Range Avoidance and Remote Point: New Algorithms and Hardness

Shengtang Huang[1]
School of the Gifted Young, University of Science and Technology of China, Hefei, Anhui, China

Xin Li
Department of Computer Science, Johns Hopkins University, Baltimore, MD, USA

Yan Zhong
Department of Computer Science, Johns Hopkins University, Baltimore, MD, USA

Abstract

The Range Avoidance (Avoid) problem $\mathcal{C}$-Avoid$[n, m(n)]$ asks that, given a circuit in a class $\mathcal{C}$ with input length n and output length $m(n) > n$, find a string not in the range of the circuit. This problem has been a central piece in several recent frameworks for proving circuit lower bounds and constructing explicit combinatorial objects. Previous work by Korten (FOCS' 21) and by Ren, Santhanam, and Wang (FOCS' 22) showed that algorithms for Avoid are closely related to circuit lower bounds. In particular, Korten's work reinterpreted an earlier result from bounded arithmetic, originally proved by Jeřábek (Ann. Pure Appl. Log. 2004), as an equivalence in computational complexity between the existence of $\mathbf{FP}^{\mathbf{NP}}$ algorithms for the general Avoid problem and $2^{\Omega(n)}$ lower bounds against general Boolean circuits for the class $\mathbf{E}^{\mathbf{NP}}$. In this work, we significantly complement these works by generalizing the equivalence result to restricted circuit classes and obtain the following:

- For any constant depth unbounded fan-in circuit class $\mathcal{C} \supseteq \mathsf{AC}^0$, there is an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for $\mathcal{C}$-Avoid$[n, n^{1+\varepsilon}]$ (for any constant $\varepsilon > 0$) if and only if $\mathbf{E}^{\mathbf{NP}}$ cannot be computed by $\mathcal{C}$ circuits of size $2^{o(n)}$. This addresses an open problem by Korten (Bulletin of EATCS' 25).
- If $\mathbf{E}^{\mathbf{NP}}$ cannot be computed by $o(2^n/n)$ size formulas, then there is an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for NC^0-Avoid$[n, 2n]$. Note that by an extension of Ren, Santhanam, and Wang (FOCS' 22), an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for NC^0_4-Avoid$[n, n + n^\delta]$ for any constant $\delta \in (0, 1)$ implies $\mathbf{E}^{\mathbf{NP}}$ cannot be computed by $o(2^n/n)$ size formulas.

These results yield the first characterizations of $\mathbf{FP}^{\mathbf{NP}}$ $\mathcal{C}$-Avoid algorithms for low-complexity circuit classes such as AC^0.

We also consider the average-case analog of Avoid, the Remote Point (Remote-Point) problem, and establish:

- For some suitable function $c(n)$ and constant $\gamma > 0$, there is an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for Remote-Point$[n, n^{6+\gamma}, c(O_\gamma(\log n))]$ if and only if $\mathbf{E}^{\mathbf{NP}}$ cannot be $(1/2 - c(n))$-approximated by circuits of size $2^{o(n)}$.

Finally, we also present two improved algorithms for NC^0-Avoid:

- A family of $2^{n^{1-\frac{\varepsilon}{k-1}+o(1)}}$ time algorithms for NC^0_k-Avoid$[n, n^{1+\varepsilon}]$ for any $\varepsilon > 0$, exhibiting the first subexponential-time algorithm for any super-linear stretch.
- Faster local algorithms for NC^0_k-Avoid$[n, n+1]$ running in time $O(n2^{\frac{k-2}{k-1}n})$, improving the naive $2^n \cdot \mathrm{poly}(n)$ bound.

2012 ACM Subject Classification Theory of computation → Complexity classes; Theory of computation → Circuit complexity; Theory of computation → Pseudorandomness and derandomization; Theory of computation → Expander graphs and randomness extractors

Keywords and phrases Circuit Lower Bounds, Range Avoidance Problem, Remote Point Problem

Digital Object Identifier 10.4230/LIPIcs.ITCS.2026.79

[1] Work done while visiting Johns Hopkins University.

17th Innovations in Theoretical Computer Science Conference (ITCS 2026).
Editor: Shubhangi Saraf; Article No. 79; pp. 79:1–79:19
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2025/049/`

Funding *Xin Li*: Supported by NSF CAREER Award CCF-1845349 and NSF Award CCF-2127575.
Yan Zhong: Supported by NSF CAREER Award CCF-1845349.

Acknowledgements We thank the anonymous ITCS 2026 reviewers for their thoughtful comments, the ITCS 2025 reviewers for their helpful feedback on a preliminary version of this work, and the FOCS 2025 reviewers for identifying bugs in the original proofs of the equivalence related to the Remote-Point problem and the equivalence involving exponential-size circuits for NC^1.
We would also like to thank Hanlin Ren for helpful discussions on range avoidance and Kuan Cheng for discussions on decoding in AC^0. We are grateful to Erfan Khaniki for pointing out a historical inaccuracy in an earlier version of this manuscript – the arguments underlying what we had referred to as "Korten's reduction" already appeared in Jeřábek 's earlier work, as carefully credited in [28]. Accordingly, referring to it solely as "Korten's reduction" is not historically accurate.

1 Introduction

The *Range Avoidance* problem (Avoid for short) is a total search problem introduced in [25, 28, 36], which has recently garnered significant attention. This interest stems from several natural motivations, such as identifying natural total search problems in the polynomial hierarchy (more specifically $\mathbf{\Sigma}_2$) and compelling applications in proof complexity. Notably, Korten [28] demonstrated that numerous explicit constructions of important combinatorial objects can be reduced to instances of Avoid. These include optimal Ramsey graphs, expander graphs, rigid matrices, and hard functions, among others.

At its core, the Range Avoidance problem captures a broad class of objects whose existence is typically proven via the probabilistic method [12]. As such, solving Avoid offers a potentially unified way for constructing these objects explicitly. We now define the problem formally.

▶ **Definition 1.1** (Avoid). *The* range avoidance problem*, denoted by Avoid, is the total search problem in which, given a Boolean circuit $C : \{0,1\}^n \to \{0,1\}^m$ for $m := m(n)$*[2]*$> n$, output any $y \in \{0,1\}^m \setminus \mathrm{Range}(C)$, where* $\mathrm{Range}(C) := \{C(x) \mid x \in \{0,1\}^n\}$.

Closely related is the more general Remote-Point[3] problem, which is studied extensively in previous works [25, 8, 9] and can be thought as the "average-case analog" of Avoid.

▶ **Definition 1.2** (Remote-Point). *Given a code where the encoding function is represented by a circuit $C : \{0,1\}^n \to \{0,1\}^m$ for $m := m(n) > n$ and the codewords are the range of the circuit, find an m-bit string that is far from all codewords in Hamming distance.*

While the original formulation of Avoid allows arbitrary circuits, subsequent work initiated by [36] has focused on the problem for restricted circuit classes.

▶ **Definition 1.3.** *Let $\mathcal{C}$ be a (multi-output) circuit class,*

- $\mathcal{C}$**-Avoid**$[n, m]$ *is the class of Avoid problems where the circuits are in $\mathcal{C}$, with input length n and output length m;*
- $\mathcal{C}$**-Remote-Point**$[n, m, c(n)]$ *denotes the class of Remote-Point problems where the underlying circuits belong to $\mathcal{C}$, with input length n and output length m, and where the desired output has relative Hamming distance $1/2 - c(n)$ from every string in the range of circuits in $\mathcal{C}$.*

[2] The function $m(n)$ is called the *stretch* of the circuit.
[3] We sometimes use RPP as a shorthand for Remote-Point.

A prominent motivation for studying $\mathcal{C}$-Avoid is its implication for circuit lower bounds. In particular, [36] showed that for any circuit class $\mathcal{C}$ satisfying the *universality property* – namely, the *truth table generator* $\mathsf{TT}_{\mathcal{C}}$ (i.e., a circuit that, given an encoding of a circuit $C \in \mathcal{C}$, outputs C's truth table) is itself computable by $\mathcal{C}$ circuits (e.g., $\mathsf{AC}^0, \mathsf{TC}^0, \mathsf{NC}^1$) – efficient algorithms for $\mathcal{C}$-Avoid imply circuit lower bounds for $\mathcal{C}$. Specifically, solving $\mathcal{C}$-Avoid in $\mathbf{FP}$ (resp. $\mathbf{FP^{NP}}$) implies that $\mathbf{E}$ (resp. $\mathbf{E^{NP}}$) does not have $\mathcal{C}$ circuits.[4] Analogously, $\mathbf{FP}$ (resp. $\mathbf{FP^{NP}}$) algorithms for $\mathcal{C}$-Remote-Point imply average-case $\mathcal{C}$ circuit lower bounds, which are central questions in the area of average-case complexity that have resulted in a large body of works improving correlation bounds for various models of computation (e.g., [4, 7, 6, 3, 32]). On the other hand, these results also imply that it is potentially hard to design efficient algorithms for $\mathcal{C}$-Avoid even when $\mathcal{C}$ is restricted, hence many algorithms given in previous work are *conditional*.

Furthermore, these works also demonstrate that Avoid is already extremely interesting and useful for restricted classes of circuits, for example, even when the circuit is in the class NC^0, and even when each output bit only depends on at most 4 input bits. Below, we use NC^0_k to stand for circuits in NC^0 where each output bit depends on at most k input bits. The same notation goes for the class NC^1. In this sense, the work of [36] shows that, suppose for every constant $\varepsilon > 0$, there is an $\mathbf{FP}$ (resp. $\mathbf{FP^{NP}}$) algorithm for NC^0_4-Avoid$[n, n+n^{\varepsilon}]$, then for every $k \geq 1$, there is an $\mathbf{FP}$ (resp. $\mathbf{FP^{NP}}$) algorithm for NC^1_k-Avoid; and for every $\gamma > 0$, there is a family of functions in $\mathbf{E}$ (resp. $\mathbf{E^{NP}}$) that cannot be computed by Boolean circuits of depth $n^{1-\gamma}$. Furthermore, [15] showed that constructing binary linear codes achieving the Gilbert-Varshamov bound or list-decoding capacity, and constructing rigid matrices reduce to NC^0_4-Avoid; and [13] showed that constructing rigid matrices reduces even to NC^0_3-Avoid.

