Electronic Colloquium on Computational Complexity, Report No. 196 (2026)

Many Proof Complexity Generators Inside One Demi-Bits Generator

Xin Li
Johns Hopkins University
lixints@cs.jhu.edu

Hanlin Ren*
Institute for Advanced Study
h4n1in.r3n@gmail.com

Yan Zhong
Johns Hopkins University
yanzhong.cs@gmail.com

September 19, 2026

Abstract

For a propositional proof system $\mathcal{P}$ and a polynomial-time function $G : \{0,1\}^n \to \{0,1\}^N$ ($N > 10n$), we say that G is a *proof complexity generator* against $\mathcal{P}$ if $\mathcal{P}$ cannot efficiently prove the (suitably encoded) statement "$y \notin \mathrm{Range}(G)$" for every $y \in \{0,1\}^N$, and G is a *demi-bits generator* against $\mathcal{P}$ if $\mathcal{P}$ cannot efficiently prove the statement "$y \notin \mathrm{Range}(G)$" for a noticeable fraction of $y \in \{0,1\}^N$. As can be seen from the definitions, proof complexity generators are qualitatively stronger objects than demi-bits generators.

Our main result is that, perhaps counter-intuitively, every demi-bits generator "contains" exponentially many proof complexity generators. In fact, a random subset of output bits of a demi-bits generator forms a proof complexity generator with constant probability. This result is an extremely simple corollary of Pajor's Lemma (a strengthening of the well-known Sauer–Shelah Lemma).

This result allows us to exhibit the hardness of the Range Avoidance problem (Avoid) in several new, restricted settings of interest:

- We show that demi-bits generators computable in NC^0 imply the hardness of NC^0-Avoid up to constant factors in the stretch, demonstrating a barrier to further improvements on the recent progress on this problem (Korten–Pitassi–Impagliazzo, FOCS'25; Guruswami–Lyu–Yuan, SODA'26).
- Given a linear space $V \subseteq \mathrm{GF}(2)^n$ of dimension k, the XOR-Remote-Point problem asks to find a vector far from V (Alon–Panigrahy–Yekhanin, RANDOM'09). Assuming a demi-hardness version of LPN (Learning Parity with Noise), we show that XOR-Remote-Point cannot be solved by efficient nondeterministic algorithms.
- An intriguing challenge in circuit complexity is to build a partial Boolean function on a given domain that has high circuit complexity (Arvind–Srinivasan, ICS'10; Chen–Huang–Li–Ren, STOC'23). Even for hardness against *polynomial-size DNFs*, no efficient algorithm is known for this task. We show that under a version of the random k-SAT Hypothesis against AM algorithms, such hard functions cannot be constructed in nondeterministic polynomial time.

Along the way, we introduce the notion of *zero-error disperser families* that can transform demi-bits generators into proof complexity generators, and show that this family can be computed by projections (i.e., without any circuit complexity overhead). Using a different instantiation of this family, we construct proof complexity generators from *barely non-trivial* demi-bits generators $G : \{0,1\}^n \to \{0,1\}^N$, where the number of hard-to-prove statements of the form "$y \notin \mathrm{Range}(G)$" just slightly exceeds 2^n (which is the number of *false* statements of this form).

*Hanlin Ren is supported by the Massive Dynamics Member Fund at the Institute for Advanced Study.

ISSN 1433-8092

Contents

1 Introduction

The *Range Avoidance* problem (Avoid) is the following computational problem: Given the description of a circuit $C : \{0,1\}^n \to \{0,1\}^\ell$ as input where $\ell > n$, the task is to find some ℓ-bit string y that is not in the range of C. Introduced in [KKMP21], this problem has received significant attention recently due to its connections to explicit construction problems [Kor21; RSW22; GLW22; GGNS23; CHLR23; HLZ26] and circuit lower bounds [Kor21; CHR24; Li24; KP24]. We refer to [Kor25] for a comprehensive survey on this problem.

It was observed in [Kor21] that the explicit construction problems for many important objects in mathematics and theoretical computer science reduce to Avoid; hence, a deterministic algorithm for Avoid would immediately resolve a plethora of them in one shot. These include constructions of Ramsey graphs [Erd59], rigid matrices [Val77], two-source extractors [CG88; CZ19; Li23], good binary codes [Gil52; Var64; Ta-17], and much more, all with near-optimal parameters.

Example 1.1. The most notorious example of explicit construction problems is, perhaps, *circuit lower bounds*. Already in 1949, Shannon [Sha49] proved that almost all Boolean functions $f : \{0,1\}^n \to \{0,1\}$ require circuits of $\Omega(2^n/n)$ size to compute. However, despite decades of efforts, the best lower bounds for explicit functions against general circuits are still only slightly above $3n$ [FGHK16; LY22].

Fix a size bound $s \ll 2^n/n$ and consider the following *truth table generator* $\mathsf{TT} : \{0,1\}^{O(s \log s)} \to \{0,1\}^{2^n}$ that takes the description of a size-s circuit as input and outputs the length-2^n truth table of this circuit. It is easy to see that Range(TT) consists of exactly the truth tables with circuit complexity at most s. Hence, proving a circuit lower bound is equivalent to solving Avoid on the particular input instance TT. For example, a deterministic algorithm for solving Avoid on TT is equivalent to a circuit lower bound for $\mathsf{E} := \mathsf{DTIME}[2^{O(n)}]$ [Kor21; RSW22].

Example 1.2. Fix a finite field $\mathbb{F}$. A matrix $A \in \mathbb{F}^{n \times n}$ is (r, Δ)-*rigid* if it has Hamming distance at least Δ to every matrix of rank at most r. Rigid matrices were introduced by Valiant [Val77] where it was shown that a random matrix is rigid with high probability and that an explicit construction of rigid matrices would imply several new circuit lower bounds. Since then, matrix rigidity has become an important topic in complexity theory (for a recent survey, see [Ram20]). However, it is still open to construct a rigid matrix with good parameters, say a $(0.1n, 0.1n^2)$-rigid matrix over $\mathbb{F} = \mathrm{GF}(2)$.

Constructing a rigid matrix in the above parameter regime reduces to the following instance C_{Rigid} of Avoid [Kor21; GLW22]. The input of C_{Rigid} consists of two matrices $X, Y \in \mathbb{F}^{n \times 0.1n}$ and a sparse matrix A containing at most $0.1n^2$ nonzeros, and the output is the n^2-bit string describing $XY^T + A$. One can encode A efficiently so that the total input length is less than n^2, hence C_{Rigid} is a valid Avoid instance.

Ren, Santhanam, and Wang [RSW22] initiated the study of Avoid for *restricted* circuit classes. One motivation for such a study is that reductions from concrete explicit construction problems to Avoid tend to produce restricted circuits: For example, using tools from succinct data structures [Pătr08; Yu22], Guruswami, Lyu, and Wang [GLW22] showed that the circuit C_{Rigid} in Example 1.2 can be computed in NC^1. Furthermore, applying a result in [RSW22] (proven by the *randomized encoding* technique [IK00; IK02; AIK06]), they further reduced the circuit complexity to NC^0_4 (i.e., each output bit in the Avoid instance only depends on *four* input bits). The circuit complexity of C_{Rigid} can be reduced even further to NC^0_3 [GGNS23]. Thus, a seemingly promising research program towards explicit constructions is:

> For each explicit construction problem Π of interest, reduce Π to an Avoid instance $C_\Pi : \{0,1\}^n \to \{0,1\}^\ell$ with the *smallest possible circuit complexity* and the *largest possible stretch function* $\ell = \ell(n)$. Then, design efficient deterministic algorithms for Avoid that can handle larger and larger circuit classes and smaller and smaller stretch functions.

This program turns out to be highly non-trivial to realize even for the simplest circuit class one may consider: functions that only depend on $k = O(1)$ many input bits (NC^0_k). Guruswami, Lyu, and Wang [GLW22] made the first progress on the NC^0_k-Avoid problem by providing an $\mathsf{FP}^{\mathsf{NP}}$ algorithm

when the stretch is $\ell(n) \geq \Omega_k(n^{k-1})$. The complexity of NC^0_k-Avoid has received much attention recently [GGNS23; KS25b; HLZ26], and the state-of-the-art algorithm runs in deterministic polynomial time when the stretch is $\ell(n) \geq \Omega_k(n^{(k-1)/2} \log n)$ [KPI25; GLY26].

1.1 Special Cases of Range Avoidance

Prior to the introduction of the Range Avoidance problem [KKMP21], there were already several special cases studied in the literature. We discuss two of them studied in this paper. In some sense, both were motivated by (our inability to prove) circuit lower bounds and explicit constructions.

The remote point problem. Given a linear space $L \subseteq \mathrm{GF}(2)^\ell$ of dimension $n < \ell$, the *Remote Point* problem (Remote-Point) asks to find a vector $v \in \mathrm{GF}(2)^\ell$ that is far from L. This problem was introduced by Alon, Panigrahy, and Yekhanin [APY09], motivated by the task of explicitly constructing rigid matrices [Val77]. They proposed the remote point problem as "an intermediate challenge that is less daunting than" that of constructing rigid matrices and provided a deterministic algorithm achieving certain parameters. Unfortunately, to the best of our knowledge, algorithmic progress on Remote-Point so far has not resulted in better constructions of rigid matrices.

For any linear space $L \subseteq \mathrm{GF}(2)^\ell$ of dimension n, there is a circuit $C : \{0,1\}^n \to \{0,1\}^\ell$ consisting of only XOR gates such that $\mathrm{Range}(C) = L$. This motivates a more general version of the remote point problem [KKMP21]: Given the description of a circuit $C : \{0,1\}^n \to \{0,1\}^\ell$ (from some circuit class $\mathscr{C}$) and a parameter $\delta \in (0, 1/2)$ as input, find a string $y \in \{0,1\}^\ell$ that is δ-far from $\mathrm{Range}(C)$. Under suitable parameter regimes, the existence of such y can be proven by a union bound, and the proof gives a reduction from Remote-Point to Avoid. Following [CHLR23], in the rest of this paper, we will use Remote-Point to denote the general version with a circuit as input, and refer to the special case studied in [APY09] as XOR-Remote-Point.

Finding hard partial truth tables. Fix a circuit class $\mathscr{C}$ and a size bound s. In the *Hard Partial Truth Table* problem, given a "domain" $x_1, x_2, \ldots, x_L \in \{0,1\}^n$ ($L \gg s$), the goal is to output a "partial truth table" $b_1, b_2, \ldots, b_L \in \{0,1\}$ such that the partial function $\{x_i \mapsto b_i\}$ is hard against size-s $\mathscr{C}$ circuits; that is, for any such circuit $C : \{0,1\}^n \to \{0,1\}$, there exists an index $i \in [L]$ such that $C(x_i) \neq b_i$. It is easy to see that under suitable choice of parameters, this problem also reduces to Avoid (see, e.g., [CHLR23]). We denote this problem by $\mathscr{C}$-Partial-Hard.

This problem was introduced in [AS10] under the name "circuit lower bounds with help functions," where it was observed that for $\mathscr{C} = \mathsf{AC}^0$, this problem reduces to the remote point problem in certain parameter regimes. This problem turns out to be stubbornly hard even for simple circuit classes $\mathscr{C}$ such as *polynomial-size DNFs*. Even though it is trivial to prove a $2^{\Omega(n)}$ size lower bound for DNFs, we are not aware of any polynomial-time algorithm for DNF-Partial-Hard in the regime of $s \ll L \leq \mathrm{poly}(n)$.

Besides being interesting in its own right, this problem was also motivated by closures of non-uniform circuit classes under uniform reductions: a polynomial-time algorithm for $\mathscr{C}$-Partial-Hard implies a problem $L \in \mathsf{E}$ that cannot be (polynomial-time) mapping reduced to any problem in (the non-uniform complexity class) $\mathscr{C}$ (see [AS10, Theorem 5]). The reduction is allowed to run in (uniform) polynomial time (which can potentially be more powerful than $\mathscr{C}$), hence such statements seem strictly stronger than circuit lower bounds for $\mathscr{C}$. By providing an $\mathsf{FP}^{\mathsf{NP}}$ algorithm for ACC^0-Partial-Hard in certain parameter regimes, [CHLR23] proved that there is a language in E^{NP} that does not mapping reduce to (non-uniform) ACC^0. However, due to our inability to solve DNF-Partial-Hard in polynomial time, even the following innocent-looking question is open:

Question 1.3. *Is it possible that for every language $L \in \mathsf{E}$, there is a language L' computable by a (non-uniform family of) polynomial-size DNF, such that $L \leq_m L'$?*

1.2 Is Avoid Hard?

Our inability to solve restricted versions of Avoid leads to the possibility that these versions might be *hard* to solve. For example, could it be possible that there exists no deterministic polynomial-time algorithm for NC^0_{100}-Avoid with stretch $n^{1.01}$?

At first glance, Avoid seems to be an easy problem since it can be solved by the following trivial randomized algorithm: output a random string $y \leftarrow \{0,1\}^\ell$ and it is not in Range(C) with high probability. However, since it is unclear how to *verify* whether a string y is in Range(C) or not (indeed, this is NP-hard), it is also unclear how to turn this trivial randomized algorithm into a deterministic algorithm, even under standard derandomization assumptions [NW94; IW97; Uma03].

Somewhat surprisingly, we now have cryptographic evidence that this trivial randomized algorithm for Avoid *cannot* be made deterministic. The first such evidence was given by Ilango, Li, and Williams [ILW23]. They showed that if subexponentially-secure indistinguishability obfuscation exists [BGI+12; GGH+16; JLS21] and $\mathsf{NP} \neq \mathsf{coNP}$, then there is no deterministic polynomial-time algorithm for Avoid. Based on this work, Chen and Li [CL24] showed that Avoid does not have *nondeterministic polynomial-time* (SearchNP) algorithms under certain hardness assumptions about LPN (Learning Parity with Noise) or LWE (Learning with Errors) against nondeterministic algorithms. More recently, a pair of concurrent works [Ila25; RWZ26] strengthened the results in [CL24], showing that Avoid $\notin$ SearchNP if there exist *demi-bits generators* with suitable stretch [Rud97] (an assumption weaker than that used in [CL24]). We will discuss this assumption soon in Section 1.3.

Hardness for restricted circuits? In light of the aforementioned research program, it is important to understand for which circuit classes $\mathscr{C}$ and which stretch functions $\ell(n)$ the $\mathscr{C}$-Avoid problem is hard. Except for [ILW23], all other three works imply hardness of restricted Avoid.

- The construction in [CL24] crucially relies on the structure of LWE and LPN. In particular, under a nondeterministic variant of the LPN assumption, [CL24] showed that the Remote-Point problem for $\mathsf{XOR} \circ \mathsf{AND}_{O(\log n)}$ circuits (i.e., $O(\log n)$-degree polynomials over GF(2)) is hard for SearchNP algorithms. However, [CL24] mentioned that the hardness of XOR-Remote-Point, i.e., the original "Remote Point Problem" considered in [APY09], remains open.

- Given a secure demi-bits generator $G : \{0,1\}^n \to \{0,1\}^N$ and a pairwise independent hash family $\mathcal{H} = \{h : \{0,1\}^N \to \{0,1\}^m\}$ (where $m > n$), [RWZ26] shows that any SearchNP algorithm fails to solve Avoid on some instance of the form $h \circ G$ where $h \in \mathcal{H}$. Pairwise independent hash families can be computed by GF(2)-linear circuits (circuits consisting of only XOR gates), therefore under suitable assumptions, [RWZ26] obtained hardness of Avoid where each output bit is computable in $\mathsf{XOR} \circ \mathsf{AND}_{O(1)}$ (i.e., a constant-degree GF(2)-polynomial).

- Given a secure demi-bits generator $G : \{0,1\}^n \to \{0,1\}^N$, it is implicit in [Ila25] that any SearchNP algorithm fails to solve Avoid on some instance of the form

$$C_{x_1,\ldots,x_t}(s,i) := G(s) \oplus x_i,$$

  where $t = O(N)$ and $x_1, \ldots, x_t \in \{0,1\}^N$. This allows [Ila25] to refute the *Oracle Derandomization Hypothesis* of Fortnow and Santhanam [FS11] under suitable assumptions.

Unfortunately, one can see that due to the overhead in the constructions, none of the above works have the potential of implying the hardness of NC^0-Avoid. Ilango's result is the closest one: assuming demi-bits generators computable in NC^0, [Ila25] implies that there is no SearchNP algorithm for Avoid on circuits with $O(\log N)$ locality. Furthermore, the techniques of [RWZ26; Ila25] do not seem applicable to the Remote Point Problem.

1.3 Demi-Bits Generators and Proof Complexity Generators

The above line of work on the hardness of Avoid can be equivalently stated as connections between demi-bits generators and proof complexity generators, both of which are important objects in their own right. In this sub-section, a *generator* is an efficiently computable function $G : \{0,1\}^n \to \{0,1\}^N$ with $N > n$, and an *adversary* trying to break the generator G is a *nondeterministic* polynomial-time algorithm $\mathcal{A}$ trying to *certify* a certain string $y \in \{0,1\}^N$ is not in Range(G). Equivalently, one can think of the adversary as a (Cook–Reckhow [CR79]) *proof system* by treating nondeterministic bits as "proofs" that $y \notin \text{Range}(G)$ (see Definition 2.3). We also assume that nondeterministic adversaries discussed here correspond to *sound* proof systems, i.e., for every $y \in \text{Range}(G)$, they never prove the incorrect statement "$y \notin \text{Range}(G)$".

Demi-bits generators are generators G such that there is no efficient nondeterministic adversary $\mathcal{A}$ that accepts a large fraction of strings $y \in \{0,1\}^N$ but rejects every string $y \in \text{Range}(G)$. That is, there is no sound proof system that proves (a suitably encoded version of) "$y \notin \text{Range}(G)$" efficiently for a large fraction of $y \in \{0,1\}^N$. Introduced by Rudich [Rud97], demi-bits generators can be seen as the analogue of cryptographic *hitting set generators* against efficient nondeterministic adversaries.[1]

Demi-bits generators are also related to the *natural proof* barrier of Razborov and Rudich [RR97]. To prove a circuit lower bound, we need to design a property $\mathcal{P}$ over truth tables of Boolean functions such that every Boolean function satisfying $\mathcal{P}$ requires large circuits. We say that $\mathcal{P}$ is a *natural property* [RR97] if it satisfies the following two requirements:

- **Constructivity:** Given the length-2^n truth table of a Boolean function f, it is possible to decide whether f satisfies $\mathcal{P}$ in poly(2^n) time.
- **Largeness:** A constant fraction of length-2^n truth tables satisfy $\mathcal{P}$.

Example 1.4. The main result of [RR97] is that if secure one-way functions exist, then there is no natural property $\mathcal{P}$ for proving lower bounds against general circuits ($\mathsf{P}/_{\text{poly}}$). Roughly speaking, this is because the existence of one-way functions implies that TT, the truth table generator, is a cryptographic hitting set generator [GGM86], while every natural property against $\mathsf{P}/_{\text{poly}}$ can be used to break TT as a hitting set generator.

Example 1.5. Under the LPN assumption, the circuit C_{Rigid} constructed in Example 1.2 is a cryptographic pseudorandom generator. (Proof sketch: if one can distinguish $XY^T + A$ from random where $X, Y \in \mathbb{F}_2^{N \times N^{0.1}}$ and A is some sparse random noise, then one can apply a hybrid argument and, given X, distinguish $Xs + e$ from random, where $s \in \mathbb{F}_2^{N^{0.1}}$ and e is some sparse random noise.)

Therefore, there is a "natural proof" barrier to constructing a rigid matrix: if there is a property $\mathcal{P}$ of $N \times N$ matrices that is constructive (runs in poly(N) time), large (accepts a constant fraction of random matrices), and only accepts matrices that are rigid, then one can use $\mathcal{P}$ to break the LPN assumption.

