$ lower bounds for both branching programs and B_2-formulas are well-known ([Nec66], [Weg87, p.253–255, p.422]). First, we observe:

Proposition 1.17. *There are explicit obstructions against B_2-formulas of size $o(n)$, and branching programs of size $o(n)$.*

However, we can show that any n^{ε} improvement in this simple result implies a breakthrough in formula or branching program complexity:

Theorem 1.18. *The following are equivalent:*

1. *There is an $\alpha > 0$ and an explicit obstruction against B_2-formulas of size $n^{1+\alpha}$.*

2. *For all k, there is an explicit obstruction against B_2-formulas of size n^k.*
3. *There is an $\varepsilon > 0$ such that E cannot be computed by $2^{\varepsilon n}$-size formulas, even infinitely often.*

The same equivalence also holds with "branching programs" in place of "B_2-formulas."

Therefore, making lower bounds for branching programs and/or B_2-formulas *explicit*, for any superlinear polynomial size bounds, would imply exponential size lower bounds for E. Such lower bounds, in turn, are equivalent to the existence of optimal pseudorandom generators for the corresponding model, by standard results [NW94, IW97, STV01]. Thus, explicit obstructions would also yield black-box derandomizations in this case.

1.4 Intuition

In this section we give a high-level description of our proof techniques.

The Key Technical Ingredient: An $O(\log n)$-seed Pseudorandom Restriction Generator. Our primary technical contribution is a *pseudorandom restriction generator* for subquadratic-size formulas with only $O(\log n)$ seed length. Our starting point is the classical result that the "shrinkage exponent" of De Morgan formulas is 2, first proved by Håstad [Hås98], with logarithmic factors later removed by [Tal14].

Roughly speaking, Håstad's result says that the formula complexity of any function f is expected to "shrink" by a factor of about p^2, when each variable is left unassigned with probability p, and is otherwise assigned a random bit. Such a variable restriction is called a *p-regular random restriction*. A uniform random p-regular restriction requires $\Theta(pn \log(1/p))$ random bits to encode. Our pseudorandom restriction generator provides a *derandomized* version of the [Hås98, Tal14] shrinkage theorem with an optimal $O(\log n)$ seed length, albeit with an $n^{o(1)}$-size loss on the expected size.[7] (A formal theorem statement can be found in Section 3.) In particular, it produces an explicit collection $\mathcal{C}$ of $\mathrm{poly}(n)$ partial assignments (random restrictions) $\rho : [n] \to \{0, 1, \star\}$ such that:

(a) For every formula F, applying a random partial assignment ρ from $\mathcal{C}$ "shrinks" the formula complexity of F by about p^2 (with some $n^{o(1)}$ loss), and

(b) A random ρ from $\mathcal{C}$ has decent probability of leaving at least $pn/2$ variables unset.

The collection $\mathcal{C}$ is highly explicit; we can generate any bit of its description with a $\mathrm{poly}(\log n)$ size circuit.

Our construction improves generators implicit in [OPS19] which have $O(\log^2 n)$ seed length. In more detail, our construction follows the iterated pseudorandom restriction paradigm of [IMZ12, HS17, OPS19]. In each stage, a q-regular random restriction $\boldsymbol{\varphi}_i$ is applied to the remaining variables, where q is a non-zero constant. By composing r independent stages of random restrictions, one obtains a p-regular random restriction $\boldsymbol{\rho} = \boldsymbol{\varphi}_r \circ \cdots \circ \boldsymbol{\varphi}_1$ with $p = q^r$. The key observation is that, in each stage, $\boldsymbol{\varphi}_i$ only has to be $1/q^2$-wise independent, which takes only $O(\log n)$ random bits if q is a non-zero constant. The total number of stages is $r = \log_q(p) \le O(\log n)$, and the total seed length is $O(\log^2 n)$.

To obtain an $O(\log n)$ total seed length, our improved construction uses a different parameter setting: we set r to $O(\log n / \log\log n)$, and set q to be about $1/(\log n)^\alpha$ for small enough $\alpha > 0$. We then need to sample each restriction $\boldsymbol{\varphi}_i$ using only $O(\log\log n)$ random bits, while still ensuring the shrinkage property. To achieve this, our one-stage restriction construction uses the following two ideas:

[7] Observe that $\Omega(\log n)$ random bits are required for pseudorandom restrictions for all $p \le n^{-\Omega(1)}$. To see this, assume a pseudorandom restriction $\boldsymbol{\rho}$ has $o(\log n)$ seed-length. Find a restriction ϕ in its support with maximum $|\phi^{-1}(\star)| \ge pn$. Consider a formula F of maximum size $L(F) = 2^{\Omega(pn)}$ that only depends on $|\phi^{-1}(\star)|$. Then F does not shrink under ϕ. Hence the expected size of F under $\boldsymbol{\rho}$ is at least $n^{-o(1)} \cdot L(F) \gg p^2 L(F) + p\sqrt{L(F)}$.

(1) Let $S = \mathrm{poly}(1/q)$. Fix a small set of restrictions $f_1, \dots, f_S \in \{0, 1, \star\}^{\mathrm{poly}(1/q)}$, such that a random f_i ($i \in [S]$) has a shrinkage property on $\mathrm{poly}(1/q)$-variable $1/q^2$-size formulas almost as good as q-regular random restrictions. Such $f_1, \dots, f_S$ exist, and can be found deterministically in $2^{\mathrm{poly}(1/q)} < \mathrm{poly}(n)$ time by brute force (for α small enough). After that, specifying an f_i only takes about $\log S$ bits, which is exponentially shorter than the seed-length of sampling a uniform $1/q^2$-wise independent q-regular restriction.

(2) Construct n-variable restrictions, by composing a random ($\mathrm{poly}(1/q)$-variable) restriction f_i with an almost pairwise independent hash function $h : [n] \to [\mathrm{poly}(1/q)]$, samplable using only $O(\log \mathrm{poly}(1/q) + \log \log n) \le O(\log \log n)$ bits. For a $1/q^2$-size formula F, with at least $1 - \mathrm{poly}(1/q)$ probability, the variables that F depends on are hashed into distinct variables. This allows us to extend the shrinkage property on $\mathrm{poly}(1/q)$-variable formulas to the shrinkage property on n-variable formulas.

Lower Bounds, QD Algorithms, and Explicit Obstructions From the Generator. Now we briefly discuss how our new optimal pseudorandom restriction generator implies some of our other results.

Lower Bounds for $\mathsf{MKtP}[c \log n]$. Let c be a sufficiently large constant. We first provide intuition for the proof that $\mathsf{MKtP}[c \log n]$ cannot be computed by probabilistic formulas of size $N^{1.9}$. Suppose otherwise that $\mathsf{MKtP}[c \log n]$ can be computed by a probabilistic formula $\mathcal{F}$ of size $n^{1.9}$. Applying our pseudorandom restriction generator with parameter $p = n^{-0.98}$, we show that there must be a restriction ρ with the following properties: (a) $L(F|_\rho) < 1$ with probability 0.9 for a random $F \sim \mathcal{F}$ and (b) $|\rho^{-1}(\star)| \ge pn/2 > n^{0.01}$. Define x^ρ to be the input obtained by filling every $\star$ of ρ with 0, we show that the efficiency of our generator implies that $\mathsf{MKtP}[c \log n](x^\rho) = 1$.

Item (a) implies that $\mathcal{F}$ must output the same value on every input consistent with ρ, but $\mathcal{F}$ must output 1 on every x^ρ. Therefore, $\mathcal{F}$ must output 1 on every input that is consistent with ρ. But there are at least $2^{n^{\Omega(1)}}$ many inputs consistent with ρ, and (by a counting argument) at least one of them must make $\mathsf{MKtP}[c \log n]$ output 0. This is a contradiction.

Hitting Set for Quantified Derandomization. Now we sketch how to obtain a quantified derandomization algorithm for generalized probabilistic formulas. For simplicity we focus on the case of size $n^{1.9}$, and $B(n) = 2^{n^{0.01}}$. Apply the pseudorandom restriction generator again with $p = n^{-0.98}$, and fix a restriction ρ satisfying (a) and (b). Observe that for generalized probabilistic formulas, (a) implies for every input x consistent with ρ, $\mathcal{F}(x) \neq 0$. Therefore, we can simply let the hitting set be the collection of x^ρ (as defined above) for all ρ generated by our pseudorandom restriction generator.

Explicit Obstructions. We sketch how to obtain an explicit obstruction for $n^{1.9}$-size formulas. As before, we use our pseudorandom restriction generator with $p = n^{-0.98}$. We show that for all formula F of size $n^{1.9}$, there is a restriction ρ from our generator such that $|\rho^{-1}(\star)| > 0$, and $L(F_\rho) < 1$. For each ρ and each $i \in \rho^{-1}(\star)$, we set $x^{\rho,i}$ be the input obtained by extending ρ so that the i-th variable is set to 1 and all other $\star$'s filled with 0's. We add both $(x^\rho, \mathsf{PARITY}(x^\rho))$ and $(x^{\rho,i}, \mathsf{PARITY}(x^{\rho,i}))$ to our list S.

We show this is an *anti-checker* for the PARITY function (hence it is an explicit obstruction). In particular, for every formula F, since $L(F_\rho) < 1$, F must output the same value on both x^ρ and $x^{\rho,i}$ for $i \in \rho^{-1}(\star)$, but their parities are different in S.

Better Explicit Obstructions Imply Breakthrough Lower Bounds. Finally, we sketch how an explicit obstruction for $n^{2+\varepsilon}$ formulas implies strong formula lower bounds. The idea is that, if such an obstruction exists *and* $2^{O(n)}$ time computations have $2^{o(n)}$-size formulas, then there is a formula F of only $n^{o(1)}$ size whose truth table is the output of the $\mathrm{poly}(n)$-time algorithm printing the obstruction. Using linear hashing tricks, we can use $O(n^2)$-size PARITY formulas to construct a formula of size $n^{2+o(1)}$ which agrees with the obstruction on all of its input-output pairs, a contradiction. For branching programs and B_2-formulas, PARITY can be implemented with only $O(n)$ extra size, improving the threshold for those two cases.

1.4.1 Comparison With Some "Barrier" Results in Fine-Grained Complexity.

We note a relationship between our results and some in fine-grained complexity. For some central problems in fine-grained complexity (see e.g., [BI15, AHVW16, AB17, AB18, AR18, CGL+19]), an n^c time algorithm is known, and it is known that $n^{c-\varepsilon}$ time (even $n^c/(\log n)^8$ in some cases [AB18]) would imply a complexity breakthrough, via an abnormally-faster SAT algorithm for some circuit class. This is generally viewed as an impossibility result, because the SAT algorithms may not exist.

In the settings of this paper, we strongly **believe** that $n^{c+\varepsilon}$ bounds will hold for our tasks, and it is not outlandish to think they can be also be proved. However, our results show that achieving an $n^{c+\varepsilon}$ bound suffice for proving super-polynomial bounds. Both types of results can be interpreted in a similar light. The fine-grained results say that we know an algorithm runs in n^c time, but improving (reducing) that exponent will probably require radically new techniques (if you believe it can be improved at all). In our case, we know a lower bound of n^c, and improving (increasing) that exponent may also require radically new techniques.

Organization. In Section 2 we introduce the necessary preliminaries. In Section 3, we present the main technical construction of our paper: an $O(\log n)$-seed length pseudorandom restriction generator for formulas of sub-quadratic size. In Section 4, we discuss sharp thresholds for quantified derandomization of probabilistic formulas. In Section 5, we present sharp thresholds for lower bounds against probabilistic formulas. In Section 6, we establish sharp thresholds for explicit obstructions against (deterministic) formulas and branching programs.

2 Preliminaries

2.1 Notation

We use $\widetilde{O}(f)$ as shorthand for $O(f \cdot \mathrm{polylog}(f))$ throughout the paper. All logarithms are base-2. We use n to denote the number of input bits. We say a language $L \subseteq \{0,1\}^\star$ is $f(n)$-sparse if $|L_n| \leq f(n)$, where $L_n = L \cap \{0,1\}^n$. We assume knowledge of basic complexity theory (see [AB09, Gol08]).

We use $\mathbf{U}_\ell$ to denote the uniform distribution over $\{0,1\}^\ell$. The *statistical distance* between two random distributions X_1 and X_2 is

$$|X_1 - X_2| := \frac{1}{2} \sum_x \big| \Pr[X_1 = x] - \Pr[X_2 = x] \big|.$$

We consider De Morgan formulas (i.e. formulas with AND, OR, NOT gates). The *size* of a formula F, denoted by $L(F)$, is the number of leaves in F. For a Boolean function f we also use $L(f)$ to denote the size of the smallest formula that computing f. We say a formula is *read-once* if each input variable appears at most once as a leaf in the formula.

