Range Avoidance for Low-depth Circuits and Connections to Pseudorandomness

Venkatesan Guruswami [*] Xin Lyu[†] Xiuhan Wang[‡]

Abstract

In the range avoidance problem, the input is a multi-output Boolean circuit with more outputs than inputs, and the goal is to find a string outside its range (which is guaranteed to exist). We show that well-known explicit construction questions such as finding binary linear codes achieving the Gilbert-Varshamov bound or list-decoding capacity, and constructing rigid matrices, reduce to the range avoidance problem of log-depth circuits, and by a further recent reduction [Ren, Santhanam, and Wang, ECCC 2022] to NC^0_4 circuits where each output depends on at most 4 input bits.

On the algorithmic side, we show that range avoidance for NC^0_2 circuits can be solved in polynomial time. We identify a general condition relating to correlation with low-degree parities that implies that any almost pairwise independent set has some string that avoids the range of every circuit in the class. We apply this to NC^0 circuits, and to small width CNF/DNF and general De Morgan formulae (via a connection to approximate-degree), yielding non-trivial small hitting sets for range avoidance in these cases.

Keywords: Pseudorandomness, Explicit constructions, Low-depth circuits, Boolean function analysis, Hitting sets.

[*]UC Berkeley. Email: `venkatg@berkeley.edu`. Research supported in part by NSF CCF-2210823 and a Simons Investigator Award.

[†]UC Berkeley. Email: `xinlyu@berkeley.edu`.

[‡]Tsinghua University. Email: `wangxh19@mails.tsinghua.edu.cn`.

ISSN 1433-8092

1 Introduction

We study a basic computational problem in circuit analysis called the *range avoidance* problem (which we call AVOID henceforth): given the description of a multi-output Boolean circuit C mapping n input bits to $m := m(n) > n$ output bits[1], find a $y \in \{0,1\}^m$ that is outside the range of C (i.e., $C(x) \neq y$ for every $x \in \{0,1\}^n$). This is a total search problem that has been the subject of a few recent works [KKMP21, Kor21, RSW22], which highlight its significance and connections to central themes in computational complexity including circuit complexity, proof complexity, and pseudorandomness.

To gain some intuition about the problem, note that AVOID can be trivially solved by a Monte Carlo algorithm: a random guess would solve AVOID with probability $1 - 2^{n-m} \geq \frac{1}{2}$. There is also a straightforward $\mathsf{ZPP}^{\mathsf{NP}}$ algorithm for AVOID: the algorithm just repeatedly samples a string $y \in \{0,1\}^m$ and tests if $y \in \mathrm{Range}(C)$ by calling the NP oracle. Remarkably, the work by Korten [Kor21] showed that if we can deterministically solve AVOID, then we can obtain explicit constructions of many important objects in CS theory and mathematics, including Ramsey graphs, two-source extractors, rigid matrices, Boolean functions hard against polynomial-size circuits, etc. These reductions put AVOID in a central position among several notoriously hard explicit construction questions that have resisted attack for decades.

In this work, we study AVOID problem for low-depth Boolean circuits (in particular, NC^0 and NC^1 circuits). For every constant $k \geq 1$, we say a circuit $C : \{0,1\}^n \to \{0,1\}^m$ is an NC^0_k-AVOID instance, if each output bit of C depends on at most k input bits. Similarly, we say C is an NC^1_k instance, if each output bit of C can be computed by a $(k \log n)$-depth Boolean circuit of fan-in two. A recent work by Ren, Santhanam and Wang [RSW22] demonstrates some attractive motivations to study AVOID problem for these weak circuit models. In particular, they showed the following.

Theorem 1 (Theorem 5.8 of [RSW22]). *Suppose there is an* FP *(resp.* $\mathsf{FP}^{\mathsf{NP}}$*)*[2] *algorithm for* NC^0_4-AVOID. *Then the following statements are true.*

- *For every $k \geq 1$, there is an* FP *(resp.* $\mathsf{FP}^{\mathsf{NP}}$*) algorithm for* NC^1_k-AVOID.
- *For every $\varepsilon > 0$, there is a family of functions in* E *(resp.* E^{NP}*) that does not have Boolean circuits of depth $n^{1-\varepsilon}$.*

Item (1) shows that NC^0_4-AVOID is as hard as NC^1-AVOID. Item (2) shows that finding explicit Boolean functions hard against low-depth circuits can be reduced to NC^0_4-AVOID. Together, these connections demonstrate that studying AVOID for weak circuit classes is already challenging and fruitful. This suggests two new research directions to approach AVOID from above and below: (i) we can show the "usefulness" of AVOID for "weak" circuit classes by reducing further explicit construction problems to it, and (ii) starting from weak circuit classes such as NC^0_2, we can try to design algorithms for AVOID of increasingly powerful models. Ultimately, we aim for an AVOID algorithm for a circuit class expressive enough to capture some elusive explicit construction questions.

1.1 Our Results

In this work, we make progress on both directions mentioned above. On the one hand, we reduce a sample of famous explicit construction problems to NC^1-AVOID. This improves the previous

[1] The function $m(n)$ is called the *stretch* of the circuit.

[2] Recall that $\mathsf{FP}, \mathsf{FP}^{\mathsf{NP}}$ are function classes analogue of the decision problem classes $\mathsf{P}, \mathsf{P}^{\mathsf{NP}}$.

results by Korten [Kor21], who only showed reductions to AVOID of general polynomial-size circuits. Reducing the explicit construction problems to NC^1-AVOID makes them potentially more tractable.

On the other hand, towards solving AVOID of low-depth circuits *unconditionally*, we offer two approaches to design deterministic algorithms for AVOID of low-depth circuits. We give a simple deterministic algorithm for NC^0_2-AVOID, and a novel approach to construct *hitting sets* for AVOID instances. This is to say, for a class of circuits $\mathcal{C} \subseteq \{C : \{0,1\}^n \to \{0,1\}^m\}$ that satisfy certain conditions, we can deterministically construct a set $S \subseteq \{0,1\}^m$ of size $|S| = \mathrm{poly}(m)$, such that for every $C \in \mathcal{C}$, we have $S \not\subseteq \mathrm{Range}(C)$. Note that a hitting set construction implies an $\mathsf{FP}^{\mathsf{NP}}$ algorithm to solve AVOID of $\mathcal{C}$. It is incomparable to an FP algorithm, because the hitting set is *oblivious* to the actual circuit, and the same hitting set can work for a broad class of "weak" circuits.

In the following, we elaborate on our contributions and their implications.

1.1.1 Reductions to NC^1-AVOID

As our first set of results, we reduce a sample of famous and central explicit construction questions to NC^1_k-AVOID for constant k. In particular, we consider the following explicit construction tasks.

- Rigid matrices. A matrix $M \in \mathbb{F}_2^{n\times n}$ is called (ε, δ)-rigid, if one cannot reduce the rank of M to εn by alternating at most $\frac{\delta n^2}{\log n}$ entries in M. The motivation to study explicit constructions of rigid matrices is due to its connection to circuit lower bounds [Val77].
- Binary linear codes which meet the Gilbert-Varshamov bound (the best known rate vs. distance trade-off for binary codes which is achieved by random linear codes). This is an outstanding challenge that has been open for much of coding theory's history. Recently there has been impressive progress in the low-rate regime [TS17], but the general question remains a tantalizing challenge at the intersection of coding theory and pseudorandomness.
- Binary linear codes that achieve list-decoding capacity. While there are explicit codes over large alphabets that achieve list-decoding capacity (i.e., are decodable up to the information-theoretically largest fraction of worst-case errors with small lists) [GR08], the best known binary codes fall well short of achieving capacity [GR09].

We reduce these explicit construction questions to AVOID. We first define explicit construction problems in the complexity-theoretic language: let $\Pi \in \{$LINEAR CODE, LIST-DECODABLE CODE, RIGID MATRIX$\}$ be a property of algebraic objects. Define the Π-construction problem: given as input 1^n, output an object of size n that satisfies the property Π.

Theorem 2 (Informal)**.** *Suppose that for each $k \geq 1$, there is an* FP *(resp.* $\mathsf{FP}^{\mathsf{NP}}$*) algorithm for* NC^1_k-AVOID*. Then, there is an* FP *(resp.* $\mathsf{FP}^{\mathsf{NP}}$*) algorithm for Π-Construction for $\Pi \in \{$*LINEAR CODE, LIST-DECODABLE CODE, RIGID MATRIX$\}$.

Furthermore, by Theorem 1, the same conclusion holds if we assume the existence of an FP *(resp.* $\mathsf{FP}^{\mathsf{NP}}$*) algorithm for* NC^0_4-AVOID.

Our reductions for linear codes are new, and the reduction for rigid matrices improves a similar result in [Kor21], in the sense that we reduce the question to AVOID on *logarithmic-depth* circuits. Our technique is general enough that it can be applied to many other construction problems to give reductions to AVOID of low-depth circuits[3]. For brevity, we only present three representative examples in this paper.

Proof idea. All of the three reductions follow the same framework. To illustrate the idea, we briefly discuss the reduction for rigid matrices. We follow the idea of Korten [Kor21]. That is, we carefully construct a circuit $C : \{0,1\}^{n^2-1} \to \{0,1\}^{n^2}$, whose outputs, when interpreted as matrices in $\mathbb{F}_2^{n\times n}$, contain all "non-rigid" matrices. To design the circuit, note that if a matrix $M \in \mathbb{F}_2^{n\times n}$ is not rigid, then there is a way to compress the matrix. Namely, we can write $M = L \cdot R + S$ where L, R are $n \times \varepsilon n$ and $\varepsilon n \times n$ matrices, and S is a sparse matrix with only $\frac{\delta n^2}{\log n}$ entries being 1. Note that for $\varepsilon, \delta \in (0,1)$ sufficiently small, we can encode L, R, S with $2\varepsilon n^2 + 2\log n \cdot \frac{\delta n^2}{\log n} < n^2$ bits and recover M in polynomial time. In more detail, the encoding just stores L, R explicitly, and stores a list of $\frac{\delta n^2}{\log n}$ pairs $(x,y) \in [n]^2$, specifying the non-zero entries of S.