Rudich [Rud97] considered a notion of NP-constructivity, where the property $\mathcal{P}$ is only required to be computable in *nondeterministic* poly(2^n) time. Given the correspondence between proof systems and nondeterministic algorithms [CR79], NP-constructivity is arguably a more suitable notion than (polynomial-time) constructivity in the study of the (metamathematical) difficulty of circuit lower bounds. By definition, if TT is a secure demi-bits generator, then there is a natural proof barrier to proving circuit lower bounds, even if the property is allowed to use nondeterminism. Similarly, if C_{Rigid} is a secure demi-bits generator, then there is a natural proof barrier to constructing rigid matrices.

[1] We note that Rudich [Rud97] also introduced the notion of *super-bits generators*, which is the analogue of cryptographic pseudorandom generators against efficient nondeterministic adversaries.

Proof complexity generators against a proof system $\mathcal{P}$ are generators G satisfying the stronger requirement that $\mathcal{P}$ cannot efficiently prove "$y \notin \mathrm{Range}(G)$", for *any* $y \in \{0,1\}^N$.[2] Introduced independently in [ABRW04] and [Kra01], this notion was heavily inspired by the notion of *pseudorandomness* [Yao82]. Indeed, the motivation of [ABRW04] was to pin down a precise definition of "pseudorandom generators against proof systems," and [Kra01, Section 7] conjectured that the existence of pseudorandom generators implies the existence of proof complexity generators secure against Extended Frege. The influence of pseudorandomness goes far beyond these two papers, as can be seen, for example, in the literature studying the Nisan–Wigderson generator ([NW94]) as a proof complexity generator [ABRW04; Kra04; Raz15; Pic11; Kra11; Kra12; Pic15; Kha22]. A comprehensive survey on proof complexity generators can be found in [Kra25].

Motivated by the apparent relationship between pseudorandomness and proof complexity generators, Razborov [Raz15] asked the following question: For what family of generators $G : \{0,1\}^n \to \{0,1\}^N$ can we base their hardness as proof complexity generators on their "*computational or combinatorial*" hardness? Razborov coined this approach towards proof complexity lower bounds as the *generator approach*.[3] Razborov also conjectured that plugging suitable hard functions (in the *computational* sense) into the Nisan–Wigderson generator would result in proof complexity generators against proof systems such as Frege and Extended Frege [Raz15, Conjectures 1 and 2]. Unfortunately, progress on this conjecture has only been made for restricted proof systems such as AC^0-Frege [Pic11; Kha22].

Apparently, many generators arising from explicit construction problems possess both cryptographic hardness and proof complexity hardness. Our inability to prove circuit lower bounds *for any explicit function* and construct *any* rigid matrices suggests that the generators TT and C_{Rigid} described in Example 1.1 and Example 1.2 are plausible proof complexity generators. Indeed, great effort has been devoted to the unprovability of circuit lower bounds [Raz95; Kra97; Raz98; Raz04a; Raz04b; Kra04; Raz15; Pic15; PS19; ST21; PS21; PS22; GGRT26], which is equivalent to TT being a proof complexity generator. On the other hand, as discussed in Example 1.4 and Example 1.5, under standard cryptographic assumptions, TT and C_{Rigid} are cryptographic hitting set generators as well! Understanding the role of (computational) pseudorandomness in the generator approach is profoundly related to the natural proof barrier, see Section 1.5 for more discussions.

Finally, another motivation for [Kra01] to define proof complexity generators is the *dual weak pigeonhole principle*: For any function $f : [N] \to [2N]$, there exists at least one element $y \in [2N]$ that is not in the range of f. If $G : \{0,1\}^n \to \{0,1\}^{n+1}$ is a proof complexity generator against a certain proof system $\mathcal{P}$, then the dual weak pigeonhole principle is "effectively false" for $\mathcal{P}$! The same principle also underlies the Range Avoidance problem; indeed, Avoid on f asks precisely to find such an element $y \in [2N] \setminus \mathrm{Range}(f)$. It is easy to see (in fact, *by definition*) that

Observation 1.6 ([RSW22, Section 6])**.** Avoid $\notin \mathsf{SearchNP}$ *if and only if for every propositional proof system* $\mathcal{P}$*, there exists a proof complexity generator* G *secure against* $\mathcal{P}$*.*

This connection allows us to import the hardness results for Avoid described in Section 1.2 [CL24; RWZ26; Ila25] to the realm of proof complexity generators. These results need to assume the existence of demi-bits generators, hence they are in fact transformations from demi-bits generators to proof complexity generators. For example:

Theorem 1.7 (Informal)**.** *Let* $G : \{0,1\}^n \to \{0,1\}^N$ *be a demi-bits generator, and* $\mathcal{P}$ *be a "nice" proof system. Then:*

[2] It is more natural to talk about proof complexity generators against a *fixed* proof system instead of those against *all* proof systems. One reason is that if the proof system is allowed to be non-uniform, then it can simply hardwire a string $y \notin \mathrm{Range}(G)$ and add this fact as an axiom. There is also evidence that any uniform proof complexity generator cannot be secure against all uniform proof systems simultaneously [KS25a].

[3] We remark that our paper, as well as the recent line of works [CL24; RWZ26; Ila25], does not contribute to the exact questions asked in [Raz15] since we still need hardness assumptions against *nondeterministic* adversaries.

- ([RWZ26]) *For any suitable family of pairwise independent hash functions* $\mathcal{H} = \{h : \{0,1\}^N \to \{0,1\}^m\}$*, there exists* $h \in \mathcal{H}$ *such that* $h \circ G$ *is a proof complexity generator against* $\mathcal{P}$*.*
- ([Ila25]) *There exists a family of "shifts"* $z_1, z_2, \ldots, z_{O(n)}$ *such that the function*

$$C(s,i) := G(s) \oplus z_i$$

is a proof complexity generator against $\mathcal{P}$*.*

However, the same issue regarding circuit complexity overhead mentioned in Section 1.2 arises here as well: What is the smallest complexity overhead of constructing proof complexity generators from demi-bits generators? In particular, are there proof complexity generators computable in NC^0?

1.4 Our Results

1.4.1 Proof Complexity Generators from Demi-Bits Generators with No Overhead

Our main result is the following seemingly surprising statement:

For every demi-bits generator G*, there are lots of proof complexity generators "hidden" in* G*.*

More precisely, let $m := 10n$, $G : \{0,1\}^n \to \{0,1\}^m$, and let $I \subseteq [m]$. Define $G|_I : \{0,1\}^n \to \{0,1\}^{|I|}$ to be the function consisting of the output bits of G indexed by I. We show:

Theorem 1.8 (Main; Informal)**.** *Let* $\mathcal{P}$ *be a "well-behaved"*[4] *proof system and* G *be a demi-bits generator against* $\mathcal{P}$*. Then with constant probability over a uniformly random subset* $I \subseteq [m]$*,* $G|_I$ *is a proof complexity generator against* $\mathcal{P}$*.*

The proof of Theorem 1.8 is remarkably simple: It follows directly from the well-known *Sauer–Shelah* lemma [Sau72; She72; VA68]. Recall that a set system $\mathcal{F}$ over $[m]$ *shatters* a subset of indices $I \subseteq [m]$, if for every subset $J \subseteq I$, there exists a set $S \in \mathcal{F}$ such that $S \cap I = J$. The Sauer–Shelah lemma (in fact, its strengthening by Pajor [Paj85]) states that every set system $\mathcal{F}$ shatters at least $|\mathcal{F}|$ many subsets.

Proof Idea of Theorem 1.8. Let $H \subseteq \{0,1\}^m$ denote the set of "hard" strings for $\mathcal{P}$, i.e., the strings $z \in \{0,1\}^m$ such that $\mathcal{P}$ does not admit short proofs of "$z \notin \mathrm{Range}(G)$". If G is a demi-bits generator against $\mathcal{P}$, then $|H| \geq \Omega(2^m)$. Let I be a random subset of $[m]$, then by Pajor's lemma, H shatters I with constant probability. (Also, $|I| > n$ with high probability.)

It is easy to verify that if H shatters I, then $G|_I$ is a proof complexity generator against $\mathcal{P}$: for every $y \in \{0,1\}^I$, there exists a string $z \in H$ such that $z|_I = y$; hence, if $\mathcal{P}$ can efficiently prove "$y \notin \mathrm{Range}(G|_I)$", then $\mathcal{P}$ can efficiently prove that "$z \notin \mathrm{Range}(G)$" as well, a contradiction to $z \in H$. □

We remark that the main results in [RWZ26; Ila25] are also obtained by very simple proofs. In our opinion, the simplicity of our proofs indicates that there are potentially much more excitement in the direction of *cryptography against nondeterministic adversaries* yet to be discovered. Moreover, the versatility of nondeterministic adversaries suggests that many results there are going to be vastly different from our existing intuition in the setting of deterministic adversaries.

Theorem 1.8 gives a transformation from demi-bits generators to proof complexity generators with *zero* circuit complexity overhead, which also allows us to make progress on the hardness of special cases of the Range Avoidance problem.

[4] Thanks to the simplicity of our arguments, the definition of "well-behaved" here is much more relaxed than the definition of "nice" in Theorem 1.7; see Section 4.3 for more details.

Range avoidance for NC^0 circuits. Let $k \geq 3$ be a constant. Assuming there are demi-bits generators $G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ computable in NC^0_k for $\ell(n) \geq 10n$, we immediately obtain hardness of NC^0_k-Avoid against SearchNP algorithms. Moreover, the hard Avoid instances we obtain have the same stretch ($\Theta(\ell(n))$) as the demi-bits generator G up to a constant factor.

Corollary 1.9 (NC^0-Avoid is Hard)**.** *Let $k \in \mathbb{N}$, and $\ell(n) > n$ be a stretch function. The following holds for some universal constant $c \geq 1$: If there exists a demi-bits generator $G : \{0,1\}^n \to \{0,1\}^{c\ell(n)}$ computable in NC^0_k, then NC^0_k-Avoid$[n, \ell(n)] \notin$ SearchNP.*

Guruswami, Lyu, and Yuan [GLY26] explicitly asked the open problem of determining the "computational threshold" for NC^0_k-Avoid. Corollary 1.9 contributes to this problem by reducing it to determining the best stretch of demi-bits generators computable in NC^0_k.

Every pseudorandom generator computable in NC^0_k with stretch $\Omega_k(n^{\lceil k/2 \rceil})$ can be distinguished by a linear test with constant advantage [MST06]. To the best of our knowledge, when k is a large enough constant (say $k \geq 10$), we are not aware of attacks for *general, arbitrary* pseudorandom generators computable in NC^0_k with a smaller stretch. (In contrast, there is a long line of work on attacking specific NC^0_k PRGs such as Goldreich's PRG [CM01; BQ12; OW14; ABR16; AL18; CDM+18; OST22; Üna23].) Plugging recent algorithmic advances for NC^0_k-Avoid [KPI25; GLY26] into Corollary 1.9 immediately implies a nondeterministic adversary that achieves a better stretch: For every $k \geq 3$, [GLY26] presented a deterministic algorithm for NC^0_k-Avoid with $\Omega_k(n^{(k-1)/2} \log n)$ stretch, hence we obtain nondeterministic adversaries breaking every demi-bits generator of stretch $\Omega_k(n^{(k-1)/2} \log n)$ as well.

Finding hard partial truth tables is hard. Theorem 1.8 also allows us to explain our lack of progress in the $\mathscr{C}$-Partial-Hard problem even for very simple circuit classes such as $\mathscr{C} = \mathsf{DNF}$. In fact, we show that this problem is at least as hard as *PAC-learning* $\mathscr{C}$ by nondeterministic algorithms.[5] It is a fundamental and long-standing open problem in learning theory to design an efficient PAC-learning algorithm for DNFs, and the best algorithm to date requires $2^{\tilde{O}(n^{1/3})}$ time [KS04]. It turns out that this long-standing question explains our inability to solve DNF-Partial-Hard as well!

Theorem 1.10 (Informal)**.** *For nondeterministic algorithms, generating hard partial truth tables for a circuit class $\mathscr{C}$ is at least as hard as learning $\mathscr{C}$. More precisely:*

- *A worst-case nondeterministic algorithm for $\mathscr{C}$-Partial-Hard implies a distribution-free nondeterministic PAC-learner for $\mathscr{C}$.*
- *An errorless average-case nondeterministic heuristic for $\mathscr{C}$-Partial-Hard implies a distribution-specific nondeterministic PAC-learner for $\mathscr{C}$.*

Moreover, Daniely and Shalev-Shwartz [DS16] proved the hardness of learning DNFs under a natural assumption about the complexity of random k-SAT. Applying their reduction, we establish the hardness of DNF-Partial-Hard under a variant of the random k-SAT assumption against nondeterministic algorithms. (Since their reduction is randomized, we actually need the hardness of random k-SAT against AM algorithms.)

Corollary 1.11 (DNF-Partial-Hard Is Hard; Informal)**.** *If random k-SAT with suitable parameters is hard against AM algorithms, then there is no errorless nondeterministic heuristic that efficiently solves DNF-Partial-Hard, even on average over a certain samplable distribution.*

[5] We actually define nondeterministic PAC-learning algorithms in terms of "random right-hand-side refutation" algorithms, as these two problems are equivalent in the deterministic setting [Vad17], and the latter can be defined straightforwardly in the nondeterministic setting. See Definition 5.1 for details.

Hardness of the Remote Point problem. Theorem 1.8 also helps explain the hardness of XOR-Remote-Point, the original Remote Point problem introduced in [APY09].

To prove the hardness of the Remote Point problem, we introduce a slight generalization of demi-bits generators called remote-point demi-bits generators: these are generators $G : \{0,1\}^n \to \{0,1\}^N$ such that no proof system can prove "y is *far from* Range(G)" for a noticeable fraction of y. It is easy to see that the nondeterministic hardness of LPN implies a remote-point demi-bits generator computable using only XOR gates. Combining this with a disperser family satisfying a certain *Lipschitz property* (which is satisfied by our main construction in Theorem 1.16), we have:

Theorem 1.12 (Informal)**.** *Composing a remote-point demi-bits generator with a Lipschitz disperser family yields hard instances for the Remote Point Problem: no efficient errorless nondeterministic heuristic can find a point far from the range of the resulting circuit, even on average over the choice of disperser.*

Combining Theorem 1.12 with an assumption on the demi-hardness of LPN, we immediately obtain:

Corollary 1.13 (Hardness of Remote-Point for Linear Circuits, Informal)**.** *Under the Demi-hardness of LPN, no efficient nondeterministic heuristic can find a point far from the range of a* XOR *circuit, even on average. The same holds for k-*XOR *circuits under the Demi-hardness of Sparse LPN.*

Finally, we consider the hardness of finding *average-case hard partial truth tables* against a circuit class $\mathscr{C}$. This is the natural "hybrid" of Partial-Hard and Remote-Point [CHLR23]: Given the domain of a partial function $x_1, \ldots, x_L \in \{0,1\}^n$ and parameters $s \geq 1$, $\delta > 0$, our goal is to output a sequence of bits $b_1, \ldots, b_L \in \{0,1\}$ such that the partial function $\{x_i \mapsto b_i\}_{i \in [L]}$ is δ*-average-case hard* against $\mathscr{C}$ circuits of size s. Namely, there is no $\mathscr{C}$ circuit C of size s such that

$$\Pr_{i \leftarrow [L]}[C(x_i) = b_i] \geq 1 - \delta.$$

We denote this problem as $\mathscr{C}$-Partial-AvgHard.

It turns out that average-case hard partial truth tables are hard to construct if *agnostic PAC-learning* [KSS92] is hard for nondeterministic algorithms:

Theorem 1.14 (Informal)**.** *For nondeterministic algorithms, generating hard partial truth tables for a circuit class $\mathscr{C}$ is at least as hard as agnostic PAC-learning $\mathscr{C}$. More precisely:*

- *A worst-case nondeterministic algorithm for $\mathscr{C}$-Partial-AvgHard implies a nondeterministic distribution-free agnostic PAC-learner for $\mathscr{C}$.*
- *An errorless average-case nondeterministic heuristic for $\mathscr{C}$-Partial-AvgHard implies a nondeterministic distribution-specific agnostic PAC-learner for $\mathscr{C}$.*

Theorem 1.14 allows us to base the hardness of Partial-AvgHard for *extremely simple circuit classes*, such as k-XOR, on variants of nondeterministic hardness of LPN:

Corollary 1.15 (Hardness of k-XOR-Partial-AvgHard (Informal))**.** *Under the Sparse-Secret LPN Assumption for Random Matrices, no efficient nondeterministic heuristic can find a partial truth table that is far from every k-*XOR *function, even on average.*

1.4.2 Abstraction: Zero-Error Disperser Families

Intuitively, the results in [RWZ26; Ila25] and Theorem 1.8 suggest that proof complexity generators can be obtained by composing a certain function with a demi-bits generator. We make this intuition formal by introducing the notion of *zero-error disperser families*.

Let $\mathcal{X} \sim \{0,1\}^n$ be a random variable. A deterministic function $f : \{0,1\}^n \to \{0,1\}^m$ is a *zero-error disperser* for $\mathcal{X}$ if the support of the random variable $f(\mathcal{X})$ contains every string in $\{0,1\}^m$. We are

interested in constructing zero-error dispersers for a class $\mathcal{C}$ of random variables (weak sources) such that the function f is a zero-error disperser for every $\mathcal{X} \in \mathcal{C}$. It is well known that such dispersers cannot exist for the class of general weak random sources even if the min-entropy[6] of $\mathcal{X}$ is as large as $n-1$. Therefore, historically such zero-error dispersers have been studied for restricted classes of sources such as bit-fixing sources [GS12a; GS12b], affine sources [Li11; GS12b; LZ24], and independent sources [GS12a; Coh21]. For our applications here, however, we study a *family* of zero-error dispersers (or equivalently, seeded zero-error dispersers) for *general weak random sources* with relatively large (e.g., $n - o(n)$) min-entropy.

Specifically, we say that a distribution $\mathcal{F}$ over functions $f : \{0,1\}^n \to \{0,1\}^m$ is a family of (k, ε) zero-error dispersers if for every source $\mathcal{X} \sim \{0,1\}^n$ with min-entropy at least k, with probability at least $1-\varepsilon$ over a random function $f \sim \mathcal{F}$, f is a zero-error disperser for $\mathcal{X}$. Note that this is stronger than standard (k, ε) dispersers, which only require that the output has support size at least $(1-\varepsilon)2^m$. We also remark that such a family of zero-error dispersers follows from a *seeded randomness extractor* with sufficiently small error (e.g., when the error is smaller than $2^{-(m+1)}$ where m is the output length). However, for our applications, it is important that the disperser has the smallest possible circuit complexity.

The projection family. Let $n \geq 10m$ and denote by $\mathrm{Proj}_{n\to m}$ the family of random projections mapping n bits to m bits. That is, to sample a random function $f : \{0,1\}^n \to \{0,1\}^m$ in $\mathrm{Proj}_{n\to m}$, sample a random subset $I \subseteq [n]$ of size $|I| = m$. Suppose $I = \{i_1, i_2, \ldots, i_m\}$ with $1 \leq i_1 < i_2 < \cdots < i_m \leq n$, then define

$$f(x) := x_{i_1} x_{i_2} \ldots x_{i_m}.$$

Theorem 1.16 (Corollary of Theorem 3.3)**.** *Let $n \geq 10m$, then $\mathrm{Proj}_{n\to m}$ is an $(n-1, 1/2)$ zero-error disperser family.*

To showcase how zero-error disperser families turn a demi-bits generator into proof complexity generators, we prove the following strengthening of our main theorem Theorem 1.8. Roughly speaking, this theorem states that under suitable choice of parameters, even if we fix in advance the output length m of our desired proof complexity generator, we can still draw m output bits uniformly at random from our demi-bits generator and obtain a secure proof complexity generator.