Driven by these motivations and applications, there have been several works studying both algorithms and hardness results for Avoid and Remote-Point. On the algorithm side, [8] designed an unconditional $\mathbf{FP^{NP}}$ algorithm for ACC^0-Remote-Point$[n, \mathrm{qpoly}(n), 1/\mathrm{poly}(n)]$ ($\mathrm{qpoly}(n)$ denotes quasi-polynomial(n)), recovering the state-of-the-art average-case lower bound for ACC^0 against $\mathbf{E^{NP}}$. A recent breakthrough [5, 33] showed that $\mathsf{S_2E} \not\subset$ *i.o.*-$\mathsf{SIZE}[\frac{2^n}{n}]$[5] via a single-valued $\mathsf{FS_2P}$ algorithm to Avoid, improving over the decades' old lower bound that $\Delta_3\mathsf{E} = \mathsf{E}^{\Sigma_2} \not\subset \mathsf{SIZE}[2^{o(n)}]$ [34]. On the hardness side, Ilango, Li, and Williams [18] showed that under the assumption that subexponential secure indistinguishability obfuscation ($i\mathcal{O}$) exists [22] and $\mathbf{NP} \neq \mathbf{coNP}$, we have that Avoid $\notin \mathbf{FP}$ (i.e., there are no polynomial-time algorithms to solve Avoid). A subsequent work by Chen and Li [9] generalizes the framework and shows that under plausible cryptographic assumptions, $\mathcal{C}$-Avoid and $\mathcal{C}$-Remote-Point are not in $\mathbf{FP}$, or even not in $\mathbf{SearchNP}$, when the underlying $\mathcal{C}$ has small enough stretch (e.g., in the case of NC^0-Avoid, the hardness works for the minimal stretch $m(n) = n + 1$).

However, for certain applications (e.g., explicit constructions of important combinatorial objects) one would desire *relatively efficient* algorithms (e.g., polynomial-time algorithms or at least $\mathbf{FP^{NP}}$ algorithms). Yet even for the case of NC^0-Avoid, the current state-of-the-art results only work for large stretches. For example, the polynomial-time algorithms for NC^0_k-Avoid [15, 13] require the stretch to be at least $n^{k-1}/\log(n)$. Most recently, this was improved to $\widetilde{O}(n^{k/2})$ for even k by [27], which also improved the stretch to $(\widetilde{O}(n^{k/2+(k-2)/(2k+4)}))$ with an $\mathbf{FP^{NP}}$ algorithm for odd k. A conditional $\mathbf{FP^{NP}}$ algorithm was proposed in [36] for NC^0-Avoid with stretch $n^{1+\varepsilon}$ for any constant ε, and whether there is an unconditional

[4] The size of the circuit lower bound depends on the stretch of the Avoid instance.

[5] The prefix "*i.o.*-" indicates that $\mathsf{S_2E}$ is not infinitely often in $\mathsf{SIZE}[2^n/n]$, that is $\mathsf{S_2E}$ *eventually* requires $\mathsf{SIZE}[2^n/n]$ circuit.

$\mathbf{FP}^{\mathbf{NP}}$ algorithm for such stretch is left as a central open question in [36]. Even if one allows for subexponential ($2^{O(n^{1-\varepsilon})}$) time, the best known algorithms for NC^0_k-Avoid only works for stretch $n^{k-2+\varepsilon}$ [13].

A recent work by Kuntewar and Sarma [31] showed that the monotone version of NC^0_3-Avoid$[n, n+1]$, i.e., Monotone-NC^0_3-Avoid$[n, n+1]$ can be solved in polynomial time; the symmetric version of NC^0_3-Avoid$[n, 8n+1]$, i.e., Symmetric-NC^0_3-Avoid$[n, n+1]$ can be solved in polynomial time.

These results fall short of the above mentioned goal of a unified approach towards explicit constructions of combinatorial objects, as most interesting explicit construction problems only reduce to $\mathcal{C}$-Avoid with very small *stretch.* For example, in the case of NC^0-Avoid, to show a better circuit lower bound, one needs $m = n + n^{o(1)}$; while finding rigid matrices enough for Valiant's application needs $m = n + n^{2/3}$ [13]. This was also noted and remarked in [36].

> "We think this result reveals some fundamental difference between the small-stretch regime ($m(n) = n+1$), for which an avoidance algorithm for NC^0 implies breakthrough lower bounds, and the large-stretch regime ($m(n) = n^{1+\Omega(1)}$), for which an avoidance algorithm for NC^0 seems within reach (Theorem 3.12)."

Therefore, it is interesting and important to study the tradeoff between the stretch and the hardness for $\mathcal{C}$-Avoid when $\mathcal{C}$ is restricted (e.g., NC^0, AC^0 and ACC^0), and similarly for $\mathcal{C}$-Remote-Point as better algorithms in this case may lead to stronger average-case circuit lower bounds. In this paper, we make progress towards this direction, by establishing several new results in terms of both algorithms and hardness for $\mathcal{C}$-Avoid and $\mathcal{C}$-Remote-Point, where $\mathcal{C}$ are suitable classes of circuits.

1.1 Our Results

While as mentioned before, several previous works showed that algorithms for $\mathcal{C}$-Avoid or $\mathcal{C}$-Remote-Point with small stretch lead to circuit lower bounds, the works [23, 28, 5] remarkably showed that the converse is also true in the case where $\mathcal{C}$ is the class of unrestricted Boolean circuits. Specifically, they showed that

$$\text{Avoid} \in \mathbf{FP}^{\mathbf{NP}} \iff \mathbf{E}^{\mathbf{NP}} \not\subset i.o.\text{-}\mathsf{SIZE}[2^{o(n)}] \iff \mathbf{E}^{\mathbf{NP}} \not\subset i.o.\text{-}\mathsf{SIZE}[2^n/n]^6$$

In particular, assuming $\mathbf{E}^{\mathbf{NP}}$ does not have subexponential-size circuits implies an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for Avoid on unrestricted circuits. This assumption is significantly weaker than the classical hardness required in PRG-based approaches [21, 26], which assume that $\mathbf{E}$ lacks subexponential-size SAT-oracle circuits to derandomize $\mathbf{FZPP}^{\mathbf{NP}}$.

Thus, for unrestricted Boolean circuits, algorithms for Avoid and lower bounds for $\mathbf{E}^{\mathbf{NP}}$ are, in a precise sense, equivalent. However, such an equivalence was previously unknown for restricted circuit classes. Our first major contribution is to significantly extend previous works, by establishing (near) equivalence for certain restricted classes $\mathcal{C}$, more specifically constant depth circuits with possible augmented gates[7]. As a result, we also obtain conditional $\mathbf{FP}^{\mathbf{NP}}$ algorithms for $\mathcal{C}$-Avoid for these circuit classes $\mathcal{C}$ with suitable smaller stretch, under much weaker assumptions than those needed for general Avoid in [28]. In addition, we establish a new equivalence result between $\mathbf{FP}^{\mathbf{NP}}$ algorithms for Remote-Point and average-case general circuit lower bound for $\mathbf{E}^{\mathbf{NP}}$.

[7] Say, exact threshold gates.

1.1.1 Equivalence between $\mathrm{FP}^{\mathrm{NP}}$ $\mathcal{C}$-Avoid Algorithms and Exponential-size $\mathcal{C}$ Circuit Lower Bound against E^{NP}

As mentioned in the above paragraphs, previous works [28, 36] established the direction from Avoid algorithms to circuit lower bounds. In this work, we complete the equivalence by showing the converse direction for a range of natural restricted circuit classes.

Results for NC^0_4 Circuits with Small Stretch. Our first set of results concerns NC^0_4 circuits. We show that near-maximal formula lower bounds against $\mathbf{E}^{\mathbf{NP}}$ imply efficient algorithms for NC^0_4-Avoid with small stretch:

▶ **Theorem 1.4.** *If $\mathbf{E}^{\mathbf{NP}}$ requires near-maximum ($\Omega(2^n/n)$) size formulas*[8]*, then there is an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for NC^0-Avoid$[n, 2n]$. In particular, this implies an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for NC^0_4-Avoid$[n, 2n]$.*

Conversely, extending ideas from [36], we show:

▶ **Theorem 1.5** (Strong Version of Theorem 5.8 in [36])**.** *For any constant $\delta \in (0, 1)$, NC^0_4-Avoid$[n, n + n^\delta] \in \mathbf{FP}^{\mathbf{NP}} \implies \mathbf{E}^{\mathbf{NP}} \not\subset$ i.o.-*$\mathsf{Formula}[o(2^n/n)]$.

Together, these results nearly characterize the hardness of proving near-maximum $\mathbf{E}^{\mathbf{NP}}$ lower bounds against formulas in terms of $\mathbf{FP}^{\mathbf{NP}}$ algorithms for NC^0_4-Avoid.