Given this encoding, the reduction to AVOID is simple: we design a circuit C as follows. The input to C is a tuple $(L, R, \mathcal{S})$, where L, R are $n \times \varepsilon n$ and $\varepsilon n \times n$ matrix, respectively. $\mathcal{S}$ is a list of $\frac{n^2}{\log n}$ pairs describing a sparse matrix S. Given the tuple, the circuit C computes the matrix $L \cdot R + S$. It is easy to see that the range of C includes every non-rigid matrix. Hence, we can construct a rigid matrix by finding a matrix outside the range of C. However, it is not clear from the reduction whether C can be implemented in logarithmic depth.

In fact, computing $L \cdot R$ can be done by a logarithmic circuit easily. If the matrix S is presented in its natural form as a square matrix, adding S to $L \cdot R$ is also easy. Therefore, the main bottleneck in this reduction is to recover the sparse matrix S from its short description $\mathcal{S}$. Note that using a short encoding of S is essential for the reduction, as we need to ensure that the input length is strictly smaller than n^2. Still, there is some room for manoeuvre: it is not necessary to encode S in an information-theoretically optimal way, and we *can* afford a certain amount of redundancy, as long as the overall number of bits to encode L, R, S is bounded by $n^2 - 1$.

Succincter comes into play. We achieve the improvement by utilizing techniques from succinct data structures (see, e.g., [Pat08, Yu20]). Succinct data structures allow storage of a data set using an amount of memory that is close to the information-theoretic lower bound, but they still allow for retrieving information efficiently. In particular, there is a classic data structure [Pat08], which can store an n-bit string of Hamming weight k using $\log\binom{n}{k} + O(n/\log^2 n)$ bits. Moreover, one can recover any bit of the string by querying at most $O(\log n)$ bits in the memory. This data structure perfectly fits our purpose: we can encode the sparse matrix S by the memory configuration[4] of the data structure storing S, which is denoted by $\mathcal{S}'$ in the following. Then, we can recover each entry of S by querying $O(\log n)$ bits in $\mathcal{S}'$. By a simple construction (Lemma 26), this implies that each entry of S can be computed by a logarithmic-depth circuit given $\mathcal{S}'$.

Therefore, given L, R and $\mathcal{S}'$, there is a logarithmic-depth circuit C' that computes $L \cdot R + S$. The number of bits to describe $L, R, \mathcal{S}'$ is bounded by

$$2\varepsilon n^2 + \log\binom{n^2}{\frac{\delta n^2}{\log n}} + O(n^2/\log^2 n) < (1-\Omega(1))n^2 + O(n^2/\log^2 n) < n^2.$$

Hence, C' is a valid NC^1-AVOID instance, and any matrix outside the range of C' is $(\varepsilon n, \frac{\delta n^2}{\log n})$-rigid.

[3]However, we note that the reduction for two-source extractors in [Kor21] might be an exception. Still, by combining [Kor21] with our technique, one can reduce two-source extractor construction to NC^2-AVOID. i.e., each output can be computed by a Boolean circuit of depth $O(\log^2 n)$.

[4]In our application, we do not care about the complexity of preparing the data structure, as the AVOID problem asks one to avoid every output in the range of the circuit.

This completes the proof sketch for the rigid matrix reduction. Reductions for linear codes follow the same approach. Namely, every generator matrix M that fails to generate a desired code can be compressed, where the compression of M consists of a structured algebraic part A and a low-Hamming weight binary string B. The structured part A has an efficient encoding/decoding scheme, and the combination of A and B to recover M is also efficiently computable. Using a naive encoding scheme for B results in an inefficient (in terms of circuit depth) decoding procedure. Replacing the naive encoding scheme with the succinct data structure gives the desired efficient reduction.

1.1.2 Unconditional Algorithms for Avoid of Weak Circuits

On the positive side, we show an algorithm for NC^0_2-Avoid, as well a hitting set construction for solving Avoid of low-depth circuits and large stretch.

A polynomial time algorithm for NC^0_2-Avoid. When the given circuit $C : \{0,1\}^n \to \{0,1\}^m$ is in NC^0_2 (i.e., each output bit depends on only two input bits), we can solve Avoid of C by a simple deterministic polynomial-time algorithm.

Theorem 3. *There is a polynomial time algorithm which, given an NC^0_2 circuit $C : \{0,1\}^n \to \{0,1\}^m$ where $m > n$, outputs a string $y \in \{0,1\}^m$ that is not in the range of C.*

The idea behind Theorem 3 is simple. Let $C_1(x)$ be the first output bit of C. We observe that there is always a way to fix C_1 to a constant, so that we can reduce the problem to solving NC^0_2-Avoid for a smaller circuit $C' : \{0,1\}^{n-1} \to \{0,1\}^{m-1}$. To illustrate, suppose that $C_1(x)$ is an AND of two variables (say, x_1 and x_2). Then, by setting C_1 to 1, we have effectively restricted that x_1 and x_2 must be 1. Hence, we can replace every appearance of x_1, x_2 with constant 1 in C, and get a new NC^0_2-Avoid instance $C' : \{0,1\}^{n-2} \to \{0,1\}^{m-1}$. Suppose $y \in \{0,1\}^{m-1}$ is not in the range of C'. Then we claim that $1 \circ y$ (where $\circ$ denotes string concatenation) is not in the range of C. In fact, for $C_1(x)$ evaluating to 1, one has to set both x_1 and x_2 as 1. But then there is no way to find an input x where $C(x)_{2\ldots m} = C'(x) = y$.

The argument above illustrates one step of the reduction. To design an algorithm for NC^0_2-Avoid, we can recursively apply the reduction, until at one point where we are left with a circuit $C'' : \{0,1\}^0 \to \{0,1\}^{m-n}$. At this point, C'' always outputs a fixed string, while the number of possible outputs is $2^{m-n} > 1$, which allows us to solve Avoid for C'' trivially. Finally, we can backtrack to recover a string $y \in \{0,1\}^m$, which solves Avoid for the original circuit C.

Since the result in [RSW22] (see also Theorem 1) gives a strong evidence suggesting that solving NC^0_4-Avoid unconditionally is hard and would imply surprisingly strong circuit lower bounds, the strategy above probably fails to give an algorithm for NC^0_4-Avoid. Still, finding out the complexity of NC^0_3-Avoid remains an interesting question.

Approaching Avoid via hitting sets. We also introduce a novel technique for solving Avoid in $\mathsf{FP}^{\mathsf{NP}}$. Informally, we show that there is an $\mathsf{FP}^{\mathsf{NP}}$ algorithm for simple circuits if the stretch $m(n)$ is large enough. Here is the list of our results.

Theorem 4 (Informal)**.** *Let $m = m(n), s = s(n)$ be two non-decreasing functions and $k, w \geq 1$ be two constants. Suppose $C : \{-1,1\}^n \to \{-1,1\}^m$ is a multi-output function. There is an $\mathsf{FP}^{\mathsf{NP}}$ algorithm for $\text{Avoid}(C)$ if one of the following statements hold:*

- *Each output bit $C_i(x)$ depends on only k input bits and $m \geq 2^{4k+1}n^{k-1} + n$;*

- *Each output bit $C_i(x)$ is a width-w size-s CNF or DNF of input bits and $m \geq 32s^2n^w$;*
- *Each output bit $C_i(x)$ is a size-s De Morgan formula of input bits and $m \geq n^{\omega(\sqrt{s})}$;*
- *Each output bit $C_i(x)$ is a size-s DNF or CNF of input bits and $m \geq 2^{\omega(n^{1/2} \cdot \log(s))}$.*

Formally, our result is stronger than $\mathsf{FP}^{\mathsf{NP}}$ algorithm. Our construction is a hitting set which is independent of the circuit C. That is, we can output a set of polynomial size which always contains a solution for $\textsc{Avoid}(C)$, *without looking at the input circuit C*. We formally list our results here.

Theorem 5. *Let $m = m(n), s = s(n)$ be two non-decreasing functions and $k, w \geq 1$ be two constants. Suppose $C : \{-1,1\}^n \to \{-1,1\}^m$ is a multi-output function. The following statements hold.*

- *If each output bit $C_i(x)$ depends on only k input bits and $m \geq 2^{4k+1}n^{k-1} + n$, then there is a set $S \subseteq \{-1,1\}^m$ of size $2^{O(k)}m^2$ that is computable in polynomial time and satisfies $S \not\subseteq \mathrm{Range}(C)$.*
- *If each output bit $C_i(x)$ is a width-w size-s CNF or DNF of input bits and $m \geq 32s^2n^w$, then there is a set $S \subseteq \{-1,1\}^m$ of size $O(s^2 \log^2 m)$ that is computable in polynomial time and satisfies $S \not\subseteq \mathrm{Range}(C)$.*
- *If each output bit $C_i(x)$ is a size-s De Morgan formula of input bits and $m \geq n^{\omega(\sqrt{s})}$, then there is a set $S \subseteq \{-1,1\}^m$ of size $\mathsf{poly}(m)$ that is computable in polynomial time and satisfies $S \not\subseteq \mathrm{Range}(C)$.*
- *If each output bit $C_i(x)$ is a size-s DNF or CNF of input bits and $m \geq 2^{\omega(n^{1/2} \cdot \log(s))}$, then there is a set $S \subseteq \{-1,1\}^m$ of size $\mathsf{poly}(m)$ that is computable in polynomial time and satisfies $S \not\subseteq \mathrm{Range}(C)$.*

In all cases, the set S is independent *of the circuit C. Namely, only knowing m, n, s, k, w suffices to construct the set S.*

Perhaps surprisingly, we construct the hitting set by exploiting an interesting connection to pseudorandomness of distributions. In particular, we carry out a two-step plan as follows.

- For a class of simple circuits $\mathcal{C} \subseteq \{C : \{0,1\}^n \to \{0,1\}^m\}$, we show that if the stretch m is sufficiently large, then under any input distribution $\boldsymbol{x}$ over $\{0,1\}^n$, the output distribution $C(\boldsymbol{x})$ cannot be pairwise independent over $\{0,1\}^m$.
- On the other hand, we *can* sample a pairwise independent string of length m, with only $2 \log m$ truly random bits.

Putting two items together, we conclude that the support of a low-entropy pairwise independent distribution $\mathcal{D}$ over $\{0,1\}^m$ constitutes a hitting set for $\textsc{Avoid}$ of $\mathcal{C}$. Indeed, if the support of $\mathcal{D}$ is contained in $\mathrm{Range}(C)$ for some $C \in \mathcal{C}$, then we know that under a proper input distribution $\boldsymbol{x}$ over $\{0,1\}^n$, $C(\boldsymbol{x})$ can sample $\mathcal{D}$ perfectly, which leads to a contradiction to Item (1).