Theorem 1.17 (Informal)**.** *Let $\mathcal{P}$ be a "well-behaved" proof system and $G : \{0,1\}^n \to \{0,1\}^{20n}$ be a demi-bits generator against $\mathcal{P}$. Then with constant probability over a uniformly random subset $I \subseteq [20n]$ of size $|I| = 2n$, $G|_I$ is a proof complexity generator against $\mathcal{P}$.*

Proof Idea of Theorem 1.17. This follows easily from definition and is nearly identical to Theorem 1.8.

Let $H \subseteq \{0,1\}^{20n}$ denote the set of "hard" strings for $\mathcal{P}$, i.e., the strings $z \in \{0,1\}^{20n}$ such that $\mathcal{P}$ does not admit short proofs that $z \notin \mathrm{Range}(G)$. If G is a demi-bits generator against $\mathcal{P}$, then $|H| \geq 2^{20n}/2$, hence the uniform distribution over H has min-entropy $20n - 1$. It follows from Theorem 1.16 that a random function $f \leftarrow \mathrm{Proj}_{20n\to 2n}$ is a zero-error disperser for H with probability at least $1/2$.

It is easy to verify that if f is a zero-error disperser for H, then $f \circ G$ is a proof complexity generator against $\mathcal{P}$: for every $y \in \{0,1\}^{2n}$, there exists a string $z \in H$ such that $f(z) = y$; hence, if $\mathcal{P}$ can efficiently prove $y \notin \mathrm{Range}(f \circ G)$, then $\mathcal{P}$ can efficiently prove $z \notin \mathrm{Range}(G)$ as well, a contradiction to $z \in H$. □

Proof complexity generators from barely non-trivial demi-bits generators. In [RWZ26], the authors composed pairwise independent hash functions with demi-bits generators to obtain proof complexity generators. Inspired by their work, we show that t-wise independent hash functions are zero-error disperser families with extremely good parameters:

[6] The min-entropy of a random variable $\mathcal{X}$ is denoted as $H_\infty(\mathcal{X})$. $H_\infty(\mathcal{X}) \geq k$ iff $\forall x \in \mathrm{Supp}(\mathcal{X}), \Pr[\mathcal{X} = x] \leq 2^{-k}$.

Theorem 1.18 (Corollary of Theorem 3.7)**.** *Any family of m-wise independent hash functions $\mathcal{H} = \{h : \{0,1\}^n \to \{0,1\}^m\}$ is an $(m + \log m + O(1), 2^{-m})$ zero-error disperser family.*

This family requires large circuit complexity overhead. Nevertheless, in the regime where we do not need to optimize the circuit complexity of our generators, the extremely nice parameters of this family enable us to obtain proof complexity generators from demi-bits generators that are *barely non-trivial*.

Let $G : \{0,1\}^n \to \{0,1\}^m$ be any function and $\mathcal{P}$ be a sound proof system. If $y \in \text{Range}(G)$, then the (false) statement "$y \notin \text{Range}(G)$" is trivially unprovable in $\mathcal{P}$, since $\mathcal{P}$ is sound. Therefore, if G is injective, then there are at least 2^n statements of the form "$y \notin \text{Range}(G)$" that are infeasible to prove in $\mathcal{P}$. We say that G is a *barely non-trivial* demi-bits generator if there are at least $100n2^n$ many statements of the form "$y \notin \text{Range}(G)$" that are infeasible to prove in $\mathcal{P}$. This bound only exceeds the trivial bound of 2^n by an $O(n)$ factor, hence, as a demi-bits generator, G looks *barely non-trivial*. However, composing G with a hash function in an m-wise independent hash family, we obtain that:

Theorem 1.19 (Informal)**.** *If there is a barely non-trivial demi-bits generator against $\mathcal{P}$, then there is a proof complexity generator against $\mathcal{P}$.*

Finally, we note that Ilango's proof [Ila25] can be regarded as a construction of *seeded* zero-error disperser families as well; see Section 3.3.

1.5 Interlude: Are Lower Bound Properties Inherently Large?

A major objection to the natural proof barrier is its *largeness* requirement [Cho11; Wil16; CWY23]. For example, Williams [Wil16] showed that circuit lower bounds for NEXP are equivalent to properties that are constructive (but not necessarily large). Suppose a complexity theorist discovers a property $\mathcal{P}$ of hard Boolean functions, why does $\mathcal{P}$ have to accept many *random* functions?

Moreover, the largeness issue was exposed in the recent line of work on *hardness magnification* [OS18; OPS21; MMW19; CMMW19; CJW19; CJW20; Hir23; CHO+22; LP21; CLY22]: One can use "simple tricks" (such as kernelization) to reduce the task of proving strong lower bounds (which may be subject to the natural proof barrier) to that of proving weaker lower bounds. Moreover, in many scenarios, the weak lower bound is already known (just not for the specific hard function required for magnification) and can be proved by a natural property, hence a "simple trick" seems to bypass the barrier completely!

We suspect that the relationship between demi-bits generators and proof complexity generators is the key towards understanding the largeness condition. If it were the case that every demi-bits generator is also a proof complexity generator, then largeness would be inherent in any circuit lower bound proofs—when we prove a circuit lower bound, we break TT as a proof complexity generator, hence we also break it as a demi-bits generator, and this is equivalent to constructing an NP-natural property (with the largeness requirement). However, if demi-bits generators exist at all, then one can construct contrived demi-bits generators that are not proof complexity generators.[7] Still, Theorem 1.8 along with the recent works connecting demi-bits generators to proof complexity generators [RWZ26; Ila25] suggest that largeness might be somewhat inherent in circuit lower bound proofs.

Our results imply a version of the natural proof barrier that does not refer to the largeness of *functions* (proven to be hard) but refers to the largeness of *generators* (that the lower bound method applies to). Suppose that the truth table generator TT is a demi-bits generator. Then with constant probability over a random subset $S \subseteq \{0,1\}^n$, $\mathsf{TT}|_S$ is a proof complexity generator, which implies hardness of a large family of explicit construction problems that look very similar to circuit lower bounds (namely, hard partial truth tables over a random domain $S \subseteq \{0,1\}^n$). To summarize, the demi-hardness of

[7] Let $G : \{0,1\}^n \to \{0,1\}^N$ be a demi-bits generator, $y \in \{0,1\}^N$, and G_y be the same generator as G except that when G returns y, G_y returns some string different from y. It is trivial to show that $y \notin \text{Range}(G_y)$, i.e., to break G_y as a proof complexity generator.

TT implies infeasibility of not only proving circuit lower bounds for *random* functions, but also proving circuit lower bounds for arbitrary functions over a *random* domain.

In his influential work on "the hardest explicit constructions," Korten [Kor21] concluded[8] that to prove a circuit lower bound for E, one must use specific properties of TT that distinguish it from an arbitrary generator. Going one step further, we conclude that one must use specific properties of TT that distinguish it from *most* generators of the form $\mathsf{TT}|_S$! In any case, we believe that our work will lead to a better understanding of the role of largeness in circuit lower bound proofs.

2 Preliminaries

In this paper, a *circuit class* $\mathscr{C}$ is just any class of Boolean functions closed under projections. In contrast to standard circuit classes considered in the literature such as AC^0 and $\mathsf{P}/_{\mathrm{poly}}$, we will consider *heavily restricted* circuit classes such as NC^0_k or the class of k-bit parities.

We use $\{x_i \mapsto b_i\}_{i\in[L]}$ to denote a partial function f such that $f(x_i) = b_i$ for every $i \in [L]$.

2.1 Explicit Construction Problems

Let $\mathscr{C}$ be a (multi-output) circuit class, we consider the following problems.

- $\mathscr{C}$-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 underlying circuits belong to $\mathscr{C}$, with input length n and output length m, and where the desired output has relative Hamming distance $> c(n)$ from every string in the range of circuits in $\mathscr{C}$.

  $\mathscr{C}$-Partial-Hard$[n, s, L]$: Given L input strings $x_1, x_2, \ldots, x_L \in \{0,1\}^n$, output L bits $b_1, b_2, \ldots, b_L$ such that no circuit $C \in \mathscr{C}$ of size $\leq s$ satisfies $C(x_i) = b_i$ for all $i \in [L]$.
- $\mathscr{C}$-Partial-AvgHard$[n, s, L, \delta]$: Given L input strings $x_1, x_2, \ldots, x_L \in \{0,1\}^n$, output L bits $b_1, b_2, \ldots, b_L$ such that for every circuit $C \in \mathscr{C}$ of size $\leq s$,

$$\frac{1}{L}\left|\{i \in [L] : C(x_i) \neq b_i\}\right| \geq \delta.$$

2.2 Demi-Bits Generators

Definition 2.1 (Demi-Bits Generators)**.** Let n, m be length parameters such that $n < m$. A function $G : \{0,1\}^n \to \{0,1\}^m$ is an (s, ε)-*secure demi-bits generator* if there is no $\mathsf{NP}/_{\mathrm{poly}}$ adversary Adv of size s such that

$$\Pr_{y\leftarrow\{0,1\}^m}[\mathsf{Adv}(y) \text{ accepts}] \geq \varepsilon \quad \text{and} \quad \Pr_{x\leftarrow\{0,1\}^n}[\mathsf{Adv}(G(x)) \text{ rejects}] = 1.$$

In this paper we assume the existence of demi-bits generators for any $\mathsf{NP}/_{\mathrm{poly}}$ adversary with polynomial size, so we will just write ε-secure demi-bits generators. Equivalently, no efficient proof system can prove the statement "$y \notin \mathrm{Range}(G)$" for $\geq \varepsilon 2^m$ many strings y (every such proof system must fail on at least $(1-\varepsilon)2^m$ strings), since if there is such a proof then an $\mathsf{NP}/_{\mathrm{poly}}$ adversary can first guess such a proof and verify it. Notice that the notion of demi-bits generators is weaker when ε is larger.

We will also study the following notion of *weak* demi-bits generators. To break a K-weak demi-bits generator G, the adversary needs to prove "$y \notin \mathrm{Range}(G)$" for *all but* K many strings y.

[8] Modulo the conjecture that Avoid $\notin \mathsf{FP}$, which is now confirmed under cryptographic assumptions [ILW23].

Definition 2.2 (Weak Demi-Bits Generators)**.** Let n, m be length parameters such that $n < m$. A function $G : \{0,1\}^n \to \{0,1\}^m$ is an (s, K)-*weak demi-bits generator* if there is no $\mathsf{NP}/_{\mathrm{poly}}$ adversary Adv of size s such that

$$|\{y \in \{0,1\}^m : \mathsf{Adv}(y) \text{ rejects}\}| < K \quad \text{and} \quad \Pr_{x \leftarrow \{0,1\}^n}[\mathsf{Adv}(G(x)) \text{ rejects}] = 1.$$

Again, we assume the existence of weak demi-bits generators for any $\mathsf{NP}/_{\mathrm{poly}}$ adversary with polynomial size, so we will just write K-weak demi-bits generator. Note that a K-weak demi-bits generator is equivalent to a $(1-K/2^m)$-secure demi-bits generator. Also, any injective function $G : \{0,1\}^n \to \{0,1\}^m$ is trivially a 2^n-weak demi-bits generator. When we refer to weak demi-bits generators, we usually work in the regime where K is just slightly larger than 2^n (and we deduce hardness of Avoid even in this regime).

We will also consider demi-bits generators against uniform (i.e., NP) adversaries (or uniform propositional proof systems, see Definition 2.3). When the demi-bits generator G is non-uniform but the adversary is uniform, we assume that the adversary gets the circuit description of G as input as well.

2.3 Proof Complexity

Recall that TAUT is the coNP-complete language consisting of all tautological DNFs. In one sentence, a *proof system* is a nondeterministic algorithm for TAUT.

Definition 2.3 ([CR79])**.** A *Cook–Reckhow* proof system for a language L (usually $L = \text{TAUT}$) is a deterministic algorithm $V(\varphi, \pi)$ that takes a purported tautology φ and a purported proof π as inputs and satisfies the following requirements:

- **Completeness:** For every $\varphi \in \text{TAUT}$, there exists a proof π such that $V(\varphi, \pi)$ accepts.
- **Soundness:** For every $\varphi \notin \text{TAUT}$ and every purported proof π, $V(\varphi, \pi)$ rejects.
- **Efficiency:** $V(\varphi, \pi)$ runs in time $\mathrm{poly}(|\varphi| + |\pi|)$.

Note that it is possible that $|\pi|$ is super-polynomially longer than φ. Indeed, there exists a proof system such that every $\varphi \in \text{TAUT}$ has a polynomially long proof π if and only if $\mathsf{NP} = \mathsf{coNP}$ (such proof systems are called "p-bounded"), and the goal of propositional proof complexity is to exhibit tautologies requiring super-polynomial (or even exponential) proof length in natural proof systems. We refer to [Kra19; Seg07] for a comprehensive overview of proof complexity.

Proof complexity generators. Let $G : \{0,1\}^n \to \{0,1\}^N$ be a polynomial-size circuit and $y \in \{0,1\}^N$. We fix a propositional encoding of the statement "$y \notin \mathrm{Range}(G)$" into DNFs, such as the ones in [ABRW04]. Such an encoding is possible because this statement can be written as "$\forall x \in \{0,1\}^n$, $G(x) \neq y$", hence the propositional variables of "$y \notin \mathrm{Range}(G)$" consists of x (and possibly extension variables representing the computational history of $G(x)$). As discussed in Section 4.3, most natural encodings would fit in our purpose.

Let $\mathcal{P}$ be a proof system. We say G is a *proof complexity generator* against $\mathcal{P}$ if for every $y \in \{0,1\}^N$, the DNF encoding of "$y \notin \mathrm{Range}(G)$" does not admit polynomial-size proofs in $\mathcal{P}$. We emphasize that if y is actually in $\mathrm{Range}(G)$, then by soundness of $\mathcal{P}$, "$y \notin \mathrm{Range}(G)$" does not admit proofs in $\mathcal{P}$ (of any length). Hence this definition is equivalent to saying that for every string $y \in \{0,1\}^N \setminus \mathrm{Range}(G)$, the *tautology* "$y \notin \mathrm{Range}(G)$" is hard to prove in $\mathcal{P}$.

Again, when the proof system is uniform but the proof complexity generator G is non-uniform, we assume that the proof system gets the circuit description of G as well. This is reflected in the encoding of "$y \notin \mathrm{Range}(G)$" (which contains a description of G).

2.4 Nondeterministic Algorithms

FNP v.s. SearchNP. In this paper, we will distinguish between the two notions FNP and SearchNP.

Definition 2.4 (SearchNP [CL24])**.** Let P be a search problem and R be the binary relation defining P. We say P can be solved by a nondeterministic polynomial-time algorithm if there is a nondeterministic Turing machine M such that for every input x,

- If x has a solution, then $M(x)$ has an accepting computation path, and every accepting path will output a valid solution y, i.e., $R(x, y)$ is true.
- If x has no solution, then $M(x)$ has no accepting computation path.

The class of search problems solvable by nondeterministic polynomial-time algorithm is defined as SearchNP.

Definition 2.5 (FNP [CL24])**.** The class of search problems defined by a polynomial-time relation, i.e., $R \in \mathsf{P}$ is defined as FNP.

While it is clear that $\mathsf{FNP} \subseteq \mathsf{SearchNP}$, the following example suggests that this inclusion is strict.

Proposition 2.6 ([CL24])**.** *If* $\mathsf{P} \neq \mathsf{NP}$*, then there is a total search problem in* $\mathsf{SearchNP} \setminus \mathsf{FNP}$.

We define *errorless* SearchNP *heuristics* for search problems (such as Avoid or Partial-Hard):

Definition 2.7. Let Π be a search problem. We say a nondeterministic algorithm $\mathcal{A}$ is an *errorless heuristic* for Π if for every input instance x, every computational path of $\mathcal{A}(x)$ either outputs $\perp$ or outputs a correct Π-solution of x.

If $\mathcal{A}$ is an errorless heuristic for Π and x is an instance, then we say that $\mathcal{A}$ solves Π on input x if there exists at least one computational path on which $\mathcal{A}$ does not output $\perp$ (which means that $\mathcal{A}$ outputs a correct Π-solution of x). Otherwise, we say that $\mathcal{A}(x) = \perp$.

Let Π be a search problem, $\mathcal{D}$ be a distribution over inputs of Π, and $\varepsilon > 0$. We say that an errorless heuristic $\mathcal{A}$ *solves* Π *on* $\mathcal{D}$ *with success probability* $1 - \varepsilon$ if

$$\Pr_{x \leftarrow \mathcal{D}}[\mathcal{A}(x) = \perp] \leq \varepsilon.$$

2.5 Arthur–Merlin Protocols

An *Arthur–Merlin* protocol [Bab85] for a language L is a constant-round public-coin interactive protocol between a computationally unbounded Prover (Merlin) and a randomized polynomial-time Verifier (Arthur) that satisfies the following properties for every input x:

- *(Completeness)* If $x \in L$, then there is a Prover that makes the Verifier accept w.p. $\geq 2/3$.
- *(Soundness)* If $x \notin L$, then no Prover can make the Verifier accept w.p. $> 1/3$.

Let AM denote the set of languages with an Arthur–Merlin protocol.

In this paper, we do not require the full power of AM; we only require an average-case variant where completeness holds over random instances rather than all yes-instances:

- *(Soundness)* If $x \notin L$, then no Prover can make the Verifier accept w.p. $> 1/3$.
- *(Completeness)* If $x \leftarrow L$ is a random yes-instance, then there exists a Prover that makes the Verifier accept w.p. $\geq 2/3$ over the randomness of the instance and the Verifier's coins.

For example, when refuting random k-CNFs: soundness requires that no Prover can make the Verifier accept a satisfiable CNF w.p. $> 1/3$, while completeness only requires that the Verifier accepts w.p. $\geq 2/3$ over a random CNF and its own coins. (See, e.g., Definition 5.4.)

2.6 Nondeterministic CSP Refutation

Definition 2.8 (Nondeterministic Strong Refutation of Semi-random k-XOR)**.** A *semi-random* k-XOR *instance* ϕ on n Boolean variables consists of an adversarial k-uniform hypergraph $\mathcal{H}$ on $[n]$ with m hyperedges and i.i.d. uniform random right-hand sides $b_C \sim \{\pm 1\}$ for $C \in \mathcal{H}$. Associated to ϕ is the polynomial

$$\phi(x) \;=\; \frac{1}{m} \sum_{C \in \mathcal{H}} b_C \prod_{i \in C} x_i,$$

so that $\mathrm{val}(\phi) = \frac{1}{2} + \frac{1}{2} \max_{x \in \{\pm 1\}^n} \phi(x)$.

A *nondeterministic ε-tight refutation algorithm* is a polynomial-time Verifier V that takes as input an instance ϕ, a witness $w \in \{0,1\}^{\mathrm{poly}(n)}$, and a claimed upper bound $\alpha \in [0,1]$, such that:

- *(Soundness)* For every instance ϕ, every witness w, and every α,

$$V(\phi, w, \alpha) = 1 \implies \max_{x \in \{\pm 1\}^n} \phi(x) \;\le\; \alpha.$$

- *(Completeness)* For every adversarial $\mathcal{H}$ with m sufficiently large,

$$\Pr_{b \sim \{\pm 1\}^m} \big[\exists\, w \in \{0,1\}^{\mathrm{poly}(n)} :\; V(\phi, w, \varepsilon) = 1 \big] \;\ge\; 0.99.$$

In particular, whenever V accepts, the certified bound $\mathrm{val}(\phi) \le \frac{1}{2} + \frac{\alpha}{2}$ holds. For $\varepsilon \to 0$, this approaches the random assignment threshold of $\frac{1}{2}$.