Results for Constant Depth Circuit Classes Containing AC^0 with Polynomial Stretch. In the regime of polynomial stretch, we obtain tight equivalences for constant depth unbounded fan-in circuit classes $\mathcal{C}$ satisfying $\mathsf{AC}^0 \subseteq \mathcal{C}$:

▶ **Theorem 1.6.** *For any constant depth unbounded fan-in circuit class $\mathcal{C}$ such that $\mathsf{AC}^0 \subseteq \mathcal{C}$ (e.g., $\mathsf{AC}^0, \mathsf{ACC}^0, \mathsf{TC}^0$), $\mathbf{E}^{\mathbf{NP}}$ requires $2^{\Omega(n)}$ size $\mathcal{C}$ circuits if and only if there is an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for $\mathcal{C}$-Avoid$[n, n^{1+\varepsilon}]$ for any constant $\varepsilon > 0$.*

Moreover, we show analogous equivalences for $\mathbf{FQP}^{\mathbf{NP}}$[9] algorithms and $\mathbf{EXP}^{\mathbf{NP}}$ circuit lower bounds:

▶ **Theorem 1.7.** *For any constant depth unbounded fan-in circuit class $\mathcal{C}$ such that $\mathsf{AC}^0 \subseteq \mathcal{C}$, $\mathbf{EXP}^{\mathbf{NP}}$ requires $2^{\Omega(n)}$ size $\mathcal{C}$ circuits if and only if there is an $\mathbf{FQP}^{\mathbf{NP}}$ algorithm for $\mathcal{C}$-Avoid$[n, n^{1+\varepsilon}]$ for any constant $\varepsilon > 0$.*

These results represent the first equivalence theorems connecting algorithms for $\mathcal{C}$-Avoid with explicit lower bounds for $\mathbf{E}^{\mathbf{NP}}$ and $\mathbf{EXP}^{\mathbf{NP}}$ in restricted circuit classes.

We remark that the complexity-theoretic assumptions we made for Theorem 1.4 and Theorem 1.6 are consistent with our current knowledge of circuit lower bounds.

[8] In a preliminary version of this paper (Revision1OfTR25-049), we claim a near equivalence regarding "exponential-size NC^1 circuits". However, exponential-size NC^1 circuits actually do not make sense because if the circuit is in NC and the depth is $O(\log n)$, then the size has to be polynomial. It only makes sense to talk about exponential size AC^i circuits.

[9] $\mathbf{FQP}$ denotes the class of functions computable in *quasi-polynomial time*, i.e., time $T(n) = n^{(\log n)^{O(1)}}$.

Connections to Open Problems. Our results make progress on the following open question:

▶ **Open Problem 1.8** (Open problem 2 in [29]). *Can we reduce $\mathcal{C}$-Avoid to circuit lower bounds for $\mathcal{C}$ for any circuit class $\mathcal{C} \subseteq \mathbf{P}/\mathrm{poly}$?*

Specifically, Theorem 1.6 and Theorem 1.7 address Open Problem 1.8 in the stretch regime $m(n) = n^{1+\varepsilon}$, for any constant $\varepsilon > 0$, and any circuit classes containing AC^0. In addition, Theorem 1.4 and Theorem 1.5 also nearly pin down the hardness of proving $\mathbf{E}^{\mathbf{NP}}$ requires exponential size formulas in terms of NC^0_4-Avoid algorithm: proving such a lower bound should be no harder than proving NC^0_4-Avoid$[n, n+n^\delta] \in \mathbf{FP}^{\mathbf{NP}}$ for any $\delta \in (0,1)$, but should be no easier than NC^0_4-Avoid$[n, 2n] \in \mathbf{FP}^{\mathbf{NP}}$.

1.1.2 Equivalence between $\mathbf{FP}^{\mathbf{NP}}$ RPP Algorithms and Average-case Exponential-size Circuit Lower Bound against $\mathbf{E}^{\mathbf{NP}}$

Recall the definition of *good* function from [36].

▶ **Definition 1.9** (Good function [36]). *A function $f : \mathbb{N} \to \mathbb{N}$ is good if there is a Turing machine that, given the input n (in binary), outputs the value $f(n)$ (also in binary), and runs in time at most* $\mathrm{poly}(\log n, \log f(n))$.

The equivalence result for Avoid established in [28] naturally raises the question of whether a similar equivalence holds in the average-case setting. In this paper, we answer this question affirmatively and obtain the following theorems.

▶ **Theorem 1.10.** *Let $c : \mathbb{N} \to \mathbb{N}$ be a good and monotonically decreasing function that satisfies $c(O(\log n)) \geq 1/n$. Then $\mathbf{E}^{\mathbf{NP}}$ cannot be $(1/2 + c(n))$-approximated by $2^{o(n)}$-size general boolean circuits if and only if there is an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for RPP$[n, n^{6+\gamma}, c(O_\gamma(\log n))]$ for some constant $\gamma > 0$.*

▶ **Theorem 1.11.** *Let $c : \mathbb{N} \to \mathbb{N}$ be a good and monotonically decreasing function that satisfies $c(O(\log n)) \geq 1/n$. Then $\mathbf{EXP}^{\mathbf{NP}}$ cannot be $(1/2 + c(n))$-approximated by $2^{o(n)}$-size general boolean circuits if and only if there is an $\mathbf{FQP}^{\mathbf{NP}}$ algorithm for RPP$[n, n^{6+\gamma}, c(O_\gamma(\log n))]$ for some constant $\gamma > 0$.*

1.1.3 New NC^0-Avoid Algorithms

As our second contribution, we design a new $2^{n^{1-\frac{\varepsilon}{k-1}+o(1)}}$ time algorithm for NC^0_k-Avoid$[n, n^{1+\varepsilon}]$. This gives the first subexponential-time[10] algorithm for NC^0_k-Avoid with any super-linear stretch for any constant k.

▶ **Theorem 1.12.** *For any $\varepsilon > 0$, there exists a family of $2^{n^{1-\frac{\varepsilon}{k-1}+o(1)}}$ time algorithms for NC^0_k-Avoid$[n, n^{1+\varepsilon}]$. In addition, the algorithm can output a succinct representation of $\geq 1/2$ fraction of strings outside the range.*

Previously, the best known algorithms with comparable running time were applicable only to stretch $m(n) = \tilde{O}(n^{k/2})$ [27][11], making our result the first to achieve subexponential-time performance with superlinear stretch for all k. Subsequently, the work of [15] further improved the running time to $2^{n^{1-\frac{2\varepsilon}{k-3}+o(1)}}$.

[10] There are two notions of subexponentiality in literature: $\bigcap_{c<1} 2^{O(n^c)}$ and $\bigcup_{c<1} 2^{O(n^c)}$. Here, we denote by subexponential a function that is contained in $\bigcup_{c<1} 2^{O(n^c)}$.

[11] For the special case $k = 3$, an algorithm with comparable running time was obtained in [13].

Table 1 Range Avoidance and Remote Point Algorithms. In the 9-th row, we assert $\mathcal{C}$ is a constant depth unbounded fan-in circuit class which contains AC^0.

Problem	Algorithm	Assumption	Reference
$\textsc{Avoid}[n, n+1]$	$\mathbf{FP}^{\mathbf{NP}}$	$\mathbf{E}^{\mathbf{NP}} \not\subset i.o.\text{-}\mathsf{SIZE}[2^{o(n)}]$	[28]
$\textsc{Avoid}[n, n+1]$	$\mathbf{svFS_2P}$	–	[8, 33]
$\mathsf{NC}^0_k\text{-}\textsc{Avoid}[n, n^{1+\varepsilon}]$	$2^{n^{1-\frac{2\varepsilon}{k-3}+o(1)}}$	–	[16]
$\mathsf{NC}^0_k\text{-}\textsc{Avoid}[n, O_k(n^{(k-1)/2}\log n)]$	$\mathbf{FP}$	–	[16]
$\mathsf{NC}^0_{2t}\text{-}\mathrm{RPP}[n, O_t(n^t \log n), O(1)]$	$\mathbf{FP}$	–	[27]
$\mathsf{NC}^0\text{-}\textsc{Avoid}[n, n^{1+\varepsilon}]$	$\mathbf{FP}^{\mathbf{NP}}$	Assumption 2.19	[36]
$\mathsf{ACC}^0\text{-}\mathrm{RPP}[n, \mathrm{qpoly}(n), 1/\mathrm{poly}(n)]$	$\mathbf{FP}^{\mathbf{NP}}$	–	[8]
$\mathrm{RPP}[n, n^{6+\gamma}, c(O_\gamma(\log n)]$	$\mathbf{FP}^{\mathbf{NP}}$	$\mathbf{E}^{\mathbf{NP}} \not\subset i.o.\text{-}\mathsf{Avg}_{c(n)}\text{-}\mathsf{SIZE}[2^{o(n)}]$	Theorem 1.6
$\mathcal{C}\text{-}\textsc{Avoid}[n, n^{1+\varepsilon}]$	$\mathbf{FP}^{\mathbf{NP}}$	$\mathbf{E}^{\mathbf{NP}} \not\subset i.o.\text{-}\mathcal{C}\text{-}\mathsf{SIZE}[2^{o(n)}]$	Theorem 1.6
$\mathsf{NC}^0_4\text{-}\textsc{Avoid}[n, 2n]$	$\mathbf{FP}^{\mathbf{NP}}$	$\mathbf{E}^{\mathbf{NP}} \not\subset i.o.\text{-}\mathsf{Formula}[o(2^n/n)]$	Theorem 1.4
$\mathsf{NC}^0_k\text{-}\textsc{Avoid}[n, n^{1+\varepsilon}]$	$2^{n^{1-\frac{\varepsilon}{k-1}+o(1)}}$	–	Theorem 1.12
$\mathsf{NC}^0_k\text{-}\textsc{Avoid}[n, n+1]$	$O(n2^{\frac{k-2}{k-1}n})$	–	Theorem 1.14

a) We use $\mathbf{svFS_2P}$ to denote single-valued $\mathbf{FS_2P}$ algorithm.

Using a known connection between NC^0-Avoid and local PRGs, we show that faster Avoid algorithms would contradict plausible cryptographic assumptions.

▶ **Theorem 1.13.** *Suppose Assumption 2.20 is true, there does not exist an algorithm for* NC^0_k*-Avoid running in time* 2^{n^β} *for some constant* $0 < \beta < 1$ *that identifies* $\mathsf{negl}(n)$ *fraction of strings outside the range.*

We also design an improved algorithm for the regime of minimal stretch $m = n + 1$, improving over brute-force search.