Here, Item (2) is standard [ABI86]. We achieve Item (1) by generalizing a technique by Mossel, Shpilka and Trevisan [MST06], where the authors showed that it is impossible for NC^0_3 circuits to expand n *uniformly random* bits into a $(4n+1)$-bit string that fools every linear test (i.e., the output fails to be a low-biased distribution). We generalize the [MST06] result by considering an arbitrary distribution (instead of uniform distribution) over inputs.

We briefly describe the high-level proof strategy below. We start with a simplicity measure of Boolean functions, parameterized by an integer $d \geq 1$ and a real $\delta \in (0,1)$. A function f :

$\{0,1\}^n \to \{0,1\}$ is called (d,δ)-simple, if under any distribution $\boldsymbol{x}$ over $\{0,1\}^n$, there is a parity test over a set $S \subseteq [n]$ of size $|S| \leq d$, such that

$$\left|\Pr_{\boldsymbol{x}}\left[f(\boldsymbol{x}) = \bigoplus_{q\in S} \boldsymbol{x}_q\right] - \frac{1}{2}\right| \geq \delta.$$

The following theorem shows our general template to construct hitting sets based on simplicity of functions.

Theorem 6. *Suppose $m > n \geq 2$. Let $C : \{0,1\}^n \to \{0,1\}^m$ be a circuit and $\varepsilon > 0$ be a parameter. Suppose each output bit C_i is a (d,ε)-simple function of input bits and $m > \frac{2}{\varepsilon^2}n^d$. Then, for every distribution $\boldsymbol{x}$ over the input space $\{0,1\}^n$, the output distribution $C(\boldsymbol{x})$ is not pairwise independent.*

We prove Theorem 6 following the technique of [MST06]. Let $\boldsymbol{x}$ be sampled from an arbitrary but fixed distribution. Since there are $m \geq \frac{2}{\varepsilon^2}n^d$ outputs and each output is correlated with a parity test on at most d inputs, by pigeonhole principle, there are at least $\frac{2}{\varepsilon^2}$ output bits that are ε-correlated with the same parity test. Then we follow [MST06] and carry out a second-moment argument, which shows that there is a pair of indices $i, j \in [m]$ among the $\frac{2}{\varepsilon^2}$ outputs, such that $C_i(\boldsymbol{x})$ and $C_j(\boldsymbol{x})$ have a correlation lower-bounded by $\frac{3}{8}\varepsilon^2$, meaning that $C(\boldsymbol{x})$ does not sample a pairwise independent distribution.

Note that the argument above also shows a lower bound of the correlation between two output bits. This allows us to use an *almost* pairwise independent distribution in the final construction, which makes the size of our hitting set even smaller. See Section 5 for the details.

Instantiating Theorem 6 with some canonical circuit classes, we deduce the results listed in Theorem 5.

- The results for NC^0_k circuits and constant-width DNF/CNFs are proved by ad-hoc but straightforward arguments. We remark that [MST06] has shown that every NC^0_k function is either an $\mathbb{F}_2$ polynomial of degree $\lceil k/2 \rceil$ or correlated with a parity test on at most $\lceil k/2 \rceil$ inputs under the *uniform* distribution of inputs. We managed to prove a correlation lower bound under arbitrary distributions, but we need to use parity tests on at most $(k-1)$ inputs, which in turn determines that our construction only works for NC^0_k-Avoid with stretch at least $\Omega(n^{k-1})$. Still this is non-trivial in the sense that prior to our work, even an algorithm for NC^0_k-Avoid with stretch $o(n^k)$ appears to not have been known.

- The results for unbounded-width CNF/DNFs and small-size De Morgan formulae are proved by relating the simplicity of functions to their (large-error) *approximate degree*, a central notion in complexity theory that has been studied extensively (see, e.g., [KS04, RS10, BT21]). Specifically, to show the simplicity of a function, it suffices (and, in some sense, is necessary) to find a low-degree polynomial over reals that point-wise approximates the function within a slightly non-trivial error (e.g. within error $\frac{1}{2} - \frac{1}{n}$)[5]. This connection allows us to translate known approximate degree upper bounds for CNF/DNF [KS04] and small-size De Morgan formulae [Rei11] to the simplicity of corresponding function classes.

Discussions. We find the connection to pseudorandomness quite interesting. In some sense, following Razborov and Rudich's natural proof [RR97], our argument establishes a separation

[5]Note that the polynomial $p(x) \equiv \frac{1}{2}$ trivially $\frac{1}{2}$-approximates every Boolean function.

result for weak circuits (with large stretches) by studying a natural property about *distributions*[6] over hypercubes. Namely, we consider the property of being a pairwise independent distribution. By standard pseudorandomness constructions [ABI86], there is a low-entropy distribution that attains this property easily, while our results rule out the possibility of sampling such distributions by weak circuit classes that only receive a short random seed, even if the random seed can come from an arbitrary distribution.

We leave it as an intriguing question to further explore the potential of this framework. Namely, can we identify more (pseudorandom) property of distributions, where there exists a low-entropy (and hopefully polynomial-time constructible) distribution with this property, but every weak circuit from a class $\mathcal{C}$ fails to sample a distribution with this property, even if its input distribution can be carefully tailored?

Note that the existence of such a "pseudorandom" property usually implies an efficient statistical test to distinguish the output of $\mathcal{C}$-circuits from uniform (in our example, this is a linear test on two output bits of $\mathcal{C}$). Thus, under the cryptography assumption that NC^1 circuits can compute PRG of polynomial stretch, it seems difficult to push this technique to NC^1. Still, we note there is a gap between our results and the best-known lower bounds and pseudorandomness results: for example, we know strong lower bounds and good PRGs against AC^0 (see e.g. [**?**, TX13, Lyu22]). Moreover, when the input distribution is uniform, we have very good sampling lower bounds against AC^0 circuits of quasi-polynomial stretches [Vio10, Vio20]. If, quantitatively, solving $\mathcal{C}$-AVOID is as hard as proving lower bounds/constructing PRGs for $\mathcal{C}$, then these results suggest that one should be able to solve AC^0-AVOID of (large) quasi-polynomial stretches. However, our result can only give a hitting set construction for AC^0 circuits of sub-exponential stretch[7]. We leave it as an interesting open question to close the gap between the known pseudorandomness results and our hitting sets. Namely, can we give better hitting sets for AC^0 circuits of smaller stretch, or is there any formal evidence suggesting that AVOID of low-end models (e.g., AC^0) is strictly harder than designing PRG for AC^0?

Comparison with previous works. Attempting to solve AVOID of weak circuits with large stretch, Ren, Santhanam and Wang [RSW22] presented an algorithmic framework in $\mathsf{FP}^{\mathsf{NP}}$, which is based on Williams' algorithmic method [Wil14] and rectangular PCPs [BHPT20]. Our framework is not directly comparable to theirs. A polynomial-size hitting set construction appears to be stronger than an $\mathsf{FP}^{\mathsf{NP}}$ algorithm, as a hitting set implies an $\mathsf{FP}^{\mathsf{NP}}$ algorithm in a straightforward way. But our assumption (the existence of a proper "natural property" of distributions) is incomparable to the assumption in [RSW22].

We note that [RSW22] also showed an $\mathsf{FP}^{\mathsf{NP}}$ algorithm for De Morgan formula-AVOID with stretch $m \geq 2^{\omega(\sqrt{s}\log(s))}$ as an application of their technique. To devise the algorithm, they also used the approximate degree upper bounds [Rei11] as a key technical ingredient. For this application, our result compares favorably with theirs. First, our hitting set construction is considerably simpler and can also handle a somewhat smaller stretch: the algorithm in [RSW22] needs a "constructive version" of the approximate degree upper bounds, which roughly says that one can deterministically find a degree-$(\sqrt{s}\log s)$ polynomial approximating a given size-s De Morgan formula. The $\log(s)$ overload in turn determines that their algorithm can only handle stretches

[6]This is in contrast with the typical notion of natural proofs, where natural properties of languages/Boolean functions are considered.

[7]We explicitly give a construction for depth-2 circuits (e.g., DNFs) with stretch $2^{n^{1/2}}$. It is easy to see that we can extend our results to depth-d AC^0 circuit of stretch roughly $2^{n^{1-\Omega(1/d)}}$ by the known approximate degree upper bound for AC^0.

larger than $n^{\omega(\sqrt{s}\log s)}$. In contrast, our solution only needs the *existence* of a low-degree approximate polynomial, enabling us to construct hitting sets for stretch $n^{O(\sqrt{s})}$. Second, the framework in [RSW22] cannot obtain a non-trivial algorithm from large-error ($\varepsilon = 1 - n^{-\Omega(1)}$) approximate degree. In particular, their framework does not naturally apply to polynomial-size DNF/CNFs as our result does.

1.2 Conclusion & Open Questions

In this work, we study the range avoidance problem for low depth circuits. We reduce some explicit construction challenges to the range avoidance problem of NC^0_4 circuits. On the algorithmic side, we give a polynomial time algorithm for NC^0_2-range avoidance. We also introduce a hitting set construction for the range avoidance problem of weak circuit classes with large stretch.

As suggested by [RSW22], NC^0_4-AVOID might be hard to solve. For NC^0_3, our hitting set construct works when the stretch is at least $C \cdot n^2$ for a large constant C. For smaller stretch, the complexity of NC^0_3 is less clear. It is natural to ask:

Open Question 1. Is there a deterministic polynomial time algorithm for NC^0_3-AVOID with stretch $n^{1+o(1)}$, even when an NP oracle is available?

As we have mentioned, our hitting set construction suggests a new approach to solve AVOID for weak computational models. It naturally raises the following questions.

Open Question 2. For some weak computational models (e.g., AC^0), is there a distribution that can be efficiently sampled using a short seed but cannot be sampled by these models? They are some known sampling lower bounds for AC^0 when the input distribution is uniform [Vio10, Vio20]. Do techniques in those works help in proving a sampling lower bound under arbitrary distributions?

