Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 49 (2025)

Range Avoidance and Remote Point for Low-Depth Circuits: New Algorithms and Hardness

Xin Li [*] Yan Zhong [†]

April 25, 2025

Abstract

The Range Avoidance (Avoid) problem $\mathscr{C}$-Avoid$[n, m(n)]$ asks that, given a circuit in a class $\mathscr{C}$ 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 works by Korten (FOCS' 21) and 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 $\mathscr{C} \supseteq \mathsf{AC}^0$, there is an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for $\mathscr{C}$-Avoid$[n, n^{1+\varepsilon}]$ (for any constant $\varepsilon > 0$) if and only if $\mathbf{E}^{\mathbf{NP}}$ cannot be computed by $\mathscr{C}$ circuits of size $2^{o(n)}$.
- For any integer i, if $\mathbf{E}^{\mathbf{NP}}$ cannot be computed by $o(2^n/n)$ size NC^{i+1} circuits, then there is an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for NC^i-Avoid$[n, 2n]$. Note that by an extension of Ren, Santhanam, and Wang (FOCS' 22), an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for NC^i-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 NC^{i+1} circuits.

These results yield the first characterizations of $\mathbf{FP}^{\mathbf{NP}}$ $\mathscr{C}$-Avoid algorithms for low-complexity circuit classes such as AC^0. We also extend our results to the average-case analog of Avoid, the Remote Point (Remote-Point) problem, and establish similar equivalence between $\mathbf{FP}^{\mathbf{NP}}$ algorithms and the average-case circuit lower bounds for $\mathbf{E}^{\mathbf{NP}}$. 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.

[*]Department of Computer Science, Johns Hopkins University, lixints@cs.jhu.edu. Supported by NSF CAREER Award CCF-1845349 and NSF Award CCF-2127575.

[†]Department of Computer Science, Johns Hopkins University, yzhong36@jhu.edu. Supported by NSF CAREER Award CCF-1845349.

ISSN 1433-8092

Contents

1 Introduction

The *Range Avoidance* problem (Avoid for short) is a total search problem introduced in [KKMP21, Kor22, RSW22], 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 [Kor22] 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 [Erd47]. 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 $m := m(n)$*[1]*$> n$, 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[2] problem, which is studied extensively in previous works [KKMP21, CHLR23, CL24] 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 [RSW22] has focused on the problem for restricted circuit classes.

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

- $\mathscr{C}$**-Avoid**$[n, m]$ *is the class of Avoid problems where the circuits are in $\mathscr{C}$, with input length n and output length m;*
- $\mathscr{C}$**-Remote-Point**$[n, m, c(n)]$ *is the class of Remote-Point problems where the circuits are in $\mathscr{C}$, with input length n, output length m and whose output has relative hamming distance $1/2 - c(n)$ from any strings in the range of $\mathscr{C}$.*

A prominent motivation for studying $\mathscr{C}$-Avoid is its implication for circuit lower bounds. In particular, [RSW22] showed that for any circuit class $\mathscr{C}$ satisfying the *universality property* — namely, the *truth table generator* $\mathsf{TT}_{\mathscr{C}}$ (i.e., a circuit that, given an encoding of a circuit $C \in \mathscr{C}$, outputs C's truth table) is itself computable by $\mathscr{C}$ circuits (e.g., $\mathsf{AC}^0, \mathsf{TC}^0, \mathsf{NC}^1$) — efficient algorithms for $\mathscr{C}$-Avoid imply circuit lower bounds for $\mathscr{C}$. Specifically, solving $\mathscr{C}$-Avoid in $\mathbf{FP}$ (resp. $\mathbf{FP}^{\mathbf{NP}}$) implies that $\mathbf{E}$ (resp. $\mathbf{E}^{\mathbf{NP}}$) does not have $\mathscr{C}$ circuits.[3] Analogously, $\mathbf{FP}$ (resp. $\mathbf{FP}^{\mathbf{NP}}$) algorithms for $\mathscr{C}$-Remote-Point imply average-case $\mathscr{C}$ 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., [Che24, CR22, CLW20, CL23, LZ24]). On the other hand, these results also imply that it is potentially hard to design efficient algorithms for

[1] The function $m(n)$ is called the *stretch* of the circuit.
[2] We sometimes use RPP as a shorthand for Remote-Point.
[3] The size of the circuit lower bound depends on the stretch of the Avoid instance.

$\mathscr{C}$-AVOID even when $\mathscr{C}$ is restricted, hence many previous works also give *conditional* algorithms under various assumptions.

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 [RSW22] shows that, suppose for every constant $\varepsilon > 0$, there is an $\mathbf{FP}$ (resp. $\mathbf{FP}^{\mathbf{NP}}$) algorithm for NC^0_4-AVOID$[n, n + n^{\varepsilon}]$, then for every $k \geq 1$, there is an $\mathbf{FP}$ (resp. $\mathbf{FP}^{\mathbf{NP}}$) algorithm for NC^1_k-AVOID; and for every $\gamma > 0$, there is a family of functions in $\mathbf{E}$ (resp. $\mathbf{E}^{\mathbf{NP}}$) that cannot be computed by Boolean circuits of depth $n^{1-\gamma}$. Furthermore, [GLW22] 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 [GGNS23] 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, [CHLR23] designed an unconditional $\mathbf{FP}^{\mathbf{NP}}$ algorithm for ACC^0-REMOTE-POINT$[n, \text{qpoly}(n), 1/\text{poly}(n)]$ (qpoly(n) denotes quasi-polynomial(n)), recovering the state-of-the-art average-case lower bound for ACC^0 against $\mathbf{E}^{\mathbf{NP}}$. A recent breakthrough [CHR24, Li24] showed that $\mathsf{S_2E} \not\subset i.o.\text{-}\mathsf{SIZE}[2^n/n]$[4] via a single-valued $\mathsf{FS_2P}$ algorithm to AVOID, improving over the decades' old lower bound that $\Delta_3\mathsf{E} = \mathsf{E}^{\Sigma_2} \not\subset \mathsf{SIZE}[2^{o(n)}]$ [MVW99]. On the hardness side, Ilango, Li, and Williams [ILW23] showed that under the assumption that subexponential secure indistinguishability obfuscation ($i\mathcal{O}$) exists [JLS21] 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 [CL24] generalizes the framework and shows that under plausible cryptographic assumptions, $\mathscr{C}$-AVOID and $\mathscr{C}$-REMOTE-POINT are not in $\mathbf{FP}$, or even not in $\mathbf{SearchNP}$, when the underlying $\mathscr{C}$ 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 $\mathbf{FP}^{\mathbf{NP}}$ 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 [GLW22, GGNS23] 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 [KPI25], which also improved the stretch to $(\widetilde{O}(n^{k/2+(k-2)/(2k+4)}))$ with an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for odd k. A conditional $\mathbf{FP}^{\mathbf{NP}}$ algorithm was proposed in [RSW22] for NC^0-AVOID with stretch $n^{1+\varepsilon}$ for any constant ε, and whether there is an unconditional $\mathbf{FP}^{\mathbf{NP}}$ algorithm for such stretch is left as a central open question in [RSW22]. 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 $n^{k-2+\varepsilon}$ [GGNS23].

A recent work by Kuntewar and Sarma [KS25] 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 $\mathscr{C}$-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 $m = n + n^{2/3}$ [GGNS23]. This was also noted and remarked in [RSW22].

[4]The prefix "*i.o.*-" indicates that $\mathsf{S_2E}$ is not infinitely often in $\mathsf{SIZE}[2^n/n]$, that is $\mathsf{S_2E}$ is *almost-everywhere* hard for $\mathsf{SIZE}[2^n/n]$.

"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 $\mathscr{C}$-Avoid when $\mathscr{C}$ is restricted (e.g., NC^0, AC^0 and ACC^0), and similarly for $\mathscr{C}$-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 $\mathscr{C}$-Avoid and $\mathscr{C}$-Remote-Point, where $\mathscr{C}$ represents low-depth circuits.

1.1 Our Results

While as mentioned before, several previous works showed that algorithms for $\mathscr{C}$-Avoid or $\mathscr{C}$-Remote-Point with small stretch lead to circuit lower bounds, the works [Jeř04, Kor22, CHR24] remarkably showed that the converse is also true in the case where $\mathscr{C}$ is the class of unrestricted Boolean circuits. Specifically, they showed that

$$\text{Avoid} \in \mathbf{FP}^{\mathbf{NP}} \iff \mathbf{E}^{\mathbf{NP}} \not\subset i.o.\text{-}\mathsf{SIZE}[2^{o(n)}] \iff \mathbf{E}^{\mathbf{NP}} \not\subset i.o.\text{-}\mathsf{SIZE}[2^n/n]^{5}$$

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 [IW97, KvM02], 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 complement previous works, by establishing (near) equivalence when $\mathscr{C}$ is restricted. As a result, we also obtain conditional $\mathbf{FP}^{\mathbf{NP}}$ algorithms for $\mathscr{C}$-Avoid and $\mathscr{C}$-Remote-Point for a vast range of circuit classes $\mathscr{C}$ with suitable smaller stretch, under much weaker assumptions than those needed for general Avoid in [Kor22].

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

As mentioned in the above paragraphs, previous work [Kor22, RSW22] 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 Circuits with Small Stretch. Our first set of results concerns NC^i circuits. We show that near-maximal circuit lower bounds against $\mathbf{E}^{\mathbf{NP}}$ in NC^{i+1} imply efficient algorithms for NC^i-Avoid with small stretch:

Theorem 1.1. *For any integer i, if $\mathbf{E}^{\mathbf{NP}}$ requires near-maximum ($\Omega(2^n/n)$) size NC^{i+1} circuits, then there is an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for NC^i-Avoid$[n, 2n]$.*

Conversely, extending ideas from [RSW22] (with the proof deferred to Appendix D), we show:

[5] The original second equivalence obtained by [Kor22] is $\mathbf{E}^{\mathbf{NP}} \not\subset i.o.\text{-}\mathsf{SIZE}[2^{o(n)}] \iff \mathbf{E}^{\mathbf{NP}} \not\subset i.o.\text{-}\mathsf{SIZE}[2^n/(2n)]$, which can be strengthened by a finer encoding arguments of circuits [CHR24].

Theorem 1.2. *For any constant $\delta \in (0,1)$ and any integer i, NC^i-*AVOID*$[n, n+n^\delta] \in \mathbf{FP}^{\mathbf{NP}} \implies \mathbf{E}^{\mathbf{NP}} \not\subset$ i.o.-NC^{i+1}-$\mathsf{SIZE}[o(2^n/n)]$.*

Together, these results nearly characterize the hardness of proving near-maximum $\mathbf{E}^{\mathbf{NP}}$ lower bounds against NC^{i+1} in terms of $\mathbf{FP}^{\mathbf{NP}}$ algorithms for NC^i-AVOID.

We also generalize this characterization to the REMOTE-POINT problem:

Recall the definition of *good* function from [RSW22].

Definition 1.4 (Good function [RSW22])**.** *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 $\mathrm{poly}(\log n, \log f(n))$.*

Theorem 1.3. *For any integer i and any monotone function $c : \mathbb{N} \to \mathbb{N}$ that is good, if $\mathbf{E}^{\mathbf{NP}}$ cannot be $(1/2 + c(\frac{2^n}{2}))$-approximated by near-maximum ($\Omega(2^n/n)$) size NC^{i+1} circuits, then there is an $\mathbf{FP}^{\mathbf{NP}}$ algorithms for NC^i-REMOTE-POINT$[n, 2n, c(n)]$.*

Theorem 1.4. *Let $c : \mathbb{N} \to \mathbb{N}$ be any monotone function $c : \mathbb{N} \to \mathbb{N}$ that is good. For any constant $\delta \in (0,1)$ and any integer i, NC^i-REMOTE-POINT$[n, n+n^\delta, c(n)] \in \mathbf{FP}^{\mathbf{NP}} \implies \mathbf{E}^{\mathbf{NP}} \not\subset$ i.o.-$\mathsf{Avg}_{c(\frac{2^n}{2})}\mathsf{NC}^{i+1}$-$\mathsf{SIZE}[o(2^n/n)]$.*

Results for Circuit Classes Containing AC^0 with Polynomial Stretch. In the regime of polynomial stretch, we obtain tight equivalences for circuit classes $\mathscr{C}$ satisfying $\mathsf{AC}^0 \subseteq \mathscr{C}$:

Theorem 1.5. *For any circuit class $\mathscr{C}$ such that $\mathsf{AC}^0 \subseteq \mathscr{C}$ (e.g., $\mathsf{AC}^0, \mathsf{ACC}^0, \mathsf{TC}^0, \mathsf{NC}^1$), $\mathbf{E}^{\mathbf{NP}}$ requires $2^{\Omega(n)}$ size $\mathscr{C}$ circuits if and only if there is an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for $\mathscr{C}$-AVOID$[n, n^{1+\varepsilon}]$ for any constant $\varepsilon > 0$.*

Theorem 1.6. *For any circuit class $\mathscr{C}$ such that $\mathsf{AC}^0 \subseteq \mathscr{C}$ and any monotone function $c : \mathbb{N} \to \mathbb{N}$ that is good, $\mathbf{E}^{\mathbf{NP}}$ cannot be $(1/2 + c(2^{\frac{n}{1+\varepsilon}}))$-approximated by $2^{\Omega(n)}$ size $\mathscr{C}$ circuits if and only if there is an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for $\mathscr{C}$-RPP$[n, n^{1+\varepsilon}, c(n)]$ for any constant $\varepsilon > 0$.*

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

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

Theorem 1.8. *For any circuit class $\mathscr{C}$ such that $\mathsf{AC}^0 \subseteq \mathscr{C}$ and any monotone function $c : \mathbb{N} \to \mathbb{N}$ that is good, $\mathbf{EXP}^{\mathbf{NP}}$ cannot be $(1/2 + c(2^{\frac{n}{1+\varepsilon}}))$-approximated by $2^{\Omega(n)}$ size $\mathscr{C}$ circuits if and only if there is an $\mathbf{FQP}^{\mathbf{NP}}$ algorithm for $\mathscr{C}$-RPP$[n, n^{1+\varepsilon}, c(n)]$ for any constant $\varepsilon > 0$.*

These results represent the first equivalence theorems connecting algorithms for $\mathscr{C}$-AVOID and $\mathscr{C}$-REMOTE-POINT 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 1.5 and Theorem 1.1 are consistent with our current knowledge of circuit lower bounds.