2.7 Useful Results

Pajor's lemma. Let $H \subseteq \{0,1\}^m$, and $I \subseteq [m]$ denote a subset of indices. We say that H *shatters* I if for every string $z \in \{0,1\}^I$, there exists a string $y \in H$ such that $y|_I = z$.

Lemma 2.9 (Pajor's Lemma [Paj85])**.** *Let $H \subseteq \{0,1\}^m$. Then there are at least $|H|$ many subsets of indices I such that H shatters I.*

As a direct corollary of Pajor's lemma, we have:

Lemma 2.10 (Sauer–Shelah Lemma [Sau72; She72; VA68])**.** *Let $H \subseteq \{0,1\}^m$ such that no subset $I \subseteq [m]$ with size k shatters H. Then*

$$|H| \le \sum_{i=0}^{k-1} \binom{m}{i}.$$

Kruskal–Katona theorem. Let $\mathcal{F}$ be a collection of subsets of $[n]$, we say $\mathcal{F}$ is *downward closed* if for every two subsets $C \subseteq D \subseteq [n]$, if $D \in \mathcal{F}$, then $C \in \mathcal{F}$. For a positive real number $x > 0$ and an integer $i \le x$, define

$$\binom{x}{i} := \frac{x(x-1)\dots(x-i+1)}{i!}.$$

We need the following version of the Kruskal–Katona theorem [Kru63; Kat68; Har66; CL69], formulated by Lovász [Lov07, 13.31b]:

Theorem 2.11 (Kruskal–Katona Theorem, Lovász's Formulation)**.** *Let $\mathcal{F}$ be a collection of subsets of $[n]$ that is downward closed, and for each $k \in \mathbb{N}$ denote $\mathcal{F}_k$ to be the collection of size-k subsets in $\mathcal{F}$. For every $i > j$, if $|\mathcal{F}_i| = \binom{x}{i}$ where $x \ge i$ is real, then $|\mathcal{F}_j| \ge \binom{x}{j}$.*

Prefix sum of binomial coefficients. Let $H(p) = -p\log p - (1-p)\log(1-p)$ be the binary entropy function, then:

Proposition 2.12. *For every integers n, k such that $k \le n/2$, we have*

$$\sum_{i=0}^{k} \binom{n}{i} \le 2^{n \cdot H(k/n)}.$$

3 Disperser Families

Definition 3.1 ((k,ε) Zero-Error Disperser Family)**.** Let $\mathcal{F}$ be a distribution over functions $f : \{0,1\}^n \to \{0,1\}^m$, $k \ge m$, and $\varepsilon > 0$ be parameters. We say that $\mathcal{F}$ is a (k,ε) *zero-error disperser family*, if for every random variable $\mathcal{X}$ over $\{0,1\}^n$ with $H_\infty(\mathcal{X}) \ge k$,

$$\Pr_{f \leftarrow \mathcal{F}}[\mathrm{Supp}(f(\mathcal{X})) = \{0,1\}^m] \ge 1 - \varepsilon.$$

We say $\mathcal{F}$ is efficient if every function $f \in \mathcal{F}$ has a polynomial size description and is computable in polynomial time.

It is easy to see that in Definition 3.1, it suffices to consider random variables $\mathcal{X}$ that are uniform distributions over 2^k many strings.

3.1 Random Projections

Let $\mathrm{Proj}_{n\to m}$ be the distribution of a uniformly random projection from n bits to m bits. That is, to sample $f : \{0,1\}^n \to \{0,1\}^m$ from $\mathrm{Proj}_{n\to m}$, sample a random subset $I \subseteq [n]$ of size m, let $I = \{i_1, i_2, \ldots, i_m\}$ where $1 \le i_1 < i_2 < \cdots < i_m \le n$, and define $f(x) = x|_I := x_{i_1} x_{i_2} \ldots x_{i_m}$.

We first observe that the Sauer–Shelah Lemma implies:

Theorem 3.2. *For any $k \le n$ and some $m := \Omega(k/\log(2n/k))$, $\mathrm{Proj}_{n\to m}$ is a $(k, 1-\delta)$ zero-error disperser family with some (exponentially-small) $\delta > 0$.*

Proof. Let $\mathcal{X}$ be a random variable over $\{0,1\}^n$ with min-entropy k. Without loss of generality, we may assume that $\mathcal{X}$ is the uniform distribution over a subset of 2^k many strings, and abusing notation, we denote this size-2^k subset as $\mathcal{X}$ as well.

For some $m = \Omega(k/\log(2n/k))$, we have

$$\left(\frac{en}{m}\right)^m < 2^k.$$

By Sauer–Shelah Lemma, since

$$|\mathcal{X}| = 2^k \ge \left(\frac{en}{m}\right)^m \ge \sum_{i=0}^{m-1} \binom{n}{i},$$

there exists a subset $I \subseteq [n]$ with $|I| = m$ such that I is shattered by $\mathcal{X}$. It follows that the function $f(x) = x|_I$ in the support of $\mathrm{Proj}_{n\to m}$ satisfies that $\mathrm{Supp}(f(\mathcal{X})) = \{0,1\}^m$. □

The success probability δ in Theorem 3.2 is inherently exponentially small if $k \ll n$: Let $\mathcal{X}$ be the uniform distribution over strings whose last $n-k$ bits are 0 (but first k bits can be arbitrary). Then, a random function f from $\mathrm{Proj}_{n\to m}$ satisfies $\mathrm{Supp}(f(\mathcal{X})) = \{0,1\}^m$ only when the underlying index set I is a subset of $[k]$, which happens with probability $\approx (k/n)^m \ll o(1)$. However, when k is very close to n, by combining Pajor's Lemma and Kruskal–Katona Theorem, we can indeed show that $\mathrm{Proj}_{n\to m}$ is a good disperser family with probability close to 1 (a typical parameter setting of the following theorem is that Δ being a constant and $m = \varepsilon n/\Delta$ for some small constant $\varepsilon > 0$):

Theorem 3.3 (Random Projection Disperser Family)**.** *Let $m \le 0.4n$ and $\Delta \le o(n)$, then $\mathrm{Proj}_{n\to m}$ is an $(n-\Delta, 4m\Delta/n)$ zero-error disperser family.*

Proof. Let $\mathcal{X} \subseteq \{0,1\}^n$ be a subset of size $2^{n-\Delta}$. Let $\mathcal{F}$ be the collection of subsets of indices $I \subseteq [n]$ that are shattered by $\mathcal{X}$. Recall this means that for every $z \in \{0,1\}^I$, there exists $y \in \mathcal{X}$ such that $y|_I = z|_I$. By Pajor's Lemma, $|\mathcal{F}| \ge |\mathcal{X}| \ge 2^{n-\Delta}$. Moreover, $\mathcal{F}$ is downward closed: if $\mathcal{X}$ shatters I, then $\mathcal{X}$ shatters every subset of I.

For every $k \in \mathbb{N}$, let $\mathcal{F}_k$ denote the collection of size-k sets in $\mathcal{F}$. Suppose that $|\mathcal{F}_m| = \binom{x}{m}$ for some real number $x \ge m$. Then by Kruskal–Katona Theorem, for every integer $i \in [m, x]$, $|\mathcal{F}_i| \le \binom{x}{i}$; while if $i > x$ then there is no size-i subset in $\mathcal{F}$. It follows that

$$|\mathcal{F}| \le \sum_{i=0}^{m-1} \binom{n}{i} + \sum_{i=m}^{\lfloor x \rfloor} \binom{x}{i} \le 2^{n\cdot H(m/n)} + 2^x \le 2^{0.9999n} + 2^x.$$

Therefore $x \ge n - \Delta - 1$. It follows that a random size-m subset of $[n]$ is in $\mathcal{F}$ with probability

$$\binom{x}{m} / \binom{n}{m} \ge \left(\frac{x-m}{n-m}\right)^m \ge \left(1 - \frac{\Delta+1}{n-m}\right)^m \ge 1 - 4m\Delta/n.$$

Let $f : \{0,1\}^n \to \{0,1\}^m$ be a random projection sampled from $\mathrm{Proj}_{n\to m}$, and let $I \subseteq [n]$ be the subset of indices such that $f(x) = x|_I$. With probability at least $1 - 4m\Delta/n$, we have $I \in \mathcal{F}$, and thus $\mathcal{X}$ shatters I. This means that $\mathrm{Supp}(f(\mathcal{X})) = \{0,1\}^m$. □

3.2 t-Wise Independent Hash Families

We also show that t-wise independent hash functions are good disperser families in the suitable parameter regime. Although the circuit complexity overhead of these hash functions is much worse than projections, looking ahead, these disperser families will be used in Section 4.2 to obtain extreme hardness results for (unrestricted) Avoid from mild demi-hardness assumptions and vastly improve the main results of [RWZ26].

We start with pairwise independent hash families. We need the following easy corollary of Chebyshev Inequality.

Lemma 3.4. *Let $x_1, x_2, \ldots, x_n$ be pairwise independent variables where the marginal distribution of each x_i is the Bernoulli distribution with success probability p. Then, the probability that every x_i is equal to 0 is at most $1/(pn)$.*

Theorem 3.5. *Let $k := 2m + \log(1/\varepsilon)$. Let $\mathcal{H}$ be a family of pairwise independent hash functions where each function $h : \{0,1\}^n \to \{0,1\}^m$. Then $\mathcal{H}$ is a (k, ε) zero-error disperser family.*

Proof. Let $\mathcal{X} \subseteq \{0,1\}^n$ be a subset of size 2^k, and let $h \leftarrow \mathcal{H}$ be a random hash function. For every fixed $z \in \{0,1\}^m$, we will argue that $\Pr_{h\leftarrow\mathcal{H}}[z \notin h(\mathcal{X})] \le \varepsilon \cdot 2^{-m}$. Indeed, for every $x \in \mathcal{X}$, let I_x denote the indicator variable of $h(x) = z$, then the marginal distribution of each I_x is the Bernoulli distribution with success probability 2^{-m}, and the variables $\{I_x\}_{x\in\mathcal{X}}$ are pairwise independent. The probability that $z \notin h(\mathcal{X})$ is equal to the probability that every I_x is equal to 0, hence by Lemma 3.4 is at most $2^{m-k} \le \varepsilon \cdot 2^{-m}$. Finally, the theorem then follows from a union bound over all $z \in \{0,1\}^m$. □

For t-wise independence, we need the following tail bound, which follows from [BR94, Lemma 2.3] by substituting $\mu := np$ and $A := np$:

Lemma 3.6. *Let $t \ge 4$ be an even integer and suppose that $x_1, x_2, \ldots, x_n$ are t-wise independent random variables where the marginal distribution of each x_i is the Bernoulli distribution with success probability p. Then, the probability that every x_i is 0 is at most $8(tnp + t^2)^{t/2}/(np)^t$.*

Theorem 3.7. *Let $t \geq 6$ be an even number and $k := m(1+2/t) + \log t + O(\log(1/\varepsilon)/t) + 1$. Let $\mathcal{H}$ be a family of t-wise independent hash functions where each function $h : \{0,1\}^n \to \{0,1\}^m$. Then $\mathcal{H}$ is a (k, ε) zero-error disperser family.*

Proof. We follow the same proof as Theorem 3.5, but use Lemma 3.6 instead of Lemma 3.4.

Let $\mathcal{X} \subseteq \{0,1\}^n$ be a subset of size 2^k, and let $h \leftarrow \mathcal{H}$ be a random hash function. For every fixed $z \in \{0,1\}^m$, we will argue that $\Pr_{h \leftarrow \mathcal{H}}[z \notin h(\mathcal{X})] \leq \varepsilon \cdot 2^{-m}$. Indeed, for every $x \in \mathcal{X}$, let I_x denote the indicator variable of $h(x) = z$, then the marginal distribution of each I_x is the Bernoulli distribution with success probability 2^{-m}, and the variables $\{I_x\}_{x \in \mathcal{X}}$ are t-wise independent. The probability that $z \notin h(\mathcal{X})$ is equal to the probability that every I_x is equal to 0, hence by Lemma 3.6 is at most

$$\begin{aligned}
\frac{8(t \cdot 2^{k-m} + t^2)^{t/2}}{2^{(k-m)t}} &\leq \frac{8 \cdot (2t \cdot 2^{k-m})^{t/2}}{2^{(k-m)t}} && \text{(Since } 2^{k-m} \geq t\text{)} \\
&\leq \frac{8 \cdot (2t)^{t/2} \cdot 2^{mt/2}}{2^{kt/2}} \\
&\leq \frac{8 \cdot (2t)^{t/2} \cdot 2^{mt/2}}{2^{mt/2} 2^m (2t)^{t/2} \cdot \mathrm{poly}(1/\varepsilon)} \\
&\leq \varepsilon \cdot 2^{-m}.
\end{aligned}$$

Finally, the theorem then follows from a union bound over all $z \in \{0,1\}^m$. □

3.3 Ilango's Seeded Disperser Family

We also remark that Ilango's proof [Ila25] implicitly uses a *seeded* disperser family as well:

Theorem 3.8. *Let $\Delta \geq 1$, $\varepsilon > 0$, $d := \Delta + \log n + \log(1/\varepsilon) + O(1)$, and $\mathcal{F}$ be the following distribution of functions $f : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^n$. To sample a function $f \leftarrow \mathcal{F}$, sample $t := 2^d$ random strings ("shifts") $s_1, s_2, \ldots, s_t \leftarrow \{0,1\}^n$ and define $f(x,i) = x \oplus s_i$.*

Then $\mathcal{F}$ is an $(n - \Delta, \varepsilon)$ zero-error seeded *disperser family in the following sense. For every random variable $\mathcal{X}$ over $\{0,1\}^n$ with $H_\infty(\mathcal{X}) \geq n - \Delta$, let $\mathcal{Z}$ be the uniform distribution over $\{0,1\}^d$ independent from everything else, then*

$$\Pr_{f \leftarrow \mathcal{F}}[\mathrm{Supp}(f(\mathcal{X}, \mathcal{Z})) = \{0,1\}^n] \geq 1 - \varepsilon.$$

Proof. This is basically the same proof as in [Ila25] and [Lau83]. Again, without loss of generality, we may assume that $\mathcal{X}$ is the uniform distribution over some $X \subseteq \{0,1\}^n$ of size at least $2^{n-\Delta}$.

Let $s_1, s_2, \ldots, s_t \leftarrow \{0,1\}^n$ be sampled independently and uniformly at random. For each $z \in \{0,1\}^n$, the probability over $s \leftarrow \{0,1\}^n$ that $z \notin X \oplus s$ is at most $1 - 2^{-\Delta}$, hence the probability that $z \notin X \oplus s_i$ for every $i \in [t]$ is at most $(1 - 2^{-\Delta})^t < \varepsilon \cdot 2^{-n}$. It follows from a union bound over all $z \in \{0,1\}^n$ that with probability at least $1 - \varepsilon$, the support of $f(\mathcal{X}, \mathcal{Z})$ contains every length-n string. □

4 Hardness of Range Avoidance

Our main result is that composing zero-error disperser families with demi-bits gives us hard instances for the Range Avoidance problem. We have the following theorem.

Theorem 4.1. *Let $G : \{0,1\}^n \to \{0,1\}^N$ be a 2^k-weak demi-bits generator, and $\mathcal{F} = \{f : \{0,1\}^N \to \{0,1\}^m\}$ be an efficient (k, ε) zero-error disperser family. For any function $f \in \mathcal{F}$ define the function $C_f : \{0,1\}^n \to \{0,1\}^m$ such that $\forall x \in \{0,1\}^n, C_f(x) = f(G(x))$. Let $\mathscr{C}$ be a circuit class where $C_f \in \mathscr{C}$ for all $f \in \mathcal{F}$. Then, for any errorless* SearchNP *heuristic $\mathcal{A}$ for $\mathscr{C}$-*Avoid$[n, m]$, $\Pr_{f \leftarrow \mathcal{F}}[\mathcal{A}(C_f) \neq \perp] \leq \varepsilon$.*

Proof. Suppose for contradiction that some errorless SearchNP heuristic $\mathcal{A}$ satisfies $\Pr_f[\mathcal{A}(C_f) \neq \perp] > \varepsilon$. We construct a nondeterministic polynomial-time adversary $\mathcal{B}$ breaking the demi-bits generator G. On input $y \in \{0,1\}^N$, the adversary $\mathcal{B}$ accepts if and only if there exists $f \in \mathcal{F}$ such that some nondeterministic branch of $\mathcal{A}(C_f)$ outputs $f(y)$.

$\mathcal{B}$ rejects every $y \in \mathrm{Range}(G)$. If $y = G(s)$, then for every $f \in \mathcal{F}$ we have $C_f(s) = f(G(s)) = f(y)$, so $f(y) \in \mathrm{Range}(C_f)$. Since $\mathcal{A}$ is an errorless solver for Avoid, no accepting branch of $\mathcal{A}(C_f)$ outputs an element of $\mathrm{Range}(C_f)$, so $\mathcal{B}(y)$ rejects.

$\mathcal{B}$ accepts at least $2^N - 2^k$ strings $y \in \{0,1\}^N$. Assume to the contrary that $\mathcal{B}$ accepts $\leq 2^N - 2^k$ strings of length N. Let X be the uniform distribution over the strings in $\{0,1\}^N$ rejected by $\mathcal{B}$. Then we have $|\mathrm{Supp}(X)| \geq 2^k$ and $H_\infty(X) \geq k$. Since $\mathcal{F}$ is an efficient (k,ε) zero-error disperser family, we have

$$\Pr_{f \leftarrow \mathcal{F}}\left[\mathrm{Supp}(f(X)) = \{0,1\}^m\right] \;\geq\; 1 - \varepsilon.$$

Since $\Pr_f[\mathcal{A}(C_f) \neq \perp] > \varepsilon$, there must exist some $f \in \mathcal{F}$ such that $\mathcal{A}(C_f) \neq \perp$ and $\mathrm{Supp}(f(X)) = \{0,1\}^m$. This means that there exists some $x \in \mathrm{Supp}(X)$ such that some nondeterministic accepting branch of $\mathcal{A}(C_f)$ outputs exactly $f(x)$, hence $\mathcal{B}$ accepts x, a contradiction.

Hence $\mathcal{B}$ is a nondeterministic polynomial-time adversary that rejects all of $\mathrm{Range}(G)$ and accepts all but 2^k many strings in $\{0,1\}^N$, contradicting the demi-bits security of G. □

Remark 4.2. Recall that $G : \{0,1\}^n \to \{0,1\}^N$ is a 2^k-weak demi-bits generator if any efficient proof system must fail to prove the statement "$y \notin \mathrm{Range}(G)$" for at least 2^k strings $y \in \{0,1\}^N$. Applying any efficient function $f : \{0,1\}^N \to \{0,1\}^m$, one can "convert" a statement "$y \notin \mathrm{Range}(G)$" that is hard to prove into another statement "$f(y) \notin \mathrm{Range}(C_f)$" that is also hard to prove, where $C_f = f \circ G$. If the function f is a zero-error disperser for the set of 2^k strings y that are hard for G, then the image of these hard strings under f covers the entire $\{0,1\}^m$, and we get hard Avoid instances.

Similarly, composing zero-error disperser families with demi-bits generators gives us (non-uniform) proof complexity generators secure against every (uniform) proof system. This result can be proved by enumerating all such proof systems and using the Borel–Cantelli lemma; see [Ila25, Theorem 3.6]. Here we present an alternative proof due to Krajíček [Kra26] using the existence of optimal proof systems with one bit of advice.