Open Question 3. For a class of circuits $\mathcal{C} \subseteq \{\{0,1\}^n \to \{0,1\}^m\}$, it is easy to see (via the probabilistic method) that there *exists* a hitting set H for AVOID of $\mathcal{C}$ with size $|H| \leq \mathsf{poly}(\log|\mathcal{C}|)$. Note that such a hitting set constitutes a "universal" solution to explicit construct problems. Namely, for every explicit construction problem Π that is reducible to $\mathcal{C}$-AVOID, there is a string $x \in H$ that has the property Π. It would be interesting to identify the construction of the hitting set H itself as an explicit construction problem, and study its complexity and/or algorithms via various kinds of approaches (including the pseudorandomness approach considered in this work).

1.3 Organization

The rest of the paper is organized as follows. In Section 2, we put some preliminaries, including the problems we study and some mathematical tools used in our proofs. In Section 3, we show how to reduce some explicit construction problems to NC^0_4-AVOID. In Section 4, we give a polynomial time algorithm for NC^0_2. In Section 5, we show a general framework to construct hitting sets for AVOID by correlations with low-degree parities. Finally we show some applications of this method.

2 Preliminaries

In this section, we state necessary background knowledge and set up some useful pieces of notation.

Range Avoidance. We first define the AVOID problem, which is the primary subject of this work.

Definition 7 (AVOID [Kor21, RSW22])**.** *AVOID is the following problem: given a Boolean circuit $C : \{0,1\}^n \to \{0,1\}^m$ where $m > n$, find an m-bit string outside the range of C. If the input circuit is guaranteed to be in some circuit class $\mathcal{C}$, we also call the problem $\mathcal{C}$-AVOID.*

Definition 8 (NC circuits)**.** *For each $k \geq 1$, we define NC^0_k and NC^1_k as follows. NC^0_k contains all functions that depend on at most k input bits. For every $n \geq 1$, NC^1_k contains all n-bit functions that are computable by $(k \log n)$-depth Boolean circuits of fan-in two.*

We will be mainly interested in NC^0_k-AVOID and NC^1_k-AVOID for constant k's. When we say an explicit construction problem reduces to NC^1-AVOID, we mean there exists $k \geq 1$ such that the problem reduces to NC^1_k-AVOID.

Explicit Constructions. We study the following explicit construction problems, starting with rigid matrices.

Definition 9 (rigid matrix [Val77])**.** *Let $q \geq 1$ be a prime power and $r, s \geq 1$ be two integers. We say an $n \times n$ matrix M over $\mathbb{F}_q$ is (r,s)-rigid, if for any matrix $S \in \mathbb{F}_q^{n\times n}$ with at most s non-zero entries, the rank of $M + S$ is at least r.*

An explicit construction of $(\Omega(n^2), n^{1+\varepsilon})$-rigid matrices would imply a lower bound against linear-size, logarithmic-depth arithmetic circuits [Val77]. By probabilistic method, a random matrix is $(\Omega(n^2), \Omega(n^2/\log n))$-rigid with high probability. This motivates us to formulate the following problem.

Definition 10 (RIGID)**.** *(ε, δ, q)-RIGID is the following problem: given input 1^n, output an $n \times n$ matrix over $\mathbb{F}_q$ that is $\left(\varepsilon n, \frac{\delta n^2}{\log n}\right)$-rigid.*

The next object we consider is linear codes with good rates and distances.

Definition 11 (linear code [Ham50])**.** *Let $r, p \in (0,1), n \in \mathbb{N}$ and $k = r \cdot n$. We say an $k \times n$ matrix G of full row rank over $\mathbb{F}_2$ is a generator matrix of a (r,p)-linear code, if every two distinct codewords generated by G have Hamming distance at least pn, or equivalently, the Hamming weight of any nonzero codeword is at least pn.*

By probabilistic method, for every $r, p \in (0,1)$ such that $r < 1 - h(p)$, there is a family of linear codes with rate r and distance pn (the inequality $r < 1 - h(p)$ is called Gilbert-Varshamov bound in literature). However, despite an extensive line of efforts, an explicit construction meeting this bound remains widely open. We formulate the linear code construction in the complexity-theoretic language as follows.

Definition 12 (LINEARCODE)**.** *(r,p)-LINEARCODE is the following problem: given input 1^n, output a matrix $G \in \mathbb{F}_2^{rn\times n}$ such that G is a generator matrix of a (r,p)-linear code.*

Finally, we study linear codes with good list-decoding capacity.

Definition 13 (list-decodable code [Eli57, Woz58])**.** *Let $r, p \in (0,1), n \in \mathbb{N}$ and $k = r \cdot n$. We say an $rn \times n$ matrix G over $\mathbb{F}_2$ is a generator matrix of a (p, L)-list decodable code if for every $z \in \mathbb{F}_2^n$, the number of codewords $c \in \mathrm{Im}(G)$ within Hamming distance pn from z is at most L, i.e.*

$$|\{s \in \mathbb{F}_2^{rn} : \mathrm{wt}(sG - z) \leq pn\}| \leq L$$

where $\mathrm{wt}(s)$ denotes the number of ones in the string s.

The probabilistic method shows the existence of (r, p, L)-list decodable codes, provided that $r < 1 - h(p) - \frac{2}{\log_2 L}$. Again, finding an explicit family of linear codes approaching this limit remains an outstanding challenge.

Definition 14 (ListDecodable)**.** *(r, p, L)-ListDecodable is the following problem: given input 1^n, output a matrix $G \in \mathbb{F}_2^{rn \times n}$ such that G is a generator matrix of a (p, L)-list decodable code.*

2.1 Boolean Functions

In this subsection, we list some useful notations about Boolean functions. To represent a Boolean variable, we sometimes use $\mathbb{F}_2$ as the domain and sometimes use $\{-1, 1\}$ as the domain. When the domain is $\mathbb{F}_2$, we use 1 to represent True and 0 to represent False. When the domain is $\{-1, 1\}$, we use -1 to represent True and 1 to represent False.

Definition 15 (parity functions)**.** *For every set $S \subseteq [n]$, define the parity function $\chi_S : \{-1, 1\}^n \to \{-1, 1\}$ by*

$$\chi_S(x) = \prod_{i \in S} x_i.$$

Definition 16 (equality functions)**.** *For every set $S \subseteq [n]$ and $z \in \{-1, 1\}^n$, we define the equality function $\mathsf{EQ}_{S,z}(x)$ to be 1 if $x_i = z_i$ holds for every $i \in S$ and 0 otherwise.*

It can be easily verified that parity functions and equality functions have the following relation:

Fact 17. *For every set $S \subseteq [n]$ and $z \in \{-1, 1\}^n$, we have*

$$\mathsf{EQ}_{S,z}(x) = \frac{1}{2^{|S|}} \sum_{T \subseteq S} \left(\prod_{i \in T} z_i \right) \chi_T(x).$$

Definition 18 (inner product)**.** *For two Boolean functions $f, g : \{-1, 1\}^n \to \mathbb{R}$, we define their inner product by:*

$$\langle f, g \rangle = \frac{1}{2^n} \sum_{x \in \{-1,1\}^n} f(x) g(x).$$

For a given distribution $\varphi : \{-1, 1\}^n \to [0, 1]$ where $\varphi(x) := \Pr_{\boldsymbol{x} \sim \varphi}[\boldsymbol{x} = x]$, we define their inner product over φ by:

$$\langle f, g \rangle_\varphi = \sum_{x \in \{-1,1\}^n} f(x) g(x) \varphi(x).$$

Definition 19 (correlation)**.** *We say two Boolean functions $f, g : \{-1, 1\}^n \to \{-1, 1\}$ are ε-correlated if*

$$|\langle f, g \rangle| \geq \varepsilon.$$

For a given distribution $\varphi : \{-1, 1\}^n \to [0, 1]$, we say they are ε-correlated under φ if

$$|\langle f, g \rangle_\varphi| \geq \varepsilon.$$

2.2 Miscellaneous

Definition 20 (binary entropy)**.** *The binary entropy function $h : [0,1] \to \mathbb{R}$ is defined as*

$$h(p) := -p \log_2 p - (1-p)\log_2(1-p).$$

For a distribution $\mathcal{D}$, we use $\boldsymbol{x} \sim \mathcal{D}$ to denote that a random variable $\boldsymbol{x}$ is drawn from $\mathcal{D}$. We then define ε-biased distribution and ε-almost pairwise independent distribution here.

Definition 21 (ε-biased distribution)**.** *A distribution $\mathcal{D}$ on $\{-1,1\}^n$ is ε-biased if for every nonempty $T \subseteq [n]$, it holds that*

$$-\varepsilon \le \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}}\left[\prod_{i\in T} \boldsymbol{x}_i\right] \le \varepsilon.$$

Definition 22 (ε-almost pairwise independent distribution)**.** *A distribution $\mathcal{D}$ on $\{-1,1\}^n$ is ε-almost pairwise independent if for every two distinct indices $i, j \in [n]$ and vector $\vec{v} \in \{-1,1\}^2$, it holds that*

$$\left|\Pr_{\boldsymbol{x}\sim\mathcal{D}}[(\boldsymbol{x}_i, \boldsymbol{x}_j) = \vec{v}] - \frac{1}{4}\right| < \varepsilon.$$

We have the following standard constructions of ε-biased distribution and ε-almost pairwise independent distribution.

Theorem 23 ([NN93, AGHP92])**.** *For every $\varepsilon \in (0,1)$ and $n \in \mathbb{N}$, there is an explicit (polynomial-time computable) ε-biased distribution with support size $O(n^2/\varepsilon^2)$.*

Theorem 24 ([AGHP92])**.** *For every $\varepsilon \in (0,1)$ and $n \in \mathbb{N}$, there is an explicit (polynomial-time computable) ε-almost pairwise independent distribution with support size $O(\log^2 n/\varepsilon^2)$.*

3 Explicit Constructions Reduce to NC^0_4-Avoid

In this section, we reduce several central explicit construction problems in coding theory and complexity theory to solving Avoid for logarithmic depth circuits.

3.1 Technical Ingredients

We need the following technical tools from literature.