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

Open Problem 1.1 (Open problem 2 in [Kor25])**.** *Can we reduce $\mathscr{C}$-AVOID to circuit lower bounds for $\mathscr{C}$ for any circuit class $\mathscr{C} \subseteq \mathbf{P}/\mathrm{poly}$?*

Open Problem 1.2 (Open problem 7 in [GGNS23])**.** *Do there exist polynomial-time algorithms with* **NP** *oracles that solve* NC^0_3*-*Avoid *for stretch* $m = o(n^2/\log(n))$*?*

Specifically, Theorem 1.5 and Theorem 1.7 address Open Problem 1.1 in the stretch regime $n \mapsto n^{1+\varepsilon}$, for any constant $\varepsilon > 0$, and any circuit classes containing AC^0. In addition, Theorem 1.2 and Theorem 1.1 also nearly pin down the hardness of proving $\mathbf{E}^{\mathbf{NP}}$ requires near-maximum NC^{i+1} circuit in terms of NC^i-Avoid algorithm: proving such a lower bound should be no harder than proving $\mathsf{NC}^i[n, n+n^\delta] \in \mathbf{FP}^{\mathbf{NP}}$ for any $\delta \in (0,1)$, but should be no easier than $\mathsf{NC}^i[n, 2n] \in \mathbf{FP}^{\mathbf{NP}}$[6].

Theorem 1.5 partially addresses Open Problem 1.2. Given the high plausibility of $\mathbf{E}^{\mathbf{NP}} \not\subset \mathsf{AC}^0$-$\mathsf{SIZE}[2^{o(n)}]$ (e.g., Håstad proved 40 years ago that $\oplus$ cannot be computed by AC^0 circuits of size $2^{n^{1/(d-1)}}$ infinitely often [Has86]), one would expect there to be an $\mathbf{FP}^{\mathbf{NP}}$ algorithm even for AC^0-Avoid$[n, n^{1+\varepsilon}]$.

1.1.2 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[7] algorithm for NC^0_k-Avoid with any super-linear stretch for any constant k.

Theorem 1.9. *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 similar runtime only worked for stretch $n \mapsto n^{k-2+\varepsilon}$ [GGNS23], making our result the first to achieve subexponential-time performance with superlinear stretch for all k.

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

Theorem 1.10. *Suppose Assumption 2.11 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 1.11. *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 $\mathscr{C}$-Avoid across both small and large stretch regimes, with implications for circuit lower bounds and local PRG security.

[6] In the case of $i = 0$, the results apply to NC^0_4-Avoid.

[7] 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 $\bigcup_{c<1} 2^{O(n^c)}$.

[8] We use $\mathbf{svFS_2P}$ to denote single-valued $\mathbf{FS_2P}$ algorithm

Problem	Algorithm	Assumption	Reference
$\textsc{Avoid}[n, n+1]$	$\mathbf{FP}^{\mathbf{NP}}$	$\mathbf{E}^{\mathbf{NP}} \not\subset i.o.\text{-}\mathsf{SIZE}[2^{o(n)}]$	[Kor22]
$\textsc{Avoid}[n, n+1]$	$\mathbf{svFS_2P}$[8]	–	[CHLR23, Li24]
$\mathsf{NC}^0_k\text{-}\textsc{Avoid}[n, n^{k-1}/\log(n)]$	$\mathbf{FP}$	–	[GGNS23]
$\mathsf{NC}^0_k\text{-}\textsc{Avoid}[n, n^{k-2+\varepsilon}]$	$2^{O(n^{1-\varepsilon})}$	–	[GGNS23]
$\mathsf{NC}^0_{2t}\text{-}\mathrm{RPP}[n, O_t(n^t \log n), O(1)]$	$\mathbf{FP}$	–	[KPI25]
$\mathsf{NC}^0_{2t+1}\text{-}\textsc{Avoid}[n, \widetilde{O}(n^{t+\frac{2}{2t+3}})]$	$\mathbf{FP}^{\mathbf{NP}}$	–	[KPI25]
$\mathsf{NC}^0\text{-}\textsc{Avoid}[n, n^{1+\varepsilon}]$	$\mathbf{FP}^{\mathbf{NP}}$	Assumption 2.4	[RSW22]
$\mathsf{ACC}^0\text{-}\mathrm{RPP}[n, \mathrm{qpoly}(n), 1/\mathrm{poly}]$	$\mathbf{FP}^{\mathbf{NP}}$	–	[CHLR23]
$\mathscr{C}\text{-}\mathrm{RPP}[n, n^{1+\varepsilon}, c(n)]$	$\mathbf{FP}^{\mathbf{NP}}$	$\mathbf{E}^{\mathbf{NP}} \not\subset i.o.\text{-}\mathsf{Avg}_{c(2^{\frac{n}{1+\varepsilon}})}\text{-}\mathscr{C}\text{-}\mathsf{SIZE}[2^{o(n)}]$	Theorem 1.5
$\mathsf{NC}^i\text{-}\mathrm{RPP}[n, 2n, c(n)]$	$\mathbf{FP}^{\mathbf{NP}}$	$\mathbf{E}^{\mathbf{NP}} \not\subset i.o.\text{-}\mathsf{Avg}_{c(\frac{2^n}{2})}\text{-}\mathsf{NC}^{i+1}\text{-}\mathsf{SIZE}[o(2^n/n)]$	Theorem 1.1
$\mathsf{NC}^0_k\text{-}\textsc{Avoid}[n, n^{1+\varepsilon}]$	$2^{n^{1-\frac{\varepsilon}{k-1}+o(1)}}$	–	Theorem 1.9
$\mathsf{NC}^0_k\text{-}\textsc{Avoid}[n, n^{k-1}/\log^{k-2}(n)]$	$\mathbf{FP}$	Assumption 4.3	Theorem 4.4
$\mathsf{NC}^0_k\text{-}\textsc{Avoid}[n, n+1]$	$O(n2^{\frac{k-2}{k-1}n})$	–	Theorem 1.11

Table 1: Range Avoidance and Remote Point Algorithms. In the 9-th row, we assert $\mathsf{AC}^0 \subseteq \mathscr{C}$. Also note that taking $c(n) = 1/m(n) = 1/n^{1+\varepsilon}$ in the 9-th row and $c(n) = m(n) = 1/(2n)$ in the 10-th row recovers $\mathbf{FP}^{\mathbf{NP}}$ algorithms for $\textsc{Avoid}$ from worst-case circuit lower bounds.

1.2 Technical Overview

Equivalence between $\mathscr{C}$-$\textsc{Avoid}[n, n^{1+\varepsilon}] \in \mathbf{FP}^{\mathbf{NP}}$ and $\mathbf{E}^{\mathbf{NP}} \not\subset i.o.\text{-}\mathscr{C}\text{-}\mathsf{SIZE}[2^{o(n)}]$. We establish a tight equivalence between the complexity of solving $\mathscr{C}$-$\textsc{Avoid}[n, n^{1+\varepsilon}]$ in $\mathbf{FP}^{\mathbf{NP}}$ and proving exponential lower bounds for $\mathscr{C}$ circuits against $\mathbf{E}^{\mathbf{NP}}$, generalizing the reduction of Jeřábek and Korten [Jeř04, Kor22], who proved that $\textsc{Avoid} \in \mathbf{FP}^{\mathbf{NP}}$ if and only if $\mathbf{E}^{\mathbf{NP}} \not\subset i.o.\text{-}\mathsf{SIZE}[2^{o(n)}]$[9].

The forward direction — namely, that an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for $\mathscr{C}$-$\textsc{Avoid}$ implies exponential $\mathscr{C}$ circuit lower bounds against $\mathbf{E}^{\mathbf{NP}}$ — was largely established in [RSW22]. A key component of this argument is the *universality property* of the circuit class $\mathscr{C}$: that the truth table generator $\mathsf{TT}_{\mathscr{C}}$ can itself be computed by a circuit in $\mathscr{C}$. We strengthen and formalize this notion, showing that any circuit class $\mathscr{C}$ 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 $\mathcal{C}$ and an input x, it decodes $\mathcal{C}$ 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 $\mathscr{C}$ circuit lower bounds for functions in $\mathbf{E}^{\mathbf{NP}}$

[9]This reduction, which we refer to as *Jeřábek-Korten reduction*, was originally proved in the framework of bounded arithmetic by Jeřábek [Jeř04], and later translated to the language of computational complexity by Korten [Kor22]. Specifically, as pointed out to us by Erfan Khaniki, [Jeř04, 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 $\textsc{Avoid}$ can be extracted from the dual weak pigeonhole principle (i.e., formalization of the totality of $\textsc{Avoid}$) in S^1_2 via the Witnessing Theorem from [Kra92].

imply that $\mathscr{C}$-Avoid $\in \mathbf{FP}^{\mathbf{NP}}$, 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 $\mathscr{C}$-Remote-Point$[n, n^{1+\varepsilon}]$ problem for any $\mathscr{C}$ containing AC^0.

We first briefly recall the $\mathbf{FP}^{\mathbf{NP}}$ reduction from circuit lower bound to Avoid in [Jeř04, Kor22] 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 GGM-tree of height k. The resulting circuit has stretch $n \cdot 2^k$, and the output $y \in \mathrm{Range}(\mathsf{GGM}[C])$ can be regarded as encoding the truth table of a function g, whose input are 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 $\mathscr{C}$.

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 $y \in \mathrm{Range}(\mathsf{GGM}_{\ell,q,k}[C])$, 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.

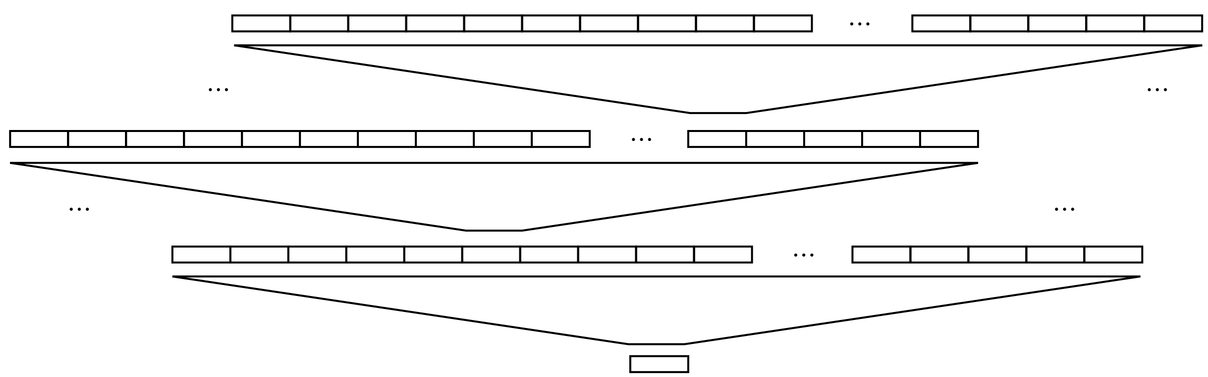

Figure 1: Generalized q-ary GGM-Tree

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 $\mathrm{Range}(C)$ by traversing the GGM-tree with an $\mathbf{NP}$ 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:

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

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

Subexponential time NC^0-Avoid algorithm for any superlinear stretch. We present the first subexponential-time algorithm for NC^0_k-$\textsc{Avoid}[n, n^{1+\varepsilon}]$, achieving runtime $2^{n^{1-\frac{x}{k-1}+o(1)}}$ 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 [RTS00] can be adapted to argue that such graphs necessarily contain Hall-violating subsets of outputs of size at most $K = n^{1-\frac{\varepsilon}{k-1}}$. 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 $\mathcal{C}'$ of size K, and enumerate all strings in $\{0,1\}^{|\Gamma(S)|}$ to find those not in the image of $\mathcal{C}'$. 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 $\mathcal{C}$.