▶ **Theorem 1.14.** *There exists a family of* $O(n \cdot 2^{\frac{(k-2)n}{k-1}})$ *time algorithms for* NC^0_k*-Avoid*$[n, n+1]$.

Previous and our algorithmic results are summarized in Table 1. Overall, these results expand the algorithmic landscape for $\mathcal{C}$-Avoid across both small and large stretch regimes, with implications for circuit lower bounds and local PRG security.

1.2 Technical Overview

Equivalence between $\mathcal{C}$-Avoid$[n, n^{1+\varepsilon}] \in \mathbf{FP}^{\mathbf{NP}}$ and $\mathbf{E}^{\mathbf{NP}} \not\subset i.o.\text{-}\mathcal{C}\text{-}\mathsf{SIZE}[2^{o(n)}]$. For a constant depth unbounded fan-in circuit class $\mathcal{C}$, we establish a tight equivalence between the complexity of solving $\mathcal{C}$-Avoid$[n, n^{1+\varepsilon}]$ in $\mathbf{FP}^{\mathbf{NP}}$ and proving exponential lower bounds for $\mathcal{C}$ circuits against $\mathbf{E}^{\mathbf{NP}}$, generalizing the reduction of Jeřábek and Korten [23, 28], who proved that Avoid $\in \mathbf{FP}^{\mathbf{NP}}$ if and only if $\mathbf{E}^{\mathbf{NP}} \not\subset i.o.\text{-}\mathsf{SIZE}[2^{o(n)}]$[12].

[12] This reduction, which we refer to as *Jeřábek -Korten reduction*, was originally proved in the framework of bounded arithmetic by Jeřábek [23], and later translated to the language of computational complexity by Korten [28]. Specifically, as pointed out to us by Erfan Khaniki, [23, Proposition 3.5] proved that the dual weak pigeonhole principle (dwPHP(PV)) is equivalent to the statement asserting the existence of Boolean functions with exponential circuit complexity in Buss' bounded arithmetic theory S^1_2 which captures polynomial time reasoning. An $\mathbf{FP}^{\mathbf{NP}}$ algorithm for Avoid can be extracted from the dual weak pigeonhole principle (i.e., formalization of the totality of Avoid) in S^1_2 via the Witnessing Theorem from [30].

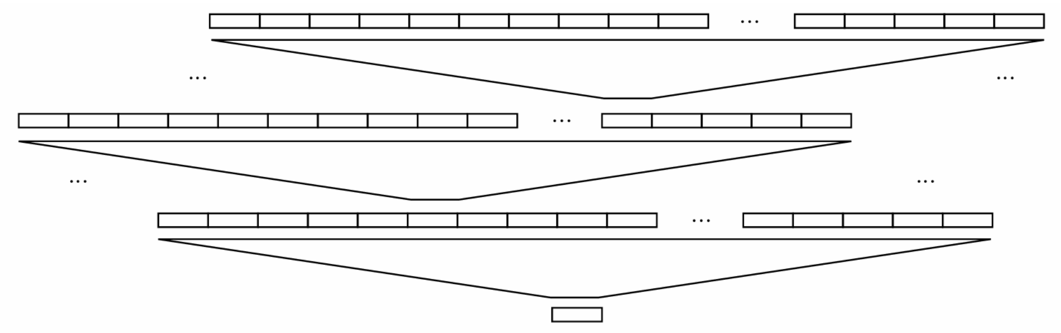

Figure 1 Generalized q-ary GGM-Tree.

The forward direction – namely, that an $\mathbf{FP^{NP}}$ algorithm for $\mathcal{C}$-Avoid implies exponential $\mathcal{C}$ circuit lower bounds against $\mathbf{E^{NP}}$ – was largely established in [36]. A key component of this argument is the *universality property* of the circuit class $\mathcal{C}$: that the truth table generator $\mathsf{TT}_{\mathcal{C}}$ can itself be computed by a circuit in $\mathcal{C}$. We strengthen and formalize this notion, showing that any circuit class $\mathcal{C}$ containing AC^0 satisfies this property. The intuition is that the universal circuit $\mathcal{U}$ acts as a decoder: given an encoding of a circuit C and an input x, it decodes C and evaluates it on x. Since decoding and simple simulation can be implemented in AC^0, this universality follows for all such classes.

The reverse direction, which shows that exponential $\mathcal{C}$ circuit lower bounds for functions in $\mathbf{E^{NP}}$ imply that $\mathcal{C}$-Avoid $\in \mathbf{FP^{NP}}$, proceeds by generalizing Korten's construction based on the GGM-tree. We illustrate the approach in the context of AC^0-Avoid$[n, n^{1+\varepsilon}]$, although the framework extends to the broader $\mathcal{C}$-Remote-Point$[n, n^{1+\varepsilon}]$ problem for any $\mathcal{C}$ containing AC^0.

We first briefly recall the $\mathbf{FP^{NP}}$ reduction from circuit lower bound to Avoid in [23, 28] which we thereafter refer to as *Jeřábek -Korten reduction*. Given an instance of Avoid$[n, 2n]$, which we call C, one constructs a new circuit $\mathsf{GGM}[C]$ by composing C along the nodes of a perfect binary tree of height k (this construction is known as the GGM-tree construction). The resulting circuit has stretch $n \cdot 2^k$, and the output $y \in \text{Range}(\mathsf{GGM}[C])$ can be regarded as encoding the truth table of a function g, whose input is the bits used to select a path in the tree. Importantly, due to redundancy and the tree structure in $\mathsf{GGM}[C]$, this output y can be computed by a relatively small-size circuit at the cost of increasing the depth. Thus, the complexity of the function g – whose truth table is y – can be bounded in terms of the complexity of C and the structure of the GGM-tree.

We generalize this framework in the following three aspects: (1) the fan-out of the tree, denoted by q; (2) the height of the tree, denoted by k; and (3) the circuit C, which we draw from a restricted circuit class $\mathcal{C}$.

Let ℓ denote the stretch of the resulting circuit after composing C through the generalized GGM-tree, which we denote by $\mathsf{GGM}_{\ell,q,k}[C]$ (see Figure 1 for an illustration). It is easy to see that $\ell = n \cdot q^k$. To analyze the complexity of any $y \in \text{Range}(\mathsf{GGM}_{\ell,q,k}[C])$, we associate it with a function $g : \{0,1\}^{\log \ell} \to \{0,1\}$ (corresponding to the structure of the GGM-tree), whose truth table is exactly y.

The circuit computing g can be constructed by composing the circuit C with k layers of multiplexing (selection) and a final indexing operation. These multiplexing and indexing subcircuits can be implemented by $O(n)$-size DNF formulas, and hence belong to any class containing DNF (such as AC^0).

Assuming $C \in \mathsf{AC}^0_d$ where AC^0_d denotes depth d AC^0 circuits, to ensure that $g \in \mathsf{AC}^0$, we must take $k = O(1)$. By setting the fan-out $q = n^{\varepsilon}$, the overall stretch becomes $\ell = n \cdot n^{k\varepsilon} = n^{1+k\varepsilon}$, and the resulting circuit g has size $O(n) + O(|C| \cdot k) = O(n^{1+\varepsilon})$.

Now suppose there exists a function $f \in \mathbf{E}^{\mathbf{NP}}$ that requires AC^0_{dk} circuits of size at least ℓ^{γ} for some constant $\gamma \in (0,1)$. Then for sufficiently large ℓ, f cannot be in the range of $\mathsf{GGM}_{\ell,q,k}[C]$, since all such y have low circuit complexity. Thus, we can use f to find a string not in Range(C) by traversing the GGM-tree with an **NP** oracle backwards. This yields an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for AC^0_d-Avoid$[n, nq]$, completing the reduction.

Altogether, this establishes a precise characterization:

$$\mathcal{C}\text{-}\textsc{Avoid}[n, n^{1+\varepsilon}] \in \mathbf{FP}^{\mathbf{NP}} \iff \mathbf{E}^{\mathbf{NP}} \not\subset i.o.\text{-}\mathcal{C}\text{-}\mathsf{SIZE}[2^{o(n)}]$$

for any $\mathcal{C}$ containing AC^0, and where the stretch satisfies $nq = n^{1+\varepsilon}$ for any arbitrary constant $\varepsilon > 0$.

Equivalence between $\mathsf{RPP}[n, n^{6+\gamma}, c(O_\gamma(\log n))] \in \mathbf{FP}^{\mathbf{NP}}$ and $\mathbf{E}^{\mathbf{NP}} \not\subset i.o.\text{-}\mathsf{Avg}_{c(n)}\text{-}\mathsf{SIZE}[2^{o(n)}]$. We try to extend the GGM-style idea to Remote-Point. Nevertheless, the original Jeřábek-Korten reduction does not work for Remote-Point. Consider the toy case of $\mathsf{GGM}_{4n,2,2}[C]$. Assume that we have an average-case hard truth table y and are not able to find a remote point of C at relative distance ρ by traversing down the tree. Divide y into two blocks y_1, y_2, each of size $2n$. Then there exists $x, x_1, x_2 \in \{0,1\}^n$ such that $C(x_1) \approx_\rho y_1$, $C(x_2) \approx_\rho y_2$, and $C(x) \approx_\rho (x_1 \circ x_2)$, where $C(x_1), C(x_2)$, and $C(x)$ respectively achieve the maximum distance from y_1, y_2, and $x_1 \circ x_2$ among all points in Range(C). However, dividing $C(x)$ into two blocks of equal size x'_1 and x'_2, it is unclear how close $C(x'_1)$ is to $C(x_1)$ and how close $C(x'_2)$ is to $C(x_2)$. In other words, it is hard to argue about the distance between y and $\mathsf{GGM}_{4n,2,2}[C](x)$.