2.2 Definitions of MCSP and MKtP

The Minimum Circuit Size Problem (MCSP) [KC00] and the Minimum Kt Complexity Problem (MKtP, [Lev84, ABK+02]) are studied in this paper. We recall their definitions.

Definition 2.1 (MCSP)**.** Let $s : \mathbb{N} \to \mathbb{N}$ satisfy $s(m) \geq m - 1$ for all m.
Problem: $\mathsf{MCSP}[s(m)]$.
Input: A function $f : \{0,1\}^m \to \{0,1\}$, presented as a truth table of $n = 2^m$ bits.
Decide: Does f have a (fan-in two) Boolean circuit C of size at most $s(m)$?

Recall that the Kt complexity (time-bounded Kolmogorov complexity) of string x is the smallest $c + \log(t)$, such that there is a Turing machine M of description length c that prints x in at most t steps.

Definition 2.2 (MKtP)**.** Let $p : \mathbb{N} \to \mathbb{N}$.
Problem: $\mathsf{MKtP}[p(n)]$.
Input: A string $x \in \{0,1\}^n$.
Decide: Does x have Kt complexity at most $p(n)$?

In the $\mathsf{Gap\text{-}MCSP}[f(n), g(n)]$ problem (respectively, $\mathsf{Gap\text{-}MKtP}[f(n), g(n)]$), we are given a string with circuit complexity (respectively, Kt complexity) which is either at least $g(n)$ or at most $f(n)$, and the goal is to distinguish between the two cases. We remark that when $p(n) \leq O(\log n)$, $\mathsf{MKtP}[p(n)] \in \mathsf{P}$.

2.3 Useful Tools

We introduce some technical tools used in this paper. The most important tool in this paper is the explicit construction of (almost) k-wise independent distributions. We recall two explicit constructions of k-wise independent spaces.

Theorem 2.3 (Explicit k-wise independent hash family, [CW79]; see also [Vad12, Corollary 3.34])**.** *For n, m, k, there is a family of k-wise independent functions $\mathcal{H} = \{h : \{0,1\}^n \to \{0,1\}^m\}$ such that every function from $\mathcal{H}$ can be described in $k \cdot \max\{n, m\}$ random bits, and evaluating a function from $\mathcal{H}$ (given its description, and given an input $x \in \{0,1\}^n$) takes time* $\mathrm{poly}(n, m, k)$.

Theorem 2.4 (Explicit k-wise ε-dependent distribution, [AGHP90])**.** *For $1 \leq k \leq n$ and $\varepsilon > 0$, there is a distribution $\mathcal{R}$ on $\{0,1\}^n$ which can be efficiently sampled using $O(k + \log\log n + \log(1/\varepsilon))$ random bits, such that for every k positions $1 \leq i_1 < i_2 < \cdots < i_k \leq n$, we have*

$$\left| \mathcal{R}\big|_{i_1, i_2, \ldots, i_k} - \mathbf{U}_k \right| \leq \varepsilon,$$

where $\mathcal{R}|_{i_1, i_2, \ldots, i_k}$ denotes the distribution of the k-bit string $x_{i_1} x_{i_2} \cdots x_{i_k}$ for $x \sim \mathcal{R}$.

Moreover, the i-th coordinate of $x \sim \mathcal{R}$ can be computed in $\mathrm{poly}(\log n, k, \log(1/\varepsilon))$ *time, given $i \in [n]$ and the seed as input.*

We also need linear-time computable error correcting codes [Spi96].

Theorem 2.5 ([Spi96])**.** *There is a linear error correcting code E (i.e., E is a linear function over $\mathbb{F}_2$) with constant rate and constant minimum relative distance, which can be computed in linear time, and by logarithmic-depth circuits of linear size.*

We will also apply *linear extractors* in our paper. Here we recall the definition.

Definition 2.6 (Min entropy and extractors)**.** The *min entropy* of a random variable X is the largest $k \in \mathbb{R}^+$ such that $\Pr[X = x] \leq 2^{-k}$ for every x in the range of X. A distribution over $\{0,1\}^n$ with min entropy at least k is called an (n, k)-source.

A function $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k, ε) *extractor*, if for every (n, k)-source X,

$$|\mathsf{Ext}(X, \mathbf{U}_d) - \mathbf{U}_m| \leq \varepsilon.$$

We say that Ext is a *linear extractor* if $\mathsf{Ext}(\cdot, u)$ is a linear function over $\mathbb{F}_2$, for every $u \in \{0,1\}^d$.

We also use explicit expander graphs.

Definition 2.7 (Expander graph family)**.** An n-vertex undirected graph G is an (n, d, λ)-*expander* if G is d-regular and $\lambda(G) \leq \lambda$, where $\lambda(G)$ denotes the second largest eigenvalue (in absolute value) of the normalized adjacency matrix of G (i.e., the adjacency matrix of G divided by d).

A well-known strongly explicit construction [GG81] suffices for our applications.

Theorem 2.8 (Strongly Explicit Expander Construction, e.g., [GG81])**.** *There exists a (λ, d)-expander family $\{G_n\}$ for some constants $d \in \mathbb{N}$ and $\lambda < 1$, such that there is an algorithm that on inputs $n, v \in [n], i \in [d]$ outputs the i-th neighbor of v in graph G_n in* $\operatorname{polylog} n$ *time.*

We also need the following expander Chernoff bound, which shows that a random walk on an expander graph behaves similarly to a sequence of i.i.d. random vertices.

Theorem 2.9 (Expander Chernoff Bound, [Gil98])**.** *Let $G = (V, E)$ be an (n, d, λ)-expander. Let $f : V \to \{0, 1\}$ be arbitrary, and let $\mu = \mathbb{E}_{v \in V} f(v)$. Let $v_1, v_2, \ldots, v_t$ be a random walk on G (where $v_1 \in V$ is uniformly chosen). For $\delta > 0$,*

$$\Pr_{v_1,\ldots,v_t}\left[\frac{1}{t}\sum_{i=1}^{t} f(v_i) < \mu - \delta\right] \leq e^{-(1-\lambda)\delta^2 t/4}.$$

We also use random restriction methods in this paper; let us recall some notation for them. Fix $n \in \mathbb{N}$. Call a function $\rho : [n] \to \{0, 1, \star\}$ an *n-variable restriction*; a distribution $\mathcal{H}$ over n-variable restrictions is called a *random restriction*. We say $\mathcal{H}$ is p-regular if $\Pr_{\rho\sim\mathcal{H}}[\rho(i) = \star] = p$ and $\Pr_{\rho\sim\mathcal{H}}[\rho(i) = 0] = \Pr_{\rho\sim\mathcal{H}}[\rho(i) = 1] = (1 - p)/2$ for every $i \in [n]$. We use $\mathcal{R}_p$ to denote the p-regular random restriction where all n coordinates are mutually independent.

Definition 2.10. We say $\mathcal{H}$ is *p-regular k-wise δ-dependent*, if for any k coordinates $i_1, \ldots, i_k$ we have $|\mathcal{D}_{i_1,\ldots,i_k} - \mathcal{R}_p| \leq \delta$, where $\mathcal{D}_{i_1,\ldots,i_k}$ is the distribution of $(\rho(i_1), \ldots, \rho(i_k))$ where $\rho \sim \mathcal{H}$. If $\delta = 0$, we say $\mathcal{H}$ is *p-regular k-wise independent*.

For a function $f : \{0, 1\}^n \to \{0, 1\}$, we use $f|_\rho$ to denote the function $\{0, 1\}^{|\rho^{-1}(\star)|} \to \{0, 1\}$ obtained by restricting f according to ρ in the natural way. The *composition* $\rho = \rho_2 \circ \rho_1$ of two restrictions $\rho_1, \rho_2 : [n] \to \{0, 1, \star\}$ is defined as

$$\rho(i) := \begin{cases} \rho_1(i), & \text{if } \rho_1(i) \neq \star \\ \rho_2(i) & \text{otherwise} \end{cases}.$$

We observe the $\circ$ operator is associative, and $(f|_{\rho_1})\big|_{\rho_2} = f|_{\rho_2\circ\rho_1}$.

3 Pseudorandom Restriction Generator with Logarithmic Seed Length

In this section we construct our main technical component: a pseudorandom restriction generator for $O(n^{2-\varepsilon})$ formulas with $O(\log n)$ seed length. We begin with the classical result that the shrinkage exponent of De Morgan formulas is 2.

Theorem 3.1 (The Shrinkage Constant of De Morgan Formulas is 2 [Hås98, Tal14])**.** *Let f be a Boolean function. For every $p > 0$,*

$$\mathop{\mathbb{E}}_{\rho\sim\mathcal{R}_p}[L(f|_\rho)] = O\Big(p^2 L(f) + p\sqrt{L(f)}\Big).$$

That is, when f is "hit" with a p-regular random restriction, we expect its formula complexity to shrink by about a p^2 factor. A uniform random p-regular restriction needs about $\Theta(pn\log(1/p)$ bits to encode. The main theorem of this section, stated below, provides a derandomized version of Theorem 3.1, with an optimal $O(\log n)$ seed length, albeit an $n^{o(1)}$-size loss on the expected shrinkage size.

Theorem 3.2. *For all sufficiently large n, and probability parameters $p \in [20/n, 1]$, there is a family of pseudorandom restrictions $\boldsymbol{\rho} : [n] \to \{0,1,\star\}$ that is samplable using $O(\log n)$ random bits, such that the following properties hold for sufficiently large n and a universal constant $C > 0$:*

(Shrinkage) *For all n-variable formulas F, $\mathbb{E}[L(F|_{\boldsymbol{\rho}})] \le \left(p^2 L(F) + p\sqrt{L(F)}\right) \cdot n^{C/\log\log n}$.*

("Close" to p-regular) $\Pr\left[|\boldsymbol{\rho}^{-1}(\star)| \ge pn/2\right] > 2/3$.

(Constructive) *There is a $\mathrm{poly}(n)$-time algorithm that outputs $\rho(1), \ldots, \rho(n)$ given the seed s. There is also a $\mathrm{polylog}\, n$-size circuit that outputs $\rho(i)$, given index $i \in [n]$ and seed s as input.*

Note that our random restriction $\boldsymbol{\rho}$ is not p-regular, but it is "close enough" on average: with $2/3$ probability, at least $p/2$ fraction of the coordinates are unrestricted. This turns out to be sufficient for us. Our proof of Theorem 3.2 will crucially exploit the following structure theorem for formulas, which decomposes a large formula into many small formulas.

Lemma 3.3 (Decomposition of Formulas [IMZ12, Tal14])**.** *Let F be a formula over variables $X = \{x_1, \ldots, x_n\}$, and $k \in \mathbb{N}$ be some parameter. There are $m \le 1 + 36 \cdot L(F)/k$ formulas over X, denoted by $G_1, \ldots, G_m$, each of size at most k, and a read-once formula F' of size m, such that $F'(G_1(x), \ldots, G_m(x)) = F(x)$ for all $x \in \{0,1\}^n$.*

3.1 One-Stage Pseudorandom Restriction

We now describe one "stage" of our pseudorandom restriction generator. As stated in the introduction, our proof follows the iterated pseudorandom restriction paradigm [IMZ12, HS17, OPS19]. We first need a small k-perfect hash family, as stated below.

Lemma 3.4 (Adapted from [AYZ95, Section 4])**.** *For $1 \le k \le n$, there is a distribution $\mathcal{H}$ over functions $h : [n] \to [k^{20}]$ which can be efficiently sampled by $O(\log k + \log\log n)$ random bits, such that for every subset $S \subseteq [n]$ of size $|S| = k$,*

$$\Pr_{h\sim\mathcal{H}}[h(i) \text{ are distinct for all } i \in S] \ge 1 - 1/k^{18}.$$

Proof. Denote $K := k^{20}$. Let $h' : [n\log K] \to \{0,1\}$ be sampled from a $(2\log K)$-wise ε-dependent distribution (Theorem 2.4) with $\varepsilon := 1/K$, using $O(\log K + \log\log(n\log K) + \log(1/\varepsilon)) = O(\log k + \log\log n)$ random bits. The function $h : [n] \to [K]$ is derived from h' by grouping $\log K$ bits together. Then h is pairwise ε-dependent. For every $S \subseteq [n]$ of size k, by the union bound, the probability that there exist two different $i, j \in S$ with $h(i) = h(j)$ is at most $\binom{k}{2} \cdot (1/K + \varepsilon) \le 1/k^{18}$. □

Now we are ready to construct one stage of the pseudorandom restriction.