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-$\textsc{Avoid}$ in stretch regimes as low as $m = n^{k-1}/\log^{k-2} n$, improving prior work [GGNS23] which required larger stretch.

Finally, we connect our algorithmic result to pseudorandomness. We show that any subexponential-time $\textsc{Avoid}$ 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-$\textsc{Avoid}[n, n+1]$ 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 such assignments, the preimage space collapses to a singleton or 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 [GLW22]. 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 Paper Organization

The rest of the paper is organized as follows. In Section 2 we give some preliminary knowledge and some primitives from prior works. In Section 3 we present the generalized Jeřábek-Korten reduction, conditional $\mathbf{FP}^{\mathbf{NP}}$ algorithms as well as the precise characterization of $\mathbf{E}^{\mathbf{NP}}$ circuit lower bound in terms of Avoid (Remote-Point) problems. In Section 4 we present the subexponential-time NC^0-Avoid algorithm for any superlinear stretch. In Section 5 we present the non-trivial algorithm for NC^0-Avoid$[n, n+1]$. Finally, we conclude in Section 6 with some open problems.

2 Preliminaries

2.1 Notations

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

Definition 2.1 (NC circuits [GGNS23])**.** *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 **FP** or even $\mathbf{FE^{NP}}$ [GGNS23, Val77, AB09, Frontier 3] is a decades-old challenge. Valiant [Val77] 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,*[10]
- *depth $O(\log^i n)$,*

[10] We also say, e.g., exponential-size AC circuits. The "polynomial size" is the default setting when we refer to AC circuits without explicitly spelling out the size.

- *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*.*

Definition 2.3 (DNF)**.** *The term* DNF *refers to* AC^0_2 $(\vee \circ \wedge)$ *circuits.*

2.3 Some Previous Results on NC^0-Avoid

NC^0-Avoid with Strong Parameters Simulates NC^1-Avoid. NC^0-Avoid with strong parameters simulates NC^1-Avoid using the randomized encoding technique [RSW22].

Theorem 2.1 (NC^0-Avoid with strong parameters simulates NC^1-Avoid [RSW22])**.** *The following is a polynomial time reduction from* NC^1*-Avoid to* NC^0*-Avoid with exact stretch computed*

1. NC^1-*Avoid*$[n, \ell] \leq_{\mathbf{FP}} \mathsf{NC}^0_4$-*Avoid*$[n, n + n^{\log_{n+\mathrm{poly}(n)}(\ell - n)}]$
2. NC^1-*Avoid*$[n, \mathrm{poly}(n)] \leq_{\mathbf{FP}} \mathsf{NC}^0_4$-*Avoid*$[n, 2n]$

In fact, in this paper we show that for any integer i, NC^{i+1}-Avoid$[n, n+1] \leq_{\mathbf{FP}} \mathsf{NC}^i$-Avoid$[n, n+1]$. The proof is deferred to Appendix B.

Matrix Rigidity and the Connection to NC^0_3-Avoid.

Theorem 2.2 ([Val77])**.** *If a family of matrices* $(M_n)_{n \geq 1}$*,* $M_n \in \mathsf{F}_2^{n \times n}$*, is* $(\varepsilon n, n^{\delta})$*-rigid for constant* $\varepsilon, \delta > 0$*, then the linear map* $x \mapsto Mx$ *requires linear* NC^1 *circuits of size* $\Omega(n \log\log(n))$*.*

Definition 2.4 (Rigid [GLW22, GGNS23])**.** *Rigid is the following problem: given input* 1^n*, output an* $n \times n$ *matrix over* F_2 *that is* $(\varepsilon n, n^{\delta})$*-rigid for constant* $\varepsilon, \delta > 0$*.*

Theorem 2.3 ([GGNS23])**.** *Rigid* $\leq_{\mathbf{FP}} \mathsf{NC}^0_3$-Avoid$[n, n + n^{2/3}]$.

An Assumption that Yields NC^0-Avoid$[n, n^{1+\varepsilon}]$ Algorithms.

Assumption 2.4 ([RSW22])**.** *For every constants* $k \geq 1$ *and* $\varepsilon > 0$*, there is an* $\mathbf{FP}^{\mathbf{NP}}$ *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.*

2.4 Universality Property and Truth Table Generator

Definition 2.5 (Universality Property [RSW22])**.** *Let* $\mathscr{C}$ *be a circuit class. We say that* $\mathscr{C}$ *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* $\mathscr{C}$ *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* $\mathscr{C}$ *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 ([CH85])**.** *The class* AC^0 *has universality property.*

Theorem 2.6 ([Bus87])**.** *The class* NC^1 *has universality property.*

In effect, any circuit class containing AC^0 has universality property. We include in Appendix A for a detailed proof.

Definition 2.6 (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* $\mathsf{TT}_{\mathscr{C}} : \{0,1\}^{O(s \log s)} \to \{0,1\}^{2^n}$ *to be the circuit that takes as input the description of a size s $\mathscr{C}$ circuit on n variables, and outputs the truth table of this $\mathscr{C}$ circuit. It is clear that if $\mathscr{C}$ has universality property, then* $\mathsf{TT}_{\mathscr{C}} \in \mathscr{C}$*.*

The following modified Theorem says that solving $\mathscr{C}$-Avoid on $\mathsf{TT}_{\mathscr{C}}$ implies $\mathscr{C}$ circuit lower bounds with tight parameters (see Appendix D for a proof).

Theorem 2.7 (Modified Theorem 5.2 of [RSW22])**.** *Let $\mathscr{C}$ 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 $\mathscr{C}$-Remote-Point$[N, f(N), c(N)]$, where each output gate has $\mathscr{C}$ 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 $\mathscr{C}$ circuits of size* $\frac{\varepsilon f^{-1}(2^n)}{\log f^{-1}(2^n)}$*.*

2.5 Bipartite Vertex Expander

Definition 2.7 (Vertex expander [Vad12])**.** *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 2.8 (Left regular bipartite graphs [Vad12])**.** *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 2.8 and Theorem 2.9 are modified from [Vad12]. Since the parameters are different from the original theorem, we include in Appendix E the corresponding proofs for completeness.

Theorem 2.8 (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 2.9 (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 2.9 (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 2.10 (Disperser graphs [Sip86, CW89])**.** *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 2.10 (Lower bounds for disperser graphs [RTS00])**.** *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 [VW23].

2.7 The Existence of PRGs in NC^0

Assumption 2.11 ([JLS21])**.** *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 Generalized GGM-Tree and Conditional $\mathbf{FP^{NP}}$ Algorithms

In light of the difficulty in obtaining an unconditional $\mathbf{FP^{NP}}$ algorithm for AC^0-Avoid$[n, \mathrm{qpoly}(n)]$ and NC^0-Avoid$[n, n + o(n)]$ [RSW22], we turn our attention to exploring which assumptions might yield such an $\mathbf{FP^{NP}}$ algorithm for AC^0-Avoid and NC^0-Avoid.

Korten [Kor22] observed that Avoid admits an $\mathbf{FZPP^{NP}}$ algorithm. Moreover, he, building on the work of Jeřábek [Jeř04], obtained a conditional derandomization of this algorithm under assumptions (e.g., $\mathbf{E^{NP}}$ requires circuits of size $2^{\Omega(n)}$) significantly weaker than those required by standard approaches (which typically demand, for example, that $\mathbf{E}$ requires SAT-oracle circuits of size $2^{\Omega(n)}$ [KvM02]). His approach, which we have dubbed Jeřábek-Korten reduction in the introduction, also inspired a recent breakthrough achieving near-maximal circuit lower bounds against $\mathbf{S_2E}$ [CHR24, Li24].

These developments motivate us to explore generalizations of Jeřábek-Korten reduction aimed at derandomizing the $\mathbf{FZPP^{NP}}$ algorithm for restricted circuit classes $\mathscr{C}$.

Jeřábek-Korten Reduction in a Nutshell. Given an Avoid instance described by a circuit $C : \{0,1\}^n \to \{0,1\}^{2n}$, Jeřábek-Korten's reduction evaluates C along a GGM-style computation tree [GGM86] to define an expanded circuit C^* (see Figure 2). A key insight is that every string $y \in \mathrm{Range}(C^*)$ has low circuit complexity Lemma 3.1. Thus, if one finds a string $y' \in \{0,1\}^{|\mathrm{Range}(C^*)|}$ with higher circuit complexity, then $y' \notin \mathrm{Range}(C^*)$. This gap can be leveraged: given such a y', and access to a circuit-inversion oracle, one can traverse the tree and extract a string not in the range of the original circuit C.

Lemma 3.1 (The output of GGM-tree has small circuit complexity [GGM86, CHR24])**.** *Let* $\mathsf{GGMEval}(C, T, x, i)$ *denote the* i*-th bit of* $\mathsf{GGM}_T[C](x)$*. There is an algorithm running in* $\widetilde{O}(|C| \cdot \log T)$ *time that, given* C, T, x, i *outputs* $\mathsf{GGMEval}(C, T, x, i)$*.*

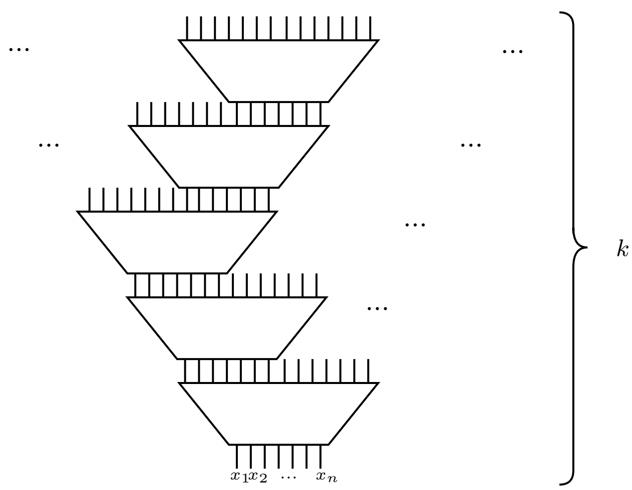

Figure 2: Apply the circuit $C : \{0,1\}^n \to \{0,1\}^{2n}$ over the GGM-tree of height k to obtain the circuit $C^* : \{0,1\}^n \to \{0,1\}^{2^k n}$.

3.1 Generalized Jeřábek-Korten Reduction

We now define a generalized GGM-tree and demonstrate that it characterizes the feasibility of solving $\mathscr{C}$-Avoid in $\mathbf{FP}^{\mathbf{NP}}$, even when $\mathscr{C}$ is as weak as AC^0. Previously, such tight correspondences were only known for unrestricted circuit classes.

Generalized GGM-tree Construction $\mathsf{GGM}_{\ell,q,k}[C]$: Given a circuit $C : \{0,1\}^n \to \{0,1\}^{nq}$ and parameters $\ell = nq^k$, construct $\mathsf{GGM}_{\ell,q,k}[C]$ as follows:

1. Assign the root vertex $(0,0)$ the value $v_{0,0} = x$.
2. Build a perfect q-ary tree of height k. Let (i,j) denote the j-th node at level i ($0 \le i \le k$, $0 \le j < q^i$).
3. At each node (i,j), compute $y = C(v_{i,j})$ and assign its h-th child the h-th block of n bits of y, for $h \in [q]$.

4. The output $\mathsf{GGM}_{\ell,q,k}[C](x)$ is the concatenation of the values at the q^k leaves.

Circuit Complexity of the Output.

Theorem 3.2. *Let $C : \{0,1\}^n \to \{0,1\}^m$ be a circuit where each output bit has circuit complexity s_C. Let $C^* = \mathsf{GGM}_{\ell,q,k}[C]$ have tree height k. Then:*

- *The output length (stretch) of C^* is $\ell = m^k/n^{k-1}$.*
- *The circuit complexity of $C^*(x)$ is at most $O(s_C \cdot k)$.*

Proof. We prove by each bullet.

- Each increase in the depth level stretches the output length by a multiplicative factor of m/n. According to the definition of the height of the tree, the final stretch $\ell = m^k/n^{k-1}$.
- To compute a specific bit of $C^*(x)$, we only iteratively apply the C for k times. The rest of the configuration operations can be implemented by size $O(s_C \cdot k)$ AC^0 circuits, as detailed in the following paragraph.