To solve this problem, we use an idea from [8] that reduces Remote-Point to Avoid, and incorporate an error-correcting code at each node of the GGM-tree to prevent the accumulation of errors across levels. To illustrate the core idea, consider first the simpler case of a code with unique decoding. Suppose at each node, we compose the circuit $C : \{0,1\}^n \to \{0,1\}^m$ with a unique decoder $\mathsf{Dec}_{\mathsf{uniq}} : \{0,1\}^m \to \{0,1\}^{qn}$ for a code with decoding radius ρ. If a string $y \in \{0,1\}^{qn}$ is not in Range$(\mathsf{Dec}_{\mathsf{uniq}} \circ C)$, then its encoding $\mathsf{Enc}(y)$ (under the code's natural encoding) is at least ρ-far from Range(C). This property effectively isolates the error at each node: the failure to find a preimage of y under $\mathsf{Dec}_{\mathsf{uniq}} \circ C$ directly implies that y is a remote-point for C, without the error propagating to its children. This allows the reduction to proceed similarly to the Avoid problem, by searching for a preimage on each node of the tree.

However, unique decoding limits us to a radius of $\rho \leq 1/4 - \varepsilon$, which is insufficient for our purposes. In the actual construction, we employ list-decoding to achieve a larger radius of $\rho = 1/2 - \varepsilon$. We use a list-decodable code with a decoder $\mathsf{Dec}_{\mathsf{list}} : \{0,1\}^m \to (\{0,1\}^{qn})^L$. At each node, applying $\mathsf{Dec}_{\mathsf{list}} \circ C$ produces a list of candidate values. We then apply a padding method to pad both the input and the output of C with extra $\log L$ bits. This enables us to select one candidate from this list to pass to the next level of the tree.

Hence we get a conditional $\mathbf{FP}^{\mathbf{NP}}$ for Remote-Point. Combining with a refinement of the result in [36] (Theorem 2.8), it yields an equivalence between the $\mathbf{FP}^{\mathbf{NP}}$ algorithm for Remote-Point and the average-case circuit lower bound for $\mathbf{E}^{\mathbf{NP}}$.

Subexponential time NC^0-Avoid algorithm for any superlinear stretch. We present the first subexponential-time algorithm for NC^0_k-Avoid$[n, n^{1+\varepsilon}]$, achieving runtime $2^{n^{1-\frac{\varepsilon}{k-1}+o(1)}}$ for any $\varepsilon > 0$. Our approach exploits structural limitations of local circuits in terms of their associated bipartite graphs to identify small subcircuits with poor expansion, enabling targeted enumeration over their input-output behavior.

The algorithm is based on the following high-level idea: every $\mathsf{NC}^0_k[n, n^{1+\varepsilon}]$ circuit corresponds to a degree-k left-regular bipartite graph with n right vertices (inputs) and $m = n^{1+\varepsilon}$ left vertices (outputs). Using standard probabilistic methods, one can show that a random left-regular bipartite graph with degree k, n right vertices and $m(n) = n^{1+\varepsilon}$ left vertices is a $(K = o(n), A = 1 - o(1))$ vertex expander – meaning that for every subset of left vertices of size $\leq K$, it has $\geq KA$ neighbors. One would expect these probabilistic arguments to be actually tight. Assuming so, we would be able to find a Hall-violating subsets (i.e., a subset of outputs whose neighbors have size smaller than the subset of outputs) in any such graphs.

Luckily, the lower bound results on disperser graphs from [35] can be adapted to argue that such graphs necessarily contain Hall-violating subsets of outputs of size at most $K = n^{1-\frac{\varepsilon}{k-1}+o(1)}$. This means that every such circuit contains a subcircuit of size K that maps a subset of inputs to outputs non-surjectively.

Our algorithm proceeds by brute-force search for such Hall-violating subsets $S \subseteq [m]$ of size K. Once a violating subset is found, we isolate the corresponding subcircuit C' of size K, and enumerate all strings in $\{0,1\}^{|\Gamma(S)|}$ to find those not in the image of C'. We then lift these local non-image strings to full-length output strings by assigning arbitrary values outside of S, yielding many globally valid strings not in the image of the full circuit C.

This gives the following guarantee: for every $\mathsf{NC}^0_k[n, n^{1+\varepsilon}]$ circuit, we can find (and succinctly represent) at least $2^{n^{1+\varepsilon}-1}$ strings outside the range of the circuit in time

$$O(2^{\binom{m}{K}}) = 2^{n^{1-\frac{\varepsilon}{k-1}+o(1)}}.$$

Under a conjectured tight bound on bipartite dispersers, we further refine this analysis to show that even smaller Hall-violating subsets exist, yielding improved runtimes of $2^{n^{1-\frac{\varepsilon}{k-2}+o(1)}}$. Notably, this leads to *polynomial-time* algorithms for NC^0_k-Avoid in stretch regimes as low as $m = n^{k-1}/\log^{k-2} n$, improving a prior work [13] which required larger stretch.

Finally, we connect our algorithmic result to pseudorandomness. We show that any subexponential-time Avoid algorithm capable of identifying a non-negligible fraction of non-image strings for NC^0_k circuits contradicts the existence of secure NC^0_k-based pseudorandom generators (PRGs) against subexponential-time adversary. In particular, under standard assumptions about local PRGs, our algorithm demonstrates that no such PRG with stretch $n^{1+\varepsilon}$ can be secure against 2^{n^γ}-time distinguishers for any $\gamma \geq 1 - \frac{\varepsilon}{k-1} + o(1)$, even with constant distinguishing advantage.

Improvement over brute-force for NC^0_k-Avoid$[n, n+1]$. We design a greedy, local algorithm for solving NC^0_k-Avoid$[n, n+1]$ that proceeds by iteratively fixing output bits to values that provably shrink the preimage space of the circuit. At each step, the algorithm selects an unfixed output bit y_i and assigns it a value such that the number of inputs consistent with all fixed output values decreases by at least a factor of $1/2$. This ensures that after at most $n+1$ such assignments, the preimage space collapses to an empty set, yielding a string outside the image of the circuit.

The core technical challenge lies in bounding the "decision space", i.e., the portion of the input space that must be explored to determine the effect of fixing an output bit. We analyze this by modeling the NC^0_k circuit as a bipartite dependency graph between input and output bits, and we introduce the notion of the *traversed space*: the subset of input variables affected by the fixed output bits. We show that after fixing t output bits, the maximum size of any connected component (i.e., subspace) in the traversed space is bounded by $2^{(k-2)t+1}$. This follows from structural properties of bounded-locality circuits and a case-based inductive argument.

Combining this with the observation that fixing each output bit reduces the entropy of the input space by one, we find that the decision space remains small as long as $t \leq n/(k-1)$. In particular, the algorithm only needs to examine subspaces of size at most

$$2^{(k-2)n/(k-1)},$$

leading to a total runtime of $O(n \cdot 2^{(k-2)n/(k-1)})$. Notably, when $k = 2$, the runtime becomes linear, reproducing the result of [15]. For larger k, this provides a non-trivial improvement over brute force.

We also show a matching lower bound for this greedy strategy: under mild assumptions on the structure of random NC^0_k circuits (specifically, that they form good bipartite vertex expanders), any such greedy algorithm necessarily explores an exponential-sized decision space in the worst case. This demonstrates that while the algorithm performs well for $k = 2$, solving NC^0_k-Avoid efficiently in the general case may require fundamentally different techniques.

1.3 Subsequent Work

Subsequent to our work, Guruswami, Lyu, and Yuan [16] presented an **FP** algorithm for NC^0_k-Avoid$[n, O_k(n^{(k-1)/2} \log n)]$, which now represents the state-of-the-art polynomial-time algorithm for NC^0-Avoid. They also obtained a $2^{,n^{1-\frac{2\varepsilon}{k-3}+o(1)}}$-time algorithm for NC^0_k-Avoid$[n, n^{1+\varepsilon}]$, offering a slight improvement over our subexponential-time algorithm. The rest of our results remain orthogonal to their work.

2 Preliminaries

2.1 Notations

We use $\mathcal{C}$ to denote a circuit class, e.g., $\mathsf{NC}^0, \mathsf{AC}^0, \mathsf{ACC}^0, \mathsf{TC}^0$, etc. We use $\mathcal{C}[n, m(n)]$ to denote $\mathcal{C}$ with input length n and output length $m(n)$. We use $\mathcal{C}_1 \circ \mathcal{C}_2$ to denote the composition of circuits from $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively. We use $\mathcal{C}_{n,s,d}$ to denote all the single-output $\mathcal{C}$ circuit of input length n, size s, and depth d. We use $\mathcal{C}$-Avoid$[n, m(n)]$ to denote $\mathcal{C}$-Avoid problem where the circuit $\mathcal{C}$ has input length n and output length $m(n)$. We call $m(n)$ the *stretch* of the $\mathcal{C}$-Avoid problem.

Given a circuit $C : \{0,1\}^n \to \{0,1\}^m$ where $m > n$. For a partial assignment of an m-bit string y, we use $y \notin \mathrm{Range}(C)$ to denote that any assignment consistent with y is not in the range of the circuit C.

We use $\leq_{\mathbf{FP}}$ (resp. $\leq_{\mathbf{FP}^{\mathbf{NP}}}$) to denote reduction in **FP** (resp. $\mathbf{FP}^{\mathbf{NP}}$).

For two strings $x, y \in \{0,1\}^N$, define the *relative Hamming Distance* to be the fraction of indices where x and y differ, formally $\delta(x,y) := \frac{1}{N}|\{i \in [N] : x_i \neq y_i\}|$. For a string $x \in \{0,1\}^N$ and a subset $S \subset \{0,1\}^N$, we say that x is ρ-close/far to S iff $\min_{y \in S} \delta(x,y) \leq \rho / \min_{y \in S} \delta(x,y) > \rho$. When $S = \{y\}$, we also say that x is ρ-close/far to y.