Lemma 3.5 (One stage of the pseudorandom restriction)**.** *Let $k \ge 2$ and $q := k^{-1/2}$. There is a pseudorandom restriction $\boldsymbol{\varphi} : [n] \to \{0,1,\star\}^n$, samplable using $O(\log k + \log\log n)$ random bits, with the properties:*

(Shrinkage) *For all n-variable formulas F, $\mathbb{E}[L(F|_{\boldsymbol{\varphi}})] \le O\left(q^2 L(F) + q\sqrt{L(F)}\right)$.*

(Regular and Almost Pairwise Independent) *$\boldsymbol{\varphi}$ is q-regular pairwise $1/k^{18}$-dependent.*

(Constructive) *There is a $\left(2^{O(k^8)} + \mathrm{poly}(k,n)\right)$-time algorithm that outputs $\varphi(1), \ldots, \varphi(n)$ given a seed s of length $O(\log k + \log\log n)$, and a $\mathrm{poly}(k, \log n)$-size circuit that outputs $\varphi(i)$, given an index $i \in [n]$ and seed s as input.*

Proof. Let $K := k^{20}, q := k^{-1/2}$. Our proof works by first fixing an optimal pseudorandom restriction on K variables using non-uniform advice. We then show how to "amplify" that into the required pseudorandom restriction on n variables, by composing our small pseudorandom restriction with a random hash function from the k-perfect hash family of Lemma 3.4.

Construction of a "Micro" Pseudorandom Restriction on K Variables. Let $\boldsymbol{f} : [K] \to [2/q]$ be a k-wise independent random function, samplable using $O(k \log k)$ random bits (Lemma 2.3), and let $\boldsymbol{g} : [K] \to [2/q]$ be a pairwise independent random function, samplable using $O(\log k)$ random bits. Think of $\boldsymbol{f}$ and $\boldsymbol{g}$ as outputting $\log(2/q)$-bit strings. Then $\boldsymbol{g} \oplus \boldsymbol{f}$ (where $\oplus$ denotes XOR on $\log(2/q)$ bits) is also a k-wise independent random function, from which we can obtain a q-regular k-wise independent random restriction $\boldsymbol{\rho} : [K] \to \{0, 1, \star\}$. For example, given $g \sim \boldsymbol{g}$, we can set $\rho(i) := \star$ if $g(i) \in \{0, 1\}$, and set $\rho(i) := g(i) \bmod 2$ if $g(i) \in [2/q] \setminus \{0, 1\}$. Let $F|_{\boldsymbol{g}\oplus\boldsymbol{f}}$ denote the restricted formula $F|_{\boldsymbol{\rho}}$ where $\boldsymbol{\rho}$ is obtained from $\boldsymbol{g} \oplus \boldsymbol{f}$.

For any K-variable formula H of size $L(H) \leq k$, by the original shrinkage theorem (Theorem 3.1), we have

$$\begin{aligned}\mathop{\mathbb{E}}_{\boldsymbol{f}}\mathop{\mathbb{E}}_{\boldsymbol{g}}[L(H|_{\boldsymbol{g}\oplus\boldsymbol{f}})] &= O\left(q^2 L(H) + q\sqrt{L(H)}\right) \\ &\leq O\left(q\sqrt{L(H)}\right),\end{aligned}$$

where the last inequality follows from $q\sqrt{L(H)} \leq q\sqrt{k} = 1$.

From $\boldsymbol{f}$, we draw S i.i.d. random samples $(f_1, \ldots, f_S)$ for a parameter S to be determined later, and let $\mathcal{F}_S$ denote the uniform distribution over $(f_1, \ldots, f_S)$. Then for any K-variable formula H of size $1 \leq L(H) \leq k$, the Chernoff-Hoeffding bound implies:

$$\begin{aligned}&\Pr_{f_1,\ldots,f_S}\left[\mathop{\mathbb{E}}_{f\sim\mathcal{F}_S}\mathop{\mathbb{E}}_{\boldsymbol{g}}[L(H|_{\boldsymbol{g}\oplus f})] \geq \mathop{\mathbb{E}}_{\boldsymbol{f}}\mathop{\mathbb{E}}_{\boldsymbol{g}}[L(H|_{\boldsymbol{g}\oplus\boldsymbol{f}})] + q\sqrt{L(H)}\right] \\ \leq\ &\exp\left(-2S \cdot \left(\frac{q\sqrt{L(H)}}{k}\right)^2\right) \leq \exp\left(-\frac{2S}{k^3}\right).\end{aligned}$$

There are $K^{O(k)} \leq 2^{O(k \log k)}$ formulas H on K variables of size at most k. Set $S := O(k^5)$. Then a union bound implies that, with at least $2/3$ probability over the choice of $(f_1, \ldots, f_S)$, for all K-variable formulas H of size at most k,

$$\mathop{\mathbb{E}}_{f\sim\mathcal{F}_S}\mathop{\mathbb{E}}_{\boldsymbol{g}}[L(H|_{\boldsymbol{g}\oplus f})] = O\left(q\sqrt{L(H)}\right).$$

We observe that such an S-tuple $(f_1, \ldots, f_S)$ can be found deterministically in $2^{O(k^8)}$ time. We simply enumerate all $2^{O(k \log k)\cdot S} \leq 2^{O(k^7)}$ possible $(f_1, \ldots, f_S)$, and output the first tuple that satisfies the above property. To verify the property holds, we enumerate all $K^{O(k)} \leq k^{O(k)}$ relevant formulas H, and compute the formula size after shrinkage under each possible $g \oplus f$ by brute force. Recall that we define $\mathcal{F}_S$ to be the uniform distribution over the elements $f_1, \ldots, f_S$ in this S-tuple.

Composing With the Hash Function. Let $\mathcal{H}$ be a family of perfect hash functions $h : [n] \to [K]$ as defined in Lemma 3.4. For a formula G of size $L(G) \le k$ on the variable set $\{x_1, \ldots, x_n\}$, we now consider its expected size under the random restriction defined by

$$\varphi(i) := \varphi_K(h(i))$$

for every $i \in [n]$, where $h \sim \mathcal{H}$ and $\varphi_K \sim \boldsymbol{g} \oplus \mathcal{F}_S$. That is, we draw a function $g \sim \boldsymbol{g}$ and a function $f \sim \mathcal{F}_S$, and let φ_K be the restriction obtained from the function $g \oplus f$ in the way we have defined above.

From the formula G, we obtain a new formula H on variables $\{y_1, \ldots, y_K\}$ by replacing every leaf variable x_i of G with the variable $y_{h(i)}$. Since there are at most k leaves, by Lemma 3.4 says that with at least $1 - 1/k^{18}$ probability over the choice of $h \sim \mathcal{H}$, this replacement does not assign any distinct x_i's to the same variable y_j. In this case, we have

$$\mathop{\mathbb{E}}_{\varphi_K \sim \boldsymbol{g} \oplus \mathcal{F}_S}[L(G|_{\varphi_K \circ h})] = \mathop{\mathbb{E}}_{\varphi_K \sim \boldsymbol{g} \oplus \mathcal{F}_S}[L(H|_{\varphi_K})] = O\left(q\sqrt{L(G)}\right).$$

Hence,

$$\mathop{\mathbb{E}}_{\boldsymbol{\varphi}}[L(G|_{\boldsymbol{\varphi}})] = \mathop{\mathbb{E}}_{h \sim \mathcal{H}} \mathop{\mathbb{E}}_{\varphi_K \sim \boldsymbol{g} \oplus \mathcal{F}_S}[L(G|_{\varphi_K \circ h})] \le \left(1 - \frac{1}{k^{18}}\right) \cdot O\left(q\sqrt{L(G)}\right) + \frac{1}{k^{18}} \cdot L(G) = O\left(q\sqrt{L(G)}\right). \tag{1}$$

For a formula F over $\{x_1, \ldots, x_n\}$ of size $L(F) > k$, we apply the decomposition of Lemma 3.3. We obtain $m = O(L(F)/k)$ formulas $G_1, \ldots, G_m$, each of size at most k, along with a read-once formula F' of size m such that $F'(G_1(x), \ldots, G_m(x)) = F(x)$ for all $x \in \{0,1\}^n$. By linearity of expectation, we have

$$\mathop{\mathbb{E}}_{\boldsymbol{\varphi}}[L(F|_{\boldsymbol{\varphi}})] \le \sum_{i=1}^{m} \mathop{\mathbb{E}}_{\boldsymbol{\varphi}}[L(G_i|_{\boldsymbol{\varphi}})] \le m \cdot O(q\sqrt{k}) = O(q^2 L(F)). \tag{2}$$

Combining (1) and (2) finishes the proof of the shrinkage property.

Note that $\varphi_K \sim \boldsymbol{g} \oplus \mathcal{F}_S$ can be specified using $O(\log k) + \log S = O(\log k)$ bits. Hence a restriction $\varphi = \varphi_K \circ h$ can be sampled by a seed s of $O(\log k + \log\log n)$ bits.

After computing $(f_1, \ldots, f_S)$ in $2^{O(k^8)}$ time, a restriction $\varphi \in \{0, 1, \star\}^n$ can be computed in $\mathrm{poly}(n, k)$ time given the seed s. After fixing a seed, we can also implement the function that maps indices i of $\log(n)$ bits to $\phi(i) \in \{0, 1, \star\}$ using a $\mathrm{poly}(\log n, k)$-size circuit (the tuple $(f_1, \ldots, f_S)$ can be hardwired into the circuit).

Since $\boldsymbol{g}$ is pairwise independent, $\boldsymbol{\varphi}_K \sim \boldsymbol{g} \oplus \mathcal{F}_S$ is also. (To see this, note that $\boldsymbol{g} \oplus f$ is pairwise independent for *any* fixed function f.) For two different $i_1, i_2 \in [n]$, with probability at least $1 - 1/k^{18}$ we have $h(i_1) \ne h(i_2)$, in which case $\boldsymbol{\varphi}_K(h(i_1)), \boldsymbol{\varphi}_K(h(i_2))$ are q-regular independent. Hence $\boldsymbol{\varphi} \sim \boldsymbol{\varphi}_K \circ \boldsymbol{h}$ is q-regular pairwise $1/k^{18}$-dependent. □

3.2 Multi-stage Pseudorandom Restriction

Now we build our final pseudorandom restriction by composing the one-stage construction from Lemma 3.5.

Proof of Theorem 3.2. Assume $L(F) \ge 1$. Set $k := (\log n)^{1/10}, q := k^{-1/2}$, and $p := q^r$, where $r = \log(p^{-1})/\log(q^{-1}) \le O(\log n/\log\log n)$. Let $\boldsymbol{\varphi}^i$ denote the distribution of the composed restriction

$$\varphi_i \circ \cdots \circ \varphi_1,$$

where each φ_j $(1 \le j \le i)$ is independently sampled from the one-stage restriction $\boldsymbol{\varphi}$ (Lemma 3.5), using $O(\log k + \log\log n) = O(\log\log n)$ bits. Our pseudorandom restriction $\boldsymbol{\rho}$ is defined as $\boldsymbol{\varphi}^r$, samplable using

$r \cdot O(\log\log n) = O(\log n)$ random bits. Since $2^{O(k^8)} = 2^{o(\log n)}$, each $\boldsymbol{\varphi}$ can be sampled in $\mathrm{poly}(n)$ time. So $\boldsymbol{\rho}$ can also be sampled in $\mathrm{poly}(n)$ time, or strongly explicitly by a $\mathrm{polylog}(n)$-size circuit (using the constructive property of the one-stage restriction).