□

Consequently, any string $y \in \{0,1\}^\ell$ with circuit complexity exceeding $O(s_C \cdot k)$ must lie outside $\mathrm{Range}(C^*)$.

Implementing the Succinct Circuit. Figure 3 illustrates a succinct circuit $g : \{0,1\}^{\log \ell} \to \{0,1\}$ whose truth table corresponds to a string $y \in \mathrm{Range}(C^*)$. For any such y, a circuit implementing g can be built using a single $\mathscr{C}$ circuit, provided that C is in $\mathscr{C}$ and k is not too large.

The key components of this construction are:

- **(Multiplexers)** Given a $\log n$-bit index i and n bits $x_1, ..., x_n$, selection can be implemented as a DNF of the form:
$$\bigvee_{j=0}^{n-1} ((i = j) \wedge x_{j+1})$$
where $(i = j)$ is computed by conjoining each bit of i with its matching bit in j (or its negation).
- **(Tree Evaluation)** Evaluating a depth-k GGM-tree over a hardwired input x can be done with $\mathscr{C}$ circuits of size $O(|C| \cdot k)$.
- **(Indexing)** Extracting an individual bit from the final output is another instance of multiplexing.

This framework enables us to recover a string not in the range of C, given one outside the range of C^*.

Modified Jeřábek-Korten Reduction. We give a variant of Jeřábek-Korten reduction that traverses the generalized GGM-tree using post-order traversal:

Definition 3.1 (Post-order traversal for perfect q-ary trees). *In the post-order traversal, a vertex u_1 precedes u_2 ($u_1 <_P u_2$) if u_1 is visited before u_2 in a depth-first search that processes children from the 0-th to the $(q-1)$-th before the parent.*

Algorithm 1: $\mathsf{Jeřábek\text{-}Korten}''(C, f)$: Modified Jeřábek-Korten Reduction for q-ary GGM-Tree

---

Input: Circuit $C : \{0,1\}^n \to \{0,1\}^{n \cdot q}$ and string $f \in \{0,1\}^\ell \setminus \mathrm{Range}(\mathsf{GGM}_{\ell,q,k}[C])$.
Output: A string $y \notin \mathrm{Range}(C)$.
Data: GGM-tree with q-ary branching and height k.

for $j \leftarrow 0$ *to* $q^k - 1$ **do**
  $v_{k,j} \leftarrow f_{[jn,(j+1)n)}$
end
for *vertex* (i, j) *in post-order traversal* **do**
  Let $v_{i,j}$ be the lexicographically smallest x such that $C(x) = v_{i+1,qj} \circ \cdots \circ v_{i+1,qj+q-1}$
  if *no such x exists* **then**
    Set remaining vertices to $\perp$ and **return** $v_{i+1,qj} \circ \cdots \circ v_{i+1,qj+q-1}$
  end
end
return $\perp$

3.2 Conditional $\mathbf{FP^{NP}}$ Algorithm for NC^i-Avoid$[n, 2n]$

In this section, we show that, for any integer i, assuming near-maximum ($\Omega(2^n/n)$) size NC^{i+1} circuit lower bound against $\mathbf{E^{NP}}$, we can obtain an $\mathbf{FP^{NP}}$ algorithm for NC^i-Avoid$[n, 2n]$.

Theorem 3.3. *For any integer i, if $\mathbf{E^{NP}}$ requires near-maximum ($\Omega(2^n/n)$) size NC^{i+1} circuits, then there is an $\mathbf{FP^{NP}}$ algorithm for NC^i-Avoid$[n, 2n]$.*

Proof. Let $C : \{0,1\}^n \to \{0,1\}^{2n}$ be a circuit in NC^i. Consider applying the generalized GGM construction $C^* = \mathsf{GGM}_{\ell,q,k}[C]$, and let $g : \{0,1\}^{\log \ell} \to \{0,1\}$ denote the succinct circuit computing the truth table of an output $y \in \mathrm{Range}(C^*)$.

We now analyze the circuit complexity of g, using the structure of NC^i.

Choose parameters as follows:

- Let $k = c \cdot \log^{i+1} \log n$ for a sufficiently large constant c;
- Then $\ell = n \cdot 2^k = n \cdot 2^{c \log^{i+1} \log n}$, so:

$$\frac{\ell}{\log \ell} = \frac{n \cdot 2^{c \log^{i+1} \log n}}{\log n + c \log^{i+1} \log n}$$

- Since $|C| = O(n)$, it follows that $O(|C| \cdot k) = O(n \log^{i+1} \log n) = o(\ell / \log \ell)$.

Thus, by Theorem 3.2, any $y \in \mathrm{Range}(C^*)$ can be computed by an NC^i circuit of size $O(n \log^{i+1} \log n)$, while any string $f \in \{0,1\}^\ell$ with circuit complexity $\Omega(\ell / \log \ell)$ lies outside the range of C^*.

Consequently, given such a string f, we can invoke Algorithm 1 to recover a string not in $\mathrm{Range}(C)$, thereby obtaining an $\mathbf{FP^{NP}}$ algorithm for NC^i-Avoid$[n, 2n]$ under the assumption that f is hard. $\square$

3.3 Conditional $\mathbf{FP}^{\mathbf{NP}}$ Algorithm for $\mathscr{C}$-AVOID$[n, n^{1+\varepsilon}]$

We now extend our generalized framework to establish an equivalence between lower bounds against a circuit class $\mathscr{C}$ and the existence of $\mathbf{FP}^{\mathbf{NP}}$ algorithms for $\mathscr{C}$-AVOID, under mild stretch.

Theorem 3.4. *Let $\mathscr{C}$ be a circuit class satisfying $\mathsf{AC}^0 \subseteq \mathscr{C}$. Then the following are equivalent:*

1. *$\mathbf{E}^{\mathbf{NP}}$ does not have $2^{o(n)}$-size $\mathscr{C}$ circuits;*
2. *For every constant $\varepsilon > 0$, there exists an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for $\mathscr{C}$-AVOID$[n, n^{1+\varepsilon}]$.*

Proof. (" $\Longleftarrow$ ") This direction follows from the universality of $\mathscr{C}$, as formalized in Theorem 2.7. Specifically, if $\mathsf{TT}_{\mathscr{C}}$ can be implemented within $\mathscr{C}$, then the existence of an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for $\mathscr{C}$-AVOID implies that $\mathbf{E}^{\mathbf{NP}}$ requires exponential-size $\mathscr{C}$ circuits. See Appendix A for a detailed proof.

(" $\Longrightarrow$ ") We now show that assuming $\mathbf{E}^{\mathbf{NP}}$ requires $2^{\Omega(n)}$-size $\mathscr{C}$ circuits, one can obtain an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for $\mathscr{C}$-AVOID$[n, n^{1+\varepsilon}]$, for any constant $\varepsilon > 0$, via the generalized GGM construction.

Let $C : \{0,1\}^n \to \{0,1\}^{n^{1+\varepsilon}}$ be an instance of $\mathscr{C}$-AVOID$[n, n^{1+\varepsilon}]$, where each output bit of C is computed by a size-$s_C = n^c$ $\mathscr{C}$ circuit of depth d.

Let us construct $C^* = \mathsf{GGM}_{\ell,q,k}[C]$ with parameters chosen as follows:

- Set $q = n^{\varepsilon}$ and $k = O(1)$;
- Then $\ell = n \cdot q^k = n^{1+k\varepsilon}$, the output length of C^*;
- By Theorem 3.2, the circuit complexity of any $y \in \text{Range}(C^*)$ is bounded by $s_{C^*} = k \cdot s_C = O(n^c)$, since k is constant.

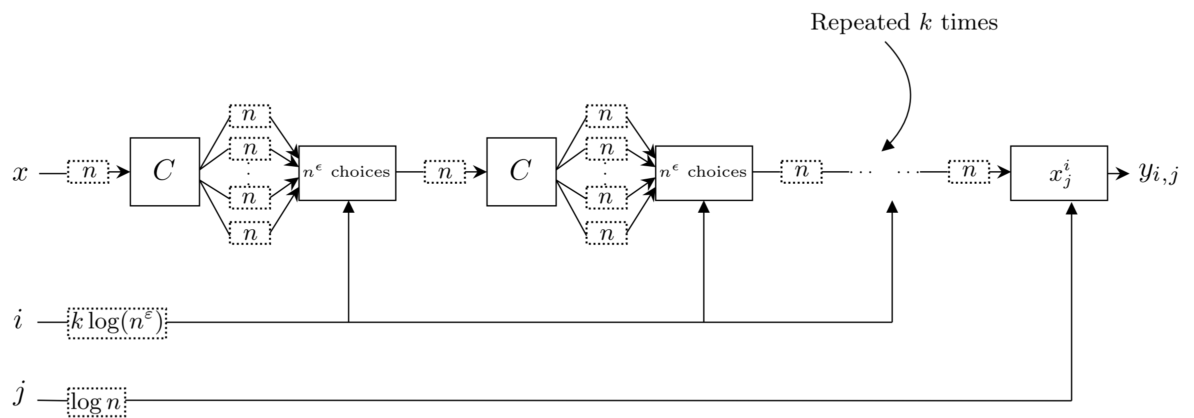

Figure 3: A succinct circuit whose truth table is y, for any y in the range of C^* [Kor22, Figure 2]. Each C circuit has input length n and output length $n^{1+\varepsilon}$ and is by overlayed by n^{ε} new C circuits in the next level. As in [Kor22], dotted boxes indicate the number of bits along a wire; x is hardwired as constants/advice for any given y. The only true inputs to this circuit are i, j.

Now suppose there exists a string $y^* \in \{0,1\}^{\ell}$ with $\mathscr{C}$ circuit complexity $\geq \ell^{\delta} = n^{\delta(1+k\varepsilon)}$ for some constant $\delta > 0$, and depth $(2+d)k+2$. Since $\delta(1+k\varepsilon) > c$ (by choosing k appropriately), it follows that $y^* \notin \text{Range}(C^*)$.

Applying Algorithm 1 on input C and y^* allows us to find a string outside $\mathrm{Range}(C)$, using an **NP** oracle and evaluation of $\mathscr{C}$ circuits of size $O(n^{\delta(1+k\varepsilon)})$. Since C and all circuits in the reduction are polynomial size (in ℓ), this yields an $\mathbf{FP}^{\mathbf{NP}}$ algorithm.

It remains to verify that the succinct circuit for y^* can be efficiently implemented by $\mathscr{C}$. As illustrated in Figure 3, each bit of y^* can be computed by:

- Selecting one of $q = n^\varepsilon$ blocks using a multiplexer implementable by a size-$O(n^\varepsilon)$ DNF;
- Applying the circuit C on the selected input block, using a size-$n^{1+\varepsilon} \cdot s_C = n^{1+\varepsilon+c}$ $\mathscr{C}$ circuit;
- Repeating for k layers of GGM-tree evaluation (multiplexing and applying C);
- Performing a final selection to extract the i-th bit from the output of the last layer.

Since $\mathscr{C}$ is closed under constant-depth composition and contains AC^0, the entire computation stays within $\mathscr{C}$, with total size $O(k \cdot n^{1+\varepsilon+c}) = O(n^{1+\varepsilon+c})$ and depth $(2+d)k+2$. Thus, we obtain the desired succinct circuit and complete the reduction.

□

The above proof also extends to the setting of $\mathbf{FQP}^{\mathbf{NP}}$ algorithms and corresponding lower bounds for $\mathbf{EXP}^{\mathbf{NP}}$. Intuitively, if one can construct the truth table of a length-ℓ function in quasi-polynomial time, then the hard function lies in **EXP**. Combined with Theorem 2.7, this yields the following theorems.

Theorem 3.5. *For any circuit class $\mathscr{C}$ such that $\mathsf{AC}^0 \subseteq \mathscr{C}$, $\mathbf{EXP}^{\mathbf{NP}}$ requires $2^{\Omega(n)}$ size $\mathscr{C}$ circuits if and only if there is an $\mathbf{FQP}^{\mathbf{NP}}$ algorithm for $\mathscr{C}$-*Avoid*$[n, n^{1+\varepsilon}]$ for any constant $\varepsilon > 0$.*

The smallest circuit class of the equivalence result is AC^0. However, it is also an intriguing question to obtain $\mathbf{FP}^{\mathbf{NP}}$ algorithm for NC^0-Avoid$[n, n^{1+\varepsilon}]$.