Theorem 4.3. *Let $G : \{0,1\}^n \to \{0,1\}^N$ be a 2^k-weak demi-bits generator, and $\mathcal{F} = \{f : \{0,1\}^N \to \{0,1\}^m\}$ be an efficient (k,ε) zero-error disperser family. Then, with probability $\geq 1 - \varepsilon$ over the choice of $f \leftarrow \mathcal{F}$, the generator $C : \{0,1\}^n \to \{0,1\}^m$ defined as*

$$C(x) := f(G(x))$$

is a proof complexity generator against all (uniform) proof systems.

Proof Sketch. Cook and Krajíček proved that there exists an optimal proof system $\mathcal{P}$ with one bit of advice [CK07, Theorem 6.6]. Since G is a secure demi-bits generator against $\mathcal{P}$, the same argument as in Theorem 4.1 shows that with probability at least $1 - \varepsilon$ over $f \leftarrow \mathcal{F}$, $f \circ G$ is a proof complexity generator against $\mathcal{P}$. Since $\mathcal{P}$ simulates every (uniform) proof system, $f \circ G$ is a proof complexity generator against every (uniform) proof system. □

In the remainder of this section, we instantiate Theorem 4.1 and Theorem 4.3 with various (candidate) demi-bits and disperser families to obtain hardness of Avoid in various settings.

4.1 Hardness of Avoid for Restricted Circuits

Thanks to the Random Projection Disperser Family, we can convert demi-bits generators in any complexity class into hard Avoid instances within the same complexity class.

Recall that for a circuit $G : \{0,1\}^n \to \{0,1\}^N$ and a subset of output indices $I \subseteq [N]$, $I = \{i_1, i_2, \ldots, i_m\}$ where $1 \le i_1 < i_2 < \cdots < i_m \le N$, we define $G|_I$ to be the circuit $G|_I : \{0,1\}^n \to \{0,1\}^m$ where for every input $x \in \{0,1\}^n$,

$$G|_I(x) = G(x)|_I = G(x)_{i_1} G(x)_{i_2} \ldots G(x)_{i_m}.$$

Theorem 4.4. *Let $G : \{0,1\}^n \to \{0,1\}^N$ be a $(1 - 2^{-\Delta})$-secure demi-bits generator with $N \ge 3n$ and $\Delta \le o(N)$. Then:*

- *For some universal constant $\alpha < 1$, let $m = \alpha N$ and assuming $m > n$, then any* SearchNP *algorithm $\mathcal{A}$ fails to solve* Avoid *on the instance $G|_I$ for some subset $I \subseteq [N]$, $|I| = m$.*
- *For any $n < m \le 0.4N$ and any errorless* SearchNP *heuristic $\mathcal{A}$ for* Avoid, $\Pr_I\left[\mathcal{A}(G|_I) \ne \bot\right] \le \frac{4m\Delta}{N}$, *where $I \subseteq [N]$ is a uniformly random subset of size $|I| = m$.*

Proof. We combine Theorem 4.1 with the Random Projection Disperser Family. For the first bullet, let $k := N - \Delta$, $m := \Omega(k/\log(N/k)) = \alpha N$, then by Theorem 3.2, $\mathrm{Proj}_{N \to m}$ is a $(k, 1-\delta)$ zero-error disperser family for some $\delta > 0$. Similarly, the second bullet follows from Theorem 3.3. □

Corollary 4.5. *Let $k \in \mathbb{N}$, and $\ell(n) > n$ be a stretch function. The following holds for some universal constant $c \ge 1$: If there exists a demi-bits generator $G : \{0,1\}^n \to \{0,1\}^{c\ell(n)}$ computable in NC^0_k, then NC^0_k-Avoid$[n, \ell(n)] \notin$ SearchNP.*

Combining the current best algorithm for NC^0-Avoid [KPI25; HLZ26; GLY26], we have:

Corollary 4.6. *For every $k \ge 3$ and $\varepsilon > 0$:*

- *Every demi-bits generator computable in NC^0_k with stretch $\Omega_k(n^{(k-1)/2} \log n)$ can be broken in nondeterministic $n^{O(k)}$ time.*
- *Every demi-bits generator computable in NC^0_k with stretch $n^{1+\varepsilon}$ can be broken in nondeterministic $2^{n^{1-\frac{2\varepsilon}{k-3}+o(1)}}$ time.*

Proof. The first bullet follows from a deterministic algorithm for NC^0_k-Avoid$[n, \Omega_k(n^{(k-1)/2} \log n)]$ in $n^{O(k)}$ time, while the second bullet follows from a deterministic algorithm for NC^0_k-Avoid$[n, n^{1+\varepsilon}]$ in $2^{n^{1-\frac{2\varepsilon}{k-3}+o(1)}}$ time [HLZ26; GLY26]. □

4.2 Hardness of Avoid from Barely Non-trivial Demi-Bits

In this section, we instantiate Theorem 4.1 and Theorem 4.3 with the t-wise independent hash families as dispersers, and derive various hardness results for Avoid.

The main result of [RWZ26] is that for every demi-bits generator $G : \{0,1\}^n \to \{0,1\}^N$, pairwise independent hash family $\mathcal{H} = \{h : \{0,1\}^N \to \{0,1\}^m\}$, and every errorless SearchNP heuristic $\mathcal{A}$, there exists a hash function $h \in \mathcal{H}$ such that $\mathcal{A}$ fails to solve Avoid on the instance $h \circ G$. Our new framework allows us to improve this theorem in two aspects: *with high probability* over $h \leftarrow \mathcal{H}$, $\mathcal{A}$ fails to solve Avoid on $h \circ G$; and this only requires G to be a *weak* demi-bits generator.

Theorem 4.7 (Strengthening of [RWZ26, Theorem 1.2])**.** *Let $G : \{0,1\}^n \to \{0,1\}^N$ be a demi-bits generator, $\mathcal{H} = \{h : \{0,1\}^N \to \{0,1\}^m\}$ be a family of pairwise independent hash functions, and $\mathcal{A}$ be an errorless* SearchNP *heuristic for* Avoid. *If $N \ge 4m$ and $m > n$, then with probability $\ge 1 - 2^{-m}$ over $h \leftarrow \mathcal{H}$, $\mathcal{A}$ fails to solve* Avoid *on the input instance $h \circ G$.*

Moreover, this is still true even if G is only a 2^{3m}-weak demi-bits generator.

Proof. We combine Theorem 4.1 with the disperser family in Theorem 3.5. By setting $\varepsilon = 2^{-m}$ in Theorem 3.5, the theorem follows. □

Similarly, we can also get the following corollaries.

Corollary 4.8. *If $\mathcal{H}$ is an m-wise independent hash family, then the above theorem still holds even if G is only a $cm2^m$-weak demi-bits generator for some universal constant $c \geq 1$.*

Proof. We combine Theorem 4.1 with the disperser family in Theorem 3.7 and set $\varepsilon = 2^{-m}$ there. □

Any injective function $G : \{0,1\}^n \to \{0,1\}^N$ is trivially a 2^n-weak demi-bits generator, since for every $y \in \text{Range}(G)$, it is impossible to prove that "$y \notin \text{Range}(G)$" in a sound proof system. In this regard, an $O(n2^n)$-weak demi-bits generator is *barely non-trivial*. However, using an n-wise independent hash family, we show that such a barely non-trivially weak demi-bits generator implies proof complexity generators:

Corollary 4.9. *For some universal constant $c \geq 1$, if there exists a $cn2^n$-weak demi-bits generator $G : \{0,1\}^n \to \{0,1\}^N$ with $N \geq 2n$, then there exists a single (non-uniform family of) proof complexity generator secure against all (uniform) proof systems.*

Proof. We combine Theorem 4.3 with the disperser family in Theorem 3.7, where we use an n-wise independent hash family, and set $m = n + 2$ and error $\varepsilon = 2^{-m}$ to get a disperser family $\mathcal{H} = \{h : \{0,1\}^N \to \{0,1\}^m\}$. By Theorem 3.7, the entropy requirement is $k \geq n + \log n + O(1)$. Therefore we only need a $O(n2^n)$-weak demi-bits generator. □

4.3 Many Proof Complexity Generators Inside One Demi-Bits Generator

In this subsection, we show that for every well-behaved proof system $\mathcal{P}$, demi-bits generators against $\mathcal{P}$ imply proof complexity generators against $\mathcal{P}$.

Definition 4.10. We say that a proof system $\mathcal{P}$ is *well-behaved* if there exists a polynomial $p(\cdot)$ such that for every two CNFs $C \subseteq C'$ (i.e., every clause appearing in C also appears in C'), if $\mathcal{P}$ admits a length-ℓ refutation of C, then $\mathcal{P}$ admits a length-poly(ℓ) refutation of C'.

That is, a proof system is *well-behaved* if adding more axioms cannot make the proof harder. This is a very natural assumption that holds for essentially every proof system of interest: to prove C' is unsatisfiable, it suffices to only look at the clauses in C. We are not aware of any proof system studied in the literature that is not well-behaved, and we believe that any such proof system needs to be very contrived.

Let $G : \{0,1\}^n \to \{0,1\}^m$ and $y \in \{0,1\}^m$, we use the notation "$y \notin \text{Range}(G)$" to represent the propositional formula asserting that there is no $z \in \{0,1\}^n$ such that $G(z) = y$. (This formula is also denoted as $\tau_y(G)$ in the literature [Kra25].) We do not specify how to encode "$y \notin \text{Range}(G)$" as a CNF here; any encoding of "$y \notin \text{Range}(G)$" satisfying the following fact would suffice, and we also remark that the encoding described in [RWZ26, Section 3.1] satisfies the fact.

Fact 4.11. *Let $G : \{0,1\}^n \to \{0,1\}^m$ be a generator and $I \subseteq [m]$. Then every clause appearing in "$y|_I \notin \text{Range}(G|_I)$" also appears in "$y \notin \text{Range}(G)$".*

Theorem 4.12. *Let $\mathcal{P}$ be a well-behaved proof system, then there is a polynomial $p(\cdot)$ such that the following holds. Let $2^{-o(N)} < \varepsilon \leq 1/2$, $n < m < N/(10\log(1/\varepsilon))$, and $G : \{0,1\}^n \to \{0,1\}^N$ be a $(1-\varepsilon)$-secure demi-bits generator against $\mathcal{P}$-proofs of length $p(\ell)$. Then w.p. at least $1 - 4m\log(1/\varepsilon)/N$ over $I \subseteq [N]$, $|I| = m$, $G|_I$ is a proof complexity generator secure against $\mathcal{P}$-proofs of length ℓ.*

Proof. Since $\mathcal{P}$ is well-behaved, there is a polynomial $p(\cdot)$ such that for every two CNFs $C \subseteq C'$, if $\mathcal{P}$ admits a length-ℓ refutation of C, then $\mathcal{P}$ admits a length-$p(\ell)$ refutation of C'.

Let $\textsc{Hard} \subseteq \{0,1\}^N$ be the set of strings $y \in \{0,1\}^N$ such that $\mathcal{P}$ does not admit length-$p(\ell)$ proofs of "$y \notin \mathrm{Range}(G)$". Since G is a $(1-\varepsilon)$-secure demi-bits generator, we have $|\textsc{Hard}| \geq \varepsilon \cdot 2^N$. It follows from the Random Projection Disperser Family that a random subset $I \subseteq [N]$ of size $|I| = m$ is shattered by $\textsc{Hard}$ with probability $\geq 1 - 4m\log(1/\varepsilon)/N$. By Fact 4.11, $G|_I$ is a proof complexity generator in the sense that for every $y \in \{0,1\}^I$, $\mathcal{P}$ does not admit a length-ℓ proof of "$y \notin \mathrm{Range}(G|_I)$". □

Remark 4.13 (Comparison with [RWZ26]). Previously, [RWZ26] proved the same theorem for every proof system $\mathcal{P}$ closed under *simple parity reductions* ([RWZ26, Definition 3.3]). Roughly speaking, this is because [RWZ26] needs to compose the demi-bits generator with a hash function in some universal hash family; since such hash functions need unbounded-fanin XOR gates to compute, the results in [RWZ26] only hold for proof systems that can reason about such XOR gates. In particular, when $\mathcal{P}$ is the *resolution* proof system, [RWZ26] fails to transform demi-bits against $\mathcal{P}$ to proof complexity generators against $\mathcal{P}$.[9] Thanks to the Random Projection Disperser Family, our results hold as long as the proof system is *well-behaved*, which is a much weaker (in fact, nearly trivial) condition compared to being closed under simple parity reductions, and includes weak proof systems such as resolution.

As a corollary, for every well-behaved proof system $\mathcal{P}$, the ability for $\mathcal{P}$ to break demi-bits generators in NC^0 coincides with its ability to break proof complexity generators in NC^0 up to a constant factor in the stretch.

Corollary 4.14. *Let $\mathcal{P}$ be a well-behaved proof system, $k \geq 2$ be a constant, and $\ell(n) > n$ be a function, then the following are equivalent:*

1. *For every constant $c \geq 1$, there is a demi-bits generator $G : \{0,1\}^n \to \{0,1\}^{c\cdot\ell(n)}$ computable in NC^0_k that is 0.99-secure against $\mathcal{P}$.*
2. *For every constant $c \geq 1$, there is a proof complexity generator $G : \{0,1\}^n \to \{0,1\}^{c\cdot\ell(n)}$ computable in NC^0_k that is secure against $\mathcal{P}$.*

Proof. (1) $\implies$ (2) follows from Theorem 4.12: If $G : \{0,1\}^n \to \{0,1\}^{1000\ell(n)}$ is a demi-bits generator that is 0.99-secure against $\mathcal{P}$, then with probability ≥ 0.9 over $I \subseteq [1000\ell(n)]$, $|I| = \ell(n)$, $G|_I$ is a proof complexity generator against $\mathcal{P}$.

(2) $\implies$ (1) follows from definition: If $\mathcal{P}$ can prove "$y \notin \mathrm{Range}(G)$" for *many* strings y, then $\mathcal{P}$ can prove "$y \notin \mathrm{Range}(G)$" for at least *some* string y. □

A similar corollary concerns Goldreich's PRG [Gol11] where the underlying hypergraph is selected uniformly at random. Let $k \geq 3$, $P : \{0,1\}^k \to \{0,1\}$ be any function, and $\ell(n) > n$ be a stretch parameter. Let $\mathcal{G}_{\ell(n)}$ denote the following distribution of candidate PRGs $G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$: For each output bit $i \in [\ell(n)]$, we select k random indices $j_1, j_2, \ldots, j_k \leftarrow [n]$ and set the i-th output bit as

$$G(x)_i := P(x_{j_1}, x_{j_2}, \ldots, x_{j_k}).$$

Corollary 4.15. *Let $k \geq 3$ be a constant, $P : \{0,1\}^k \to \{0,1\}$ be any Boolean function, $\ell(n)$ be a stretch function, and $\mathcal{P}$ be a well-behaved proof system. Then the following are equivalent:*

1. *For every constant $c \geq 1$ and $\varepsilon > 0$, w.p. at least $1-\varepsilon$ over $G \leftarrow \mathcal{G}_{c\cdot\ell(n)}$, G is a $(1-\varepsilon)$-secure demi-bits generator against $\mathcal{P}$.*
2. *For every constant $c \geq 1$ and $\varepsilon > 0$, w.p. at least $1-\varepsilon$ over $G \leftarrow \mathcal{G}_{c\cdot\ell(n)}$, G is a proof complexity generator against $\mathcal{P}$.*

[9] Resolution is *provably not* closed under simple parity reductions. The Tseitin formula is exponentially hard for resolution [Urq87], but it reduces to a trivial formula $x_1 \wedge \bar{x}_1$ under simple parity reductions.

Proof. (1) $\implies$ (2) follows from Theorem 4.12: Fix $c \geq 1$, $\varepsilon > 0$, and let $c' := 1000c\varepsilon^{-1}\log(\varepsilon^{-1})$ and $\varepsilon' := \varepsilon/3$. Suppose that $G \leftarrow \mathcal{G}_{c'\ell(n)}$ and $G' := G|_I$ for a random subset $I \subseteq [c'\ell(n)]$ with $|I| = c\ell(n)$, then the distribution of G' is exactly $\mathcal{G}_{c\ell(n)}$. Moreover, if G is a $(1-\varepsilon')$-secure demi-bits generator secure against $\mathcal{P}$ (which happens w.p. $\geq 1-\varepsilon'$), then w.p. $\geq 1-\varepsilon/3$ we have that G' is a proof complexity generator secure against $\mathcal{P}$. Hence, a generator $G' \leftarrow \mathcal{G}_{c\ell(n)}$ is a proof complexity generator secure against $\mathcal{P}$ w.p. $\geq 1-\varepsilon$.

(2) $\implies$ (1) is trivial, as every proof complexity generator against $\mathcal{P}$ is a demi-bits generator secure against $\mathcal{P}$. □

5 Hard Partial Truth Tables is Hard

Let $\mathscr{C}$ be a class of Boolean functions; the reader is encouraged to think of $\mathscr{C}$ as a circuit class with a certain resource bound, such as the class of DNFs of $n^{\log n}$ size. In this section, we first apply the Random Projection Disperser Family to establish the hardness of $\mathscr{C}$-Partial-Hard from the nondeterministic hardness of *PAC-learning* $\mathscr{C}$. Then we apply a reduction in [DS16] to connect the hardness of PAC-learning DNFs to the Random k-SAT Hypothesis Against AM.

Recall that for input-output pairs $\{(x_i, b_i)\}_{i\in[L]}$, we use the notation $\{x_i \mapsto b_i\}_{i\in[L]}$ to denote the partial function with domain $\{x_i : i \in [L]\}$ that maps each x_i to b_i. Without loss of generality, we assume that no two inputs x_i, x_j are the same, as otherwise the input becomes trivial for both PAC-learning and Partial-Hard. We also denote $\vec{x} = (x_1, \ldots, x_L)$ and $\vec{b} = (b_1, \ldots, b_L)$ for convenience of notation.

We now define nondeterministic algorithms for PAC-learning $\mathscr{C}$. The definition actually refers to a notion of *random right-hand-side (RRHS) refutation* of the dual class of $\mathscr{C}$ [Vad17]. With respect to randomized algorithms, it is known that PAC-learning $\mathscr{C}$ is equivalent to RRHS refutation of the dual class of $\mathscr{C}$ [Vad17]. To simplify our terminology, we will also refer to RRHS refutation algorithms for the dual class of $\mathscr{C}$ as "PAC-learning algorithms for $\mathscr{C}$" in the nondeterministic setting.

Definition 5.1 (Nondeterministic PAC-Learning)**.** Let $\mathscr{C}$ be a class of Boolean functions, $L \geq 1$, $p \in (0,1)$, and $\mathcal{D}$ be a distribution over $\{0,1\}^n$. A nondeterministic algorithm $\mathcal{A}$ is said to *PAC-learn* $\mathscr{C}$ over input distribution $\mathcal{D}$ using L samples with success probability p if:

- *(Soundness)* for all inputs $x_1, \ldots, x_L \in \{0,1\}^n$ and labels $b_1, \ldots, b_L \in \{0,1\}$, if there exists a function $C \in \mathscr{C}$ such that $C(x_i) = b_i$ for every $i \in [L]$, then $\mathcal{A}((x_1, b_1), \ldots, (x_L, b_L))$ rejects.
- *(Completeness)* If we sample i.i.d. $x_i \leftarrow \mathcal{D}$ and $b_i \leftarrow \{0,1\}$, then

$$\Pr[\mathcal{A}((x_1, b_1), \ldots, (x_L, b_L)) \text{ accepts}] \geq p.$$

We also say that a nondeterministic algorithm $\mathcal{A}$ *distribution-freely* PAC-learns $\mathscr{C}$ with success probability p if it also satisfies the following stronger (Completeness) condition:

- *(Completeness')* For every $x_1, \ldots, x_L \in \{0,1\}^n$, if we choose i.i.d. labels $b_1, \ldots, b_L \leftarrow \{0,1\}$, then

$$\Pr[\mathcal{A}((x_1, b_1), \ldots, (x_L, b_L)) \text{ accepts}] \geq p.$$

We also recall the definition of errorless heuristics for Partial-Hard:

Definition 5.2 (Errorless Heuristics for Partial-Hard)**.** Let $\mathscr{C}$ be a class of functions, $L \geq 1$, $p \in (0,1)$, and $\mathcal{D}$ be a distribution over $\{0,1\}^n$. A SearchNP algorithm $\mathcal{A}$ is an *errorless heuristic* for $\mathscr{C}$-Partial-Hard over $\mathcal{D}$ on domain size L with success probability p if:

- *(Soundness)* For every $x_1, \ldots, x_L \in \{0,1\}^n$, any accepting path of $\mathcal{A}(\vec{x})$ prints a sequence of bits $b_1, \ldots, b_L \in \{0,1\}$ such that the partial function $\{x_i \mapsto b_i\}$ disagrees with every function in $\mathscr{C}$.