We use PRGs to denote pseudorandom generators. We use $\mathsf{Bip}_{n,m,D}$ to be the set of bipartite multigraphs that have m left vertices and n right vertices where $m \geq n+1$ and are D-left regular. We often use capital letters for random variables and corresponding small letters for their instantiations. Let s be an integer, $\{V_1, V_2, \cdots, V_s\}$ be a set of random variables. We use $V_{[s]}$ to denote the subset $\{V_1, \cdots, V_s\}$. For any strings y_1 and y_2, let $y_1 \circ y_2$ denote the concatenation of y_1 and y_2. Let F_2 denote the binary field.

We will adopt 0-index, e.g., the first bit of s string s is s_0, the first child of a parent in a tree is its 0-th child, etc. The height of a tree is referred to as the number of edges in the longest path from the root node to any leaf node.

2.2 Formulas, NC Circuits and AC Circuits

We use standard definitions of circuit complexity classes. A Boolean circuit is a directed acyclic graph composed of logic gates with bounded fan-in (e.g., $\wedge$, $\vee$, $\neg$) computing functions over $\{0,1\}$. A family of circuits $\{C_n\}_{n\in\mathbb{N}}$ is said to compute a function $f : \{0,1\}^* \to \{0,1\}^*$ if, for every input length n, the circuit C_n correctly computes f on inputs of length n. We use the size s of a circuit as its number of gates plus the length of output, and the depth d to denote the length of the longest path between input bits and output bits.

A formula is a specific type of circuit where the fan-out of every gate is restricted to *exactly one*. This means the output of each gate can be used as the input to *at most one* other gate, *or* it may serve as *exactly one* bit of the output.

▶ **Definition 2.1** (NC circuits [13]). *The circuit class NC^i contains multi-output Boolean circuits on n inputs of depth $O(\log^i n)$ where each gate has fan-in 2. We are particularly concerned with the following classes of circuits:*

- *For every constant $k \geq 1$, NC^0_k is the class of circuits where each output depends on at most k inputs.*
- *NC^1 is the class of circuits of depth $O(\log n)$ where all gates have fan-in 2.*
- *Linear NC^1 circuits are circuits of depth $O(\log n)$ where every gate has fan-in 2 and computes an affine function, i.e., the XOR of its two inputs or its negation.*

Proving a super-linear circuit lower bound on the size of arithmetic computing an n-output function from $\mathbf{FP}$ or even $\mathbf{FE}^{\mathbf{NP}}$ [13, 39, 1, Frontier 3] is a decades-old challenge. Valiant [39] introduced the problem of explicitly constructing rigid matrices and showed that this would prove super-linear lower bounds on the size of (linear) NC^1 circuits.

▶ **Definition 2.2** (AC Circuits). *We denote by AC^i the class of Boolean functions computable by a family of circuits of:*

- *polynomial size,*
- *depth $O(\log^i n)$,*
- *unbounded fan-in $\wedge$ and $\vee$ gates,*
- *and $\neg$ gates allowed only at the input level and are not counted into the depth.*

We say a function f is in AC^i if it is computed by a family of AC^i circuits. The class AC is defined as the union $\mathsf{AC} = \bigcup_{i\geq 0} \mathsf{AC}^i$.

We use the notation AC^i_d to denote the family of AC^i circuits with depth at most d.

More generally, an AC^i-circuit of size $s(n)$, where $s(n)$ may be super-polynomial of n, is defined identically to an AC^i circuit but relaxing the size restriction from polynomial to $s(n)$.

For a *correlation* factor $2\gamma > 0$, we say that a circuit $C : \{0,1\}^n \to \{0,1\}$ $(1/2+\gamma)$-approximates a function $f : \{0,1\}^n \to \{0,1\}$ if $C(x) = f(x)$ for $(1/2+\gamma)$ fraction of inputs from $\{0,1\}^n$. Let $N := 2^n$, and the truth table of C be $\mathsf{TT}_C \in \{0,1\}^N$, the truth table of f be $\mathsf{TT}_f \in \{0,1\}^N$. Then the above is equivalent to $\delta(\mathsf{TT}_C, \mathsf{TT}_f) < (1/2-\gamma)$.

For a function $f : \{0,1\}^n \to \{0,1\}$, we define $\mathsf{SIZE}(f)$ to be the minimum size of a circuit computing f exactly. Similarly, for $\gamma > 0$, we define $\mathsf{Avg}_\gamma\text{-}\mathsf{SIZE}(f)$ to be the minimum size of a circuit that $(1/2 + \gamma)$-approximates f.

We use $\mathsf{SIZE}[s(n)]$ to denote the set of functions with boolean circuit complexity $s(n)$. We use $\mathcal{C}\text{-}\mathsf{SIZE}[s(n)]$ to denote the set of functions with $\mathcal{C}$ circuit complexity $s(n)$. We use $\mathsf{Avg}_\gamma\text{-}\mathcal{C}\text{-}\mathsf{SIZE}[s(n)]$ to denote the set of functions that can be $(1/2+\gamma)$-approximated by $\mathcal{C}$ with circuit complexity $s(n)$.

We use $\mathsf{Formula}[s(n)]$ to denote the set of functions that can be computed by size-$s(n)$ boolean formulas.

▶ **Definition 2.3** (($\mathcal{C}$) Circuit Complexity of a String)**.** *Given a bit string $s \in \{0,1\}^n$, we define the ($\mathcal{C}$) circuit complexity of s to be the smallest ($\mathcal{C}$) circuit whose truth table agrees with s for the first n indices. In particular, the formula complexity of s to be the smallest formula whose truth table agrees with s for the first n indices.*

2.3 Universality Property and Truth Table Generator

▶ **Definition 2.4** (Universality Property [36])**.** *Let $\mathcal{C}$ be a circuit class. We say that $\mathcal{C}$ has the universality property if there is a constant $c \geq 1$ such that for any good function $s : \mathbb{N} \to \mathbb{N}$, there is a sequence of $\mathcal{C}$ circuits $\{U_{s,n}\}_{n\in\mathbb{N}}$ such that the following are true:*

- *The size of $U_{s,n}$ is $s(n)^c$ and it has $O(s \log s + n)$ variables.*
- *Given an input $(\langle C \rangle, x)$, where $\langle C \rangle$ is the encoding of a $\mathcal{C}$ circuit C of size s on n variables, and $x \in \{0,1\}^n$, it accepts the input iff C accepts x.*
- *The family $U_{s,n}$ is uniform: there is a Turing machine that on input $(1^s, 1^n)$, outputs the description of $U_{s,n}$ in polynomial time.*

▶ **Theorem 2.5** ([11])**.** *The class AC^0 has universality property.*

▶ **Theorem 2.6** ([2])**.** *The class NC^1 has universality property.*

In effect, any circuit class containing AC^0 has the universality property. The readers are referred to Appendix A of the full version for a proof.

▶ **Definition 2.7** (Truth Table Generator)**.** *Let $\mathsf{TT} : \{0,1\}^{O(s\log s)} \to \{0,1\}^{2^n}$ be the circuit that takes as input the description of a size-s circuit on n variables, and outputs the truth table of this circuit. Here TT denotes* truth table*. Define $\mathsf{TT}_\mathcal{C} : \{0,1\}^{O(s\log s)} \to \{0,1\}^{2^n}$ to be the circuit that takes as input the description of a size s $\mathcal{C}$ circuit on n variables, and outputs the truth table of this $\mathcal{C}$ circuit. It is clear that if $\mathcal{C}$ has universality property, then $\mathsf{TT}_\mathcal{C} \in \mathcal{C}$.*

The following modified Theorem says that solving $\mathcal{C}$-Remote-Point on $\mathsf{TT}_\mathcal{C}$ implies $\mathcal{C}$ circuit lower bounds with tight parameters (see Appendix C of the full version for a proof).

▶ **Theorem 2.8** (Modified Theorem 5.2 of [36])**.** *Let $\mathcal{C}$ be any circuit class that has the universality property, and $c, f : \mathbb{N} \to \mathbb{N}$ be monotone functions that are good. Suppose there is an $\mathbf{FP}^{\mathbf{NP}}$ (resp. $\mathbf{FP}$, $\mathbf{FQP}^{\mathbf{NP}}$) algorithm for $\mathcal{C}$-Remote-Point$[N, f(N), c(N)]$, where each output gate has $\mathcal{C}$ circuit complexity $\mathrm{poly}(N)$. Then for some constant $\varepsilon > 0$, $\mathbf{E}^{\mathbf{NP}}$ (resp. $\mathbf{E}$, $\mathbf{EXP}^{\mathbf{NP}}$) cannot be $(1/2 + c(f^{-1}(2^n)))$ approximated by $\mathcal{C}$ circuits of size $\frac{\varepsilon f^{-1}(2^n)}{\log f^{-1}(2^n)}$.*

2.4 Error-correcting Code

Here we will quickly review the basic concepts from coding theory that will be needed for this work. A binary code $\mathcal{C}$ of block length n' is a subset of $\{0,1\}^{n'}$. We use $n = \log|\mathcal{C}|$ to denote the message length of $\mathcal{C}$, and the rate of $\mathcal{C}$ equals n/n'. Each string in $\mathcal{C}$ is called a codeword. The distance of $\mathcal{C}$ is defined as $\min_{x \neq x'} \delta(x, x')$ where $x, x' \in \mathcal{C}$.

A list decoding algorithm for a binary code $\mathcal{C}$ of block length n' needs to do the following. Given an error parameter $0 \leq \rho < 1$ and a received word $y \in \{0,1\}^{n'}$ the decoder needs to output all codewords $c \in \mathcal{C}$ such that $\delta(c, y) \leq \rho$. We say that a code $\mathcal{C}$ of block length n' is (ρ, L)-list-decodable, if for every such y, there are at most L codewords which satisfy $\delta(c, y) \leq \rho$.