Remark 3.1. *Instantiating the same framework for $\mathscr{C} = \mathsf{NC}^0$ yields that $\mathbf{E}^{\mathbf{NP}}$ requires exponential-size $(\mathsf{DNF} \circ \mathsf{NC}^0)^k \circ \mathsf{DNF}$ circuits $\implies$ an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for NC^0-*Avoid*$[n, n^{1+\varepsilon}]$.*

3.4 Generalization of Jeřábek-Korten Reduction to Remote-Point

As we mentioned in the introduction, the Remote-Point problem $\mathscr{C}$-RPP$[n, m(n), c(n)]$ is the average-case analog of $\mathscr{C}$-Avoid$[n, m(n)]$. Algorithms for Remote-Point imply average-case lower bound.

For example, by the work of [CHLR23], it is known that the state-of-the-art $\mathbf{FP}^{\mathbf{NP}}$ algorithm for ACC^0-Remote-Point recovers the best-known almost-everywhere average-case lower bounds[11] against ACC^0 circuits by Chen, Lyu, and Williams [CLW20].

However, it was not known that the reverse is true. While in this work we were not able to establish this, we were able to prove an equivalence in the polynomial-stretch regime for any circuit class containing AC^0.

Specifically, the following algorithm can be used in place of Algorithm 1 to obtain $\mathscr{C}$-Remote-Point algorithms from a suitable average-case lower bound.

[11] Typically, a strong average-case lower bound states that certain problems cannot be $1/2 + 1/s$-approximated by size-s circuits [CHLR23]

Algorithm 2: $\mathsf{Je\check{r}\acute{a}bek\text{-}Korten}^{\mathsf{Avg}}(C, f)$: Modified Jeřábek-Korten reduction for Remote-Point

Input: $C : \{0,1\}^n \to \{0,1\}^{n\cdot q}$ denotes the input circuit whose size is s_C, and $f \in \{0,1\}^\ell \backslash \mathrm{Range}(\mathsf{GGM}_{\ell,q,k}[C])$ denotes the input average-case hard truth table: let $\ell(n) = \ell$, and f cannot be $(1/2 + c(\ell^{-1}(n)))$-approximated by $\mathscr{C}$ circuits (such that $\mathsf{GGM}_{\ell,q,k}[C] \in \mathscr{C}$) of size $O(k \cdot s_C)$; `// assume that` $c(\cdot)$ `is a good function.`

Output: A string y that is $(1/2 - c(n))$-far from $\mathrm{Range}(C)$.

Data: A perfect q-ary tree of height k that contains the computational history.

```
for j ← 0 to q^k − 1 do
    v_{k,j} ← f_{[jn,(j+1)n)} ;                                  // set f to the leaves
end
for vertex (i, j) in the Post-Order Traversal do
    Set v_{i,j} be the lexicographically smallest string such that
     δ(C(v_{i,j}), v_{i+1,qj} ∘ v_{i+1,qj+1} ∘ ··· ∘ v_{i+1,qj+(q−1)}) ≤ 1/2 − c(n) ; // this step requires
     an NP oracle
    if v_{i,j} does not exist then
        Set all remaining vertices ⊥ ;
        return v_{i+1,qj} ∘ v_{i+1,qj+1} ∘ ··· ∘ v_{i+1,qj+(q−1)} ;
    end
end
return ⊥;
```

Applying Algorithm 2 to the proof of Theorem 3.3 yields the following theorem.

Theorem 3.6. *For any integer i and any monotone function $c : \mathbb{N} \to \mathbb{N}$ that is good, if $\mathbf{E}^{\mathbf{NP}}$ cannot be $(1/2 + c(\frac{2^n}{2}))$-approximated by near-maximum ($\Omega(2^n/n)$) size NC^{i+1} circuits, then there is an $\mathbf{FP}^{\mathbf{NP}}$ algorithms for NC^i-Remote-Point$[n, 2n, c(n)]$.*

Applying Algorithm 2 to the proof Theorem 3.4, we obtain the following theorem.

Theorem 3.7. *For any circuit class $\mathscr{C}$ such that $\mathsf{AC}^0 \subseteq \mathscr{C}$ and any monotone function $c : \mathbb{N} \to \mathbb{N}$ that is good, $\mathbf{E}^{\mathbf{NP}}$ cannot be $(1/2 + c(2^{\frac{n}{1+\varepsilon}}))$-approximated by $2^{\Omega(n)}$ size $\mathscr{C}$ circuits if and only if there is an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for $\mathscr{C}$-RPP$[n, n^{1+\varepsilon}, c(n)]$ for any constant $\varepsilon > 0$.*

This also extends to $\mathbf{EXP}^{\mathbf{NP}}$ circuit lower bound and $\mathbf{FQP}^{\mathbf{NP}}$ algorithms.

Theorem 3.8. *For any circuit class $\mathscr{C}$ such that $\mathsf{AC}^0 \subseteq \mathscr{C}$, $\mathbf{EXP}^{\mathbf{NP}}$ cannot be $(1/2 + c(2^{\frac{n}{1+\varepsilon}}))$-approximated by $2^{\Omega(n)}$ size $\mathscr{C}$ circuits if and only if there is an $\mathbf{FQP}^{\mathbf{NP}}$ algorithm for $\mathscr{C}$-RPP$[n, n^{1+\varepsilon}, c(n)]$ for any constant $\varepsilon > 0$.*

4 A Family of $2^{n^{1-\frac{\varepsilon}{k-1}+o(1)}}$ Time Algorithms for NC^0-Avoid$[n, n^{1+\varepsilon}]$

4.1 Algorithm

In this subsection, we present an improved subexponential-time algorithm for NC^0_k-Avoid$[n, n^{1+\varepsilon}]$.

Our algorithm operates by identifying a small Hall-violating subcircuit and solving the corresponding restricted Avoid instance. Specifically, we reduce the original instance to a smaller one of the form NC^0_k-Avoid$[n' - 1, n']$ where $n' = n^{1-\frac{\varepsilon}{k-1}}$, and then enumerate over the image of this small subcircuit. This yields a total runtime of $2^{n^{1-\frac{\varepsilon}{k-1}+o(1)}}$.

We begin by viewing the NC^0_k circuit $\mathcal{C}\colon \{0,1\}^n \to \{0,1\}^m$ as a degree-k left-regular bipartite graph between m output bits (left side) and n input bits (right side).

The key combinatorial fact we use is the following:

Lemma 4.1 (Lower bound from [RTS00]). *Let $G = (L = [M], R = [N], E)$ be a left-regular bipartite graph that is a $(K_0, \frac{N-K_0}{N})$-disperser. Then*

$$D = \bar{D} \geq \frac{\log(M/(K_0-1))}{\log(1/(1-\frac{N-K_0}{N}))+1} \geq \frac{\log(M/K_0)}{\log(N/K_0)+1}.$$

Rearranging the above, we obtain:

$$M \leq \frac{N^D}{K_0^{D-1}}.$$

Setting $K_0 = N^{1-\frac{\varepsilon}{D-1}}$, we get $M \leq N^{1+\varepsilon}$, which matches the stretch regime of interest.

Thus, any $\mathsf{NC}^0_k[n, n^{1+\varepsilon}]$ circuit must contain a subset of $K = n^{1-\frac{\varepsilon}{k-1}}$ outputs with fewer than K distinct neighbors, violating Hall's condition. Brute-force search can find such a subset and define a subcircuit $\mathcal{C}'$ of size K, which fails to be surjective. This leads to the following algorithm:

Algorithm 3: Improved Subexponential-Time Algorithm for NC^0-Avoid$[n, n^{1+\varepsilon}]$

Input: An NC^0_k circuit $\mathcal{C}\colon \{0,1\}^n \to \{0,1\}^m$, with $m \geq n^{1+\varepsilon}$ for some constant $\varepsilon > 0$.

Output: A set of strings $y_1, \ldots, y_\ell \in \{0,1\}^m$ such that $y_i \notin \mathrm{Range}(\mathcal{C})$.

1. Search over all subsets $S \subseteq [m]$ of size $K = n^{1-\frac{\varepsilon}{k-1}}$, and find one with $|\Gamma(S)| < |S|$ (guaranteed by Lemma 4.1). Let $\mathcal{C}'$ be the induced subcircuit.
2. Enumerate all $2^{|\Gamma(S)|}$ inputs and identify strings $y'_1, \ldots, y'_\ell \notin \mathrm{Range}(\mathcal{C}')$.
3. For each y'_i, construct $y_i \in \{0,1\}^m$ that agrees with y'_i on S and is * (representing arbitrary value) elsewhere.
4. Output $y_1, \ldots, y_\ell$.

Theorem 4.2. *Algorithm 3 runs in time $2^{n^{1-\frac{\varepsilon}{k-1}+o(1)}}$.*

Proof. In Step 1, we enumerate all $\binom{m}{K} \leq \left(\frac{em}{K}\right)^K = n^{\frac{k\varepsilon}{k-1}\cdot n^{1-\frac{\varepsilon}{k-1}}} = 2^{n^{1-\frac{\varepsilon}{k-1}+o(1)}}$ subsets. Step 2 performs $2^{n^{1-\frac{\varepsilon}{k-1}}}$ enumerations. Step 3 is linear in output size. Thus the total runtime is $2^{n^{1-\frac{\varepsilon}{k-1}+o(1)}}$. ☐

Remark 4.1. *When $\varepsilon = (k-1)\left(1 - \frac{\log\log n + O(1)}{\log n}\right)$, i.e., $m = n^k/\log^{k-1} n$, the algorithm runs in polynomial time.*

Tighter Bounds via Improved Disperser Assumption. If the disperser bound of Lemma 4.1 can be improved to:

$$M \leq \frac{N^{D-1}}{K_0^{D-2}}, \tag{4.1}$$

then setting $K_0 = N^{1-\frac{\varepsilon}{D-2}}$ again yields $M \leq N^{1+\varepsilon}$ (matching exactly the existence bound from Theorem 2.9), and the same algorithm applies.

Based on the above observation, we make the following assumption:

Assumption 4.3. *Let $G = (L = [M], R = [N], E)$ be a left-regular bipartite graph that is also a $(K_0, \frac{N-K_0}{N})$ disperser, then it holds that*

$$D - 1 = \bar{D} - 1 \geq \frac{\log(M/(K_0-1))}{\log(1/(1-\frac{N-K_0}{N}))+1} = \frac{\log(M/K_0)}{\log(N/K_0)+1} \ .$$

Theorem 4.4. *Suppose Assumption 4.3 is true, there exists a family of $2^{n^{1-\frac{\varepsilon}{k-2}+o(1)}}$ time algorithms for NC^0_k-Avoid$[n, n^{1+\varepsilon}]$. In particular, the family of algorithms runs in polynomial time for NC^0_k-Avoid$[n, n^{k-1}/\log^{k-2}(n)]$. In addition, the algorithm can output a succinct representation of $\geq 1/2$ fractions of strings outside the range.*

4.2 Implications for Local PRGs

Our subexponential-time Avoid algorithm has implications for local PRG constructions in NC^0.

Theorem 4.5. *Suppose there exists a $\mathscr{C}$-Avoid$[n, m(n)]$ algorithm that, in time 2^{n^γ}, outputs a succinct representation of a non-negligible fraction of non-image strings. Then no $\mathscr{C}$-based pseudorandom generator is 2^{n^γ}-secure.*

Proof. Let $\mathcal{C} \in \mathscr{C}$ be a PRG with output length $m(n)$. Let adversary $\mathcal{A}$ accept an input y iff $y \in \mathscr{C}$-Avoid$(\mathcal{C})$. Since the Avoid algorithm runs in time 2^{n^γ}, this gives a distinguisher that accepts at least $2^{m(n)-1}$ non-image strings but accepts none from the PRG, violating the security of the PRG. $\square$

Corollary 4.6. *Assuming the existence of $2^{m(n)^\beta}$-secure local PRGs in NC^0_k, there cannot exist an algorithm for NC^0_k-Avoid that runs in time 2^{n^γ} for any $\gamma < \beta$ and identifies a $\mathsf{negl}(n)$ fraction of non-image strings.*

5 A Faster Local Greedy Algorithm for NC^0_k-Avoid$[n, n+1]$

5.1 Algorithm

We present a simple greedy algorithm for NC^0_k-Avoid$[n, n+1]$ that runs in time

$$O\left(n \cdot 2^{\frac{(k-2)n}{k-1}}\right).$$

When $k = 2$, this yields a linear-time algorithm, matching the result of [GLW22].

Algorithm 4: Improved Greedy Algorithm for NC^0_k-Avoid$[n, n+1]$

Input: An NC^0_k circuit $\mathcal{C}\colon \{0,1\}^n \to \{0,1\}^m$, where $m \geq n+1$.
Output: A string $y \in \{0,1\}^m$, such that $y \notin \text{Range}(\mathcal{C})$.

while *there exists an unassigned output bit y_i and the input space is non-empty* **do**
  Assign a value to y_i such that the remaining preimage space is reduced by at least a factor of $1/2$;
end
if *all output bits are assigned* **then**
  return *the assigned output string;*
else
  Assign arbitrary values to unassigned bits and output the resulting string;
end

5.2 Analysis

Theorem 5.1. *Algorithm 4 solves* NC^0_k*-*Avoid$[n,m]$ *for* $m \geq n+1$ *in time* $O\left(n \cdot 2^{\frac{(k-2)n}{k-1}}\right)$.