- *(Completeness)* Let $x_1, \ldots, x_L \leftarrow \mathcal{D}$ be i.i.d. samples, then with probability at least p, $\mathcal{A}(\vec{x})$ has at least one accepting path.

We remark that the (Soundness) condition needs to hold for every $x_1, \ldots, x_L \in \{0,1\}^n$, hence the term "*errorless* heuristic". If for *every* $x_1, \ldots, x_L \in \{0,1\}^n$, $\mathcal{A}(\vec{x})$ has at least one accepting path, then we say $\mathcal{A}$ is a *(worst-case) algorithm* for $\mathscr{C}$-Partial-Hard.

Our main theorem in this section is that hardness of nondeterministic PAC-learning implies hardness of Partial-Hard:

Theorem 5.3. *Let $\mathscr{C}$ be a class of functions and $L, L' \geq \omega(1)$ be parameters such that $L' \leq 0.1L$. Then:*

- *Suppose there is a (worst-case)* SearchNP *algorithm $\mathcal{A}$ for $\mathscr{C}$-Partial-Hard on domain size L'. Then there is a nondeterministic polynomial-time algorithm $\mathcal{B}$ that distribution-freely PAC-learns $\mathscr{C}$ using L samples with success probability $1 - o(1)$.*
- *Let $\mathcal{D}$ be a distribution over $\{0,1\}^n$ and $\varepsilon > 0$. Suppose there is an errorless* SearchNP *heuristic $\mathcal{A}$ for $\mathscr{C}$-Partial-Hard over $\mathcal{D}$ on domain size L' with success probability $1 - \varepsilon$. Then there is a nondeterministic polynomial-time algorithm $\mathcal{B}$ that PAC-learns $\mathscr{C}$ on input distribution $\mathcal{D}$ using L samples with success probability $1 - 2^{-0.1(L/L')} - 2\varepsilon$.*

Proof. Let $\mathcal{A}$ be an errorless SearchNP heuristic (or worst-case algorithm) for $\mathscr{C}$-Partial-Hard on domain size L'. Our nondeterministic algorithm $\mathcal{B}$, given inputs $(x_1, b_1), \ldots, (x_L, b_L)$, guesses a subset $I \subseteq [L]$ of size $|I| = L'$, and verifies that there is a nondeterministic branch of $\mathcal{A}(\{x_i\}_{i \in I})$ that outputs exactly $\{b_i\}_{i \in I}$. If this is true, then $\mathcal{B}$ accepts the input, otherwise $\mathcal{B}$ rejects the input.

The soundness of $\mathcal{B}$ is easy to see: Consider any inputs $x_1, \ldots, x_L \in \{0,1\}^n$ and labels $b_1, \ldots, b_L \in \{0,1\}$. Suppose there is a function $C \in \mathscr{C}$ such that $C(x_i) = b_i$ for every $i \in [L]$. Then for every subset $I \subseteq [L]$, the partial function $\{x_i \mapsto b_i\}_{i \in I}$ agrees with the function $C \in \mathscr{C}$, and the soundness of $\mathcal{A}$ implies that $\mathcal{B}$ will never accept $\{(x_i, b_i)\}_{i \in [L]}$.

Now we argue the completeness of $\mathcal{B}$. Let $x_1, \ldots, x_L \in \{0,1\}^n$ and define

$$\textsc{Rej} := \textsc{Rej}(x_1, \ldots, x_L) := \{\vec{b} \in \{0,1\}^L : \mathcal{B}(\{(x_i, b_i)\}) \text{ rejects}\}.$$

Suppose $I \subseteq [L]$, $|I| = L'$ is a subset shattered by Rej. We argue that $\mathcal{A}(\{x_i\}_{i \in I})$ does not have any accepting branch. Indeed, if $\mathcal{A}(\{x_i\}_{i \in I})$ outputs any sequence of bits $\{b_i\}_{i \in I}$, letting $\vec{b'} \in \textsc{Rej}$ be the element with $b'_i = b_i$ for every $i \in I$, then $\mathcal{B}(\{(x_i, b_i)\})$ accepts, contradicting the definition of Rej.

- Suppose that $\mathcal{A}$ is a worst-case SearchNP algorithm for $\mathscr{C}$-Partial-Hard on domain size L'. Then for every $x_1, \ldots, x_L$, there is no subset $I \subseteq [L]$ of size L' that is shattered by $\textsc{Rej}(x_1, \ldots, x_L)$. By the Sauer–Shelah Lemma, $|\textsc{Rej}(x_1, \ldots, x_L)| \leq \sum_{i=0}^{L'-1} \binom{L}{i} \leq o(2^L)$. It follows that $\mathcal{B}$ distributional-freely PAC-learns $\mathscr{C}$ using L samples with success probability $1 - o(1)$.
- Suppose that $\mathcal{A}$ is an errorless SearchNP heuristic for $\mathscr{C}$-Partial-Hard over $\mathcal{D}$ on domain size L' with completeness $1 - \varepsilon$. Set $\Delta := 0.1(L/L')$ and $\delta := 2\varepsilon$. Assume, towards a contradiction, that w.p. at least δ over $x_1, \ldots, x_L \leftarrow \mathcal{D}$ we have $|\textsc{Rej}(x_1, \ldots, x_L)| \geq 2^{L-\Delta}$. By Theorem 3.3, $\mathrm{Proj}_{L \to L'}$ is an $(L - \Delta, 4\Delta\frac{L'}{L})$-zero error disperser family, which means that for such $(x_1, \ldots, x_L)$, a random subset $I \subseteq [L]$ with size L' shatters $\textsc{Rej}(x_1, \ldots, x_L)$ w.p. $\geq 1 - 4\Delta\frac{L'}{L}$. Since the distribution of $\{x_i\}_{i \in I}$ (where the randomness is over samples $\{x_i\}_{i \in [L]}$ and subset I) is exactly L' i.i.d. samples from $\mathcal{D}$, we have that $\mathcal{A}(\{x_i\}_{i \in [L']})$ fails to produce an answer with probability at least $\delta(1 - 4\Delta\frac{L'}{L}) \geq \varepsilon$, contradicting the completeness of $\mathcal{A}$. Hence, $\mathcal{B}$ PAC-learns $\mathscr{C}$ over input distribution $\mathcal{D}$ using L samples with success probability

$$\geq 1 - 2^{-\Delta} - \delta \geq 1 - 2^{-0.1(L/L')} - 2\varepsilon. \qquad \square$$

5.1 Hardness of DNF-Partial-Hard

The nondeterministic hardness of Partial-Hard for a circuit class $\mathscr{C}$ follows from the nondeterministic hardness of PAC-learning $\mathscr{C}$. When $\mathscr{C}$ is the class of DNFs, hardness of PAC-learning $\mathscr{C}$ follows from the hardness of random k-SAT [DS16]; we show here that the nondeterministic hardness of PAC-learning DNFs follows from the nondeterministic hardness of random k-SAT as well. Since the reduction in [DS16] uses randomness, here we need a version of the random k-SAT hypothesis against AM refuting algorithms:

Definition 5.4. Let $k \in \mathbb{N}$, $m(n) \geq n$, and $\varepsilon \in (0, 1/2)$. An AM algorithm $\mathcal{A}$ is said to *refute random k-SAT* with $m(n)$ clauses with error ε, if:

- *(Soundness)* for every k-CNF formula C with $m(n)$ clauses that is satisfiable, $\Pr[\mathcal{A}(C) \text{ accepts}] \leq \varepsilon$, where the probability is over the internal (Arthur's) randomness of $\mathcal{A}$.
- *(Completeness)* let C be a random k-CNF formula with $m(n)$ clauses, then $\Pr[\mathcal{A}(C) \text{ accepts}] \geq 1 - \varepsilon$, where the probability is over the choice of C and the internal (Arthur's) randomness of $\mathcal{A}$.

Feige [Fei02] conjectured that random 3-SAT with $O(n)$ clauses is hard to refute (against deterministic polynomial-time algorithms). O'Donnell made the same conjecture against nondeterministic polynomial-time algorithms (cf. [BGSV16; HS17]); this nondeterministic version has been useful in the context of meta-complexity [HS17; Hir18] and proof complexity [PS19] as well.

The main result in [DS16] requires hardness of random k-SAT in a somewhat different regime: for every constant $c \geq 1$, there exists a constant $k \geq 3$ such that random k-SAT with n^c clauses is hard. We make the following assumption asserting that this is still true for AM algorithms:

Assumption 5.5 (Random k-SAT Hypothesis Against AM)**.** For every constant $c \geq 1$, there exists a constant $k \geq 3$ such that no polynomial-time AM algorithm refutes random k-SAT with n^c clauses with error 0.01.

We remark that hardness of random k-SAT against AM algorithms is implied by hardness of random k-SAT against *non-uniform* nondeterministic algorithms ($\mathsf{NP}/_{\mathrm{poly}}$):

Fact 5.6. *If there is an* AM *algorithm that refutes random k-SAT with $m(n)$ clauses with error ε, then there is an $\mathsf{NP}/_{\mathrm{poly}}$ algorithm that refutes random k-SAT with $m(n)$ clauses with error $O(\varepsilon)$.*

The proof basically follows from the argument showing that $\mathsf{AM} \subseteq \mathsf{NP}/_{\mathrm{poly}}$; however, as the (Completeness) item in Definition 5.4 is only average-case, some details are different, hence we provide a proof in Section A.1. By Fact 5.6, if one believes the nondeterministic hardness of random k-SAT even against $\mathsf{NP}/_{\mathrm{poly}}$ algorithms, then Assumption 5.5 should also be treated as a "plausible" assumption. In any case, refuting Assumption 5.5 would be a breakthrough in SAT algorithms.

We also need the following reduction:

Theorem 5.7 ([DS16])**.** *Let $n, k, m \in \mathbb{N}$, $\delta > 0$, and let $t := O(2^k \log(m/\delta))$. Then the random k-SAT problem with n inputs and m clauses reduces to PAC-learning CNFs of input length $n' := 2nt$, size $s := O(nt)$, and top fan-in t, using $L := \lfloor m/t \rfloor$ samples.*

More precisely, there exists a polynomial-time randomized reduction $\mathcal{R}$ that takes a k-SAT instance φ as input and outputs a partial function $\{(x_i \mapsto b_i)\}_{i \in [L]}$, such that the following holds:

- **$\mathcal{R}$ maps satisfiable instances to easy partial functions.**

  For every φ that is satisfiable, with probability at least $1 - \delta$ over the internal randomness of $\mathcal{R}$, the partial function $\{(x_i \mapsto b_i)\} \leftarrow \mathcal{R}(\varphi)$ can be computed by CNFs of size s and top fanin t.

- **$\mathcal{R}$ maps random instances to random partial functions.**

  The following holds for some input distribution $\mathcal{D}$ over $\{0,1\}^{n'}$. Let φ be a random k-SAT instance and let $\{(x_i \mapsto b_i)\} \leftarrow \mathcal{R}(\varphi)$, then each (x_i, b_i) is i.i.d. distributed according to $\mathcal{D} \times \{0,1\}$.

For completeness, we provide a proof of Theorem 5.7 in Section A.2, verifying that the parameters indeed follow from arguments in [DS16].

We also note that by De Morgan's law, learning CNFs and learning DNFs reduce to each other with no change in parameters (size, top fanin, sample complexity, input length).

Theorem 5.8. *Let $\mathscr{C}$ denote the class of DNFs over $n' := 2nt$ inputs with size $s := O(nt)$ and top fanin $t := O(2^k \log(m/\varepsilon))$, and $\mathcal{D}$ be the distribution in Theorem 5.7. Suppose there is an* NP *algorithm that PAC-learns $\mathscr{C}$ over $\mathcal{D}$ using $L := \lfloor m/t \rfloor$ samples with success probability $\geq 1 - \varepsilon$. Then there is an* AM *algorithm for refuting random k-SAT instances with $m(n)$ clauses with error ε.*

Proof. Let $\mathcal{A}$ be an NP algorithm for PAC-learning DNF with success probability $\geq 1 - \varepsilon$. Let $\mathcal{R}$ be the randomized reduction in Theorem 5.7 with error probability $\delta := \varepsilon$. Given an input k-SAT instance, we accept if and only if $\mathcal{A}(\mathcal{R}(\varphi))$ accepts. It is easy to see that this is an AM algorithm.

If φ is satisfiable, then with probability $\geq 1 - \varepsilon$, $\mathcal{R}(\varphi)$ can be computed by CNFs of size s and top fanin t. Therefore, our algorithm accepts with probability $\leq \varepsilon$.

If φ is random, then $\mathcal{R}(\varphi)$ consists of L i.i.d. samples from $\mathcal{D} \times \{0,1\}$, where $\mathcal{D}$ is the distribution in Theorem 5.7. It follows that our algorithm accepts with probability $\geq 1 - \varepsilon$. □

Combining Theorem 5.8 with Theorem 5.3, we have:

Corollary 5.9. *Let $\mathscr{C}$ denote the class of DNFs over $n' := 2nt$ inputs with size $s := O(nt)$ and top fanin $t := O(2^k \log(m/\varepsilon))$, and $\mathcal{D}$ be the distribution in Theorem 5.7. Suppose there is an errorless* SearchNP *heuristic for $\mathscr{C}$-*Partial-Hard *over $\mathcal{D}$ on domain size $L' := \frac{m}{20t \log(1/\varepsilon)}$ with success probability $1 - \varepsilon/4$. Then there is an* AM *algorithm for refuting random k-SAT instances with $m(n)$ clauses with error ε.*

Let $\mathsf{DNF}_{t(n)}$ denote the class of DNFs over n inputs with linear size and top fanin $t(n)$. We show that $\mathsf{DNF}_{t(n)}$-Partial-Hard is hard for suitable choices of parameters $t(n)$. In fact, DNF-Partial-Hard does not even admit errorless SearchNP heuristics over the distribution $\mathcal{D}$ in Theorem 5.7:

Corollary 5.10 (DNF-Partial-Hard Is Hard)**.** *The Random k-SAT Hypothesis Against* AM *implies that, for every integer $c \geq 1$, there is an integer $q \geq 1$ such that there is no errorless* SearchNP *heuristic for $\mathsf{DNF}_{q \log n}$-*Partial-Hard *over some distribution $\mathcal{D}$ on domain size n^c with success probability* 0.999*.*

6 The Remote Point Problem

Under suitable demi-hardness assumptions, our techniques directly imply the hardness of the Remote Point problem for GF(2)-linear circuits and the hardness of Average-Case Hard Partial Truth Tables for extremely simple circuit classes such as conjunctions and narrow parities.

We begin with a standard notion of neighborhood in Hamming space.

Definition 6.1 (Hamming Ball Around a Set)**.** For a set $S \subseteq \{0,1\}^m$ and a remoteness parameter $\tau \in [0,1]$, we define the *τ-ball* around S as

$$\mathcal{B}(S, \tau) := \{z \in \{0,1\}^m : \delta(z, S) \leq \tau\},$$

where $\delta(z, S) := \min_{x \in S} \delta(z, x)$ denotes the (normalized/relative) Hamming distance from z to the nearest point in S.

Using this, we define a generalization of demi-bits generators that requires the adversary to certify remoteness from the range, rather than merely being a non-output.

Definition 6.2 (Remote-Point Demi-Bits Generators)**.** Let n, m be length parameters such that $n < m$. Let τ be a remoteness parameter. A function $G : \{0,1\}^n \to \{0,1\}^m$ is an (s, ε, τ)-*secure remote-point demi-bits generator* if there is no $\mathsf{NP}/\mathrm{poly}$ adversary Adv of size s such that

$$\Pr_{y \leftarrow \{0,1\}^m}[\mathsf{Adv}(y) = 1] \geq \varepsilon \quad \text{and} \quad \Pr_{z \leftarrow \mathcal{B}(\mathrm{Range}(G), \tau)}[\mathsf{Adv}(z) = 1] = 0.$$

In this paper, we consider security against all polynomial-size $\mathsf{NP}/\mathrm{poly}$ adversaries, so we simply write (ε, τ)-secure remote-point demi-bits generator. Equivalently, no efficient proof system can prove the (suitably encoded) statement "$\delta(y, \mathrm{Range}(G)) > \tau$" for at least $\varepsilon \cdot 2^m$ many strings y.

Our main construction also requires the disperser family to satisfy a Lipschitz property, ensuring that nearby inputs map to nearby outputs.

Definition 6.3 ((k, ε, α)-Lipschitz Zero-Error Disperser Family)**.** Let $\mathcal{F}$ be a distribution over functions $f : \{0,1\}^n \to \{0,1\}^m$, $k \geq m$, $\varepsilon > 0$, and $\alpha > 0$ be parameters. We say that $\mathcal{F}$ is a (k, ε, α)-*Lipschitz zero-error disperser family* if:

1. For every random variable $\mathcal{X}$ over $\{0,1\}^n$ with $H_\infty(\mathcal{X}) \geq k$,

$$\Pr_{f \leftarrow \mathcal{F}}[\mathrm{Supp}(f(\mathcal{X})) = \{0,1\}^m] \geq 1 - \varepsilon.$$

2. For every $f \in \mathcal{F}$ and every $x_1, x_2 \in \{0,1\}^n$,

$$\delta(f(x_1), f(x_2)) \;\leq\; \alpha \cdot \delta(x_1, x_2).$$

We note that the Random Projection Disperser Family $\mathrm{Proj}_{n \to m}$ is $(n - \Delta, 4m\Delta/n, n/m)$-Lipschitz: If $\delta(x_1, x_2) = \varepsilon$, then x_1 and x_2 differs in only $\varepsilon \cdot n$ many coordinates, which means that for any $f \in \mathrm{Proj}_{n \to m}$, $f(x_1)$ and $f(x_2)$ differs in only $\varepsilon \cdot n$ many coordinates as well, therefore $\delta(f(x_1), f(x_2)) < \varepsilon \cdot \frac{n}{m}$.

We show that composing a remote-point demi-bits generator with a Lipschitz zero-error disperser family yields hard instances for the Remote Point problem.

Theorem 6.4. *Let $G : \{0,1\}^n \to \{0,1\}^N$ be a $(1 - 2^{k-N}, \tau/\alpha)$-secure remote-point demi-bits generator, and $\mathcal{F} : \{0,1\}^N \to \{0,1\}^m$ be an efficient (k, ε, α)-Lipschitz zero-error disperser family. For any function $f \in \mathcal{F}$ define the function $C_f : \{0,1\}^n \to \{0,1\}^m$ such that $\forall x \in \{0,1\}^n, C_f(x) = f(G(x))$. Let $\mathscr{C}$ be a circuit class where $C_f \in \mathscr{C}$ for all $f \in \mathcal{F}$. Then for any errorless* $\mathsf{SearchNP}$ *heuristic $\mathcal{A}$ for $\mathscr{C}$-*Remote-Point$[n, m, \tau]$*, it holds that $\Pr_{f \leftarrow \mathcal{F}}[\mathcal{A}(C_f) \neq \perp] \leq \varepsilon$.*

Proof. Suppose for contradiction that some nondeterministic polynomial-time algorithm $\mathcal{A}$ satisfies $\Pr_{f \leftarrow \mathcal{F}}[\mathcal{A}(C_f) \neq \perp] > \varepsilon$. We construct a nondeterministic polynomial-time adversary $\mathcal{B}$ breaking the remote-point demi-bits generator G. On input $y \in \{0,1\}^N$, the adversary $\mathcal{B}$ accepts if and only if there exists $f \in \mathcal{F}$ such that some nondeterministic branch of $\mathcal{A}(C_f)$ outputs $f(y)$.