Since $\mathbb{E}[L(F|_{\boldsymbol{\varphi}})] \le \frac{c}{2}(q^2 L(F) + q\sqrt{L(F)})$ for a constant $c > 1$, we have

$$\mathbb{E}[L(F|_{\boldsymbol{\varphi}})] \le cqL(F), \tag{3}$$

and

$$\mathbb{E}[L(F|_{\boldsymbol{\varphi}})] \le c(q^2 L(F) + 1). \tag{4}$$

By induction on (3),

$$\mathop{\mathbb{E}}_{\boldsymbol{\varphi}^w}[L(F|_{\boldsymbol{\varphi}^w})] \le c^w q^w L(F). \tag{5}$$

And by induction on (4),

$$\begin{aligned} \mathop{\mathbb{E}}_{\boldsymbol{\varphi}^u}[L(F|_{\boldsymbol{\varphi}^u})] &\le c^u q^{2u} L(F) + c(1 + cq^2 + c^2q^4 + \cdots + c^{u-1}q^{2u-2}) \\ &\le c^u q^{2u} L(F) + 2c, \end{aligned} \tag{6}$$

for small enough q. Combining (5) and (6) with $w := r - u$, we have

$$\begin{aligned} \mathbb{E}[L(F|_{\boldsymbol{\rho}})] &= \mathop{\mathbb{E}}_{\boldsymbol{\varphi}^{r-u}} \mathop{\mathbb{E}}_{\boldsymbol{\varphi}^u}[L(F|_{\boldsymbol{\varphi}^{r-u} \circ \boldsymbol{\varphi}^u})] \\ &\le c^{r-u} q^{r-u} \cdot (c^u q^{2u} L(F) + 2c) \\ &\le c^r \cdot q^r \cdot O(q^u L(F) + q^{-u}). \end{aligned}$$

Since $\min_{0 \le u \le r}\{q^u L(F) + q^{-u}\} \le O(q^r L(F) + q^{-1}\sqrt{L(F)})$, we have

$$\mathbb{E}[L(F|_{\boldsymbol{\rho}})] \le O\left(c^r q^{2r} L(F) + c^r q^{r-1}\sqrt{L(F)}\right) \le (p^2 L(F) + p\sqrt{L(F)}) \cdot n^{O(1/\log\log n)}. \tag{7}$$

Note that $\rho(i) = \star$ iff $\varphi_1(i) = \varphi_2(i) = \cdots = \varphi_r(i) = \star$. Since $\varphi_1, \ldots, \varphi_r$ are independent, and each of them is q-regular pairwise $1/k^{18}$-dependent, for every $1 \le i < j \le n$, we have

$$\Pr[\boldsymbol{\rho}(i) = \boldsymbol{\rho}(j) = \star] \le (q^2 + 1/k^{18})^r \le q^{2r}\exp(rq^{-2}/k^{18}) \le p^2(1 + o(1)),$$

and

$$\begin{aligned} \Pr[\boldsymbol{\rho}(i) = \star] &\ge (q - 1/k^{18})^r \ge p(1 - o(1)), \\ \Pr[\boldsymbol{\rho}(i) = \star] &\le (q + 1/k^{18})^r \le p(1 + o(1)). \end{aligned}$$

Hence,

$$np(1 - o(1)) \le \mathbb{E}[|\boldsymbol{\rho}^{-1}(\star)|] \le np(1 + o(1)),$$

and

$$\mathrm{Var}[|\boldsymbol{\rho}^{-1}(\star)|] = \mathbb{E}[|\boldsymbol{\rho}^{-1}(\star)|^2] - \mathbb{E}[|\boldsymbol{\rho}^{-1}(\star)|]^2 \le np(1 - p + o(1)) + n^2p^2o(1).$$

By Chebyshev's inequality,

$$\Pr[|\boldsymbol{\rho}^{-1}(\star)| \ge np/2] \ge 1 - \frac{\mathrm{Var}[|\boldsymbol{\rho}^{-1}(\star)|]}{(np/2 - o(1))^2} > 2/3.$$

This completes the proof. □

4 Sharp Thresholds for Quantified Derandomization

In this section we present sharp threshold results for quantified derandomization.

4.1 Threshold Theorems for Quantified Derandomization

To prove our threshold theorems for quantified derandomization, we need the following construction of linear extractors with a short seed.

Theorem 4.1 (Adaptation of [Li16, Theorem 3.14])**.** *For all constant $\beta \in (0,1)$ and sufficiently large n, there is an explicit (n^{β}, ε) linear extractor $\{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $\varepsilon = n^{-\beta}$, $d \leq (1+O(\beta)) \log n$, and $m = n^{\beta/2}$.*

The theorem above is already implicit in [Li16]. We provide a proof in Appendix A for completeness.

We start with a lemma along the lines of Goldreich and Wigderson [GW14], showing that, given a poly-time algorithm for QD that finds any input on which the given probabilistic formula has non-zero probability of outputting 1, we can obtain a full derandomization.

Lemma 4.2. *Suppose there is a constant $\alpha > 0$, and a deterministic poly-time algorithm that, given any generalized probabilistic formula $\mathcal{F}$ of size $n^{2+\alpha}$ with $B(n) = 2^{n^{\alpha}}$ exceptional inputs, finds an input $x \in \{0,1\}^n$ on which $\mathcal{F}$ has non-zero probability of outputting 1. Then, there is a poly-time algorithm that finds an accepting assignment to any polynomial-size formula having at most $2^n/3$ falsifying assignments.*

Proof. Our proof follows [GW14]; the idea is to apply a linear extractor appropriately. Let F be a (deterministic) polynomial size formula on m variables, which has at least $B(m) = 0.9 \cdot 2^m$ YES-inputs (we amplified the acceptance probability from $2/3$ to 0.9 by a standard argument). For simplicity, we assume the size of F is m, since otherwise we could add "dummy variables"[8] so that m equals the size of F, without affecting the fraction of YES-inputs. Now we will deterministically find a YES-input to F, by reducing to a quantified derandomization instance.

Let β be a positive constant to be specified later. Let $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be the (n^{β}, ε) linear extractor from Theorem 4.1, where $m = n^{\beta/2}$, $\varepsilon \leq 0.1$, and $d \leq (1 + O(\beta)) \log n$. We choose β small enough that $d \leq (1+\alpha) \log n$, and $\beta \leq \alpha$.

Define the probablistic formula $\mathcal{G} := F(E(x, \mathbf{U}_d))$ on input $x \in \{0,1\}^n$. Each output bit of $E(x,s)$ is the XOR of a subset of input bits, which can be computed by a n^2-size formula. So for each $s \in \{0,1\}^d$, the size of $F(E(x,s))$ is at most $n^2 \cdot m \leq n^{2+\alpha}$.

We claim that there are less than $2^{n^{\beta}} \leq 2^{n^{\alpha}}$ strings $x \in \{0,1\}^n$ such that $\mathbb{E}[F(E(x,\mathbf{U}_d))] \leq 2/3$. Assume not; then the uniform distribution $\mathcal{X}$ over all x such that $\mathbb{E}[F(E(x,\mathbf{U}_d))] > 2/3$ has min-entropy at least n^{β}. Hence $\mathcal{X}$ is an (n, n^{β}) source, which implies that the statistical distance between $E(\mathcal{X}, \mathbf{U}_d)$ and $\mathbf{U}_m$ is at most 0.1, since E is an $(n^{\beta}, 0.1)$-extractor. Therefore, $\mathbb{E}[F(E(\mathcal{X},\mathbf{U}_d))] \geq \mathbb{E}[F(\mathbf{U}_m)] - 0.1 \geq 0.9 - 0.1 = 0.8$, contradicting $\mathbb{E}[F(E(x,\mathbf{U}_d))] \leq 2/3$ for all x in the support of $\mathcal{X}$.

Thus there are at most $2^{n^{\alpha}}$ exceptional inputs on which the generalized formula $\mathcal{G}$ evaluates to **0** or **?**. By the hypothesis, on such a $\mathcal{G}$ we can deterministically find an input x where $\mathcal{G}$ has positive probability of outputting **1**. Then, by enumerating all $s \in \{0,1\}^d$, we can find a YES-input $E(x,s)$ to F. □

Reminder of Theorem 1.10. *Suppose there are $\varepsilon > 0$, $\alpha \in (0,1)$, and a poly-time algorithm for QD of generalized probabilistic formulas of size $n^{2-\alpha+\varepsilon}$ with $B(n) = 2^{n^{\alpha}}$ exceptional inputs. Then, there is*

[8] That is, if there are $n < m$ variables in the m-size formula, we add $m - n$ extra variables which are not accessed by the formula.

a poly-time algorithm finding a satisfying assignment to any poly-size formula with at most $2^n/3$ falsifying assignments. As a corollary, NP *does not have n^k-size formulas for all k.*

Moreover, if the above algorithm is black-box*, then there is a poly-time algorithm that, for* every polynomial-size formula*, additively approximates its acceptance probability up to any desired constant additive error.*

Proof. We show that, assuming the hypothesis of this theorem, we can achieve the hypothesis of Lemma 4.2, which in turn implies a (one-sided) derandomization of polynomial-size formulas with $2^n/3$ falsifying assignments. By [MW18, Theorem 1.1], this implies that NP does not have n^k-size formulas for all k.

Recall the requirement of Lemma 4.2: we need to show that there is a polynomial-time algorithm such that, on every generalized formula of size $n^{2+\alpha}$ with at most 2^{n^α} exceptions, the algorithm finds an input $x \in \{0,1\}^n$ on which $\mathcal{F}$ has non-zero probability of outputting $\mathbf{1}$.

Let $\beta > 0$ be a constant to determined later, and let $\mathcal{F}(x) = \{(p_i, F_i(x))\}$ be a generalized probabilistic formula on m variables. In particular, each $F_i(x)$ is evaluated on input x with probability p_i. We assume each F_i has size $m^{2+\beta}$, and that there are at most 2^{m^β} exceptional inputs on which $\mathcal{F}$ evaluates to $\mathbf{0}$ or $\mathbf{?}$.

Let $n := m^{1/t}$ for a constant $t \in (0,1)$ to be determined. For simplicity, we assume m divides n. We construct a new probabilistic formula $\mathcal{G}(y)$ over n Boolean variables y as follows: denote $y = (x_1, x_2, \ldots, x_{n/m})$ with $|x_j| = m$ for every $j \in [n/m]$, and define

$$\mathcal{G}(y) := \{(p_i, G_i(y))\}, \text{ where } G_i(y) := \bigvee_{j\in[n/m]} F_i(x_j).$$

Observe that, for every $j \in [n/m]$, $\mathbb{E}_{G_i \sim \mathcal{G}}\, G_i(y) \geq \mathbb{E}_{F_i \sim \mathcal{F}}\, F_i(x_j)$; that is, $\mathcal{G}$ is only more likely to output $\mathbf{1}$. Thus there are at most $\left(2^{m^\beta}\right)^{n/m} = 2^{n^{1-t+\beta t}}$ inputs on which $\mathcal{G}$ evaluates to $\mathbf{0}$ or $\mathbf{?}$.

Now, $\mathcal{G}(y)$ is a probabilistic formula of size at most $n/m \cdot m^{2+\beta} = n^{1+t+\beta t}$. By assumption, we can find a satisfying assignment y to $\mathcal{G}$ in polynomial time, as long as $1+t+\beta t \leq 2-\alpha+\varepsilon$ and $\alpha \geq 1-t+\beta t$. Setting $\beta := \alpha\varepsilon/4$ and $t := (1-\alpha)/(1-\beta)$, both inequalities are satisfied. Hence by assumption, we can deterministically find a $y \in \{0,1\}^n$ on which $\mathcal{G}$ has value $\mathbf{1}$ or $\mathbf{?}$; that is, the probability $\mathcal{G}(y)$ outputs $\mathbf{1}$ is at least $1/3$. Recalling $y = (x_1, \ldots, x_{n/m})$ for $x_i \in \{0,1\}^m$, there is at least one x_i on which $\mathcal{F}$ outputs $\mathbf{1}$ with non-zero probability. The proof then follows from Lemma 4.2.

Moreover, if the QD algorithm is black-box, then the above algorithm immediately implies a hitting set generator for all polynomial-size formulas with at most $2^n/3$ falsifying assignments. By [GW14, Theorem 2.1] (which relies on [GVW11]), this implies a *two-sided* derandomization algorithm for polynomial-size formulas, which allows us to estimate the acceptance probability of any polynomial-size formula up to any constant additive error. □

Remark 4.3. *In fact, in the hypothesis of Theorem 1.10, we can assume that the generalized probabilistic formula has support size at most $n^{1-\alpha+\varepsilon}$. To see this, we note that we can assume $\mathcal{F}$ has support size $m^{1+\beta}$ (which follows from the small seed length of the extractor used in proving Lemma 4.2). Then we observe that the support size of $\mathcal{G}$ is also $m^{1+\beta} = n^{t+\beta t} \leq n^{1-\alpha+\varepsilon}$.*

Reminder of Theorem 1.11. *Suppose there are $\varepsilon > 0$, $\alpha \in (0,1)$, and a $2^{n^{o(1)}}$-time algorithm for QD of generalized probabilistic formulas of size $n^{2-\alpha+\varepsilon}$, with $B(n) = 2^{n^\alpha}$ exceptional inputs. Then, there is a $2^{n^{o(1)}}$-time algorithm that finds a satisfying assignment to every polynomial-size formula having at most $2^n/3$ falsifying assignments. As a corollary,* NE $\not\subset$ NC^1.

Moreover, if the above algorithm is black-box*, then there is a $2^{n^{o(1)}}$-time algorithm that, for* every polynomial-size formula*, additively approximates its acceptance probability up to any constant error.*

Proof Sketch. The proof follows precisely the same strategy as Theorem 1.10. The $\mathsf{NE} \not\subset \mathsf{NC}^1$ consequence follows from [Wil13, Wil14]. □

4.2 Quantified Derandomization of Probabilistic Formulas

Complementing the previous theorem, we can prove the following quantified derandomization for generalized probabilistic formulas.