Proof. We first argue that the algorithm always finds a valid non-image string. After at most n fixings of output bits, the input space is reduced to a singleton, so the output string obtained is guaranteed to lie outside the image of the circuit.

To analyze the running time of Algorithm 4, we model the input-output behavior of $\mathcal{C}$ via random variables:

- Let $X = (X_1, \ldots, X_n)$ denote the input bits,
- and $Y = (Y_1, \ldots, Y_m)$ denote the output bits.

Each output bit Y_i is computed as:

$$Y_i = f_i\left(X_{\sigma_i(1)}, \ldots, X_{\sigma_i(k)}\right),$$

where $f_i \colon \{0,1\}^k \to \{0,1\}$ is a Boolean function and $\sigma_i \colon [k] \to [n]$ indicates the input positions read.

A string $y \notin \mathrm{Range}(\mathcal{C})$ iff $H_\infty(\mathcal{C}^{-1}(y)) = 0$. Thus, the algorithm can be viewed as a process that reduces the min-entropy of X by successively fixing values of Y.

Let us define the following useful notion of *traversed space*.

Definition 5.1 (Traversed Space $\mathcal{T}(t)$)**.** *After fixing t output bits, the corresponding input space can be decomposed into mutually independent subspaces $T_1, \ldots, T_s$, each over disjoint sets of input variables. Define:*

$$\mathcal{T}(t) := \{T_1, \ldots, T_s\}, \quad w(\mathcal{T}(t)) := \max_{i \in [s]} |T_i|.$$

Claim 2. *For all t, we have $w(\mathcal{T}(t)) \leq 2^{(k-2)t+1}$.*

Proof. We proceed inductively. There are two main cases at each step:

Case 1: The inputs to the new output bit are disjoint from the inputs of all previously traversed output bits. In this case, the decision of which boolean value to assign to the current output bit only depends on a constant-sized space of 2^k values.

Case 2: Suppose $\ell \in (0,k]$ of the inputs to the new output bit overlap with previously seen input variables. Then, setting this output bit potentially increases the size of some traversed subspace. Specifically, the space increases by a factor of at most $2^{k-\ell}$, but since our choice of the fixing of the output bit always reduces the preimage size by at least half, the net increase is bounded by:

$$w(\mathcal{T}(t)) \leq w(\mathcal{T}(t-1)) \cdot 2^{k-\ell} \cdot \frac{1}{2} \leq 2^{(k-2)t+1},$$

by induction.

□

This shows that the traversed space grows at most exponentially with rate $(k-2)t$. On the other hand, fixing t output bits reduces the input space size to at most 2^{n-t}. The algorithm terminates once the traversed space size exceeds the input space size, which occurs when

$$2^{(k-2)t+1} \geq 2^{n-t} \quad \Longrightarrow \quad t \geq \frac{n}{k-1}.$$

Thus, the worst-case number of steps is $\frac{n}{k-1}$, and in each step we consider a subspace of size $2^{(k-2)t+1}$, yielding a total running time of $O\left(n \cdot 2^{(k-2)n/(k-1)}\right)$. □

5.3 Lower Bound

The following result shows that Algorithm 4 has exponential worst-case runtime, giving evidence of the intrinsic hardness of NC^0_k-Avoid$[n, O(n)]$.

Theorem 5.3. *Algorithm 4 runs in exponential time in the worst case for* NC^0_k-*Avoid*$[n, O(n)]$.

Proof. By Theorem 2.8, a random $\mathsf{NC}^0_k[n, O(n)]$ circuit is an $(\Omega(n), k-1-\varepsilon)$-bipartite expander with probability at least $1/2$, where ε is constant arbitrarily close to 0. Fix such a circuit. For an arbitrary subset of output bits of size $\Omega(n)$, the induced subgraph on inputs and outputs is nearly a tree, with only $O(1)$ cycles. This is the worst-case scenario in the above case analysis of Algorithm 4:

- there will be only a single subspace in $\mathcal{T}(t)$;
- there are almost no cycles in the subcircuit, there is no means to additively reduce the size of $\mathcal{T}(t)$.

These essentially imply that the upper bound on $w(\mathcal{T}(t))$ could be tight if at each step of the fixing we reduce the input space by roughly $1/2$. This happens in the following instances.

Assuming each predicate f_i is a random Boolean function (say, implemented by resilient functions), then when we iteratively fix each output bit, no matter which bit value we assign to the next unfixed bit, with high probability, the queried space increases by a factor of 2^{k-2}. Thus, the number of configurations to track grows exponentially, and the traversed space size reaches $2^{\Omega(n)}$.

From the output string's perspective, this means that every $\Omega(n)$-bit projection of the image is nearly uniform. Hence, no partial assignment over $\Omega(n)$ output bits can efficiently help identify a non-image string, and the algorithm explores exponentially many paths. □

Remark 5.1. *Note that no unconditional exponenetial-time lower bound can be shown for any* NC^0-*Avoid algorithms in the constant-stretch regime. Indeed, since* NC^0-*Avoid* $\in \mathbf{F\Sigma}_2$ *[Kor22], it follows that if* $\mathbf{P} = \mathbf{NP}$*, then* NC^0-*Avoid* $\in \mathbf{FP}$*. Thus, an unconditional exponential-time lower bound would imply* $\mathbf{NP} \neq \mathbf{P}$.

6 Conclusion and Open Problems

Open Problem 1. In [GLW22], it was shown that $\mathcal{C} = \mathsf{NC}^0_k[n, \Omega(n^{k-1})]$ cannot sample $O(1)$-almost pairwise independent distribution (and therefore also $O(1)$-biased distribution) under any input distribution. Therefore, one could use the support of any $O(1)$-biased distribution as a hitting set for the strings outside $\mathrm{Range}(\mathcal{C})$. On the other hand, it is known that the support size of any ε-biased distribution is $O(n^2/\varepsilon^2)$.

There is some slackness left in their method. Given circuits that cannot sample $1/\mathrm{poly}(n)$-biased distribution under any input distribution, the support of $1/\mathrm{poly}(n)$-biased distribution would also form a polynomial-sized hitting set for such circuits. The same approach could work for a stretch regime $m(n) < \Omega(n^{k-1})$ as long as $\mathsf{NC}^0_k[n, m(n)]$ cannot sample $1/\mathrm{poly}(n)$-biased distribution under any input distribution. In particular, this would yield an $\mathbf{FP^{NP}}$ algorithm for smaller stretch.

A Lower Bound. Note that one could use Vazirani's XOR lemma to show that $\mathsf{NC}^0_k[n, n+1]$ circuit cannot sample $2^{-(n+3)/2}$-biased distribution. Recall

Lemma 6.1 (Vazirani's XOR Lemma)**.** *Let $Z_1, \cdots, Z_m$ be 0-1 random variables that are ε-biased for linear tests. Then, this distribution of $(Z_1, \cdots, Z_m)$ is $\varepsilon \cdot 2^{m/2}$-close to uniform.*

Observe that $H_\infty(\mathcal{C}(\mu)) \le n$ while $H_\infty(\mathbf{U}_{n+1}) = n+1$. Therefore, we have $\mathcal{C}(\mu)$ is $(\le 1/2)$-close to uniform. Hence, it holds that $\varepsilon \le 2^{-(n+1)/2-1} = 2^{-(n+3)/2}$.

The above upper bound and lower bound lead to the following question.

Question 1. Identify the stretch regime of $m(n)$ where $\mathsf{NC}^0_k(\mu)$ circuits cannot sample $1/\mathrm{poly}(n)$-biased distribution for any input distribution μ.

Open Problem 2.

- **(Hardness)** Improve the stretch for the hardness of NC^0-Avoid problem: by [CL24], we know that NC^1-Avoid$[n, n+1] \notin$ **SearchNP**. Under randomized encoding techniques Theorem 2.1, this also implies that NC^0_4-Avoid$[n, n+1] \notin$ **SearchNP**. Can we prove that under plausible assumptions NC^0-Avoid$[n, O(n)] \notin$ **SearchNP**, or even for some small constant, NC^0-Avoid$[n, n^{1+\varepsilon}] \notin$ **SearchNP**.

- **(Algorithms)** In the work, we show that there is a $2^{n^{1-\frac{\varepsilon}{k-1}+o(1)}}$ time algorithm for NC^0_k-Avoid$[n, n^{1+\varepsilon}]$. Does there exist a $2^{n^{o(1)}}$ time algorithm for NC^0_k-Avoid$[n, n^{1+\varepsilon}]$ for some $\varepsilon > 0$? If so, then assuming ETH (Exponential Time Hypothesis) [IPZ98, IP01], NC^0_k-Avoid$[n, n^{1+\varepsilon}] \in$ **SearchNP**. In addition, we give a conditional $\mathbf{FP^{NP}}$ algorithm for NC^0-Avoid$[n, 2n]$, are there unconditional $\mathbf{FP^{NP}}$ algorithm for NC^0_k-Avoid$[n, n^{1+\varepsilon}]$ for some $\varepsilon > 0$

Open Problem 3. 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 1. *$\exists \delta$ s.t., $\mathbf{E^{NP}}$ requires 2^{n^δ} size ACC^0 circuit complexity if and only if there is an $\mathbf{FP^{NP}}$ 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 1 is true and leveraging on existing ACC^0 circuit lower bound against $\mathbf{E^{NP}}$ [Wil14, CLW20], the reduction directly yields an $\mathbf{FP^{NP}}$ 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 would need the depth of the tree to be super-constant.

Open Problem 4. Recall that [Jeř04, Kor22, CHR24] proved the following equivalence result.

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

The second equivalence is a hardness amplification result. Is there such a similar amplification result for restricted circuit classes? Given Theorem 1.5 and that AC^0-Avoid algorithm for smaller stretch implies stronger lower bounds according to Theorem 2.7, the answer could be negative.

Acknowledgements

We would like to thank Hanlin Ren for helpful discussions. 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 [Kor22]. Accordingly, referring to it solely as "Korten's reduction" is not historically accurate.

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, USA, 1st edition, 2009. 10

[AIK06] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in NC^0. *SIAM Journal on Computing*, 36(4):845–888, 2006. doi:10.1137/S0097539705446950. 30, 31

[BHPT24] Amey Bhangale, Prahladh Harsha, Orr Paradise, and Avishay Tal. Rigid matrices from rectangular pcps. *SIAM Journal on Computing*, 53(2):480–523, 2024. doi:10.1137/22M1495597. 32

[Bus87] 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, page 123–131, New York, NY, USA, 1987. Association for Computing Machinery. doi:10.1145/28395.28409. 12

[CH85] 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, 28

[Che24] 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. 1

[CHLR23] 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, page 1058–1066, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3564246.3585147. 1, 2, 6, 18

[CHR24] 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, page 1990–1999, New York, NY, USA, 2024. Association for Computing Machinery. 2, 3, 13, 14, 24

[CL23] Eshan Chattopadhyay and Jyun-Jie Liao. Hardness against linear branching programs and more. In *Proceedings of the Conference on Proceedings of the 38th Computational Complexity Conference*, CCC '23, Dagstuhl, DEU, 2023. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. 1

[CL24] 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, page 620–629, New York, NY, USA, 2024. Association for Computing Machinery. 1, 2, 24, 32

[CLW20] 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. 1, 18, 24

[CR22] 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. 1

[CW89] 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. 13

[Ebe84] W. Eberly. Very fast parallel matrix and polynomial arithmetic. In *25th Annual Symposium onFoundations of Computer Science, 1984.*, pages 21–30, 1984. 36

[Erd47] 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. 1

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, August 1986. doi:10.1145/6490.6503. 14

[GGNS23] 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. 2, 5, 6, 8, 10, 11, 32

[GLW22] 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. 2, 9, 11, 21, 23, 32

[Has86] J Hastad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, STOC '86, page 6–20, New York, NY, USA, 1986. Association for Computing Machinery. doi:10.1145/12130.12132. 5

[ILW23] 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, page 1076–1089, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3564246.3585187. 2

[IP01] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. 24

[IPZ98] 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. 24

[IW97] 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, page 220–229, New York, NY, USA, 1997. Association for Computing Machinery. doi:10.1145/258533.258590. 3

[Jeř04] 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. 3, 6, 7, 13, 24

[JLS21] 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, page 60–73, New York, NY, USA, 2021. Association for Computing Machinery. 2, 13

[KKMP21] 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. 1

[Kor22] O. Korten. The hardest explicit construction. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 433–444, Los Alamitos, CA, USA, 2022. IEEE Computer Society. 1, 3, 6, 7, 13, 17, 23, 24, 25, 35