$\mathcal{B}$ rejects every y such that $\delta(y, \mathrm{Range}(G)) \leq \tau/\alpha$. If there exists some seed $s \in \{0,1\}^n$ such that $\delta(y, G(s)) \leq \tau/\alpha$, then for every $f \in \mathcal{F}$ we have $C_f(s) = f(G(s))$ and $\delta(C_f(s), f(y)) \leq \tau$, hence $\delta(f(y), \mathrm{Range}(C_f)) \leq \tau$. Since $\mathcal{A}$ is an errorless solver for $\mathscr{C}$-Remote-Point, no accepting branch of $\mathcal{A}(C_f)$ outputs an element τ-close to $\mathrm{Range}(C_f)$, so $\mathcal{B}(y)$ rejects.

$\mathcal{B}$ accepts at least $2^N - 2^k$ strings in $\{0,1\}^N$. Assume to the contrary that $\mathcal{B}$ accepts $< 2^N - 2^k$ strings of $\{0,1\}^N$. Let X be the uniform distribution over the strings in $\{0,1\}^N$ rejected by $\mathcal{B}$. Then we have $|\mathrm{Supp}(X)| \geq 2^k$ and $H_\infty(X) \geq k$. Since $\mathcal{F}$ is an efficient (k, ε, α)-Lipschitz zero-error disperser family, we have

$$\Pr_{f \leftarrow \mathcal{F}}\left[\mathrm{Supp}(f(X)) = \{0,1\}^m\right] \geq 1 - \varepsilon.$$

Since $\Pr_f[\mathcal{A}(C_f) \neq \perp] > \varepsilon$, there must exist some $f \in \mathcal{F}$ such that $\mathcal{A}(C_f) \neq \perp$ and $\mathrm{Supp}(f(X)) = \{0,1\}^m$. This means that there exists some $x \in \mathrm{Supp}(X)$ such that some non-deterministic accepting branch of $\mathcal{A}(C_f)$ outputs exactly $f(x)$, hence $\mathcal{B}$ accepts x, a contradiction.

Hence $\mathcal{B}$ is a nondeterministic polynomial-time adversary that rejects all strings τ/α-close to $\mathrm{Range}(G)$ and accepts $\geq 1 - 2^{k-N}$ fraction of $\{0,1\}^N$, contradicting the demi-bits security of G. □

6.1 Hardness of XOR-Remote-Point

Theorem 6.4 allows us to base the hardness of XOR-Remote-Point on (demi-hardness variants of) the Learning Parity with Noise (LPN) assumption. LPN is a standard assumption in cryptography dating back to [BFKL93]: Roughly speaking, given a set of m random linear equations over n variables in GF(2) where $m \gg n$, the LPN assumption states that it is infeasible to distinguish between the case that the equations are randomly generated and that there exists a solution satisfying a $(1-\mu)$ fraction of equations simultaneously. We also consider a sparse variant of LPN formulated in [Fei02; Ale11; ABW10], where each equation only mentions $k = O(1)$ many variables.

The LPN-style demi-bits assumption can be viewed as a nondeterministic analogue of the standard LPN assumption, with the mild relaxation that the noise vector is required to be sparse (bounded Hamming weight) rather than i.i.d. Bernoulli.

Assumption 6.5 (LPN-Style Demi-Bits [BFKL93; CL24; RWZ26])**.** For some (public) matrix $A \in \mathbb{F}_2^{m \times n}$, there is no polynomial-size non-uniform nondeterministic circuit $\mathcal{B} : \{0,1\}^m \to \{0,1\}$ such that $\mathcal{B}$ accepts a constant fraction of random strings but rejects every string of the form $A\vec{s} + \vec{e}$, where $\vec{e} \in \mathbb{F}_2^m$ is $(\mu \cdot m)$-sparse and $\vec{s} \in \mathbb{F}_2^n$.

It is easy to see that LPN-Style Demi-Bits Assumption implies a remote-point demi-bits generator computable by GF(2)-linear circuits (circuits with only XOR gates).

Analogously, we define the *Sparse LPN-Style Demi-Bits* assumption by further restricting each row of A to be k-sparse, mirroring the relationship between standard LPN and Sparse LPN [Ale11]. This can be viewed as a nondeterministic analogue of the Sparse LPN assumption.

Assumption 6.6 (Sparse LPN-Style Demi-Bits [Ale11])**.** For some (public) matrix $A \in \mathbb{F}_2^{m \times n}$ where each row of A has Hamming weight $\leq k$, there is no polynomial-size non-uniform nondeterministic circuit $\mathcal{B} : \{0,1\}^m \to \{0,1\}$ such that $\mathcal{B}$ accepts a constant fraction of random strings but rejects every string of the form $A\vec{s} + \vec{e}$, where $\vec{e} \in \mathbb{F}_2^m$ is $(\mu \cdot m)$-sparse and $\vec{s} \in \mathbb{F}_2^n$.

Similarly, the Sparse LPN-Style Demi-Bits Assumption implies a GF(2)-linear circuit computable in NC^0_k that is a remote-point demi-bits generator.

Thanks to LPN-Style Demi-Bits Assumption, and the Random Projection Disperser Family, we are able to show the hardness of XOR-Remote-Point based on Theorem 6.4.

Theorem 6.7 (Hardness of XOR-Remote-Point)**.** *Let $c \geq 10$, $\Delta \geq 1$, $\tau > 0$ be constants and let $N > cn$. If there exists a $(1 - 2^{-\Delta}, \tau/c)$-secure remote-point demi-bits generator $G : \{0,1\}^n \to \{0,1\}^N$ computable using only XOR gates, then there is a distribution $\mathcal{D}$ of GF(2)-linear circuits $C : \{0,1\}^n \to \{0,1\}^{N/c}$ (efficiently samplable given G) such that no errorless SearchNP heuristic for XOR-Remote-Point$[n, N/c, \tau]$ has success probability $> 4\Delta/c$ on $\mathcal{D}$.*

Proof. This follows from instantiating Theorem 6.4 with the remote-point demi-bits generator G and the Random Projection Disperser Family $\mathrm{Proj}_{N\to(N/c)}$ which is a $(N-\Delta, 4\Delta/c, c)$-Lipschitz zero-error disperser family. □

Corollary 6.8. *Fix constants $c \geq 10$, parameter $\tau > 0$, and let $N > cn$. If Assumption 6.5 holds with $\mu = \tau/c$ and $m = N$, then there is no errorless* SearchNP *heuristic for* XOR-Remote-Point$[n, N/c, \tau]$ *with success probability ε over the distribution $\mathcal{D}$ from Theorem 6.7.*

Based on Theorem 6.7, we comment on the optimality of the state-of-the-art algorithm for XOR-Remote-Point.

Remark 6.9 (Is the [APY09] Algorithm Optimal?). Given a GF(2)-linear circuit $C : \{0,1\}^n \to \{0,1\}^m$ with $m \geq 2n$, the XOR-Remote-Point algorithm in [APY09] finds a string $y \in \{0,1\}^m$ that is τ-far from $\mathrm{Range}(C)$, where $\tau = \Omega((\log n)/n)$. On the other hand, LPN with noise rate $\mu = O((\log n)/n)$ and $m \geq n/(1-\mu)$ is solvable in polynomial time via Gaussian elimination [Pra62]. LPN with noise rate $\mu = O((\log^2 n)/n)$ or even $\mu = O((\log^{1+\varepsilon} n)/n)$ are not known to be polynomial-time solvable [BLSV18; AMR25; BCLV26]. However, LPN with noise rate $\mu = O((\log^2 n)/n)$ is in $\mathsf{BPP}^{\mathsf{SZK}} \subseteq \mathsf{AM} \cap \mathsf{coAM}$ [BLVW19], hence it is likely insecure against nondeterministic adversaries.

It still seems possible that LPN with noise rate $\mu = \mathrm{polylog}(n)/n$ is secure against nondeterministic adversaries. If this is true, it would imply that the distance parameter in the [APY09] algorithm for XOR-Remote-Point cannot be improved to $\log^{\omega(1)} n$.

The same framework extends to k-XOR-Remote-Point by replacing the LPN-style demi-bits assumption with its sparse variant (Assumption 6.6).

Corollary 6.10. *Fix constants $c \geq 10$, $k \geq 3$, parameter $\tau > 0$, and let $N > cn$. If Assumption 6.6 holds with $\mu = \tau/c$ and $m = N$, then there is no errorless* SearchNP *heuristic for k-*XOR-Remote-Point$[n, N/c, \tau]$ *with success probability ε over the distribution $\mathcal{D}$ from Theorem 6.7.*

Remark 6.11. [GLY26] gives a polynomial-time algorithm for k-XOR-Remote-Point$[n, \tilde{\Theta}(n^{k/2}), \varepsilon]$. This may be essentially optimal: note that nondeterministically strongly refuting k-XOR (in the sense of Definition 2.8 where "strong" means certifying the system is far from satisfiable) is equivalent to breaking k-XOR Remote-Point Demi-Bits against NP. By Corollary 6.10, any improvement in stretch would yield improved nondeterministic algorithms for strongly refuting semi-random k-XOR instances, which in turn imply better such algorithms for refuting semi-random CSPs [Fei02; CCF10]. The current $\tilde{O}(n^{k/2})$ threshold for strong refutation is believed to be tight for polynomial-time algorithms [KMOW17], and no nondeterministic approach is known to improve upon it. (See Appendix B for a table of the literature on refuting random, semi-random, and smoothed CSPs, covering both weak and strong refutation as well as deterministic and nondeterministic algorithms.)

While nondeterministic algorithms that work with polynomially fewer constraints do exist [FKO06; GKM22], they only achieve *weak* refutation (i.e., they only certify the system is *unsatisfiable*, not *far from satisfiable*). These algorithms are based on even-cover witnesses, where each disjoint even cover certifies that at most one constraint must be violated. Strong refutation by this counting argument would require $\Omega(m)$ disjoint even covers, hence covers of constant length, but by Feige's conjecture [Fei08] (resolved in [GKM22]), constant-length even covers already require $m \geq \widetilde{\Omega}(n^{k/2})$ hyperedges — precisely the spectral threshold at which deterministic algorithms suffice.

6.2 Average-Case Hard Partial Truth Tables is Hard

With respect to randomized algorithms, agnostic PAC-learning $\mathscr{C}$ is equivalent to strong RRHS-refutation of the dual class $\mathscr{C}^*$. This extends [Vad17], who established the analogous equivalence in the realizable setting; see Appendix C for a self-contained proof. We note that [KL18] obtained a similar result (in different terminology) in the agnostic setting, but their equivalence is distribution-specific, whereas ours is distribution-free. The proof techniques also differ: we use Yao's next-bit prediction

combined with agnostic boosting, while [KL18] use a hybrid argument combined with Kalai–Kanade boosting [KK09].

To simplify our terminology, we will also refer to strong RRHS refutation algorithms for the dual class of $\mathscr{C}$ as "Agnostic PAC-learning algorithms for $\mathscr{C}$" in the nondeterministic setting.

Definition 6.12 (Nondeterministic τ-Agnostic PAC-Learning)**.** Let $\mathscr{C}$ be a class of Boolean functions, $L \geq 1$, $p \in (0,1)$, and $\mathcal{D}$ be a distribution over $\{0,1\}^n$. A SearchNP algorithm $\mathcal{A}$ is said to *τ-agnostic PAC-learn* $\mathscr{C}$ over input distribution $\mathcal{D}$ using L samples with success probability p if:

- *(Soundness)* for every inputs $x_1, \ldots, x_L \in \{0,1\}^n$ and labels $b_1, \ldots, b_L \in \{0,1\}$, if there exists a function $C \in \mathscr{C}$ such that $C(x_i) = b_i$ for $1 - \tau$ fraction of $i \in [L]$, then $\mathcal{A}((x_1, b_1), \ldots, (x_L, b_L))$ rejects.
- *(Completeness)* If we sample i.i.d. $x_i \leftarrow \mathcal{D}$ and $b_i \leftarrow \{0,1\}$, then

$$\Pr[\mathcal{A}((x_1, b_1), \ldots, (x_L, b_L)) \text{ accepts}] \geq p.$$

Given $q \in (0,1]$ and $\eta \in [0,1)$, we say that $\mathcal{A}$ has (q,η)-typical completeness over $\mathcal{D}$ if

- *$((q,\eta)$-Typical Completeness)*

$$\Pr_{\vec{x} \leftarrow \mathcal{D}^L}\left[\Pr_{\vec{b} \leftarrow \{0,1\}^L}[\mathcal{A}((x_1, b_1), \ldots, (x_L, b_L)) \text{ accepts}] \geq q\right] \geq 1 - \eta.$$

Notice that (q,η)-typical completeness implies ordinary completeness with success probability at least $(1-\eta)q$.

We also say that a SearchNP algorithm *distribution-freely* agnostic PAC-learns $\mathscr{C}$ with success probability p if it also satisfies the following stronger (Completeness) condition:

- *(Completeness')* For every $x_1, \ldots, x_L \in \{0,1\}^n$, if we sample i.i.d. labels $b_1, \ldots, b_L \leftarrow \{0,1\}$, then

$$\Pr[\mathcal{A}((x_1, b_1), \ldots, (x_L, b_L)) \text{ accepts}] \geq p.$$

Before connecting Definition 6.12 with the hardness of Partial-AvgHard, we recall the definition of errorless heuristics for Partial-AvgHard.

Definition 6.13 (Errorless Heuristics for Partial-AvgHard)**.** Let $\mathscr{C}$ be a class of functions, $L \geq 1$, $\tau \in (0,1)$, $p \in (0,1)$, and $\mathcal{D}$ be a distribution over $\{0,1\}^n$. A SearchNP algorithm $\mathcal{A}$ is an *errorless heuristic* for $\mathscr{C}$-Partial-AvgHard with remoteness τ, success probability p, and domain size L over $\mathcal{D}$ if:

- *(Soundness)* For every $x_1, \ldots, x_L \in \{0,1\}^n$, every accepting path of $\mathcal{A}(\vec{x})$ outputs bits $b_1, \ldots, b_L \in \{0,1\}$ such that the partial function $\{x_i \mapsto b_i\}_{i \in [L]}$ has relative distance $\geq \tau$ from every function in $\mathscr{C}$.
- *(Completeness)* When $x_1, \ldots, x_L \leftarrow \mathcal{D}$ are i.i.d. samples, the algorithm $\mathcal{A}(\vec{x})$ has at least one accepting path with probability at least p.

If $\mathcal{A}(\vec{x})$ has at least one accepting path for *every* $x_1, \ldots, x_L \in \{0,1\}^n$, we say $\mathcal{A}$ is a *worst-case algorithm* for $\mathscr{C}$-Partial-AvgHard.

The main result of this section shows that nondeterministic hardness of agnostic PAC-learning yields nondeterministic hardness of Partial-AvgHard:

Theorem 6.14. *Let $\mathscr{C}$ be a class of functions and $L, L' \geq \omega(1)$ be parameters such that $L' \leq 0.1L$. Then:*

- *Suppose there is a (worst-case)* SearchNP *algorithm $\mathcal{A}$ for $\mathscr{C}$-*Partial-AvgHard *with remoteness τ on domain size L'. Then there is a* SearchNP *algorithm $\mathcal{B}$ that distribution-freely $\frac{\tau L'}{2L}$-agnostic PAC-learns $\mathscr{C}$ using L samples with success probability $1 - o(1)$.*
- *Let $\mathcal{D}$ be a distribution over $\{0,1\}^n$ and $\varepsilon \in (0, 3/5)$. Suppose there is an errorless* SearchNP *heuristic $\mathcal{A}$ for $\mathscr{C}$-*Partial-AvgHard *with remoteness τ over $\mathcal{D}$ on domain size L' with success probability $1 - \varepsilon$. Then there is a* SearchNP *algorithm $\mathcal{B}$ that $\frac{\tau L'}{2L}$-agnostic PAC-learns $\mathscr{C}$ on input distribution $\mathcal{D}$ using L samples with success probability at least $(1 - 5\varepsilon/3)(1 - 2^{-0.1(L/L')}) \geq 1 - 2^{-0.1(L/L')} - 2\varepsilon$. Moreover, $\mathcal{B}$ has $(1 - 2^{-0.1(L/L')}, 5\varepsilon/3)$-typical completeness.*

Proof. Let $\mathcal{A}$ be an errorless SearchNP heuristic (or a worst-case algorithm) for $\mathscr{C}$-Partial-AvgHard with remoteness τ on domain size L'. Our SearchNP algorithm $\mathcal{B}$, given inputs $(x_1, b_1), \ldots, (x_L, b_L)$, guesses a subset $I \subseteq [L]$ of size $|I| = L'$, and verifies that there is a nondeterministic branch of $\mathcal{A}(\{x_i\}_{i \in I})$ that outputs a sequence $\{b'_i\}_{i \in I}$ that is $(\tau/2)$-close to $\{b_i\}_{i \in I}$. If this is true, then $\mathcal{B}$ accepts the input, otherwise $\mathcal{B}$ rejects the input.

Soundness of $\mathcal{B}$: Consider any inputs $x_1, \ldots, x_L \in \{0,1\}^n$ and labels $b_1, \ldots, b_L \in \{0,1\}$. Suppose there is a function $C \in \mathscr{C}$ such that $C(x_i) = b_i$ for $\geq 1 - \frac{\tau L'}{2L}$ fraction of $i \in [L]$. Then for every subset $I \subseteq [L]$ of size L', the partial function $\{(x_i \mapsto b_i)\}_{i \in I}$ agrees with C for $\geq 1 - \tau/2$ fraction of $i \in I$. Now suppose $\mathcal{A}(\{x_i\}_{i \in I})$ outputs bits $\{b'_i\}_{i \in I}$ with $\delta(\vec{b}', \vec{b}|_I) < \tau/2$. By the triangle inequality,

$$\delta(\vec{b}', C|_I) \leq \delta(\vec{b}', \vec{b}|_I) + \delta(\vec{b}|_I, C|_I) < \tau/2 + \tau/2 = \tau,$$

contradicting the soundness of $\mathcal{A}$ (which guarantees $\delta(\vec{b}', C|_I) \geq \tau$). Hence $\mathcal{B}$ never accepts $\{(x_i, b_i)\}_{i \in [L]}$.

Completeness of $\mathcal{B}$: Let $x_1, \ldots, x_L \in \{0,1\}^n$ and define

$$\textsc{Rej} := \textsc{Rej}(x_1, \ldots, x_L) := \{\vec{b} \in \{0,1\}^L : \mathcal{B}(\{(x_i, b_i)\}) \text{ rejects}\}.$$

Suppose $I \subseteq [L]$, $|I| = L'$ is a subset shattered by Rej. We argue that $\mathcal{A}(\{x_i\}_{i \in I})$ does not have any accepting branch. Indeed, if $\mathcal{A}(\{x_i\}_{i \in I})$ outputs any sequence of bits $\{b'_i\}_{i \in I}$, there exists $\vec{b} \in \textsc{Rej}$ such that $b_i = b'_i$ for every $i \in I$. Then $\delta(\vec{b}', \vec{b}|_I) = 0 < \tau/2$, so $\mathcal{B}(\{(x_i, b_i)\})$ accepts, contradicting $\vec{b} \in \textsc{Rej}$.