Reminder of Theorem 1.9. *Let $1 \le s(n) \le n/20$. There is a polynomial time computable hitting set for generalized probabilistic formulas of size $n^{2-K/\log\log n}/s(n)^2$ with $B(n) = 2^{s(n)}$ exceptional inputs, where $K > 0$ is a universal constant.*

Proof. Let $\mathcal{F}$ be a generalized probabilistic formula with $B(n) = 2^{s(n)}$ exceptional inputs, where each formula F in its support has size $L(F) \le n^{2-K/\log\log n}/s(n)^2$. Let $\boldsymbol{\rho}$ be the pseudorandom restriction guaranteed by Theorem 3.2 with parameter $p := 20s(n)/n$. Let $K := 4C$, where C is the universal constant from Theorem 3.2. Then by the shrinkage property of the pseudorandom restriction, for every formula F with $L(F) \le n^{2-K/\log\log n}/s(n)^2$, we have

$$\mathop{\mathbb{E}}_{\rho}[L(F|_\rho)] \le (p\sqrt{L(F)} + p^2 L(F)) \cdot n^{C/\log\log n} \le O(n^{-C/\log\log n}) \le 1/18$$

for large enough n. Therefore

$$\mathop{\mathbb{E}}_{\rho}\mathop{\mathbb{E}}_{F\sim\mathcal{F}}[L(F|_\rho)] = \mathop{\mathbb{E}}_{F\sim\mathcal{F}}\mathop{\mathbb{E}}_{\rho}[L(F|_\rho)] \le 1/18.$$

By Markov's inequality, with at least $2/3$ probability over the choice of ρ, we have $\mathbb{E}_{F\sim\mathcal{F}}[L(F|_\rho)] \le 1/6$, which (again by Markov's inequality) implies $\Pr_{F\sim\mathcal{F}}[L(F|_\rho) < 1] \ge 5/6$. Note that when $L(F|_\rho) < 1$, $F|_\rho$ is simply a constant function.

By the "close to p-regular" property of the pseudorandom restriction, $\Pr_\rho\left[|\rho^{-1}(\star)| \ge np/2\right] \ge 2/3$. Hence there is a ρ from our pseudorandom restriction generator such that $|\rho^{-1}(\star)| \ge np/2 > 10s(n)$, and $\Pr_{F\sim\mathcal{F}}[F|_\rho \text{ is a constant}] \ge 5/6$. Let $p_0 := \Pr_{F\sim\mathcal{F}}[F|_\rho \text{ is the constant } 1]$. Then for every $x \in \{0,1\}^n$ consistent with ρ,

$$\begin{aligned}\mathop{\mathbb{E}}_{F\sim\mathcal{F}}[F(x)] &= \mathop{\mathbb{E}}_{F\sim\mathcal{F}}\left[F(x)\middle|F|_\rho \text{ is not constant}\right] \cdot \Pr_{F\sim\mathcal{F}}[F|_\rho \text{ is not constant}] + 1\cdot p_0 \\ &\in [p_0, p_0 + 1/6].\end{aligned}$$

The number of such x is at least $2^{10s(n)}$, while the number of x that make $\mathcal{F}$ output **?** and **0** is at most $2^{s(n)}$. Therefore $p_0 + 1/6 \ge 2/3$, and $\mathcal{F}$ on such an x has value either **?** or **1**. □

Finally, we observe that if the given probabilistic formula satisfies the standard BPP-style promise (for each input, it either outputs 1 with probability $\ge 2/3$, or outputs 0 with probability $\ge 2/3$), we can obtain a slightly better hitting set construction than that of Theorem 1.9.

Theorem 4.4. *For $1 \le b(n) \le n/20$, there is a polynomial time computable hitting set for probabilistic formulas of size $n^{2-K/\log\log n}/b(n)$ with $B(n) = 2^{b(n)}$ exceptional inputs, where $K > 0$ is a universal constant.*

Proof. Let $\mathcal{F}$ be a probabilistic formula with $B(n) = 2^{b(n)}$ exceptional inputs, where each formula F in the support of $\mathcal{F}$ satisfies $L(F) \le n^{2-K/\log\log n}/b(n)$. We set $K := 4C$, where $C > 0$ is the universal constant in Theorem 3.2. Let $\boldsymbol{\rho}$ be the pseudorandom restriction from Theorem 3.2 with parameter $p := 20b(n)/n$.

By the shrinkage property of Theorem 3.2, for each F with $L(F) \leq n^{2-K/\log\log n}/b(n)$,

$$\begin{aligned}\mathop{\mathbb{E}}_{\rho}[L(F|_\rho)] &\leq \left(p^2 L(F) + p\sqrt{L(F)}\right) \cdot n^{C/\log\log n} \\ &\leq O\left(b(n) \cdot n^{-C/\log\log n}\right).\end{aligned}$$

By Markov, $\Pr_\rho[L(F|_\rho) \geq b(n)] < 1/60$ for large enough n. Hence, by Markov's inequality,

$$\begin{aligned}\Pr_{\rho}\left[\Pr_{F\sim\mathcal{F}}\left[L(F|_\rho) \geq b(n)\right] \geq 1/20\right] &\leq 20 \cdot \mathop{\mathbb{E}}_{\rho}\left[\Pr_{F\sim\mathcal{F}}\left[L(F|_\rho) \geq b(n)\right]\right] \\ &= 20 \cdot \mathop{\mathbb{E}}_{F\sim\mathcal{F}}\left[\Pr_{\rho}\left[L(F|_\rho) \geq b(n)\right]\right] < 1/3. \qquad (8)\end{aligned}$$

By the "close to p-regular" property of Theorem 3.2,

$$\Pr_{\rho}[|\rho^{-1}(\star)| \geq np/2] > 2/3. \qquad (9)$$

Comparing (9) with (8), we know that there exists a restriction ρ such that

$$|\rho^{-1}(\star)| \geq np/2 = 10b(n)$$

and

$$\Pr_{F\sim\mathcal{F}}\left[L(F|_\rho) \geq b(n)\right] < 1/20. \qquad (10)$$

Fix such a ρ, and let $X \subseteq \{x_1, \dots, x_n\}$ denote the set of input variables x_j such that

$$\Pr_{F\sim\mathcal{F}}[F|_\rho \text{ contains } x_j] \geq 1/4. \qquad (11)$$

For all $x_j \in X$, by (10), the conditional probability

$$\Pr_{F\sim\mathcal{F}}\left[F|_\rho \text{ contains } x_j \,\middle|\, L(F|_\rho) < b(n)\right] \geq 1/4 - 1/20 = 1/5.$$

Therefore

$$\begin{aligned}b(n) &\geq \mathop{\mathbb{E}}_{F\sim\mathcal{F}}\left[L(F|_\rho) \,\middle|\, L(F|_\rho) < b(n)\right] \\ &\geq \sum_{x_j\in X} \Pr_{F\sim\mathcal{F}}\left[F|_\rho \text{ contains } x_j \,\middle|\, L(F|_\rho) < b(n)\right] \\ &\geq (1/5) \cdot |X|,\end{aligned}$$

implying

$$|X| \leq 5b(n).$$

We claim that, under restriction ρ, the output value of $\mathcal{F}$ does not depend on the coordinate set $\rho^{-1}(\star)\backslash X$. Otherwise, there would be two strings $x_1, x_2 \in \{0,1\}^n$ consistent with ρ that only differ at some bit position $j \in \rho^{-1}(\star)\backslash X$, such that $\mathcal{F}(x_1) \neq \mathcal{F}(x_2)$. However, by (11), $\left|\Pr_{F\sim\mathcal{F}}[F(x_1) = 1] - \Pr_{F\sim\mathcal{F}}[F(x_2) = 1]\right| \leq 1/4$, which is smaller than the $1/3$ gap of the probabilistic formula. Since $|\rho^{-1}(\star)\backslash X| \geq 10b(n) - 5b(n) > b(n)$, we have $2^{|\rho^{-1}(\star)\backslash X|} > B(n)$, so $\mathcal{F}$ must evaluate to the majority output value on all inputs that are consistent with ρ. Therefore the following set of strings is a hitting set: the set of all restrictions ρ with all $\star$ positions filled in with zeroes. □

Remark 4.5. *Our construction actually implies a* two-sided *black-box non-adaptive quantified derandomization algorithm. We can enumerate all restrictions ρ satisfying $|\rho^{-1}(\star)| \geq np/2$. Comparing (9) and (8), we know the majority of these restrictions satisfy condition (10). Hence it follows from the discussion above that the majority value of $\mathcal{F}(\rho)$ equals the majority value of $\mathcal{F}$.*

5 Sharp Thresholds for Hardness Magnification

In this section we establish sharp threshold results for hardness magnification. We first prove our magnification theorems, and then establish corresponding lower bounds, which are just "an epsilon away" from the magnification results.

5.1 Magnification Theorems

We first establish the magnification theorems, which are an adaptation of the main results of [CJW19]. We need the following theorem from [CJW19].

Theorem 5.1 ([CJW19, Theorem 3.6, Lemma 5.1, Lemma 5.2], adapted)**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a function of sparsity S_{sparse}, where $\log n \le \log(S_{\mathsf{sparse}}) \le n^{1-\Omega(1)}$.*

There is a function $H : \{0,1\}^{\Theta(\log S_{\mathsf{sparse}})} \to \{0,1\}$, computable in nondeterministic $O(n)$ time with one oracle query to f, and a randomized $O(n)$-time algorithm O_{rand} computing f with error probability 0.01 using $O(\log n)$ random bits, $O(\log S_{\mathsf{sparse}})$ bits of advice, and only $O(1)$ non-adaptive oracle queries to H. In particular, for each fixed choice of randomness, each bit of the queries of O_{rand} to H is a parity of a subset of the input x, and the algorithm O_{rand} simply returns the majority of the answers it receives from the oracles.

Moreover, if $f = \mathsf{MKtP}[S_{\mathsf{sparse}}]$, then we can assume $H \in \mathsf{E}$, and we can assume $H \in (\Sigma_2 P)^{\oplus \mathsf{P}}$ if $f = \mathsf{MCSP}[S_{\mathsf{sparse}}]$.

Now we are ready to prove Theorem 1.3.

Reminder of Theorem 1.3.

1. *If there is an $\varepsilon > 0$ such that for all small enough $\alpha > 0$, $\mathsf{MCSP}[n^\alpha]$ does not have $n^{2+\varepsilon}$-size probabilistic formulas, then $\oplus\mathsf{P} \not\subset \mathsf{NC}^1$.*

2. *If there is an $\varepsilon > 0$ such that for all small enough $\alpha > 0$, $\mathsf{MKtP}[n^\alpha]$ does not have $n^{2+\varepsilon}$-size probabilistic formulas, then $\mathsf{EXP} \not\subset \mathsf{NC}^1$.*

3. *If there is an $\varepsilon > 0$ and a family of languages $\{L_\beta\}$ (indexed over $\beta \in (0,1)$) such that L_β is a 2^{n^β}-sparse NP language and for all β, L_β does not have $n^{2+\varepsilon}$-size probabilistic formulas, then NP does not have n^k-size formulas, for all k.*

Proof Sketch. The proof is a minor adaptation of the main results in [CJW19]; we mainly have to observe that their proofs extend to superquadratic-size probabilistic formulas.

We first describe the proof of item (3). By Theorem 5.1, for a constant $\beta >$ and every 2^{n^β}-sparse NP language $L_\beta : \{0,1\}^n \to \{0,1\}$, there is an NP function $H : \{0,1\}^\star \to \{0,1\}$, and a (nonuniform) randomized algorithm O^H_{rand} with error probability 0.01 that can decide L_β using $O(1)$ non-adaptive oracle queries to H of length only $O(n^\beta)$.

Moreover, after we fix the randomness, each bit of each query to H has the form $\bigoplus_{i \in S} x_i$ for some subset $S \subseteq [n]$ of the input coordinates; i.e., each bit is a MOD-2 sum of some input bits. Since PARITY can be computed by De Morgan formulas of size $O(n^2)$, each bit of each query can be computed in $O(n^2)$ size.

Now, assume every problem in NP has n^k-size formulas, for some k. Then, the H-oracle on $O(n^\beta)$-bit input can be computed by some $O(n^{\beta k})$-size formula. Therefore the randomized algorithm O^H_{rand} deciding L_β can be implemented with a probabilistic formula of size $O(n^2) \cdot O(n^{\beta k}) \le n^{2+\varepsilon}$, by setting $\beta \le \varepsilon/2k$. This contradicts our assumption.