[Kor25] 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. 4

[KPI25] O. Korten, T. Pitassi, and R. Impagliazzo. Stronger cell probe lower bounds via local prgs. In *Electron. Colloquium Comput. Complex.*, 2025. 2, 6

[Kra92] 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. 6

[KS25] N. Kuntewar and J. Sarma. Range avoidance in boolean circuits via turan-type bounds. In *Electron. Colloquium Comput. Complex.*, 2025. 2

[KvM02] 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. 3, 13

[Li23] Xin Li. Two Source Extractors for Asymptotically Optimal Entropy, and (Many) More . In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1271–1281, Los Alamitos, CA, USA, November 2023. IEEE Computer Society. doi:10.1109/FOCS57990.2023.00075. 35

[Li24] 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, page 2000–2007, New York, NY, USA, 2024. Association for Computing Machinery. 2, 6, 13

[LZ24] 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. 1

[MVW99] 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, page 210–220, Berlin, Heidelberg, 1999. Springer-Verlag. 2

[RSW22] 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. 1, 2, 3, 4, 6, 11, 12, 13, 30, 31, 32, 33, 36

[RTS00] 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. 8, 13, 20

[Sip86] M Sipser. Expanders, randomness, or time versus space. In *Proc. of the Conference on Structure in Complexity Theory*, page 325–329, Berlin, Heidelberg, 1986. Springer-Verlag. 13

[Vad12] Salil P. Vadhan. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012. doi:10.1561/0400000010. 12

[Val77] Leslie G Valiant. Graph-theoretic arguments in low-level complexity. In *International Symposium on Mathematical Foundations of Computer Science*, pages 162–176. Springer, 1977. 10, 11

[VW23] 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. 13

[Wil14] Ryan Williams. Nonuniform ACC circuit lower bounds. *J. ACM*, 61(1), January 2014. doi:10.1145/2559903. 24

A Universality Property of Low-Depth Circuits

The following theorem is implicit in [CH85].

Theorem A.1. *Any circuit class containing* AC^0 *has the universality property.*

Proof. We show that for any circuit $C \in \mathscr{C}_{n,s,d}$, where $\mathscr{C}$ is any circuit class containing AC^0, there exists a circuit $U_{n,s,d} \in \mathscr{C}$ that satisfies the three conditions of the universality property as defined in Definition 2.5.

We first need the following definition about the succinct encoding of C.

Definition A.1 (Encoding Format (Size $O(s \log s)$))**.** *Let the circuit C have n inputs, m gates, s wires (i.e., total* ***fan-in*** *across all gates is s), and depth d. We encode the circuit as a list of gates: Each gate descriptor includes:*

- **Gate type:** 2–3 *bits.*
- **List of fan-in wires:** *each wire is indexed by a $\log s$-bit value pointing to: either an input x_i, or another gate g_j.*

Note that the number of bits for the gate is:

$$O(1 + (\text{fan-in}) \cdot \log s)$$

Summing over all gates:

$$\sum_{\text{gates}} \text{fan-in}(g) = s \quad \Longrightarrow \quad \text{Total encoding size} = O(s \log s)$$

Then the following universal circuit construction applies.

General Universal Circuit Construction for $\mathscr{C} \supseteq \mathsf{AC}^0$. Consider the following set-up of parameters:

- Input size: n
- Wire bound: s
- Depth bound: d (can be constant or more, depending on the class)

Let C be any circuit in $\mathscr{C}$ with those bounds. We construct a *universal circuit* $U_{n,s,d}$ with the following properties:

Inputs:

- $x_1, \ldots, x_n$: regular inputs
- $\langle C \rangle$: an encoding of a circuit C of size (wires) $\leq s$, depth $\leq d$, using a total of $O(s \log s)$ bits

Outputs:

- The output(s) of the simulated circuit $C(x)$

Universal Gate Module. For each gate in the simulated circuit, the universal circuit will include a *universal gate module* that:

- **Reads** the gate type from the encoding
- **Selects** the inputs using a list of $\log s$-bit selectors
- **Evaluates** the function ($\wedge$, $\vee$, $\neg$) as per the encoding

Input selection is done via a *selector tree* or *multiplexer* circuit using control bits from the encoding. This works in any class that can simulate a selector (e.g., AC^0).

Layered Construction (Depth-Universal Simulation). For a depth-d circuit C, simulate it layer-by-layer:

- Build d layers in the universal circuit
- Each layer contains $O(s)$ universal gate modules
- Layer i reads inputs from layer $i-1$ or from the original inputs

This preserves depth:

- If $\mathscr{C}$ has constant depth, depth remains constant
- If $\mathscr{C}$ allows polylog-depth, so does $U_{n,s,d}$

Final Construction: Universal Circuit $U_{n,s,d}$. Let $\mathscr{C}$ be any circuit class containing AC^0, and let s and d be polynomially bounded functions of n.

Then we can construct a uniform family of universal circuits $\{U_{n,s,d}\}$ such that:

- Each $U_{n,s,d}$ has:
  - n regular inputs
  - $O(s \log s)$ encoding inputs
  - $O(s)$ auxiliary gates
  - Depth $O(d)$
- For any circuit $C \in \mathscr{C}$ with n inputs, $\leq s$ wires, and depth $\leq d$, and for any input $x \in \{0,1\}^n$, we have:

$$U_{n,s,d}(x, \langle C \rangle) = C(x)$$

This universal circuit simulates *any circuit from* $\mathscr{C}$ with specified resource bounds, given only its *succinct encoding* and input.

$\square$

B NC^{i+1}-AVOID$[n, n+1] \leq_{\mathbf{FP}} \mathsf{NC}^i$-AVOID$[n, n+1]$

[RSW22] showed Theorem 2.1 (NC^1-Avoid$[n, n+1] \leq_{\mathbf{FP}} \mathsf{NC}^0$-Avoid$[n, n+1]$) based on the fact that every function in NC^1 has a *perfect randomized encoding* in NC^0_4 [AIK06]. Below we first recall the definition of perfect randomized encoding and then extend the results in [AIK06] to NC Hierachy. The simulation result NC^{i+1}-Avoid$[n, n+1] \leq_{\mathbf{FP}} \mathsf{NC}^i$-Avoid$[n, n+1]$ follows from the same proof strategy in [RSW22].

Definition B.1. *Let $\ell = \ell(n)$, $m = m(n)$ be good functions, and consider functions*

$$f_n : \{0,1\}^n \to \{0,1\}^\ell \text{ and } \hat{f}_n : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^{\ell+m}.$$

We say that $\hat{f}$ is a perfect randomized encoding of f if there is a polynomial-time computable decoder $\mathsf{Dec} : \{0,1\}^{\ell+m} \to \{0,1\}^\ell$ such that for every $x \in \{0,1\}^n$ and $y \in \{0,1\}^{\ell+m}$, $f(x) = \mathsf{Dec}(y)$ iff there is $r \in \{0,1\}^m$ such that $\hat{f}(x, r) = y$.

Theorem B.1 (Recursive Perfect Randomized Encodings for NC Hierarchy)**.** *For any integer i, any function $f \in \mathsf{NC}^{i+1}$ admits a perfect randomized encoding computable in NC^i. That is,*

$$\mathsf{NC}^{i+1} \subseteq \mathrm{PREN}(\mathsf{NC}^i),$$

where $\mathrm{PREN}(\mathscr{C})$ denotes the class of functions that have a perfect randomized encoding computable in the circuit class $\mathscr{C}$.

Proof Sketch. We proceed by induction on i.

- **Base Case** ($i = 0$)**:** Applebaum, Ishai, and Kushilevitz [AIK06] construct a perfect randomized encoding in NC^0 for every function in NC^1 via $\oplus$ branching programs and randomizing polynomials.

- **Inductive Step:** Suppose the claim holds for some $i \geq 0$; that is, $\mathsf{NC}^{i+1} \subseteq \mathrm{PREN}(\mathsf{NC}^i)$. Let $f \in \mathsf{NC}^{i+2}$. Since NC^{i+2} circuits can be composed from polynomially many NC^{i+1} subcircuits, write
$$f(x) = C_{\text{top}}(C_1(x), \ldots, C_m(x)),$$
where $C_j, C_{\text{top}} \in \mathsf{NC}^{i+1}$. By the inductive hypothesis, each C_j has a perfect randomized encoding $\widehat{C_j}(x, r_j)$ in NC^i. Let $y_j := C_j(x)$ and define a perfect randomized encoding $\widehat{C_{\text{top}}}(y_1, \ldots, y_m, r_{\text{top}})$ for C_{top} using the inductive hypothesis again.

  Define the randomized encoding of f as:
$$\hat{f}(x, r_1, \ldots, r_m, r_{\text{top}}) := \widehat{C_{\text{top}}}(\widehat{C_1}(x, r_1), \ldots, \widehat{C_m}(x, r_m), r_{\text{top}}).$$

  Since composition of perfect randomized encodings preserves all the property of perfect randomized encodings [AIK06, Lemma 4.11], the result $\hat{f}$ is a perfect randomized encoding of f in NC^{i+1}.

Hence, we conclude that for any integer i, we have

$$\mathsf{NC}^{i+1} \subseteq \mathrm{PREN}(\mathsf{NC}^i).$$

□

Corollary B.2 (NC^i-Avoid with strong parameters simulates NC^{i+1}-Avoid)**.** *There is a polynomial time reduction from NC^{i+1}-Avoid$[n, n+1]$ to NC^i-Avoid$[n, n+1]$*

Proof Sketch. The proof follows from the same proof strategy in [RSW22, Theorem 5.8] given Theorem B.1.

For any integer i, let $f : \{0,1\}^n \to \{0,1\}^\ell$ be the input of the range avoidance problem where each output gate of d can be computed by an NC^{i+1} of size $s = \mathrm{poly}(n)$. Let $m := \mathrm{poly}(n, \ell, s) \leq \mathrm{poly}(n)$. Let the function $\hat{f} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^{\ell+m}$, where $\hat{f} \in \mathsf{NC}^i$ is the randomized encoding of f from Theorem B.1. Let y be a non-output of $\hat{f}$, then $z = \mathsf{Dec}(y)$ is a non-output of f. Since Dec is computable in polynomial-time, the reduction works in polynomial time. Setting $\ell = n+1$ completes the proof. □

C Reductions Between Avoid Instances via Direct-Sum

In this section, we present a reduction between instances of $\mathscr{C}$-Avoid, focusing on how to relate instances with varying input/output lengths.

We present a direct-sum-type reduction that improves upon prior reductions in the literature.

Theorem C.1. *For any constant $\delta \in (0,1)$ and any circuit class $\mathscr{C}$, it holds that*

$$\mathscr{C}\text{-}\textsc{Avoid}[n, n+n^{\delta}] \leq_{\mathbf{FP}^{\mathbf{NP}}} \mathscr{C}\text{-}\textsc{Avoid}[n, n+1].$$

Specializing to $\mathscr{C} = \mathsf{NC}^0_k$, this reduction yields several consequences when combined with results from [RSW22, GLW22, GGNS23].

For instance, [GGNS23] showed that explicitly constructing rigid matrices sufficient for Valiant's program reduces to NC^0_3-Avoid$[n, n+n^{2/3}]$. Moreover, improving the current $\mathbf{FP}^{\mathbf{NP}}$ constructions of rigid matrices [BHPT24] would follow from an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for NC^0_3-Avoid$[n, n+n^{12/17-\varepsilon}]$ for any constant $\varepsilon > 0$.

By Theorem C.1, we obtain that even solving NC^0_3-Avoid$[n, n+n^{\delta}]$ for any constant $\delta \in (0,1)$ is already sufficient to yield such constructions — though this suggests that doing so is likely as hard as solving the hardest case which has the minimum stretch NC^0_3-Avoid$[n, n+1]$, a stretch regime believed to lie beyond **SearchNP** [CL24].[12]

This reduction also applies to other explicit construction problems reducible to small-stretch NC^0_k-Avoid, including:

- constructing binary linear codes approaching the Gilbert–Varshamov bound,
- list-decodable codes achieving list-decoding capacity,
- optimal Ramsey graphs.[13]

Hence, this result is both a positive and negative message: on the one hand, it shows the potential power of solving small-stretch Avoid instances; on the other hand, it aligns with the growing evidence that these instances are unlikely to be in **SearchNP**.

In the following, we present the proof of Theorem C.1.