▶ **Definition 2.9** ((n, n', ρ, L)-code)**.** *For a binary code $\mathcal{C}$ of block length n' and message length n, an encoding function for $\mathcal{C}$ is a bijection* $\mathsf{Enc} : \{0,1\}^n \to \mathcal{C}$ *(assume w.l.o.g that n is an integer), which can also be extended as an injection from $\{0,1\}^n$ to $\{0,1\}^{n'}$. Since $\mathcal{C}$ and* Enc *are essentially the same object, we will use* Enc *to refer to $\mathcal{C}$.*

Suppose that Enc *is (ρ, L)-list-decodable, and use* $\mathsf{Dec} : \{0,1\}^{n'} \to (\{0,1\}^n)^L$ *to denote the list decoding algorithm for it. Then we call that* $(\mathsf{Enc}, \mathsf{Dec})$ *is a (n, n', ρ, L)-code, which means that* Enc *has message length n and block length n', as well as its list decoding algorithm* Dec*.*

We often need to select a specific block of the list decoded from the codeword. So we define the following notation:

▶ **Definition 2.10** (Selector of list-decoding)**.** *For a (n, n', ρ, L)-code* $(\mathsf{Enc}, \mathsf{Dec})$*, its selector* $\mathsf{Sel}_{\mathsf{Dec}} : \{0,1\}^{n'} \times [L] \to \{0,1\}^n$ *outputs the z-th block of* $\mathsf{Dec}(w)$ *over the input $w \in \{0,1\}^{n'}$ and $z \in [L]$. W.l.o.g, assume that $\log L$ is an integer, and we also view the input domain as $\{0,1\}^{n'+\log L}$ where the first n' bits form the codeword, and the remaining $\log L$ bits represent an integer in $[L]$.*

The classic Johnson bound [24] implies that *non-explicitly* a binary code of relative distance $1/2 - \varepsilon^2$ is $(1/2 - \varepsilon, 1/\varepsilon^2)$-list-decodable. When we require that both the encoding and list-decoding algorithms run efficiently, Guruswami and Rudra [17, 14] showed that:

▶ **Theorem 2.11** (Theorem 13 of [17])**.** *Given an integer $n > 1$ and reals $\gamma > 0$ and $0 < \varepsilon < 1/2$, there exists an explicit binary code* Enc *with message length n and block length at most $(1/\gamma)^{O(1)} \cdot (n^3/\varepsilon^{3+\gamma})$, which is $\left(\frac{1}{2} - \varepsilon, \left(\frac{1}{\gamma\varepsilon}\right)^{O(1/\gamma)}\right)$-list-decodable and the list decoding algorithm* Dec *runs in time $\left(\frac{n}{\gamma\varepsilon}\right)^{O(1/\gamma)}$.*

Specifically, there exists a $\big(n, O(n^{3(c+1)+\gamma}), 1/2 - n^{-c}, \mathrm{poly}(n)\big)$-code $(\mathsf{Enc}, \mathsf{Dec})$ *for any constant $c, \gamma > 0$, where both* Enc *and* Dec *run in $\mathrm{poly}(n)$ time.*

2.5 Bipartite Vertex Expander

▶ **Definition 2.12** (Vertex expander [38])**.** *A digraph G is a (K, A) vertex expander if for all sets S of at most K vertices, the neighborhood $N(S) = \{u : \exists v \in S \text{ s.t. } (u, v) \in E\}$ is of size at least $A \cdot |S|$.*

▶ **Definition 2.13** (Left regular bipartite graphs [38])**.** *Let $\mathsf{Bip}_{n,m,D}$ be the set of bipartite multigraphs that have m left vertices and n right vertices where $m \geq n + 1$ and are D-left-regular, meaning that every vertex on the left has D neighbors, but vertices on the right may have varying degrees.*

We use (K, A)-$\mathsf{Bip}_{n,m,D}$ to denote $G \in \mathsf{Bip}_{n,m,D}$ that are also (K, A) vertex expander. The following Theorem 2.14 and Theorem 2.15 are modified from [38].

▶ **Theorem 2.14** (Existence of $(\Omega(n), D-1-\varepsilon)$-$\mathsf{Bip}_{n,m,D}$)**.** *For every constant D, $0 < \varepsilon < 1$, there exists a constant $\alpha > 0$ such that for all n, $m = O(n)$, a uniformly random graph from $\mathsf{Bip}_{n,m,D}$ is an $(\alpha n, D-1-\varepsilon)$ vertex expander with probability at least $1/2$.*

▶ **Theorem 2.15** (Existence of $(o(n), 1)$-$\mathsf{Bip}_{n,m,D}$)**.** *For every constant D and every $0 < \beta < 1$, there exists a function $A = n^{1-\beta/(D-2)}$ such that for all n, and $m = n^{1+\beta}$, a uniformly random graph from $\mathsf{Bip}_{n,m,D}$ is an $(A, 1)$ vertex expander with probability at least $1/2$.*

The following definition of Hall-violating set stems from Hall's matching theorem.

▶ **Definition 2.16** (Hall-violating set)**.** *In a bipartite graph G with bipartite classes L and R, a set $H \subseteq L$ is a Hall-violating set if $|N(H)| < |H|$.*

Disperser graphs are special cases of bipartite expanders.

▶ **Definition 2.17** (Disperser graphs [37, 10])**.** *A bipartite graph $G = (V_1 = [N], V_2 = [M], E)$ is a (K, ε)-disperser graph, if for every $X \subseteq V_1$ of cardinality K, $|\Gamma(X)| > (1-\varepsilon)M$ (i.e., every large enough set in V_1 misses less than an ε fraction of the vertices of V_2). The size of G is $|E(G)|$.*

The following theorem gives necessary conditions for G to be a disperser.

▶ **Theorem 2.18** (Lower bounds for disperser graphs [35])**.** *Let $G = (V_1 = [N], V_2 = [M], E)$ be a (K, ε)-disperser. Denote by $\bar{D}$ the average degree of a vertex in V_1.*

1. *Assume that $K < N$ and $\lceil \bar{D} \rceil \leq \frac{(1-\varepsilon)M}{2}$ (i.e., G is not trivial). If $\frac{1}{M} \leq \varepsilon \leq \frac{1}{2}$, then $\bar{D} = \Omega(\frac{1}{\varepsilon} \cdot \log \frac{N}{K})$, and if $\varepsilon > \frac{1}{2}$, then $\bar{D} = \Omega(\frac{1}{\log(1/(1-\varepsilon))} \cdot \log \frac{N}{K})$.*
2. *Assume that $K \leq \frac{N}{2}$ and $\bar{D} \leq \frac{M}{4}$. Then, $\frac{\bar{D}K}{M} = \Omega(\log \frac{1}{\varepsilon})$.*

2.6 Local Algorithms

A local algorithm for Avoid problems probes very few bits to determine any particular output bit of the string out of the range. A local algorithm for a related problem Missing-String was proposed in [40].

2.7 Some Assumptions

▶ **Assumption 2.19** ([36])**.** *For every constants $k \geq 1$ and $\varepsilon > 0$, there is an $\mathbf{FP^{NP}}$ algorithm that given any k-uniform directed hypergraph G and any predicate $P : \{0,1\}^k \to \{0,1\}$, outputs a P-sparsifier of G with error $\varepsilon = 0.5$ using $\tilde{O}(n)$ hyperedges.*

▶ **Assumption 2.20** ([22])**.** *There exists a boolean function $G : \{0,1\}^n \to \{0,1\}^m$ where $m = n^{1+\tau}$ for some constant $\tau > 0$, and where each output bit computed by G depends on a constant number of input bits, such that the following computational indistinguishability holds:*

$$\{G(\sigma) \mid \sigma \leftarrow \{0,1\}^n\} \approx_c \{y \mid y \leftarrow \{0,1\}^m\}$$

The subexponential security of PRG requires the above indistinguishability to hold for adversaries of size 2^{n^β} for some constant $\beta > 0$, with negligible distinguishing advantage.

3 Conclusion and Open Problems

Open Problem 1.

- **(Hardness)** Improve the stretch for the hardness of NC^0-Avoid problem: by [9], we know that $\mathsf{NC}^1\text{-Avoid}[n, n+1] \notin \mathbf{SearchNP}$. Under randomized encoding techniques [36], this also implies that $\mathsf{NC}^0_4\text{-Avoid}[n, n+1] \notin \mathbf{SearchNP}$. Can we prove that under plausible assumptions $\mathsf{NC}^0\text{-Avoid}[n, O(n)] \notin \mathbf{SearchNP}$, or even for some small constant ε, NC^0_k-Avoid$[n, n^{1+\varepsilon}] \notin \mathbf{SearchNP}$ when k is large.
- **(Algorithms)** In the work, we show that there is a $2^{n^{1-\frac{\varepsilon}{k-1}+o(1)}}$ time algorithm for $\mathsf{NC}^0_k\text{-Avoid}[n, n^{1+\varepsilon}]$. Does there exist a $2^{n^{o(1)}}$ time algorithm for $\mathsf{NC}^0_k\text{-Avoid}[n, n^{1+\varepsilon}]$ for some $\varepsilon > 0$? If so, then assuming ETH (Exponential Time Hypothesis) [19, 20], $\mathsf{NC}^0_k\text{-Avoid}[n, n^{1+\varepsilon}] \in \mathbf{SearchNP}$.

Open Problem 2. In this work, we only prove equivalence results for polynomial stretch. Can we extend such equivalence to quasipolynomial stretch? Ideally, we would be able to prove the following conjecture.