For Item (1) and (2), we can still apply Theorem 5.1, and use the "moreover" part. Note that, assuming $\oplus\mathsf{P} \subset \mathsf{NC}^1$ (respectively, $\mathsf{EXP} \subset \mathsf{NC}^1$), it follows that the H-oracle on $\widetilde{O}(n^{\alpha})$-bit input can be computed by $n^{O(\alpha)}$-size formulas, because the complexity class $(\Sigma_2\mathsf{P})^{\oplus\mathsf{P}}$ collapses to NC^1, under the assumption that $\oplus\mathsf{P} \subset \mathsf{NC}^1$. □

We note that even slightly super-quadratic lower bounds would suffice.

Reminder of Theorem 1.4. *For any unbounded function $f(n)$, the following hold.*

1. *If there is a $d > 0$ such that $\mathsf{MCSP}[(\log n)^d]$ does not have $n^2(\log n)^{f(n)}$-size probabilistic formulas, then $\oplus\mathsf{P} \not\subset \mathsf{NC}^1$.*
2. *If there is a $d > 0$ such that, $\mathsf{MKtP}[(\log n)^d]$ does not have $n^2(\log n)^{f(n)}$-size probabilistic formulas, then $\mathsf{EXP} \not\subset \mathsf{NC}^1$.*

Proof Sketch. The proof is the same as the first two items of Theorem 1.3. Assuming the hypothesis $\oplus\mathsf{P} \subset \mathsf{NC}^1$ ($\mathsf{EXP} \subset \mathsf{NC}^1$, resp.), the H-oracle on $\widetilde{O}((\log n)^d)$-bit input can be computed by formulas of size $(\log n)^{O(d)} \ll (\log n)^{f(n)}$. □

5.2 Nearly Matching Lower Bounds

Now we are ready to prove the nearly matching lower bounds. We first state the following connection between hitting sets for QD and sparse lower bounds. The promise problems Gap-MKtP and Gap-MCSP are defined in Section 2.2.

Theorem 5.2 (Hitting Sets Imply Sparse Lower Bounds). *Let $\mathcal{C}$ be a circuit class that is closed under negation. Let $S \subseteq \{0,1\}^n$ be a hitting set against $\mathcal{C}$-circuit with $B(n)$ exceptional inputs.*

- *If every string $s \in S$ has Kt complexity at most $A(n)$, then $\mathsf{Gap\text{-}MKtP}[A(n), \log B(n)] \notin \mathcal{C}$.*
- *If every string $s \in S$ has circuit size at most $A(n)$, then $\mathsf{Gap\text{-}MCSP}[A(n), \frac{\log B(n)}{10 \log\log B(n)}] \notin \mathcal{C}$.*

Proof. To prove the first item, suppose circuit $C \in \mathcal{C}$ computes $\mathsf{Gap\text{-}MKtP}[A(n), \log B(n)]$. Since there are at most $B(n)$ strings with Kt complexity $\leq \log B(n)$, C only accepts no more than $B(n)$ inputs. By the definition of S, there exists $s \in S$ such that C rejects s. However, every $s \in S$ has Kt complexity at most $A(n)$ and should be accepted by C, a contradiction.

The second item can be proved similarly. □

Combined with Theorem 4.4, we immediately prove Theorem 1.5 as a corollary.

Proof of Theorem 1.5. Every string in the hitting set from Theorem 3.2 is generated by a $\mathrm{poly}(n)$ time algorithm on a $O(\log n)$-bit seed, so they all have Kt complexity $O(\log n)$. And, these strings all have circuit size $\mathrm{polylog}\, n$. The proof then follows from Theorem 5.2. □

5.3 Trade-off Between Sparsity and Formula Lower Bounds

There is a gap between the known formula lower bounds for sparse and non-sparse NP languages (near n^2 versus near n^3). We can prove a trade-off between lower bounds and sparsity, in particular for the MCSP problem on different circuit size parameters.

Theorem 5.3 (Local PRG against De Morgan Formulas, [CKLM19]). *For any size parameter $s \geq n^{\Omega(1)}$, there is a PRG $G : \{0,1\}^r \to \{0,1\}^n$ with seed length $r \leq s^{1/3+o(1)}$, such that*

- *For every De Morgan formula of size at most s,*

$$\left| \mathop{\mathbb{E}}_{z \sim \{0,1\}^r}[F(G(z))] - \mathop{\mathbb{E}}_{x \sim \{0,1\}^n}[F(x)] \right| \leq 0.1.$$

- *For every seed $z \in \{0,1\}^r$, the function $g_z : \{0,1\}^{\log n} \to \{0,1\}$ defined as $g_z(j) = G(z)_j$ can be computed by a circuit of size at most $s^{1/3+o(1)}$.*

Remark 5.4. *The original statement of [CKLM19] assumed $s \geq n$, which came from [IMZ12, Lemma 4.8]. In fact, this assumption can be weakened to $s \geq n^{\delta}$ for some constant δ, without affecting the original proof.*

Theorem 5.5. *Let $0 < \alpha < 1$. $\mathsf{MCSP}[n^{\alpha}]$ on input length $n = 2^m$ does not have $n^{2+\alpha-\varepsilon}$-size probabilistic formulas, for any $\varepsilon > 0$.*

Proof. Let $\mathcal{F}$ be a probabilistic formula with error $1/3$ (see Definition 1.1), where each formula F in its support satisfies $L(F) \leq n^{2+\alpha-\varepsilon}$.

Let $\boldsymbol{\rho}$ be the pseudorandom restriction from Theorem 3.2 with parameter $p := 2n^{\alpha+\varepsilon/5-1}$. Then, by the shrinkage property of Theorem 3.2, for each formula F,

$$\mathop{\mathbb{E}}_{\rho}[L(F|_{\rho})] \leq 2p^2 \cdot n^{2+\alpha} \cdot n^{o(1)} = n^{3\alpha - 3\varepsilon/5 + o(1)}.$$

Then

$$\mathop{\mathbb{E}}_{\rho} \mathop{\mathbb{E}}_{F \sim \mathcal{F}}[L(F|_{\rho})] = \mathop{\mathbb{E}}_{F \sim \mathcal{F}} \mathop{\mathbb{E}}_{\rho}[L(F|_{\rho})] \leq n^{3\alpha - 3\varepsilon/5 + o(1)}.$$

By the "close to p-regular" property of Theorem 3.2,

$$\Pr_{\rho}\left[|\rho^{-1}(\star)| \geq n^{\alpha+\varepsilon/5} \right] \geq 2/3.$$

Applying Markov's inequality (as in the proof of Theorem 1.9), we can show there is a restriction ρ such that $|\rho^{-1}(\star)| \geq n^{\alpha+\varepsilon/5}$, and $\Pr_{F\sim\mathcal{F}}[L(F|_{\rho}) \leq s(n)] \geq 0.99$ for some $s(n) = n^{3\alpha-3\varepsilon/5+o(1)}$. Moreover, by our construction, ρ is fully explicit: there is a $\mathrm{polylog}(n)$-size circuit computing $\rho(i)$ on input $i \in [n]$.

Let $\boldsymbol{x}$ be the n-bit string generated by randomly and independently filling in the $\star$ positions of ρ. By a counting argument, with probability at least 0.99, the minimum circuit size of $\boldsymbol{x}$ is at least

$$\frac{|\rho^{-1}(\star)|}{10 \log |\rho^{-1}(\star)|} \geq n^{\alpha+\varepsilon/5-o(1)}.$$

Let $G : \{0,1\}^r \to \{0,1\}^n$ be the local PRG from Theorem 5.3 with size parameter $s(n) = n^{3\alpha-3\varepsilon/5+o(1)}$. Then, if $L(F|_{\rho}) \leq s(n)$, then

$$\left| \mathop{\mathbb{E}}_{z \sim \{0,1\}^r}[F|_{\rho}(G(z))] - \mathop{\mathbb{E}}_{x \sim \{0,1\}^n}[F|_{\rho}(x)] \right| \leq 0.1.$$

Since $\mathcal{F}$ computes $\mathsf{MCSP}[n^{\alpha}]$, we have

$$\mathop{\mathbb{E}}_{x \sim \{0,1\}^n}[\mathcal{F}|_{\rho}(x)] \leq 0.01.$$

Hence

$$\mathop{\mathbb{E}}_{F \sim \mathcal{F}} \mathop{\mathbb{E}}_{x \sim \{0,1\}^n}[F|_{\rho}(x)] = \mathop{\mathbb{E}}_{x \sim \{0,1\}^n} \mathop{\mathbb{E}}_{F \sim \mathcal{F}}[F|_{\rho}(x)] \leq 0.01 + 0.99 \cdot \frac{1}{3} = 0.34.$$

And

$$\begin{aligned}
&\mathop{\mathbb{E}}_{F\sim\mathcal{F}}\ \mathop{\mathbb{E}}_{z\sim\{0,1\}^r}[F|_\rho(G(z))] \\
&\leq \Pr_{F\sim\mathcal{F}}[L(F|_\rho) > s(n)] + \mathop{\mathbb{E}}_{F\sim\mathcal{F}}\left[\mathop{\mathbb{E}}_{z\sim\{0,1\}^r}[F|_\rho(G(z))]\middle| L(F|_\rho)\leq s(n)\right]\cdot \Pr_{F\sim\mathcal{F}}[L(F|_\rho)\leq s(n)] \\
&\leq \Pr_{F\sim\mathcal{F}}[L(F|_\rho) > s(n)] + \mathop{\mathbb{E}}_{F\sim\mathcal{F}}\left[\mathop{\mathbb{E}}_{x\sim\{0,1\}^n}[F|_\rho(x)] + 0.1\middle| L(F|_\rho)\leq s(n)\right]\cdot \Pr_{F\sim\mathcal{F}}[L(F|_\rho)\leq s(n)] \\
&\leq 0.01 + \mathop{\mathbb{E}}_{F\sim\mathcal{F}}\ \mathop{\mathbb{E}}_{x\sim\{0,1\}^n}[F|_\rho(x)] + 0.1 \\
&\leq 0.45.
\end{aligned}$$

However, for each z, string $G(z)\circ\rho$ has circuit size at most $s^{1/3+o(1)} \leq n^{\alpha-\varepsilon/5+o(1)}$, and should be accepted by $\mathcal{F}$. So $\mathbb{E}_{F\sim\mathcal{F}}[F|_\rho(G(z))] \geq 2/3$, a contradiction. □

6 Sharp Thresholds for Explicit Obstructions

In this section we give our construction and sharp threshold for explicit obstructions against De Morgan formulas.

Reminder of Theorem 1.14. *For a universal constant $K > 0$, there is an explicit obstruction against De Morgan formulas of size $n^{2-K/\log\log n}$.*

Proof. We apply the pseudorandom restriction generator from Theorem 3.2 with probability parameter $p(n) := 20/n$. More precisely, we will produce a so-called *anti-checker* for the PARITY function. We use Q to denote the set of restrictions generated by the pseudorandom restriction generator; note that $|Q| \leq \mathrm{poly}(n)$.

For each restriction $\rho\in Q$ and each $i\in\{0\}\cup[n]$, we define an n-bit input $x^{\rho,i}$:

$$\text{for } j\in[n],\quad x^{\rho,i}_j := \begin{cases}\rho(j) & \rho(j)\neq\star,\\ 1 & \rho(j)=\star \text{ and } j=i,\\ 0 & \rho(j)=\star\text{ and } j\neq i.\end{cases}$$

Note that $x^{\rho,0}$ corresponds to filling in all $\star$'s of ρ with 0's. We set our obstruction to be

$$S := \{(x^{\rho,i}, \mathsf{PARITY}(x^{\rho,i})) : \rho\in Q, i\in\{0,1,\ldots,n\}\}.$$

Clearly, S is consistent (but it may contain repetitions), and it can be constructed in $\mathrm{poly}(n)$ time.

Now let F be a formula of size $N^{2-K/\log\log N}$. Let $K = 4C$. Applying Markov's inequality, there is some restriction ρ from the generator such that $|\rho^{-1}(\star)| \geq pn/2 = 10$, and $L(F|_\rho) < 1$, i.e., $F|_\rho$ is constant (we argue such a ρ exists in the proof of Theorem 1.9). Now, let i be such that $\rho(i)=\star$, and consider two inputs $x^{\rho,0}$ and $x^{\rho,i}$ from S. Since $F|_\rho$ is constant, F outputs the same value on these two inputs, yet they have different parities, which completes the proof. □

Now we turn to showing how a slightly improved explicit obstruction implies exponential-size formula lower bounds. We will need a way to efficiently construct a perfect linear hash function for a given input set.