Lemma 25 ([Pat08], Theorem 1)**.** *Given an array of n elements from an alphabet Σ, and let $f_\sigma > 0$ be the number of occurences of letter σ in the array. There is a data structure storing the array with at most*

$$O(|\Sigma|\log n) + \sum_{\sigma\in\Sigma} f_\sigma \log_2 \frac{n}{f_\sigma} + O(n/\log^2 n)$$

bits of memory. Moreover, there is an algorithm that, upon receiving an index $i \in [n]$, queries at most $O(\log n)$ bits in the data structure and returns the i-th entry of the array.

Note that the $\sum_{\sigma\in\Sigma} f_\sigma \log_2 \frac{n}{f_\sigma}$ term is the entropy of the array, i.e. the information-theoretical lower bound to store the array.

The query process of the data structure can be modeled as a depth-$O(\log n)$ decision tree. The following lemma converts it to an NC^1 circuit to suit our purpose.

Lemma 26. *Every function that can be computed by a depth-d decision tree can be computed by a depth-$2d$ circuit.*

Proof. Prove by induction. When $d = 1$, the output then only depends on a single bit and can be trivially computed by a circuit of depth 2.

Suppose the statement holds for d. Let T be a depth-$(d+1)$ decision tree. Suppose the root of T queries x_i and proceeds to the left or right subtree, depending on whether x_i is 0 or 1. By our assumption, the two subtrees can be computed by circuits of depth $2d$. Let the circuits be C_0, C_1. Then we construct a circuit C as

$$C(x) := (x_i \land C_1(x)) \lor (\neg x_i \land C_0(x)).$$

It is easy to verify that $C(x)$ computes $T(x)$ correctly and has depth $2(d+1)$, which completes the proof. $\square$

The following lemma asserts that the summation of integers is in NC^1.

Lemma 27 ([Sav76]). *Iteratively adding (i.e., summing up) n n-bit integers can be done in NC^1.*

3.2 Rigid Matrices

In this subsection, we reduce constructing rigid matrices to NC^1-AVOID.

Theorem 28. *For any fixed ε, δ such that $\varepsilon + \delta < \frac{1}{2}$ and any prime power q, (ε, δ, q)-RIGID reduces in polynomial time to NC^1-AVOID.*

Proof. WLOG we can assume n is sufficiently large since for small values of n we can solve the problem by brute force and reduce to a trivial instance. Let M be an $n \times n$ matrix over $\mathbb{F}_q$ that is not $\left(\varepsilon n, \frac{\delta n^2}{\log n}\right)$ rigid. That is, M can be written as $X + S$ where X has rank at most εn and S has at most $\frac{\delta n^2}{\log n}$ non-zero entries. X can be equivalently expressed as the product of an $n \times \varepsilon n$ matrix L and $\varepsilon n \times n$ matrix R. S can be encoded by the data structure in Lemma 25. Given this observation, we construct a circuit as follows. We interpret the input bits as the encoding of L, R and the data structure encoding S. For the output bit representing M_{ij}, we first compute $\sum_{k \in [\varepsilon n]} L_{ik} R_{kj}$, which can be done by a circuit of $O(\log n)$-depth. Then we compute S_{ij} by making a query to the data structure, which can also be done by a circuit of $O(\log n)$-depth by Lemma 26. Finally, we compute M_{ij} by XOR-ing these two results.

To encode L and R, we need $2\varepsilon n^2 \log q$ bits. One way to encode S is by storing the indices of the non-zero entries and their values, which requires $2 \log n + \log q$ bits per non-zero entry. Thus the optimal encoding requires at most $(2 \log n + \log q) \frac{\delta n^2}{\log n}$ bits, and our data structure needs

$$(2 \log n + \log q) \frac{\delta n^2}{\log n} + 2q \log n + O(n^2 / \log^2 n) < 2\delta n^2 \log q$$

number of bits. Note that the number of output bits is $n^2 \log q$. As $\varepsilon + \delta < \frac{1}{2}$, the number of input bits is less than the number of output bits. Thus the resulting instance is a valid NC^1-AVOID instance, and any string outside the range of the circuit must be $\left(\varepsilon n, \frac{\delta n^2}{\log n}\right)$-rigid. $\square$

3.3 Linear Codes Achieving the Gilbert-Varshamov Bound

Constructing binary codes that approach the Gilbert-Varshamov bound is a famous open problem in coding theory. Below we show that this task also reduces to NC^1-AVOID.

Theorem 29. *For any $r, p \in (0,1)$ such that $r < 1 - h(p)$, (r,p)-*LINEARCODE *reduces in polynomial time to* NC^1-AVOID.

Proof. WLOG we can assume n is sufficiently large since for small values of n we can solve the problem by brute force and reduce to a trivial instance. Let $k = rn$. We first describe a method to compress any "bad" generator matrix into at most $(kn-1)$ bits. Suppose $G \in \mathbb{F}_2^{k\times n}$ generates a code that is not (r,p)-linear code. It implies there is a nonzero $z \in \mathbb{F}^k$ such that the vector $s = z \cdot G \in \mathbb{F}_2^n$ contains at most pn ones. Let $i \in [k]$ be such that $z_i = 1$. Let $G_{-i} \in \mathbb{F}_2^{(k-1)\times n}$ be the matrix obtained by removing the i-th row of G. Note that we can recover G from (G_{-i}, s, z, i): we simply let z_{-i} be the vector obtained by removing the i-th coordinate of z. Then we can calculate the i-th row of G by $z_{-i} \cdot G_{-i} + s$.

Then we design the circuit C as follows. C interprets its input as a tuple (G_{-i}, S, z_{-i}, i), which contains the following pieces of information.

- A $(k-1)\times n$ matrix G_{-i}.
- A data structure as in Lemma 25 encoding a string $s \in \mathbb{F}_2^n$ of Hamming weight at most pn.
- A vector $z_{-i} \in \mathbb{F}_2^{k-1}$.
- An index $i \in [k]$ (encoded by its $\log k$-bit binary representation).

The output of C is a $k \times n$ matrix G. For each $j \in [k]$, the j-th row of G is computed as follows. First, if $j \neq i$, depending on whether $j < i$, C outputs the j-th or $(j-1)$-th row of G_{-i}. The i-th row is computed by calculating $z_{-i} \cdot G_{-i}$ and adding it with the string encoded by S. By Lemma 25 and Lemma 26, each output of C can be computed by an NC^1-circuit.

Finally, note that the number of input bits is

$$(k-1)n + h(p)n + O(n/\log^2 n) + (k-1) + \log k < kn.$$

The inequality is valid provided that n is sufficiently large (in particular, if it holds that $r + h(p) < 1 - O(1/\log^2 n)$). Therefore, we conclude that C is a valid NC^1-AVOID instance. It is clear that any solution to AVOID(C) is a generator matrix for a (r,p)-linear code. □

3.4 List-Decodable Codes

Constructing binary codes that achieve list-decoding capacity remains an outstanding challenge. Below we show that, again, this task reduces to range avoidance of log-depth circuits.

Theorem 30. *For any fixed r, p, L such that $r < 1 - h(p) - \frac{2}{\lceil \log_2 L \rceil}$, (r,p,L)-*LISTDECODABLE *reduces in polynomial time to* NC^1-AVOID.

Proof. WLOG we can assume n is sufficiently large since for small n's we can solve the problem by brute force and reduce to a trivial instance. Let $k = rn$. Let $G \in \mathbb{F}_2^{k\times n}$ be a generator matrix of a code that is not (p,L)-list decodable. That is, there is a center $z \in \mathbb{F}_2^n$ and L codewords that lie in the Hamming ball with center z and radius pn. Among these codewords we can pick $t := \lceil \log_2 L \rceil$ ones that are linearly independent, namely, $y_1 + z, y_2 + z, \ldots, y_t + z$. Then we can select $k - t$ rows in G, denoted by $g_1, g_2, \ldots, g_{k-t}$ such that $\{g_1, g_2, \ldots, g_{k-t}, y_1 + z, y_2 + z, \ldots, y_t + z\}$ is a linearly independent set. We use a length-k binary string s to encode which rows we have selected. For the remaining rows, they must be some linear combinations of $\{g_1, g_2, \ldots, g_{k-t}, y_1 + z, y_2 + z, \ldots, y_t + z\}$, and we can use t vectors $c_1, c_2, \ldots, c_t \in \mathbb{F}_2^k$ to encode the coefficients.

Then we construct a circuit as follows. We interpret the inputs as:

- A vector $z \in \mathbb{F}_2^n$;
- t data structures as in Lemma 25 encoding $y_1, y_2, \ldots, y_t \in \mathbb{F}_2^n$ where each y_i has at most pn 1's;
- A binary string $s \in \mathbb{F}_2^k$ with $k - t$ 1's;
- $(k - t)$ vectors $g_1, g_2, \ldots, g_{k-t} \in \mathbb{F}_2^n$;
- t vectors $c_1, c_2, \ldots, c_t \in \mathbb{F}_2^k$.

We first compute $g_{k-t+i} := y_i + z$ for $i \in [t]$, where each coordinate of each y_i is computed by making a query to the data structure. This step can be done in $O(\log n)$-depth by Lemma 26. For the output bit representing G_{ij}, we first compute s_i and the partial sum $p = s_1 + s_2 + \ldots + s_i$, which can be done in $(\log n)$-depth by Lemma 27. If $s_i = 1$, we output the j-th coordinate in g_p. Otherwise, the index of the coefficient vectors corresponding to this row is $q := i - p$. We then output $\sum_{\ell \in [k]} c_{q\ell} g_{\ell j}$, which can be done in $O(\log n)$-depth. Thus the circuit can be implemented in $O(\log n)$-depth. By the previous argument, any G that is not a generator matrix of a (p, L)-list decodable code lies in the range of this circuit.

Note that the number of input bits is

$$\begin{aligned}
&n + t(h(p)n + O(n/\log^2 n)) + k + (k-t)n + tk \\
&< 2n + t(h(p)n + O(n/\log^2 n) + k - n) + nk \\
&< tn\left(\frac{2}{t} + h(p) + O(1/\log^2 n) + r - 1\right) + nk \\
&< nk,
\end{aligned}$$

meaning the resulting instance has less number of inputs than outputs and is hence a valid AVOID instance. □

3.5 Reduction to NC^0_4-AVOID

In this subsection, we use the reduction by Ren, Santhanam and Wang [RSW22] (i.e., Theorem 1) to further reduce these explicit construction problems to NC^0_4-AVOID. This result shows that solving even NC^0_4-AVOID would have unexpected consequences in pseudorandomness and complexity theory.