▶ **Conjecture 3.1.** *$\exists\delta$ s.t., $\mathbf{E}^{\mathbf{NP}}$ requires 2^{n^δ} size ACC^0 circuit complexity if and only if there is an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for AC^0-Avoid$[n, \mathrm{qpoly}(n)]$, where each output bit is computed by a $\mathrm{qpoly}(n)$ size ACC^0 circuit.*

Assuming Conjecture 3.1 is true and leveraging on existing ACC^0 circuit lower bound against $\mathbf{E}^{\mathbf{NP}}$ [41, 6], the reduction directly yields an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for ACC^0-Avoid$[n, \mathrm{qpoly}(n)]$ where each output bit is computed by a $\mathrm{qpoly}(n)$-size ACC^0 circuit.

We remark that the technique in this paper seems to fall short of achieving this, as to condense a hard function of large quasi-polynomial stretch using Jeřábek -Korten's reduction, one seems to need the depth of the tree to be super-constant.

Open Problem 3. Recall that [23, 28, 5] proved the following equivalence result.

$$\text{Avoid} \in \mathbf{FP}^{\mathbf{NP}} \iff \mathbf{E}^{\mathbf{NP}} \not\subset i.o.\text{-}\mathsf{SIZE}[2^{o(n)}] \iff \mathbf{E}^{\mathbf{NP}} \not\subset i.o.\text{-}\mathsf{SIZE}[2^n/n].$$

The second equivalence is a hardness amplification result.

1. Is there such a similar amplification result for restricted circuit classes? Given Theorem 1.6 and that AC^0-Avoid algorithm for smaller stretch implies stronger lower bounds according to Theorem 2.8, the answer could be negative.
2. Is there such an average-case to average-case hardness amplification pheonomemon, possibly by proving reduction between different instances of Remote-Point? It is unclear how to generalize the $\mathbf{FP}^{\mathbf{NP}}$ reduction of Avoid from any polynomial stretch to minimal stretch to Remote-Point.

References

1 Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, USA, 1st edition, 2009.

2 S. R. Buss. The boolean formula value problem is in alogtime. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, pages 123–131, New York, NY, USA, 1987. Association for Computing Machinery. doi:10.1145/28395.28409.

3 Eshan Chattopadhyay and Jyun-Jie Liao. Hardness against linear branching programs and more. In *38th Computational Complexity Conference (CCC 2023)*, Leibniz International Proceedings in Informatics (LIPIcs), pages 9:1–9:27. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.CCC.2023.9.

4 Lijie Chen. Nondeterministic quasi-polynomial time is average-case hard for ACC circuits. *SIAM Journal on Computing*, 0(0):FOCS19–332–FOCS19–397, 2024. doi:10.1137/20M1321231.

5 Lijie Chen, Shuichi Hirahara, and Hanlin Ren. Symmetric exponential time requires near-maximum circuit size. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, pages 1990–1999, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3618260.3649624.

6 Lijie Chen, Xin Lyu, and R. Ryan Williams. Almost-Everywhere Circuit Lower Bounds from Non-Trivial Derandomization . In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1–12, Los Alamitos, CA, USA, November 2020. IEEE Computer Society. doi:10.1109/FOCS46700.2020.00009.

7 Lijie Chen and Hanlin Ren. Strong average-case circuit lower bounds from nontrivial derandomization. *SIAM Journal on Computing*, 51(3):STOC20–115–STOC20–173, 2022. doi:10.1137/20M1364886.

8 Yeyuan Chen, Yizhi Huang, Jiatu Li, and Hanlin Ren. Range avoidance, remote point, and hard partial truth table via satisfying-pairs algorithms. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, pages 1058–1066, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3564246.3585147.

9 Yilei Chen and Jiatu Li. Hardness of range avoidance and remote point for restricted circuits via cryptography. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, pages 620–629, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3618260.3649602.

10 A. Cohen and A. Wigderson. Dispersers, deterministic amplification, and weak random sources. In *30th Annual Symposium on Foundations of Computer Science*, pages 14–19, 1989. doi:10.1109/SFCS.1989.63449.

11 Stephen A. Cook and H. James Hoover. A depth-universal circuit. *SIAM Journal on Computing*, 14(4):833–839, 1985. doi:10.1137/0214058.

12 Paul Erdös. Some remarks on the theory of graphs. *Bulletin of the American Mathematical Society*, 53:292–294, 1947. URL: https://api.semanticscholar.org/CorpusID:14215209.

13 Karthik Gajulapalli, Alexander Golovnev, Satyajeet Nagargoje, and Sidhant Saraogi. Range avoidance for constant depth circuits: Hardness and algorithms. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2023, September 11-13, 2023, Atlanta, Georgia, USA*, volume 275 of *LIPIcs*, pages 65:1–65:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.APPROX/RANDOM.2023.65.

14 Venkatesan Guruswami. List decoding of binary codes–a brief survey of some recent results. In *International Conference on Coding and Cryptology*, pages 97–106. Springer, 2009. doi:10.1007/978-3-642-01877-0_10.

15 Venkatesan Guruswami, Xin Lyu, and Xiuhan Wang. Range Avoidance for Low-Depth Circuits and Connections to Pseudorandomness. In Amit Chakrabarti and Chaitanya Swamy, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2022)*, volume 245 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:21, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.APPROX/RANDOM.2022.20.

16 Venkatesan Guruswami, Xin Lyu, and Weiqiang Yuan. Cell-probe lower bounds via semi-random csp refutation: Simplified and the odd-locality case. *arXiv preprint arXiv:2507.22265*, 2025. doi:10.48550/arXiv.2507.22265.

17 Venkatesan Guruswami and Atri Rudra. Soft decoding, dual bch codes, and better list-decodable e-biased codes. In *2008 23rd Annual IEEE Conference on Computational Complexity*, pages 163–174, 2008. doi:10.1109/CCC.2008.13.

18 Rahul Ilango, Jiatu Li, and R. Ryan Williams. Indistinguishability obfuscation, range avoidance, and bounded arithmetic. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, pages 1076–1089, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3564246.3585187.

19 R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? In *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No.98CB36280)*, pages 653–662, 1998. doi:10.1109/SFCS.1998.743516.

20 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. doi:10.1006/JCSS.2000.1727.

21 Russell Impagliazzo and Avi Wigderson. P = bpp if e requires exponential circuits: derandomizing the xor lemma. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, STOC '97, pages 220–229, New York, NY, USA, 1997. Association for Computing Machinery. doi:10.1145/258533.258590.

22 Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, pages 60–73, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3406325.3451093.

23 Emil Jeřábek. Dual weak pigeonhole principle, boolean complexity, and derandomization. *Annals of Pure and Applied Logic*, 129(1):1–37, 2004. doi:10.1016/j.apal.2003.12.003.

24 S. Johnson. A new upper bound for error-correcting codes. *IRE Transactions on Information Theory*, 8(3):203–207, 1962. doi:10.1109/TIT.1962.1057714.

25 Robert Kleinberg, Oliver Korten, Daniel Mitropolsky, and Christos Papadimitriou. Total Functions in the Polynomial Hierarchy. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 44:1–44:18, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2021.44.

26 Adam R. Klivans and Dieter van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM Journal on Computing*, 31(5):1501–1526, 2002. doi:10.1137/S0097539700389652.

27 O. Korten, T. Pitassi, and R. Impagliazzo. Stronger cell probe lower bounds via local prgs. In *Electron. Colloquium Comput. Complex.*, 2025.

28 Oliver Korten. The hardest explicit construction. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 433–444, Los Alamitos, CA, USA, 2021. IEEE Computer Society. doi:10.1109/FOCS52979.2021.00051.

29 Oliver Korten. Range avoidance and the complexity of explicit constructions. *The Computational Complexity Column by Michal Koucky, Bulletin of the European Association for Theoretical Computer Science*, 145:94–134, 2025. URL: http://eatcs.org/beatcs/index.php/beatcs/article/view/825.

30 Jan Krajíček. No counter-example interpretation and interactive computation. In Yiannis N. Moschovakis, editor, *Logic from Computer Science*, pages 287–293, New York, NY, 1992. Springer New York.

31 N. Kuntewar and J. Sarma. Range avoidance in boolean circuits via turan-type bounds. In *Electron. Colloquium Comput. Complex.*, 2025.

32 Xin Li and Yan Zhong. Explicit Directional Affine Extractors and Improved Hardness for Linear Branching Programs. In Rahul Santhanam, editor, *39th Computational Complexity Conference (CCC 2024)*, volume 300 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:14, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2024.10.

33 Zeyong Li. Symmetric exponential time requires near-maximum circuit size: Simplified, truly uniform. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, pages 2000–2007, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3618260.3649615.

34 Peter Bro Miltersen, N. V. Vinodchandran, and Osamu Watanabe. Super-polynomial versus half-exponential circuit size in the exponential hierarchy. In *Proceedings of the 5th Annual International Conference on Computing and Combinatorics*, COCOON'99, pages 210–220, Berlin, Heidelberg, 1999. Springer-Verlag. doi:10.1007/3-540-48686-0_21.

35 Jaikumar Radhakrishnan and Amnon Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM Journal on Discrete Mathematics*, 13(1):2–24, 2000. doi:10.1137/S0895480197329508.

36 Hanlin Ren, Rahul Santhanam, and Zhikun Wang. On the range avoidance problem for circuits. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 640–650, 2022. doi:10.1109/FOCS54457.2022.00067.

37 M Sipser. Expanders, randomness, or time versus space. In *Proc. of the Conference on Structure in Complexity Theory*, pages 325–329, Berlin, Heidelberg, 1986. Springer-Verlag.

38 Salil P. Vadhan. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012. doi:10.1561/0400000010.

39 Leslie G Valiant. Graph-theoretic arguments in low-level complexity. In *International Symposium on Mathematical Foundations of Computer Science*, pages 162–176. Springer, 1977. doi:10.1007/3-540-08353-7_135.

40 Nikhil Vyas and Ryan Williams. On Oracles and Algorithmic Methods for Proving Lower Bounds. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 99:1–99:26, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2023.99.

41 Ryan Williams. Nonuniform ACC circuit lower bounds. *J. ACM*, 61(1), January 2014. doi:10.1145/2559903.