Lemma 6.1 (Efficient Linear Hashing)**.** *There is a deterministic algorithm which takes m distinct strings $s_1, \dots, s_m \in \{0,1\}^n$ as input, and in $\mathrm{poly}(mn)$ time outputs $t = O(\log m)$ strings $w_1, \dots, w_t \in \{0,1\}^n$, such that $H(s_i) \neq H(s_j)$ for all $i \neq j$, where $H : \{0,1\}^n \to \{0,1\}^t$ is defined as*

$$H(x) = (\langle x, w_1\rangle, \langle x, w_2\rangle, \dots, \langle x, w_t\rangle).$$

Proof. Let $E : \{0,1\}^n \to \{0,1\}^{n/c_r}$ be the linear error correcting code from Theorem 2.5, where the constant $c_r \in (0,1)$ is the rate of the code. For any two different $s_i, s_j \in \{0,1\}^n$, we have

$$\Pr_{v_0 \in [n/c_r]}[E(s_i)_{v_0} \neq E(s_j)_{v_0}] \geq c_d,$$

where constant $c_d \in (0,1)$ is the relative distance of E.

Let G be a (strongly explicit) expander graph (with constant parameters $\lambda < 1, d \in \mathbb{N}$) from Theorem 2.8 with $[n/c_r]$ as vertices. Let $t = c \cdot \log m$ for a constant c to be specified later. Given t elements $v_1, v_2, \dots, v_t$ from $[n/c_r]$, we define the hash function

$$H_v(x) := (E(x)_{v_1}, E(x)_{v_2}, \dots, E(x)_{v_t}) \in \{0,1\}^t.$$

For any two different $s_i, s_j \in \{0,1\}^n$, by the Expander Chernoff Bound (Theorem 2.9), if $v = (v_1, v_2, \dots, v_t)$ is a random walk on G, then

$$\begin{aligned}\Pr_v[H_v(s_i) = H_v(s_j)] &\leq \Pr_v\left[\frac{1}{t}\sum_{k=1}^{t}[E(s_i)_{v_k} \neq E(s_j)_{v_k}] < c_d/2\right] \\ &\leq e^{-(1-\lambda)(c_d/2)^2 t/4}.\end{aligned}$$

Choosing $t = c \cdot \log m$ for a sufficiently large constant c, we have

$$\Pr_v[\text{for every two } 1 \leq i < j \leq m,\ H_v(s_i) \neq H_v(s_j)] \geq 0.99$$

by union bound. Note that a walk v on G of length t can be specified by $O(\log(n/c_r) + t) = O(\log(mn))$ bits. We enumerate all $O(\log(mn))$-bit strings and find the lexicographically smallest, such that H_v satisfies the required condition. Then we can output the strings $w_1, \dots, w_t \in \{0,1\}$ which define this linear function H_v. □

Reminder of Theorem 1.15. *The following are equivalent:*

(1) There is an $\alpha > 0$ and an explicit obstruction against formulas of size $n^{2+\alpha}$.

(2) For all k, there is an explicit obstruction against formulas of size n^k.

(3) There is an $\varepsilon > 0$ and a function in E that does not have $2^{\varepsilon n}$-size formulas, even infinitely often.

Proof. **(1) $\Longrightarrow$ (3).** By assumption, there is an explicit obstruction against $n^{2+\alpha}$-size formulas, consisting of n^c input/output pairs, for some constants $\alpha > 0, c \geq 1$. Suppose for contradiction that E has $2^{\varepsilon n}$ size formulas, for all $\varepsilon > 0$.

We run the poly-time algorithm computing the obstruction set S on 1^n, then feed S to the algorithm of Lemma 6.1 to choose a perfect linear hash $H : \{0,1\}^n \to \{0,1\}^{dc \log n}$ that maps all n^c inputs from S to distinct strings, for some universal constant d.

For all n, define the language $L_n \subseteq \{0,1\}^{dc\log n}$, where a string $y \in L_n$ if and only if there is an $x \in \{0,1\}^n$ with $H(x) = y$ such that $(x, 1) \in S$. Define $L := \bigcup L_n$. Because the obstruction can be produced in polynomial time, and the hash H is computable in polynomial time, L can be decided in $\text{poly}(n^c)$ time. Hence $L \in \mathsf{E}$. By assumption, L has formulas of size $2^{\varepsilon(dc\log n)} \leq n^{\alpha/2}$, for $\varepsilon > 0$ sufficiently small.

Now define an n-variable function F by

$$F(x) := L(H(x)).$$

Each output bit of H is a PARITY of a subset of n input bits, which can be implemented by a formula of size n^2. Thus F has formula complexity at most $\leq n^{\alpha/2} \cdot n^2$. Because H maps all of S to distinct strings, the function F defined above agrees with all input/output pairs in S. This implies that F does not have formulas of size $n^{2+\alpha}$, a contradiction.

(3) $\implies$ (2). Let L be a function in E that does not have $2^{\varepsilon n}$-size formulas, even infinitely often. In other words, for all but finitely many n, $L \cap \{0,1\}^n$ has formula complexity greater than $2^{\varepsilon n}$.

For every $k > 1$, we can construct an explicit obstruction against n^k-size formulas as follows. Set $m := (2k/\varepsilon)\log n$. For all $x \in \{0,1\}^m$, include the pair $(x0^{n-m}, L(x))$ in the obstruction, which can be computed in $\text{poly}(n^c)$ time. Now suppose a formula F agrees with all such input-output pairs. Then for all x, the formula $F(x0^{n-m})$ computes $L(x)$ on input length m, implying that F itself must have size at least $2^{\varepsilon m} = n^{2k}$.

(2) $\implies$ (1). Trivial. □

Reminder of Theorem 1.16. *If there is an unbounded function $f(n)$ and an explicit obstruction against formulas of size $n^2 \cdot (\log n)^{f(n)}$, then $\mathsf{E} \not\subset \mathsf{NC}^1$.*

Proof Sketch. The proof is essentially the same as the ((1) $\implies$ (3)) case of Theorem 1.15. If we assume $\mathsf{E} \subset \mathsf{NC}^1$ (instead merely assuming that E has $2^{\varepsilon n}$-size formulas), then the function F constructed has formula complexity at most $n^2(c\log n)^t$ for some constant t, which is smaller than $n^2 \cdot (\log n)^{f(n)}$ for sufficiently large n. But F agrees with all input/output pairs in the obstruction, a contradiction. □

Reminder of Proposition 1.17. *There are explicit obstructions against B_2-formulas of size $o(n)$, and branching programs of size $o(n)$.*

Proof. Note that an $o(n)$-size B_2-formula (or $o(n)$-size branching program) does not depend on all of its inputs. In such a case, our obstruction can simply be the set $S = \{(0^n, 0)\} \cup \{(0^{i-1}10^{n-i}, 1) \mid i \in [n]\}$. □

It is an interesting question to ask: to what degree can this simple proposition be improved? One interesting open problem is whether we can get explicit obstructions against cn-size B_2-formulas or branching programs, for every constant $c \geq 1$. We can show that improving the proposition in the exponent would imply a breakthrough:

Reminder of Theorem 1.18. *The following are equivalent:*

- *There is an $\alpha > 0$ and an explicit obstruction against B_2-formulas of size $n^{1+\alpha}$.*
- *For all k, there is an explicit obstruction against B_2-formulas of size n^k.*
- *There is an $\varepsilon > 0$ such that E cannot be computed by $2^{\varepsilon n}$-size formulas, even infinitely often.*

The same equivalence also holds with "branching programs" in place of "B_2-formulas."

Proof Sketch. The proof is analogous to the equivalence proved in Theorem 1.15. The key difference is that PARITY has $O(n)$-size branching programs and B_2-formulas. Therefore the function F implemented in that proof has B_2-formula complexity at most $n^{1+\alpha/2}$, and branching program complexity at most $n^{1+\alpha/2}$. □

By standard hardness-to-randomness connections, we also have:

Proposition 6.2 ([NW94, IW97, STV01]). E *does not have* $2^{\varepsilon n}$*-size formulas (respectively, branching programs) for some* $\varepsilon > 0$ *(even infinitely often) if and only if for all* k*, there is an* $O(\log n)$*-seed PRG fooling* n^k*-size formulas (respectively, branching programs) that is computable in* $\mathrm{poly}(n)$*-time.*

Thus the problem of computing non-trivial explicit obstructions for these computational models is already equivalent to constructing pseudorandom generators.

7 Open Problems

Our results suggest several new directions and interesting open problems. The most interesting open question would be to prove some super-polynomial lower bounds (e.g., $\mathsf{EXP} \not\subset \mathsf{NC}^1$) via the approaches suggested in this work. In particular:

(1) Can we prove $\mathsf{MCSP}[\mathrm{polylog}(n)]$ does not have $n^{2.01}$-size probabilistic formulas?

(2) Is there a quantified derandomization algorithm for n^2-size probabilistic formulas with at most $2^{n^{0.01}}$ unsatisfying assignments?

(3) Can we construct an explicit obstruction against $n^{2.01}$-size formulas?

Since a positive answer to any of the above questions would imply breakthrough complexity separations, pessimists may believe they are out of reach. Can we provide any formal mathematical reasons to justify such a belief? As far as we can tell, the standard barriers such as diagonalization [BGS75], algebrization [AW09], and natural proofs [RR97] do not directly stand in the way of resolving problems such as (1), (2), and (3).[9] In recent work, Chen *et al.* [CHO+20] formulated a "locality barrier" which suggests that inherently new techniques are required to resolve question (1) above. However, its results do not apply to questions such as (2) and (3).

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009.

[AB17] Amir Abboud and Arturs Backurs. Towards hardness of approximation for polynomial time problems. In *ITCS*, pages 11:1–11:26, 2017.

[AB18] Amir Abboud and Karl Bringmann. Tighter connections between Formula-SAT and shaving logs. In *ICALP*, pages 8:1–8:18, 2018.

[9] For example, Problem (1) focuses on the special function $\mathsf{MCSP}[\mathrm{polylog}(n)]$, which would apparently violate the largeness condition of natural proofs. Problem (2) is based on constructing a derandomization algorithm. Problem (3) asks for an efficient algorithm that constructs bad inputs for small formulas where we already know lower bounds. None of these tasks apparently fit within the framework of natural proofs.

[ABK$^+$02] Eric Allender, Harry Buhrman, Michal Koucký, Dieter van Melkebeek, and Detlef Ronneburger. Power from random strings. *SIAM Journal on Computing*, 35(6):1467–1493, 2006. Preliminary version in FOCS'02.

[AGHP90] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple constructions of almost k-wise independent random variables. In *FOCS*, pages 544–553, 1990.

[AH19] Eric Allender and Shuichi Hirahara. New insights on the (non-)hardness of circuit minimization and related problems. *TOCT*, 11(4):27:1–27:27, 2019.

[AHK17] Eric Allender, Dhiraj Holden, and Valentine Kabanets. The minimum oracle circuit size problem. *Computational Complexity*, 26(2):469–496, 2017.

[AHVW16] Amir Abboud, Thomas Dueholm Hansen, Virginia Vassilevska Williams, and Ryan Williams. Simulating branching programs with edit distance and friends: or: a polylog shaved is a lower bound made. In *Proc. of the 48th STOC*, pages 375–388, 2016.

[AK10] Eric Allender and Michal Koucký. Amplifying lower bounds by means of self-reducibility. *J. ACM*, 57(3):14:1–14:36, 2010.

[AR18] Amir Abboud and Aviad Rubinstein. Fast and deterministic constant factor approximation algorithms for LCS imply new circuit lower bounds. In *ITCS*, pages 35:1–35:14, 2018.

[AW09] Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. *TOCT*, 1(1):2:1–2:54, 2009.

[AYZ95] Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *JACM*, 42(4):844–856, 1995.

[BGS75] Theodore P. Baker, John Gill, and Robert Solovay. Relativizations of the $\mathsf{P} = ?\mathsf{NP}$ question. *SIAM J. Comput.*, 4(4):431–442, 1975.

[BI15] Arturs Backurs and Piotr Indyk. Edit Distance Cannot Be Computed in Strongly Subquadratic Time (unless SETH is false). In *Proc. of the 47th STOC*, pages 51–58, 2015.

[CGL$^+$19] Lijie Chen, Shafi Goldwasser, Kaifeng Lyu, Guy N. Rothblum, and Aviad Rubinstein. Fine-grained complexity meets IP = PSPACE. In *SODA*, pages 1–20, 2019.

[CHO$^+$20] Lijie Chen, Shuichi Hirahara, Igor Carboni Oliveira, Ján Pich, Ninad Rajgopal, and Rahul Santhanam. Beyond natural proofs: Hardness magnification and locality. In *11th Innovations in Theoretical Computer Science Conference, ITCS*, pages 70:1–70:48, 2020.

[CI17] Mahdi Cheraghchi and Piotr Indyk. Nearly optimal deterministic algorithm for sparse Walsh-Hadamard transform. *ACM Trans. Algorithms*, 13(3):34:1–34:36, 2017.