Specifically, combining Theorem 1 with our reductions (Theorem 28, 29 and 30), we get the following corollary.

Corollary 31. *Suppose there is a polynomial-time deterministic algorithm for* NC^0_4*-AVOID. Then the following are true.*

- *For every ε, δ such that $\varepsilon + \delta < \frac{1}{2}$, there is a family of $\left(\varepsilon n, \frac{\delta n^2}{\log n}\right)$-rigid matrices that are computable in deterministic polynomial time.*
- *For every rate $r \in (0,1)$ and $p < 1 - h(r)$, there is a family of (r,p)-linear code, whose generator matrices are computable in deterministic polynomial time.*
- *For every rate $r \in (0,1)$ and parameters $p \in (0,1), L \geq 1$ such that $r < 1 - h(p) - \frac{2}{\lceil \log_2 L \rceil}$, there is a family of (r,p,L)-list-decodable code, whose generator matrices are computable in deterministic polynomial time.*

From an algorithmic perspective, Corollary 31 provides a potential approach to attack these notoriously hard explicit construction problems. From a pessimistic viewpoint, Corollary 31 gives further evidence supporting the hardness of solving NC^0_4 unconditionally.

4 A Polynomial Time Algorithm for NC^0_2-Avoid

In this section, we give a polynomial time algorithm for NC^0_2-Avoid.

Reminder of Theorem 3. *There is a polynomial time algorithm which, given an NC^0_2 circuit $C : \{0,1\}^n \to \{0,1\}^m$ where $m > n$, outputs a string $y \in \{0,1\}^m$ that is not in the range of C.*

Proof. There are 4 types of functions that depend on at most two variables:

- Functions that are either constant 0 or constant 1.
- Functions that depend on a single variable.
- AND-type or OR-type functions.
- XOR-type functions.

Let $C_1, \ldots, C_m$ denote the output bits of the circuit C. Starting with $y = \star^m$, our algorithm will inspect each $C_i, i \in [m]$ in the increasing order and gradually fill in bits in y. For each C_i, the algorithm fixes y_i to a constant $0/1$, which also implicitly restricts the input space: that is, suppose we fix $y_i = b$. Then, requiring $C_i(x) = b$ makes the set of "feasible" $x \in \{0,1\}^n$ smaller. We show that eventually we can find a string $y \in \{0,1\}^m$ such that $C^{-1}(y) = \varnothing$, which solves Avoid of C.

In more detail, starting with $y = \star^m$. We enumerate $i = 1, \ldots, m$ and do the following.

- If C_i is a constant, say, $C_i(x) \equiv b$ for all $x \in \{0,1\}^n$, we can set $y_i = \neg b$ and other output bits arbitrarily. It is easy to see that y is not in the range of C.
- Suppose C_i depends only on one input variable. Write $C_i(x) = x_{j_i} \oplus b_i$. Then we can set $y_i = 0$. For every $x \in \{0,1\}^n$ that $C_i(x) = y_i$, we have $x_{j_i} = b_i$. Then we replace every appearance of x_{j_i} in C_k $(k > i)$ with b_i.
- Suppose C_i is an AND-type circuit. Namely, $C_i(x) = (x_{j_{1i}} \oplus b_{1i}) \wedge (x_{j_{2i}} \oplus b_{2i})$. Then we set $y_i = 1$. For every $x \in \{0,1\}^n$ that $C_i(x) = y_i$, we have $x_{j_{1i}} = \neg b_{1i}$ and $x_{j_{2i}} = \neg b_{2i}$. Then we replace every appearance of $x_{j_{1i}}, x_{j_{2i}}$ with $\neg b_{1i}, \neg b_{2i}$, respectively.

- Suppose C_i is an OR-type function. Namely, $C_i(x) = (x_{j_{1i}} \oplus b_{1i}) \vee (x_{j_{2i}} \oplus b_{2i})$. Then we can set $y_i = 0$. For every $x \in \{0,1\}^n$ that $C_i(x) = y_i$, we have $x_{j_{1i}} = b_{1i}$ and $x_{j_{2i}} = b_{2i}$. Then we replace every appearance of $x_{j_{1i}}, x_{j_{2i}}$ with corresponding constants.
- It remains to handle the case that C_i is an XOR-type function. Suppose $C_i(x) = x_{j_{1i}} \oplus x_{j_{2i}} \oplus b_i$. We can set $y_i = b_i$. For every $x \in \{0,1\}^n$ that $C_i(x) = y_i$, we have $x_{j_{1i}} = x_{j_{2i}}$. Then, for each $x_{j_{2i}}$ appearing in C_k for some $k > i$, we can replace $x_{j_{2i}}$ with $x_{j_{1i}}$. After that, we can remove $x_{j_{2i}}$.

In summary, if there is an output bit that is constant, then we can construct a solution to Avoid accordingly. Otherwise, we can remove at least one input variable by fixing y_i to a constant. Since $m > n$, there must be a $i \in [m]$ where C_i is a constant (because each non-constant function will incur one removal of input variable). Therefore, the procedure above always succeeds in finding a string $y \in \{0,1\}^m$ such that $C^{-1}(y) = \emptyset$. This algorithm is clearly deterministic and runs in polynomial time. □

5 A Hitting Set Construction for Avoid

In this section, we use $\{-1,1\}$ to represent True and False, respectively. By Boolean function we mean functions of the form $f\colon \{-1,1\}^n \to \{-1,1\}$.

5.1 The General Template

We first show the general framework to construct hitting sets for Avoid instances of weak circuits. We start with a definition of "simple functions".

Definition 32. *Let $f : \{-1,1\}^n \to \{-1,1\}$ be a Boolean function. Let $d \in \mathbb{N}$ and $\delta > 0$. We say f is a (d,δ)-simple function, if for any distribution φ over $\{-1,1\}^n$, there is a set $S \subseteq [n]$ of size at most d such that the correlation between χ_S and f under φ is at least δ. That is,*

$$|\langle f, \chi_S \rangle_\varphi| := \left| \sum_x \varphi(x) f(x) \chi_S(x) \right| \geq \delta.$$

Suppose $\mathcal{F} \subseteq \{f : \{-1,1\}^n \to \{-1,1\}\}$ is a collection of functions. We say $\mathcal{F}$ is a (d,δ)-simple collection, if each function in $\mathcal{F}$ is (d,δ)-simple.

For intuition, it is easy to see that every k-bit function $f : \{-1,1\}^k \to \{-1,1\}$ is $(k, 2^{-k})$-simple.

The meta-construction of hitting set. The following theorem shows our "meta-construction" of hitting set: roughly, for every Avoid-instance $C : \{-1,1\}^n \to \{-1,1\}^m$, if the stretch $m = m(n)$ is sufficiently large (relative to the "simplicity" of the function class), then the support of an almost pairwise independent distribution would be a hitting set for Avoid(C).

Theorem 33. *Suppose $m > n \geq 2$. Let $C : \{-1,1\}^n \to \{-1,1\}^m$ be a circuit and $\varepsilon > 0$ be a parameter. Suppose each output bit C_i is a (d,ε)-simple function of input bits and $m > \frac{2}{\varepsilon^2} n^d$. Let $\mathcal{D}$ be any $\frac{3}{8}\varepsilon^2$-almost pairwise independent distribution over $\{0,1\}^m$. Then, the support of $\mathcal{D}$ is a hitting set for* Avoid(C)*. That is,* $\mathrm{supp}(\mathcal{D}) \not\subseteq \mathrm{Range}(C)$.

Using a standard construction of ε-almost pairwise independent distributions (Theorem 24), the support of $\mathcal{D}$ has size bounded by $O(\log^2 m/\varepsilon^4)$. Therefore, by Theorem 33 we can construct a hitting set of size $O(\log^2 m/\varepsilon^4)$ for $\textsc{Avoid}(C)$. Remarkably, the hitting set is *oblivious* to the circuit C: one can construct it without actually looking into C.

Proof of Theorem 33. Suppose by contradiction that there exists a pairwise independent distribution $\mathcal{D}$ such that $\mathrm{supp}(\mathcal{D}) \subseteq \mathrm{Range}(C)$. Then, every string $y \in \mathrm{supp}(\mathcal{D})$ is an output of C. This implies that under a proper input distribution φ over $\{-1,1\}^n$, the output distribution of $\{C(x) : x \sim \varphi\}$ is exactly $\mathcal{D}$, which is a $\frac{3}{8}\varepsilon^2$-almost pairwise independent distribution. In the following, we show that C *cannot* sample a $\frac{3}{8}\varepsilon^2$-almost pairwise independent distribution under *any* input distribution. This would lead to a contradiction and complete the proof.

Let φ be a distribution supported on $\{-1,1\}^n$. Given φ, every output of C is correlated with χ_S for some $|S| \le d$ by Definition 32. By pigeonhole principle, there must be $\frac{m}{2\cdot n^d} > \frac{1}{\varepsilon^2} =: t$ outputs $C_1, C_2, \ldots, C_t$ that are correlated with the same set S. By negating the output if necessary, we can assume WLOG that

$$\Pr_{x\sim\varphi}[C_i(x) = \chi_S] \ge \frac{1}{2} + \varepsilon, \quad \forall i \in [t].$$

Define

$$Z(x) = |\#\{i \in [t] : C_i(x) = 0\} - \#\{i \in [t] : C_i(x) = 1\}|.$$

We note that

$$\mathop{\mathbb{E}}_{x\sim\varphi}[Z(x)] \ge \mathop{\mathbb{E}}_{x\sim\varphi}[|\#\{i \in [t] : C_i(x) = \chi_S(x)\}| - |\#\{i \in [t] : C_i(x) \ne \chi_S(x)\}|] \ge 2\varepsilon t.$$

Define $Z_{i,j}(x)$ to be 1 if $C_i(x) = C_j(x)$ and -1 otherwise. Then clearly $Z_{i,i}(x) = 1$. Note that $Z(x)^2 = \sum_{i,j} Z_{i,j}(x)$, then

$$\mathop{\mathbb{E}}_{x\sim\varphi}\left[\sum_{i,j} Z_{i,j}(x)\right] = \mathop{\mathbb{E}}_{x\sim\varphi}[Z(x)^2] \ge \mathop{\mathbb{E}}_{x\sim\varphi}[Z(x)]^2 \ge 4\varepsilon^2 t^2 = 4t.$$