Proof of Theorem C.1. Construct $s = n^{d/(d+1)}$ copies of $\mathcal{C} \in \mathscr{C}$ of input size $n^{1/(d+1)}$, each with stretch $n^{1/(d+1)} + 1$. Concatenating them yields a circuit $\mathcal{C}'$ with input size n and output size $n + n^{d/(d+1)}$. Given $y \notin \text{Range}(\mathcal{C}')$, we can partition y into s equal-sized blocks and use an **NP**-oracle to find a block not in $\text{Range}(\mathcal{C})$ in time $O(s)$. $\square$

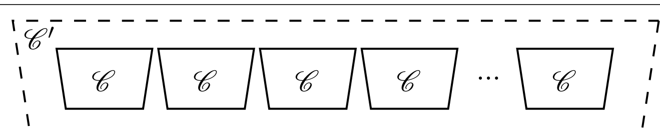

Figure 4: Concatenating small instances (circuits) with small stretch to a larger instance (circuit) with larger stretch.

[12] Precisely speaking, [CL24] only shows that it is likely that $\mathsf{NC}^0_4[n, n+1]$-Avoid $\notin$ **SearchNP**.

[13] While we are not aware of a formal reduction for Ramsey graphs in the literature, we provide one in Appendix F.

D Missing Proofs

D.1 Proof of Theorem 1.2

We restate Theorem 1.2:

Theorem D.1. *For any constant $\delta \in (0,1)$ and any integer i, an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for NC^i-AVOID$[n, n+n^\delta]$ implies that $\mathbf{E}^{\mathbf{NP}}$ requires $\Omega(2^n/n)$-size NC^{i+1} circuits.*

Proof. By Theorem C.1, an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for NC^i-AVOID$[n, n+n^\delta]$ implies an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for NC^i-AVOID$[n, n+1]$. Therefore, it suffices to prove the result assuming such an algorithm exists for NC^i-AVOID$[n, n+1]$.

Moreover, by Corollary B.2, we have a polynomial-time reduction:

$$\mathsf{NC}^{i+1}\text{-AVOID}[n, n+1] \leq_{\mathbf{FP}} \mathsf{NC}^i\text{-AVOID}[n, n+1],$$

so it suffices to assume an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for NC^{i+1}-AVOID$[n, n+1]$.

We now restate and prove a version of the implication of $\mathscr{C}$-AVOID algorithms to circuit lower bounds based on *universality property* of the circuit classes from [RSW22], with tightened parameters.

Theorem D.2 (Refinement of Theorem 5.2 from [RSW22])**.** *Let $\mathscr{C}$ be any circuit class that has the universality property, and $f : \mathbb{N} \to \mathbb{N}$ be a monotone function that is good. Suppose there is an $\mathbf{FP}^{\mathbf{NP}}$ (resp. $\mathbf{FP}$, $\mathbf{FQP}^{\mathbf{NP}}$) algorithm for $\mathscr{C}$-REMOTE-POINT$[N, f(N), c(N)]$, where each output gate has $\mathscr{C}$ 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 $\mathscr{C}$ circuits of size $\frac{\varepsilon f^{-1}(2^n)}{\log f^{-1}(2^n)}$.*

Proof. Consider the truth table mapping:

$$\mathsf{TT}_{\mathscr{C}} \colon \{0,1\}^N \to \{0,1\}^{2^n},$$

which maps the encoding $\langle C \rangle$ of a single-output $\mathscr{C}$ circuit of size $s = s(n)$ to its truth table. By the universality of $\mathscr{C}$, there exists a constant c such that $N = O(s \log s)$. In particular,

$$N \leq f^{-1}(2^n) \cdot \left(1 - \frac{\log\log f^{-1}(2^n)}{\log f^{-1}(2^n)}\right) < f^{-1}(2^n),$$

for sufficiently large n.

Thus, the output length 2^n satisfies:

$$2^n > f(N).$$

Moreover, each output bit of $\mathsf{TT}_{\mathscr{C}}$ can be computed by a $\mathscr{C}$ circuit of size $\mathrm{poly}(N)$, since evaluating C on any input is efficient by assumption.

Applying the $\mathbf{FP}^{\mathbf{NP}}$ algorithm for $\mathscr{C}$-AVOID$[N, f(N)]$, we can find a string $y \notin \mathrm{Range}(\mathsf{TT}_{\mathscr{C}})$. This string represents the truth table of a Boolean function $f \colon \{0,1\}^n \to \{0,1\}$ that cannot be computed by any $\mathscr{C}$ circuit of size s. Since the AVOID algorithm runs in $\mathbf{FP}^{\mathbf{NP}}$, the function f is in $\mathbf{FE}^{\mathbf{NP}}$.

By the definition of $\mathscr{C}$-REMOTE-POINT$[N, f(N), c(N)]$, the output of the algorithm on the instance C, which we call y, has relative hamming distance $\geq 1/2 - c(N)$ from $\mathrm{Range}(C)$. Then it holds that $\mathrm{Range}(C)$ and y agrees on $\leq 1/2 + c(f^{-1}(2^n))$ fraction of inputs. $\square$

Finally, since NC^{i+1} satisfies the universality property by Theorem 2.7, applying the above theorem with $\mathscr{C} = \mathsf{NC}^{i+1}$ and $f(N) = N+1$ implies that an $\mathbf{FP}^{\mathbf{NP}}$ algorithm for NC^{i+1}-AVOID$[n, n+1]$ yields a function in $\mathbf{E}^{\mathbf{NP}}$ requiring circuit size $\Omega(2^n/n)$ in NC^{i+1}, as desired. $\square$

E Bipartite Vertex Expanders in Various Parameter Regimes

E.1 Proof of Theorem 2.8

We restate Theorem 2.8 for convenience.

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

Proof. We generate a uniformly random graph $G \leftarrow \mathsf{Bip}_{n,m,D}$ by independently selecting D random neighbors on the right for each left vertex $v \in [m]$.

Let p_K denote the probability that there exists a subset $S \subseteq [m]$ of size $|S| = K \leq \alpha n$ whose neighborhood $N(S)$ has size less than $(D-1-\varepsilon)K$. Fix such an S, and consider the multiset $V_1, \dots, V_{KD} \in [n]$ of all neighbors of vertices in S, chosen independently with replacement.

We define V_i to be a *repeat* if $V_i \in \{V_1, \dots, V_{i-1}\}$. Then, for all i, even conditioned on $V_1, \dots, V_{i-1}$, the probability that V_i is a repeat is at most $(i-1)/n \leq KD/n$.

Hence, the number of repeats among $V_1, \dots, V_{KD}$ stochastically dominates the number of collisions in a balls-and-bins process with KD balls and n bins. Therefore,

$$\begin{aligned}\mathbf{Pr}\left[|N(S)| \leq (D-1-\varepsilon)K\right] &\leq \mathbf{Pr}\left[\text{At least } (1+\varepsilon)K \text{ repeats among } V_1, \dots, V_{KD}\right] \\ &\leq \binom{KD}{(1+\varepsilon)K}\left(\frac{KD}{n}\right)^{(1+\varepsilon)K}.\end{aligned}$$

Now summing over all such sets S, we obtain:

$$\begin{aligned} p_K &\leq \binom{m}{K}\binom{KD}{(1+\varepsilon)K}\left(\frac{KD}{n}\right)^{(1+\varepsilon)K} \\ &\leq \left(\frac{me}{K}\right)^K\left(\frac{De}{1+\varepsilon}\right)^K\left(\frac{KD}{n}\right)^{(1+\varepsilon)K} \\ &= \left(\frac{K^\varepsilon \cdot me^2D^{2+\varepsilon}}{(1+\varepsilon)n^{1+\varepsilon}}\right)^K \leq \left(\frac{\alpha^\varepsilon \cdot me^2D^{2+\varepsilon}}{(1+\varepsilon)n}\right)^K, \end{aligned}$$

where in the last step we used the assumption that $K \leq \alpha n$. Since $m = O(n)$, choosing α small enough ensures $p_K \leq 4^{-K}$. Therefore,

$$\mathbf{Pr}_{G\sim\mathsf{Bip}_{n,m,D}}[G \text{ is not an } (\alpha n, D-1-\varepsilon) \text{ expander}] \leq \sum_{K=1}^{\lceil \alpha n\rceil} 4^{-K} < \frac{1}{2}. \tag{E.1}$$

□

E.2 Proof of Theorem 2.9

We restate Theorem 2.9.

Theorem E.2 (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 $1/2$.*

Proof. The argument closely follows the proof of Theorem 2.8. Fix a subset $S \subseteq [m]$ of size K, and consider its multiset of neighbors. The probability that $N(S) \leq (D-1)K$ is at most the probability that there are at least $(D-1)K$ repeats among the KD chosen neighbors.

Using the same reasoning as above:

$$\begin{aligned} p_K &\leq \binom{m}{K}\binom{KD}{(D-1)K}\left(\frac{KD}{n}\right)^{(D-1)K} \\ &\leq \left(\frac{me}{K}\right)^K \left(\frac{KDe}{(D-1)K}\right)^{(D-1)K} \left(\frac{KD}{n}\right)^{(D-1)K} \\ &= \left(\frac{e^D D^{D+1} K^{D-2} m}{(D-1)^{D-1} n^{D-1}}\right)^K . \end{aligned}$$

Now, since $m = n^{1+\beta}$, this quantity becomes small as long as

$$K \ll \left(\frac{n^{D-1}}{m}\right)^{1/(D-2)} = n^{1-\frac{\beta}{D-2}}.$$

Thus, for all $K \leq A := n^{1-\beta/(D-2)}$, we get $p_K \leq 4^{-K}$. As before, summing over $K \leq A$ implies that with high probability, the graph is an $(A, 1)$-vertex expander. $\square$

F Reducing Explicit Construction of Optimal Ramsey Graphs to NC^0_4-Avoid

The current state-of-the-art explicit construction of a $(\log^{O(1)} n)$-Ramsey graph is due to [Li23]. It is well-known that an explicit construction of a two-source extractor with parameters $(\log n + 2\log(1/\varepsilon(n)) + 3, \varepsilon(n))$ and constant error $\varepsilon(n) = O(1)$ would imply an explicit $O(\log n)$-Ramsey graph.

In this section, we show that constructing such two-source extractors can be reduced in polynomial time to the problem of finding strings outside the range of circuits in the class NC^0_4-Avoid. Our approach closely follows the strategy of [Kor22], who constructed circuits for Avoid instances.

Theorem F.1. *Let $\varepsilon(n)$ be any efficiently computable function satisfying $1/n^c < \varepsilon(n) < 1/2$ for some constant $c > 0$ and sufficiently large n. Then, the problem of explicitly constructing a $(\log n + 2\log(1/\varepsilon(n)) + 3, \varepsilon(n))$-two-source extractor reduces in polynomial time to NC^0_4-Avoid.*

Proof. The high-level idea is to encode a partial truth table of a candidate extractor on "bad" sources, i.e., sources on which the extractor fails to produce an ε-biased output. We then build a circuit that takes this partial truth table as input and computes the coefficients of a polynomial that interpolates exactly the points in the bad source. Any string outside the image of this circuit corresponds to a set of coefficients whose polynomial disagrees with every such bad source, thereby certifying the extractor as valid.

Consider the function $f : \{0,1\}^n \to \{0,1\}^n$ defined as:

$$f(x) = \sum_{i=1}^{2^{2k}} \alpha_i x^{i-1},$$

and define $g(x) = f(x) \bmod 2$, where arithmetic is over a suitable extension field.

The input to the circuit consists of:

1. The two sources X, Y, each of size 2^k, where each element is an n-bit string. These require $2 \cdot 2^k \cdot n = 2^{k+1}n$ bits.
2. A single bit $b \in \{0,1\}$ indicating the biased output value.
3. The coefficients β_i for encoding the outputs on bad sources, which require $2^{2k}(2n-1)$ bits.
4. A string $S \in \{0,1\}^{2^{2k}}$ of Hamming weight $(1/2 - \varepsilon) \cdot 2^{2k}$, specifying the support of the bad outputs. This can be encoded using at most $2^{2k}(1-\varepsilon^2) + \log(2^{2k})$ bits (via standard entropy bounds).

The total number of *input bits* is:

$$2^{k+1}n + 1 + 2^{2k}(2n-1) + 2^{2k}(1-\varepsilon^2) + 2k.$$

The number of *output bits* is:

$$2^{2k} \cdot n,$$

corresponding to the full truth table of $f(x)$.

By choosing parameters such that:

$$2^{2k}\varepsilon^2 - 2k - 1 - 2^{k+1}n > 0,$$

we ensure that the number of inputs is strictly less than the number of outputs, making the construction amenable to the AVOID framework.

Computing the coefficients α_i from the evaluations of $f(x)$ can be done via polynomial interpolation, specifically by inverting a Vandermonde matrix. This procedure is known to be in NC^1 [Ebe84]. Finally, by applying the known reduction from NC^1-AVOID to NC^0_4-AVOID given in [RSW22], we conclude that explicitly constructing optimal two-source extractors (and thus optimal Ramsey graphs) reduces to NC^0_4-AVOID. □

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il