[CILM18] Marco L. Carmosino, Russell Impagliazzo, Shachar Lovett, and Ivan Mihajlin. Hardness amplification for non-commutative arithmetic circuits. In *33rd Computational Complexity Conference, CCC*, pages 12:1–12:16, 2018.

[CJW19] Lijie Chen, Ce Jin, and Ryan Williams. Hardness Magnification for all Sparse NP Languages. In *FOCS*, pages 1240–1255, 2019.

[CKLM19] Mahdi Cheraghchi, Valentine Kabanets, Zhenjian Lu, and Dimitrios Myrisiotis. Circuit lower bounds for MCSP from local pseudorandom generators. In *ICALP*, pages 39:1–39:14, 2019.

[CMMW19] Lijie Chen, Dylan M. McKay, Cody D. Murray, and R. Ryan Williams. Relations and equivalences between circuit lower bounds and Karp-Lipton theorems. In *CCC*, pages 30:1–30:21, 2019.

[CSS16] Ruiwen Chen, Rahul Santhanam, and Srikanth Srinivasan. Average-case lower bounds and satisfiability algorithms for small threshold circuits. In *31st Conference on Computational Complexity, CCC*, pages 1:1–1:35, 2016.

[CT19] Lijie Chen and Roei Tell. Bootstrapping results for threshold circuits ”just beyond” known lower bounds. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 34–41, 2019.

[CW79] J. Lawrence Carter and Mark N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979.

[FGHK16] 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 *FOCS*, pages 89–98, 2016.

[GG81] Ofer Gabber and Zvi Galil. Explicit constructions of linear-sized superconcentrators. *Journal of Computer and System Sciences*, 22(3):407–420, 1981.

[Gil98] David Gillman. A Chernoff bound for random walks on expander graphs. *SIAM Journal on Computing*, 27(4):1203–1220, 1998.

[Gol08] Oded Goldreich. *Computational complexity - a conceptual perspective*. Cambridge University Press, 2008.

[GVW11] Oded Goldreich, Salil Vadhan, and Avi Wigderson. Simplified derandomization of bpp using a hitting set generator. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, pages 59–67. Springer, 2011. Preliminary version in ECCC, TR00-004, 2000.

[GW14] Oded Goldreich and Avi Widgerson. On derandomizing algorithms that err extremely rarely. In *Proc. of the 47th STOC*, pages 109–118, 2014.

[Hås98] Johan Håstad. The shrinkage exponent of de Morgan formulas is 2. *SIAM J. Comput.*, 27(1):48–64, 1998.

[Hir18] Shuichi Hirahara. Non-black-box worst-case to average-case reductions within NP. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 247–258, 2018.

[HOS18] Shuichi Hirahara, Igor Carboni Oliveira, and Rahul Santhanam. Np-hardness of minimum circuit size problem for OR-AND-MOD circuits. In *33rd Computational Complexity Conference, CCC*, pages 5:1–5:31, 2018.

[HP15] John M. Hitchcock and Aduri Pavan. On the np-completeness of the minimum circuit size problem. In *35th IARCS Annual Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS*, pages 236–245, 2015.

[HS17] Shuichi Hirahara and Rahul Santhanam. On the average-case complexity of MCSP and its variants. In *32nd Computational Complexity Conference, CCC*, pages 7:1–7:20, 2017.

[HW16] Shuichi Hirahara and Osamu Watanabe. Limits of minimum circuit size problem as oracle. In *31st Conference on Computational Complexity, CCC*, pages 18:1–18:20, 2016.

[IM02] Kazuo Iwama and Hiroki Morizumi. An explicit lower bound of 5n - o(n) for boolean circuits. In *MFCS*, pages 353–364, 2002.

[IMZ12] Russell Impagliazzo, Raghu Meka, and David Zuckerman. Pseudorandomness from shrinkage. *J. ACM*, 66(2):11:1–11:16, April 2019. Preliminary version in FOCS'12.

[IPS97] Russell Impagliazzo, Ramamohan Paturi, and Michael E. Saks. Size-depth tradeoffs for threshold circuits. *SIAM J. Comput.*, 26(3):693–707, 1997.

[IW97] Russell Impagliazzo and Avi Wigderson. P = BPP if E requires exponential circuits: Derandomizing the XOR lemma. In *STOC*, pages 220–229, 1997.

[KC00] Valentine Kabanets and Jin-Yi Cai. Circuit minimization problem. In *STOC*, pages 73–79, 2000.

[KRT17] Ilan Komargodski, Ran Raz, and Avishay Tal. Improved average-case lower bounds for de Morgan formula size: Matching worst-case lower bound. *SIAM Journal on Computing*, 46(1):37–57, 2017.

[Lev84] Leonid A. Levin. Randomness conservation inequalities; information and independence in mathematical theories. *Information and Control*, 61(1):15 – 37, 1984.

[Li16] Xin Li. Improved two-source extractors, and affine extractors for polylogarithmic entropy. In *FOCS*, pages 168–177, 2016.

[LR01] Oded Lachish and Ran Raz. Explicit lower bound of $4.5n - o(n)$ for boolean circuits. In *Proc. of 33rd STOC*, pages 399–408, 2001.

[LW13] Richard J. Lipton and Ryan Williams. Amplifying circuit lower bounds against polynomial time, with applications. *Computational Complexity*, 22(2):311–343, 2013.

[LY94] Richard J. Lipton and Neal E. Young. Simple strategies for large zero-sum games with applications to complexity theory. In *STOC*, pages 734–740, 1994.

[MMW19] Dylan M. McKay, Cody D. Murray, and R. Ryan Williams. Weak lower bounds on resource-bounded compression imply strong separations of complexity classes. In *STOC*, pages 1215–1225, 2019.

[MP20] Moritz Müller and Ján Pich. Feasibly constructive proofs of succinct weak circuit lower bounds. *Ann. Pure Appl. Log.*, 171(2), 2020.

[Mul11] Ketan Mulmuley. On P vs. NP and geometric complexity theory: Dedicated to sri ramakrishna. *J. ACM*, 58(2):5:1–5:26, 2011.

[MW17] Cody D. Murray and R. Ryan Williams. On the (non) np-hardness of computing circuit complexity. *Theory of Computing*, 13(1):1–22, 2017.

[MW18] Cody Murray and R. Ryan Williams. Circuit lower bounds for nondeterministic quasi-polytime: an easy witness lemma for NP and NQP. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 890–901, 2018.

[Nec66] E. I. Nechiporuk. On a boolean function. *Doklady of the Academy of Sciences of the USSR*, 169(4):765–766, 1966. English translation in Soviet Mathematics Doklady 7:4, pages 999–1000.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994.

[Oli19] Igor Carboni Oliveira. Randomness and intractability in Kolmogorov complexity. In *ICALP*, pages 32:1–32:14, 2019.

[OPS19] Igor Carboni Oliveira, Ján Pich, and Rahul Santhanam. Hardness magnification near state-of-the-art lower bounds. In *Proc. of the 34th CCC*, pages 27:1–27:29, 2019.

[OS18] Igor Carboni Oliveira and Rahul Santhanam. Hardness magnification for natural problems. In *FOCS*, pages 65–76, 2018.

[RR97] Alexander A. Razborov and Steven Rudich. Natural proofs. *J. Comput. Syst. Sci.*, 55(1):24–35, 1997.

[Spi96] Daniel A. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Trans. Information Theory*, 42(6):1723–1731, 1996.

[Sri03] Aravind Srinivasan. On the approximability of clique and related maximization problems. *J. Comput. Syst. Sci.*, 67(3):633–651, 2003.

[STV01] Madhu Sudan, Luca Trevisan, and Salil P. Vadhan. Pseudorandom generators without the XOR lemma. *J. Comput. Syst. Sci.*, 62(2):236–266, 2001.

[SU05] Ronen Shaltiel and Christopher Umans. Simple extractors for all min-entropies and a new pseudorandom generator. *Journal of the ACM (JACM)*, 52(2):172–216, 2005.

[Tal14] Avishay Tal. Shrinkage of De Morgan formulae by spectral techniques. In *Proc. of 55th FOCS*, pages 551–560, 2014.

[Tel18] Roei Tell. Quantified derandomization of linear threshold circuits. In *Proc. of the 50th STOC*, pages 855–865, 2018.

[Tel19] Roei Tell. Improved bounds for quantified derandomization of constant-depth circuits and polynomials. *Computational Complexity*, 28(2):259–343, 2019.

[Tra84] B. A. Trakhtenbrot. A survey of russian approaches to perebor (brute-force searches) algorithms. *IEEE Ann. Hist. Comput.*, 6(4), October 1984.

[Vad12] Salil P. Vadhan. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012.

[Weg87] Ingo Wegener. *The Complexity of Boolean Functions*. John Wiley & Sons Ltd, 1987.

[Wil13] Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. *SIAM Journal on Computing*, 42(3):1218–1244, 2013.

[Wil14] Ryan Williams. Nonuniform acc circuit lower bounds. *Journal of the ACM (JACM)*, 61(1):2, 2014.

A Construction of Linear Extractors with Short Seed

In this section we prove Theorem 4.1, which is implicit in [Li16]. We first need the concepts of strong seeded extractor and condenser.

Definition A.1 (Strong seeded extractor)**.** A function $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k,ε) *strong seeded extractor*, if for every min-entropy k source X,

$$|\mathsf{Ext}(X, \mathbf{U}_d) - (\mathbf{U}_m, \mathbf{U}_d)| \leq \varepsilon,$$

where $\mathbf{U}_m$ is the uniform distribution on m bits and $\mathbf{U}_d$ is the uniform distribution on d bits independent of X. We say that the function is a linear strong seeded extractor if the function $\mathsf{Ext}(\cdot, u)$ is a linear function over $\mathbb{F}_2$, for every $u \in \{0,1\}^d$.

Definition A.2 (Condenser)**.** A function $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is an $k \to_\varepsilon k'$ condenser if for every X with min-entropy at least k, $C(X,Y)$ is ε-close to a distribution with min-entropy k', when Y is uniformly distributed on $\{0,1\}^d$. A condenser is explicit if it is computable in polynomial time. A condenser is called lossless if $k' = k + d$.

Next we need the following two constructions of linear extractors and linear condensers.

Lemma A.3 ([SU05])**.** *For every $n \in \mathbb{N}$, constant $\delta > 0, \varepsilon \geq 2^{-k^{\delta/4}}$, and $k \geq \log^{4/\delta} n$ there is an explicit (k,ε) strong linear seeded extractor $\{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = O(\log n + \frac{\log n}{\log k} \log(1/\varepsilon))$ and $m = k^{1-\delta}$.*

Lemma A.4 ([CI17])**.** *For any constant $\alpha > 0$ and any $n \in \mathbb{N}, k \leq n, \varepsilon > 0$ there is an explicit strong (k,ε)-lossless condenser $\mathsf{Cond} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d \leq (1+1/\alpha)(\log(nk/\varepsilon) + O(1))$ and $m \leq (1+\alpha)k$. Moreover, Cond is a linear function for every fixed choice of the seed.*

Now we are ready to combine the above two constructions to prove Theorem 4.1.

Reminder of Theorem 4.1. *For every constant $0 < \beta < 1$ and sufficiently large n, there is an explicit (n^β, ε) linear extractor $\{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d \leq (1 + O(\beta)) \log n$, $\varepsilon = n^{-\beta}$ and $m = n^{\beta/2}$.*

Proof. Denote $k := n^\beta$. Given any (n,k) source X, first use Lemma A.4 and take $\leq (1+1/\alpha)(\log(2nk/\varepsilon) + O(1))$ bits of seed to condense X into an (n',k) source Y with length $n' \leq (1+\alpha)k$, and error $\varepsilon/2$. Then use Lemma A.3 (with parameter $\delta := 1/2$) to extract $m = \sqrt{k}$ bits from Y with error $\varepsilon/2$, using another $O(\log n' + \frac{\log n'}{\log k} \log(2/\varepsilon))$ bits of seed. Since both the condenser and the extractor are linear and strong, the composed extractor is also a strong linear seeded extractor.

By setting $\alpha := 1/\beta$, the total seed length is at most

$$\begin{aligned}
&(1+1/\alpha)(\log(2nk/\varepsilon) + O(1)) + O(\log n' + \frac{\log n'}{\log k} \log(2/\varepsilon)) \\
\leq\ & (1+1/\alpha)(\log(nk^2) + O(1)) + O(\log n') \\
\leq\ & (1+\beta)((1+2\beta)\log n + O(1)) + O(\log((1+1/\beta)n^\beta)) \\
\leq\ & (1+O(\beta)) \cdot \log n,
\end{aligned}$$

for sufficiently large n. □

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il