It then follows that

$$\mathop{\mathbb{E}}_{x\sim\varphi}\left[\sum_{i\ne j} Z_{i,j}(x)\right] \ge 3t.$$

Hence, there must be some $i \ne j$ such that $\mathbb{E}_{x\sim\varphi}[Z_{i,j}(x)] \ge \frac{3t}{t(t-1)} > \frac{3\varepsilon^2}{2}$, meaning that

$$\Pr_{x\sim\varphi}[C(x)_i = C(x)_j] - \Pr_{x\sim\varphi}[C(x)_i \ne C(x)_j] = \mathop{\mathbb{E}}_{x\sim\varphi}[C(x)_i \cdot C(x)_j] > \frac{3\varepsilon^2}{2}.$$

By averaging principle, either $\Pr_{x\sim\varphi}[(C(x)_i, C(x)_j) = (1,1)]$ or $\Pr_{x\sim\varphi}[(C(x)_i, C(x)_j) = (-1,-1)]$ is greater than $\frac{1}{4} + \frac{3\varepsilon^2}{8}$. This contradicts to the fact that $C(\varphi)$ samples a $\frac{3\varepsilon^2}{8}$-almost pairwise independent distribution. $\square$

In the following, we show that for many natural circuit classes (NC^0_k for constant k, constant-width CNF/DNFs, small-size De Morgan formulae, etc.), functions computable in these classes are (d,δ)-simple with interesting parameters. Consequently, it allows us to apply Theorem 33 to construct hitting set for $\textsc{Avoid}$ problem of those circuit classes (for large enough stretch).

5.2 Applications

NC^0_k **circuits.** Our first application is a hitting set for NC^0_k-AVOID with stretch $m(n) \geq \omega(n^{k-1})$.

Lemma 34. *For every $k \geq 2$ and every k-bit Boolean function $f : \{-1,1\}^k \to \{-1,1\}$ that is not $\chi_{[k]}$ or $-\chi_{[k]}$, the following is true. For any distribution $\varphi : \{-1,1\}^k \to [0,1]$, there is some $S \subsetneq [k]$ and some z and such that*

$$\left| \sum_{x \in \{-1,1\}^k} \varphi(x) f(x) \mathsf{EQ}_{S,z}(x) \right| \geq 2^{-2k}.$$

Proof. Let $k \geq 2$ and $\delta = 2^{-2k}$. Suppose there is a function $f : \{-1,1\}^k \to \{-1,1\}$ and a distribution φ violating the statement of lemma. We derive a contradiction in the following.

We say that two inputs x, y are *adjacent* if they only differ at one coordinate. Suppose there are two adjacent x, y such that they differ at the i-th coordinate and $|\varphi(x) - \varphi(y)| \geq \delta$. We construct $S = [k] \setminus \{i\}$ and observe that

$$\left| \sum_{z \in \{-1,1\}^k} \varphi(z) f(z) \mathsf{EQ}_{S,x}(z) \right| = |\varphi(x) f(x) + \varphi(y) f(y)| \geq |\varphi(x) - \varphi(y)| \geq \delta.$$

Since we assumed that f and φ violate the lemma statement, we have

Observation 1: $|\varphi(x) - \varphi(y)| < \delta$ holds for every adjacent $x, y \in \{-1,1\}^k$.

Next, since f is not $\chi_{[k]}$ or $-\chi_{[k]}$, there must be adjacent inputs x, y such that $f(x) = f(y)$. Suppose they differ at the i-th coordinate. Let $S = [k] \setminus \{i\}$. Similarly, we have

$$\left| \sum_{z \in \{-1,1\}^k} \varphi(z) f(z) \mathsf{EQ}_{S,x}(z) \right| = |\varphi(x) f(x) + \varphi(y) f(y)| = \varphi(x) + \varphi(y).$$

Having assumed that f and φ violate the lemma statement, we have

Observation 2: There are adjacent $x, y \in \{-1,1\}^k$ such that $\varphi(x) \leq \varphi(x) + \varphi(y) \leq \delta$.

Finally, for each $z \in \{-1,1\}^k$, let $\mathrm{dis}(x,z)$ denote the Hamming distance between x and z. By two observations above, we have

$$\sum_{z \in \{-1,1\}^k} \varphi(z) \leq \sum_{z \in \{-1,1\}^k} (\mathrm{dis}(x,z) + 1) \cdot \delta \leq k 2^k \delta \leq k 2^{-k} < 1.$$

This contradicts to the fact that φ is a distribution. □

Lemma 35. *For every $k \geq 2$, every k-bit Boolean function $f : \{-1,1\}^k \to \{-1,1\}$ that is not $\chi_{[k]}$ or $-\chi_{[k]}$ is $(k-1, 2^{-2k})$-simple.*

Proof. By Lemma 34 and Fact 17, there exists some $S \subsetneq [k]$ and $z \in \{-1,1\}^k$ such that

$$\left| \sum_{x \in \{-1,1\}^k} \varphi(x) f(x) \mathsf{EQ}_{S,z}(x) \right| = \left| \sum_{x \in \{-1,1\}^k} \varphi(x) f(x) \sum_{T \subseteq S} \frac{1}{2^{|S|}} \left(\prod_{i \in T} z_i \right) \chi_T(x) \right| \geq 2^{-2k}.$$

As $\prod_{i \in T} z_i$ can only be ± 1, by averaging principle, there exists some $T \subseteq S \subsetneq [k]$ such that

$$|\langle f, \chi_T \rangle_\varphi| = \left| \sum_{x \in \{-1,1\}^k} \varphi(x) f(x) \chi_T(x) \right| \geq 2^{-2k}. \qquad \square$$

Corollary 36. *Let $C : \{-1,1\}^n \to \{-1,1\}^m$ be a NC^0_k circuit where $m > 2^{4k+1} n^{k-1} + n$, then there is a set $S \subseteq \{-1,1\}^m$ of size $2^{O(k)} m^2$ that is computable in polynomial time and $S \not\subseteq \mathrm{Range}(C)$.*

Proof. Let S be the support of a $\left(\frac{3}{2} \cdot 2^{-4k}\right)$-biased distribution over $\{-1,1\}^m$. By Theorem 23, it is of size $2^{O(k)} m^2$ and can be computed in polynomial time. We consider two cases.

- Suppose there are at least $n+1$ outputs that are parities of exactly k input bits. Without loss of generality assume they are $C_1, \ldots, C_{n+1}$. In this case, we interpret each C_i as a function $C_i : \mathbb{F}_2^n \to \mathbb{F}_2$ by identifying $-1, 1$ with $1, 0$ respectively. Then we know for each $i \in [n+1]$, C_i is an affine function of input bits. Since there are only n input bits, $C_1, \ldots, C_{n+1}$ are linearly dependent. That is, there is $\emptyset \neq J \subseteq [n+1]$ such that $\prod_{i \in J} C_i(x)$ is a constant. On the other hand, as S is the support of a $\frac{3}{2} \cdot 2^{-4k}$-biased distribution, there must be $y^1, y^2 \in S$ such that
$$\prod_{i \in J} y_i^1 \neq \prod_{i \in J} y_i^2.$$
Therefore, at least one of y^1, y^2 is not in the range of C.

- It remains to consider the case that there are at least $m - n$ outputs that are not in the form $\pm \chi_S$ where $|S| = k$. In this case, the correctness follows directly by combining Theorem 33 and Lemma 35. $\square$

Constant-width CNF/DNFs. Next, we apply our construction to constant-width CNF and DNFs.

Lemma 37. *For every function $f : \{-1,1\}^n \to \{-1,1\}$ that can be computed by a width-w size-s CNF/DNF, and any distribution $\varphi : \{-1,1\}^n \to [0,1]$, there exists a set $S \subseteq [n]$ and some z such that $|S| \leq w$ and*

$$\left| \sum_{x \in \{-1,1\}^n} \varphi(x) f(x) \mathsf{EQ}_{S,z}(x) \right| \geq \frac{1}{4s}.$$

Proof. WLOG we assume f is a DNF. Suppose $f = \bigvee_{i=1}^s C_i$, where each C_i is a logical AND over at most w literals (i.e., variables or their negations). We can first assume $\Pr_{x \sim \varphi}[f(x) = \mathsf{True}] \in \left(\frac{1}{4}, \frac{3}{4}\right)$, or otherwise we can set $S = \emptyset$ and $z = 0^n$ such that

$$\left| \sum_{x \in \{-1,1\}^n} \varphi(x) f(x) \mathsf{EQ}_{S,z}(x) \right| = \left| \Pr_{x \sim \varphi}[f(x) = \mathsf{False}] - \Pr_{x \sim \varphi}[f(x) = \mathsf{True}] \right| > \frac{1}{2} > \frac{1}{4s}.$$

By averaging principle, there exists $i \in [n]$ such that $\Pr_{x \sim \varphi}[C_i(x) = \mathsf{True}] \geq \frac{1}{4s}$. Let S be the variables in C_i and z be an arbitrary assignment satisfying C_i, then we have $|S| \leq w$ and

$$\left| \sum_{x \in \{-1,1\}^n} \varphi(x) f(x) \mathsf{EQ}_{S,z}(x) \right| \geq \frac{1}{4s}. \qquad \square$$

Lemma 38. *Every function $f : \{-1,1\}^n \to \{-1,1\}$ that can be computed by a width-w size-s CNF/DNF is $\left(w, \frac{1}{4s}\right)$-simple.*

Proof. By Lemma 37 and Fact 17, there exists some $S \subseteq [n]$ and $z \in \{-1,1\}^n$ such that $|S| \le w$ such that

$$\left|\sum_{x\in\{-1,1\}^n} \varphi(x)f(x)\mathsf{EQ}_{S,z}(x)\right| = \left|\sum_{x\in\{-1,1\}^n} \varphi(x)f(x)\sum_{T\subseteq S}\frac{1}{2^{|S|}}\left(\prod_{i\in T} z_i\right)\chi_T(x)\right| \ge \frac{1}{4s}.$$

As $\prod_{i\in T} z_i$ can only be ± 1, by averaging principle, there exists some $T \subseteq S$ such that $|T| \le |S| \le w$ and

$$|\langle f, \chi_T\rangle_\varphi| = \left|\sum_{x\in\{-1,1\}^n} \varphi(x)f(x)\chi_T(x)\right| \ge \frac{1}{4s}. \qquad \square$$

Corollary 39. *Let $C : \{-1,1\}^n \to \{-1,1\}^m$ be a circuit where $m > 32s^2n^w$ and each output can be computed by a width-w size-s CNF/DNF, then there is a set $S \subseteq \{-1,1\}^m$ of size $O(s^2\log^2 m)$ that is computable in polynomial time and $S \not\subseteq \mathrm{Range}(C)$.*

Proof. Let S be the support of a $\left(\frac{3}{8}\cdot\frac{1}{16s^2}\right)$-almost pairwise independent distribution. By Theorem 24, it is of size $O(s^2\log^2 m)$ and can be computed in polynomial time. The correctness directly follows from Theorem 33 and Lemma 38. $\square$

5.3 Simplicity of Functions from Approximate Degree

In this section, we derive the simplicity of functions (as per Definition 32) by connecting it with (large-error) approximate degree of Boolean functions, a well-studied complexity measure of Boolean functions in literature (see, e.g., [KS04, RS10, Rei11, BT21]).

Definition 40. *Let $f : \{-1,1\}^n \to \{-1,1\}$ be a Boolean function. For any $\varepsilon \in [0,1)$, the ε-approximate degree of f, denoted by $\deg_\varepsilon(f)$, is the least $d \in \mathbb{N}$ such that there is a degree-d real polynomial $p : \mathbb{R}^n \to \mathbb{R}$ satisfying $|p(x) - f(x)| \le \varepsilon$ for every $x \in \{-1,1\}^n$.*

In the literature, when ε is not specified, it is typically set as $\varepsilon = \frac{1}{3}$. However, for us it is also beneficial to study case that ε is very close to 1 (Note that the zero polynomial trivially 1-approximates every Boolean function).

We show that upper bounds for ε-approximate degree translate to simplicity of functions.

Theorem 41. *Suppose $n \ge 10$. Let $f : \{-1,1\}^n \to \{-1,1\}$ be a Boolean function. Let $\varepsilon \in (0,1), d \in \mathbb{N}$ be such that $\deg_{1-\varepsilon}(f) \le d$. Then f is $\left(d, \frac{\varepsilon}{3n^{d/2}}\right)$-simple.*

Proof. Let $p(x) = \sum_{S\subseteq[n],|S|\le d} \widehat{p}(S)\cdot\chi_S(x)$ be a degree-d real polynomial such that $|f(x) - p(x)| \le 1 - \varepsilon$ holds for every $x \in \{-1,1\}^n$. By Parseval's identity, we have

$$\sum_{S\subseteq[n]} \widehat{p}(S)^2 = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\{-1,1\}^n}[p(\boldsymbol{x})^2] \le (1+1-\varepsilon)^2 \le 4.$$

By Cauchy-Schwarz inequality, we have

$$4\cdot 2n^d \ge \left(\sum_{S\subseteq[n]}\widehat{p}(S)^2\right)\left(\sum_{S\subseteq[n],|S|\le d} 1\right) \ge \left(\sum_{S\subseteq[n],|S|\le d} |\widehat{p}(S)|\right)^2.$$

Therefore,

$$\sum_{S\subseteq[n],|S|\le d} |\widehat{p}(S)| \le 3n^{d/2}.$$

Note that from $f(x) \in \{-1,1\}$ and $|p(x) - f(x)| \le 1-\varepsilon$ we have $f(x)p(x) \ge \varepsilon$. Hence, for every distribution φ over $\{-1,1\}^n$, we have

$$\sum_{x\in\{-1,1\}^n} \varphi(x)\cdot f(x)\cdot p(x) \ge \sum_{x\in\{-1,1\}^n} \varphi(x)\cdot\varepsilon = \varepsilon.$$

Write $p(x) = \sum_{S\subseteq[n],|S|\le d} \widehat{p}(S)\cdot\chi_S(x)$. By averaging principle, there is $S \subseteq [n]$, $|S| \le d$ such that

$$\sum_{x\in\{-1,1\}^n} \varphi(x)\cdot f(x)\cdot\chi_S(x) \ge \frac{\varepsilon}{3n^{d/2}}.$$

Since this argument holds for every distribution φ, we conclude that f is $\left(d, \frac{\varepsilon}{3n^{d/2}}\right)$-simple. $\square$

Approximate degree of natural circuit classes. We have the following known upper bounds on approximate degree.

- For f being a size-s De Morgan formula, following a lone ling of efforts [Rei11, Tal17], we now know that $\deg_{1/3}(f) = O(\sqrt{s})$. Consequently, f is $\left(O(\sqrt{s}), n^{-O(\sqrt{s})}\right)$-simple.
- For f being a CNF/DNF of unbounded width and size s, it is known that $\deg_{1-\frac{1}{s}}(f) = O(\sqrt{n\log s})$ [KS04, dW08]. Consequently, f is $\left(\sqrt{n\log(s)}, n^{-O(n^{1/2}\log s)}\right)$-simple.

Combining these approximate degree upper bounds with Theorem 33 and 41, as well as the construction from Theorem 24, the following corollary is immediate.

Corollary 42. *Let $m = m(n), s = s(n)$ be two non-decreasing functions. Suppose $C : \{-1,1\}^n \to \{-1,1\}^m$ is a multi-output function. The following statements hold.*

- *If each output bit $C_i(x)$ is a size-s De Morgan formula of input bits and $m \ge n^{\omega(\sqrt{s})}$, then there is a set $S \subseteq \{-1,1\}^m$ of size $\mathsf{poly}(m)$ that is computable in polynomial time and satisfies $S \not\subseteq \mathrm{Range}(C)$.*
- *If each output bit $C_i(x)$ is a size-s DNF or CNF of input bits and $m \ge n^{\omega(\sqrt{n\log(s)})}$, then there is a set $S \subseteq \{-1,1\}^m$ of size $\mathsf{poly}(m)$ that is computable in polynomial time and satisfies $S \not\subseteq \mathrm{Range}(C)$.*

In both cases, the set S is independent *of the circuit C.*

References

[ABI86] Noga Alon, László Babai, and Alon Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *J. Algorithms*, 7(4):567–583, 1986.

[AGHP92] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple construction of almost k-wise independent random variables. *Random Struct. Algorithms*, 3(3):289–304, 1992.

[BHPT20] Amey Bhangale, Prahladh Harsha, Orr Paradise, and Avishay Tal. Rigid matrices from rectangular pcps or: Hard claims have complex proofs. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 858–869. IEEE, 2020.

[BT21] Mark Bun and Justin Thaler. The large-error approximate degree of ac^0. *Theory Comput.*, 17:1–46, 2021.

[dW08] Ronald de Wolf. A note on quantum algorithms and the minimal degree of epsilon-error polynomials for symmetric functions. 2008.

[Eli57] Peter Elias. List decoding for noisy channels. 1957.

[GR08] Venkatesan Guruswami and Atri Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Trans. Inf. Theory*, 54(1):135–150, 2008.

[GR09] Venkatesan Guruswami and Atri Rudra. Better binary list decodable codes via multilevel concatenation. *IEEE Trans. Inf. Theory*, 55(1):19–26, 2009.

[Ham50] R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, 1950.

[KKMP21] Robert Kleinberg, Oliver Korten, Daniel Mitropolsky, and Christos H. Papadimitriou. Total functions in the polynomial hierarchy. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 44:1–44:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[Kor21] Oliver Korten. The hardest explicit construction. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 433–444. IEEE, 2021.

[KS04] Adam R. Klivans and Rocco A. Servedio. Learning DNF in time $2^{\tilde{o}(n^{1/3})}$. *J. Comput. Syst. Sci.*, 68(2):303–318, 2004.

[Lyu22] Xin Lyu. Improved pseudorandom generators for AC^0 circuits. *Electron. Colloquium Comput. Complex.*, TR22-021, 2022.

[MST06] Elchanan Mossel, Amir Shpilka, and Luca Trevisan. On epsilon-biased generators in nc^0. *Random Struct. Algorithms*, 29(1):56–81, 2006.

[NN93] Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM J. Comput.*, 22(4):838–856, 1993.

[Pat08] Mihai Patrascu. Succincter. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 305–313. IEEE Computer Society, 2008.

[Rei11] Ben Reichardt. Reflections for quantum query algorithms. In Dana Randall, editor, *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 560–569. SIAM, 2011.

[RR97] Alexander A Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997.

[RS10] Alexander A. Razborov and Alexander A. Sherstov. The sign-rank of ac^0. *SIAM J. Comput.*, 39(5):1833–1855, 2010.

[RSW22] Hanlin Ren, Rahul Santhanam, and Zhikun Wang. On the range avoidance problem for circuits. *Electron. Colloquium Comput. Complex.*, (48), 2022.

[Sav76] John E. Savage. *The complexity of computing*. Wiley New York, 1976.

[Tal17] Avishay Tal. Formula lower bounds via the quantum method. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 1256–1268. ACM, 2017.

[TS17] Amnon Ta-Shma. Explicit, almost optimal, epsilon-balanced codes. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, page 238–251, New York, NY, USA, 2017. Association for Computing Machinery.

[TX13] Luca Trevisan and Tongke Xue. A derandomized switching lemma and an improved derandomization of AC0. In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, K.lo Alto, California, USA, 5-7 June, 2013*, pages 242–247. IEEE Computer Society, 2013.

[Val77] Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In Jozef Gruska, editor, *Mathematical Foundations of Computer Science 1977, 6th Symposium, Tatranska Lomnica, Czechoslovakia, September 5-9, 1977, Proceedings*, volume 53 of *Lecture Notes in Computer Science*, pages 162–176. Springer, 1977.

[Vio10] Emanuele Viola. The complexity of distributions. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 202–211. IEEE Computer Society, 2010.

[Vio20] Emanuele Viola. Sampling lower bounds: Boolean average-case and permutations. *SIAM J. Comput.*, 49(1):119–137, 2020.

[Wil14] Ryan Williams. Nonuniform acc circuit lower bounds. *J. ACM*, 61(1), jan 2014.

[Woz58] John M Wozencraft. List decoding. *Quarterly Progress Report*, 48:90–95, 1958.

[Yu20] Huacheng Yu. Nearly optimal static las vegas succinct dictionary. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1389–1401. ACM, 2020.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il