- Suppose that $\mathcal{A}$ is a worst-case SearchNP algorithm for $\mathscr{C}$-Partial-AvgHard$[n, s, L', \tau]$. Then for every $x_1, \ldots, x_L$, there is no subset $I \subseteq [L]$ of size L' shattered by $\textsc{Rej}(x_1, \ldots, x_L)$. By the Sauer–Shelah Lemma, $|\textsc{Rej}(x_1, \ldots, x_L)| \leq \sum_{i=0}^{L'-1} \binom{L}{i} \leq o(2^L)$. It follows that $\mathcal{B}$ distribution-freely PAC-learns $\mathscr{C}$ using L samples with success probability $1 - o(1)$.
- Suppose that $\mathcal{A}$ is an errorless SearchNP heuristic for $\mathscr{C}$-Partial-AvgHard$[n, s, L', \tau]$ over $\mathcal{D}$ with completeness $1 - \varepsilon$. Set $\Delta := 0.1(L/L')$ and define

$$q := \Pr_{(x_1, \ldots, x_L) \leftarrow \mathcal{D}^L}\left[|\textsc{Rej}| \geq 2^{L - \Delta}\right].$$

Fix any $\vec{x} = (x_1, \ldots, x_L)$ such that $|\textsc{Rej}| \geq 2^{L-\Delta}$. By Theorem 3.3, a random subset $I \subseteq [L]$ of size L' is shattered by Rej with probability at least $1 - 4\Delta\frac{L'}{L} = 0.6$.

Since the marginal distribution of $(x_i)_{i \in I}$ is exactly $\mathcal{D}^{L'}$, we obtain

$$q\left(1 - 4\Delta\frac{L'}{L}\right) \leq \Pr_{\vec{z} \leftarrow \mathcal{D}^{L'}}[\mathcal{A}(\vec{z}) \text{ has no accepting path}] \leq \varepsilon.$$

Hence, $q \le 5\varepsilon/3$. For every $\vec{x}$ such that $|\textsc{Rej}(\vec{x})| < 2^{L-\Delta}$, we have

$$\Pr_{\vec{b}\leftarrow\{0,1\}^L}\left[\mathcal{B}\big((x_i,b_i)_{i\in[L]}\big) \text{ accepts}\right] = 1 - \frac{|\textsc{Rej}(\vec{x})|}{2^L} > 1 - 2^{-\Delta}.$$

Hence, it holds that

$$\Pr_{\vec{x}\leftarrow\mathcal{D}^L}\left[\Pr_{\vec{b}\leftarrow\{0,1\}^L}[\mathcal{B}((x_1,b_1),\ldots,(x_L,b_L)) \text{ accepts}] \ge 1 - 2^{-\Delta}\right] \ge 1 - q \ge 1 - 5\varepsilon/3.$$

Thus $\mathcal{B}$ has $\big(1-2^{-\Delta}, 5\varepsilon/3\big)$-typical completeness. Finally,

$$\Pr_{\substack{\vec{x}\leftarrow\mathcal{D}^L\\ \vec{b}\leftarrow\{0,1\}^L}}[\mathcal{B}((x_1,b_1),\ldots,(x_L,b_L)) \text{ accepts}] \ge (1-q)(1-2^{-\Delta}) \ge 1 - 2^{-0.1(L/L')} - 2\varepsilon.$$

Hence, $\mathcal{B}$ PAC-learns $\mathscr{C}$ over input distribution $\mathcal{D}$ using L samples with success probability

$$1 - 2^{-0.1(L/L')} - 2\varepsilon. \qquad \square$$

6.2.1 Hardness of k-XOR-Partial-AvgHard

To prove the hardness of k-XOR-Partial-AvgHard, we introduce an assumption closely related to Assumption 6.6, that essentially states the hardness of agnostic learning k-XOR.

Assumption 6.15 (Sparse-Secret LPN-Style Demi-Bits for Random Matrices [Ale11])**.** Let $k = k(n)$ satisfy $\omega(1) \le k \le n$, $\mu = \mu(n) \in (0, 1/2)$, $\delta \in (0, 1/2)$, and $m = m(n) = \text{poly}(n)$. Fix a distribution $\mathcal{D}$ (say, the uniform distribution) over $\mathbb{F}_2^n$. We additionally assume

$$m(1 - H(\mu)) \ge \log_2\left(\sum_{j=0}^{k}\binom{n}{j}\right) + 1, \tag{1}$$

where $H(\cdot)$ is the binary entropy function. For a random matrix $A \in \mathbb{F}_2^{m\times n}$ with rows drawn i.i.d. from $\mathcal{D}$, with probability $\ge 1/2$ over A, there is no polynomial-size non-uniform nondeterministic circuit $\mathcal{B} : \{0,1\}^m \to \{0,1\}$ such that $\mathcal{B}$ accepts at least a $1/2 - \delta$ fraction of random strings but rejects every string of the form $A\vec{s} + \vec{e}$, where $\vec{e} \in \mathbb{F}_2^m$ is $(\mu \cdot m)$-sparse and $\vec{s} \in \mathbb{F}_2^n$ is k-sparse.

The condition (1) guarantees that the assumption is not information-theoretically vacuous: the number of strings of the form $A\vec{s} + \vec{e}$ is at most $2^{mH(\mu)} \cdot \left(\sum_{j=0}^{k}\binom{n}{j}\right) \le 2^{m-1}$. If $\mu < 1/2$ is a constant and $k \le n/2$, it suffices that $m = \Omega(k \log(en/k))$. Also note that the assumption is meaningful only in the regime $k = \omega(1)$. If $k = O(1)$, one can always break the assumption in deterministic $O(n^k)$ time.

The two assumptions are not directly comparable: Assumption 6.6 restricts the rows of A to be k-sparse and allows an unrestricted secret, whereas the present assumption allows unrestricted rows but requires the secret to be k-sparse. In addition, the present assumption requires hardness for at least half of the random matrices rather than for one fixed matrix. Concretely, by Theorem 6.14, nondeterministic hardness of Partial-AvgHard for a circuit class $\mathscr{C}$ follows from nondeterministic hardness of agnostically PAC-learning $\mathscr{C}$. When $\mathscr{C}$ is the class of k-XORs, the relevant LPN-style assumption must restrict the secret $\vec{s}$ to be k-sparse, since $a \mapsto \langle a, \vec{s}\rangle$ is a k-XOR function under precisely this restriction. No sparsity condition on the rows of A is required for this implication.

Corollary 6.16. *Assuming the Sparse-Secret LPN Assumption for Random Matrices with distribution $\mathcal{D}$, let $m = L$, and assume that $10 \mid L$. For any constant $\tau \in (20\mu, 1/2)$, there is no errorless* SearchNP *heuristic for k-XOR-Partial-AvgHard$[n, s, L/10, \tau]$ over $\mathcal{D}$ with success probability at least $3/4$.*

Proof. Suppose for contradiction that such a heuristic $\mathcal{A}$ exists. Set $L' := L/10$ and $\varepsilon := 1/4$. Then $\Delta := 0.1L/L' = 1$. By Theorem 6.14, there is a SearchNP algorithm $\mathcal{B}$ that $\tau/20$-agnostically PAC-learns k-XOR using L samples and has $(1/2, 5/12)$-typical completeness. Equivalently,

$$\Pr_{A \leftarrow \mathcal{D}^L}\left[\Pr_{y \leftarrow \{0,1\}^L}\left[\mathcal{B}\big((a_i, y_i)_{i\in[L]}\big) \text{ accepts}\right] \geq \frac{1}{2}\right] \geq \frac{7}{12},$$

where $a_1, \ldots, a_L$ are the rows of A.

For each fixed matrix A, define

$$\mathcal{B}_A(y) := \mathcal{B}\big((a_i, y_i)_{i\in[L]}\big).$$

Since A is fixed, it can be hardwired into $\mathcal{B}$, so $\mathcal{B}_A$ is a polynomial-size non-uniform nondeterministic circuit.

We claim that $\mathcal{B}_A$ rejects every sparse-secret LPN string. Indeed, suppose $y = A\vec{s} + \vec{e}$, $|\vec{s}| \leq k$, and $|\vec{e}| \leq \mu L$. The function $f_{\vec{s}}(a) := \langle a, \vec{s}\rangle$ is a k-XOR, and $\delta(y, (f_{\vec{s}}(a_i))_{i\in[L]}) \leq \mu < \tau/20$. Therefore, the soundness of the agnostic learner implies that $\mathcal{B}_A(y)$ rejects.

Thus, for at least a $7/12$ fraction of $A \leftarrow \mathcal{D}^L$, there is a sound nondeterministic circuit $\mathcal{B}_A$ accepting at least half of all uniformly random right-hand sides.

On the other hand, Sparse-Secret LPN Assumption for Random Matrices states that a set of matrices of probability at least $1/2$ admits no sound circuit accepting even a $1/2 - \delta$ fraction of uniformly random right-hand sides. Since $7/12 + 1/2 > 1$, these two sets of matrices intersect. For a matrix in the intersection, $\mathcal{B}_A$ accepts at least $1/2 \geq 1/2 - \delta$ of all random strings while rejecting every sparse-secret LPN string, contradicting the assumption. □

6.2.2 Hardness from Random k-SAT

Finally, we observe that in suitable parameter regimes, DNF-Partial-Hard reduces to both AND-Partial-AvgHard and XOR-Partial-AvgHard. This is because both conjunctions and parities can weakly approximate DNFs:

Fact 6.17. *Let $F = T_1 \vee T_2 \vee \cdots \vee T_t$ be a DNF with $t \geq 1$ terms, each of width at most w. Then, for every distribution $\mathcal{D}$ over the input space, there exist a conjunction C of width at most w and a possibly complemented parity function P on at most w variables that both have $\Omega(1/t)$ correlation with F over $\mathcal{D}$:*

$$\Pr_{x\leftarrow\mathcal{D}}[C(x) = F(x)] \geq \frac{1}{2} + \frac{1}{O(t)}; \qquad \text{([Sch90, Section 3.5])}$$

$$\text{and } \Pr_{x\leftarrow\mathcal{D}}[P(x) = F(x)] \geq \frac{1}{2} + \frac{1}{O(t)}. \qquad \text{([Jac97, Fact 8])}$$

Here constant functions are allowed.

This allows us to derive hardness of AND-Partial-AvgHard and XOR-Partial-AvgHard directly from Corollary 5.9.

Corollary 6.18. *Under the Random k-SAT Hypothesis Against AM, for every constant $\eta \in (0,1)$ and integer $c \geq 1$, there are an integer $q \geq 1$ and an input distribution $\mathcal{D}$ over $\{0,1\}^n$ such that neither $\mathsf{AND}_{\lceil n^\eta\rceil}$-Partial-AvgHard nor $\mathsf{XOR}_{\lceil n^\eta\rceil}$-Partial-AvgHard has an errorless SearchNP heuristic over $\mathcal{D}$ on domain size n^c, with remoteness $\frac{1}{2} - \frac{1}{q\log n}$ and success probability 0.999.*

Proof. We first claim that every errorless SearchNP heuristic producing labels at relative distance at least $\frac{1}{2} - \frac{1}{Kt}$ from all width-w conjunctions also gives an errorless SearchNP heuristic for Partial-Hard

against size-s DNFs with top fanin at most t and term width at most w, for a sufficiently large absolute constant K. Here conjunctions may use negated literals and constants. Indeed, let $\{x_i \mapsto b_i\}_{i\in[L]}$ be a partial function computable by such a DNF F, and let $\mathcal{D}$ be the uniform distribution on the indexed samples. Then by Fact 6.17, there is a width-w conjunction C such that

$$\Pr_{i\leftarrow[L]}[C(x_i) = b_i] \geq \frac{1}{2} + \frac{1}{Kt}.$$

Hence, any partial function at distance at least $\frac{1}{2} - \frac{1}{Kt}$ from all width-w conjunctions cannot be computed by such a DNF.

Fix η, c, choose integers $a > 1/\eta$ and $d > ac$, and choose k from the hypothesis for density exponent d. Let $t := O(d2^k \log n)$. By Corollary 5.9, to refute random k-SAT instances on n variables with n^d clauses with error 0.01 by an AM algorithm, it suffices to design an errorless $\mathsf{SearchNP}$ heuristic for $\mathscr{C}$-Partial-Hard over $\mathcal{D}$ on domain size

$$L' := \left\lfloor \frac{n^d}{20t \log 100} \right\rfloor$$

with success probability at least 0.9975, where $\mathscr{C}$ is the class of DNFs over $n' := 2nt = O_{k,d}(n \log n)$ inputs with size $s := O(nt)$ and top fanin t, and $\mathcal{D}$ is the distribution in Theorem 5.7.

Set $\widetilde{n} = n^a$ and pad each input x to $(x, \neg x, 1, 0, \ldots, 0) \in \{0,1\}^{\widetilde{n}}$; let $\widetilde{\mathcal{D}}$ be the resulting distribution. For all sufficiently large n, $2n' + 2 \leq \widetilde{n}$, $n' + 1 \leq \lceil \widetilde{n}^\eta \rceil$, $\widetilde{n}^c \leq L'$. Each DNF term has width at most n' after simplification, and the padding realizes negated literals and constants. Choose an integer q large enough that $Kt \leq q \log \widetilde{n}$. If the stated $\mathsf{AND}_{\lceil \widetilde{n}^\eta \rceil}$-Partial-AvgHard heuristic exists, apply it to the padded first $\widetilde{n}^c$ of the L' inputs and extend its output by arbitrary labels. By the first claim, no DNF in $\mathscr{C}$ computes these labels, already on the first $\widetilde{n}^c$ inputs. Its success probability remains $0.999 \geq 0.9975$, giving the required $\mathscr{C}$-Partial-Hard heuristic and contradicting the hypothesis. For other input lengths, define $\widetilde{\mathcal{D}}$ using $n = \lfloor \widetilde{n}^{1/a} \rfloor$; the contradiction uses $\widetilde{n} = n^a$.

The proof for $\mathsf{XOR}_{\lceil \widetilde{n}^\eta \rceil}$-Partial-AvgHard is analogous: use the parity guarantee in Fact 6.17 and the constant-one coordinate to realize complemented parities with at most one additional input. □

However, compared to Corollary 6.16, one weakness of Corollary 6.18 is that it only derives hardness of Partial-AvgHard with an inapproximability parameter very close to 1/2.

AI Disclosure

During the preparation of this work, large language models (Gemini and Claude) were employed to aid in literature review, parameter computation, and the proofs in Section 3. The authors verified the correctness and originality of all content including references, and take full responsibility for the contents in this paper.

Acknowledgments

We thank Yunqi Li for discussions regarding [APY09], Jan Krajíček for pointing us to a proof of Theorem 4.3 using an optimal proof system with one bit of advice, and anonymous reviewers for helpful comments.

References

[ABR16] Benny Applebaum, Andrej Bogdanov, and Alon Rosen. "A dichotomy for local small-bias generators". In: *Journal of Cryptology* 29.3 (2016), pp. 577–596 (cit. on p. 7).

[ABRW04] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. "Pseudorandom generators in propositional proof complexity". In: *SIAM Journal on Computing* 34.1 (2004), pp. 67–88 (cit. on pp. 5, 12).

[ABW10] Benny Applebaum, Boaz Barak, and Avi Wigderson. "Public-key cryptography from different assumptions". In: *STOC*. ACM, 2010, pp. 171–180 (cit. on p. 27).

[AGK21] Jackson Abascal, Venkatesan Guruswami, and Pravesh K. Kothari. "Strongly refuting all semi-random Boolean CSPs". In: *SODA*. SIAM, 2021, pp. 454–472 (cit. on p. 43).

[AIK06] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. "Cryptography in NC^0". In: *SIAM J. Comput.* 36.4 (2006), pp. 845–888 (cit. on p. 1).

[AL18] Benny Applebaum and Shachar Lovett. "Algebraic attacks against random local functions and their countermeasures". In: *SIAM Journal on Computing* 47.1 (2018), pp. 52–79 (cit. on p. 7).

[Ale11] Michael Alekhnovich. "More on average case vs approximation complexity". In: *Comput. Complex.* 20.4 (2011), pp. 755–786 (cit. on pp. 27, 31).

[AMR25] Damiano Abram, Giulio Malavolta, and Lawrence Roy. "Slightly sublinear trapdoor hash functions and PIR from low-noise LPN". In: *TCC (1)*. Lecture Notes in Computer Science. Springer, 2025, pp. 280–300 (cit. on p. 28).

[AOW15] Sarah R. Allen, Ryan O'Donnell, and David Witmer. "How to refute a random CSP". In: *FOCS*. IEEE, 2015, pp. 689–708 (cit. on pp. 42, 43).

[APY09] Noga Alon, Rina Panigrahy, and Sergey Yekhanin. "Deterministic approximation algorithms for the nearest codeword problem". In: *APPROX-RANDOM*. Vol. 5687. Lecture Notes in Computer Science. Springer, 2009, pp. 339–351 (cit. on pp. 2, 3, 8, 28, 33).

[AS10] Vikraman Arvind and Srikanth Srinivasan. "Circuit lower bounds, help functions, and the remote point problem". In: *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*. 2010, pp. 383–396 (cit. on p. 2).

[Bab85] László Babai. "Trading group theory for randomness". In: *STOC*. ACM, 1985, pp. 421–429 (cit. on p. 13).

[BCLV26] Andrej Bogdanov, Rohit Chatterjee, Yunqi Li, and Prashant Nalini Vasudevan. "Decoding balanced linear codes with preprocessing". In: *ITCS*. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2026, 23:1–23:23 (cit. on p. 28).

[BFKL93] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. "Cryptographic primitives based on hard learning problems". In: *CRYPTO*. Berlin, Heidelberg: Springer-Verlag, 1993, pp. 278–291. ISBN: 3540577661 (cit. on pp. 27, 47, 48).

[BGI+12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. "On the (im)possibility of obfuscating programs". In: *J. ACM* 59.2 (2012), 6:1–6:48 (cit. on p. 3).

[BGSV16] Peter Bürgisser, Oded Goldreich, Madhu Sudan, and Salil Vadhan. "Complexity theory". In: *Oberwolfach Reports* 12.4 (2016), pp. 3049–3099 (cit. on p. 24).

[BLSV18] Zvika Brakerski, Alex Lombardi, Gil Segev, and Vinod Vaikuntanathan. "Anonymous IBE, leakage resilience and circular security from new assumptions". In: *EUROCRYPT (1)*. Lecture Notes in Computer Science. Springer, 2018, pp. 535–564 (cit. on p. 28).

[BLVW19] Zvika Brakerski, Vadim Lyubashevsky, Vinod Vaikuntanathan, and Daniel Wichs. "Worst-case hardness for LPN and cryptographic hashing via code smoothing". In: *EUROCRYPT (3)*. Lecture Notes in Computer Science. Springer, 2019, pp. 619–635 (cit. on p. 28).

[BM16] Boaz Barak and Ankur Moitra. "Noisy tensor completion via the sum-of-squares hierarchy". In: *COLT*. JMLR Workshop and Conference Proceedings. JMLR.org, 2016, pp. 417–445 (cit. on p. 43).

[BQ12] Andrej Bogdanov and Youming Qiao. "On the security of Goldreich's one-way function". In: *computational complexity* 21.1 (2012), pp. 83–127 (cit. on p. 7).

[BR94] Mihir Bellare and John Rompel. "Randomness-efficient oblivious sampling". In: *FOCS*. IEEE Computer Society, 1994, pp. 276–287 (cit. on p. 16).

[CCF10] Amin Coja-Oghlan, Colin Cooper, and Alan Frieze. "An efficient sparse regularity concept". In: *SIAM Journal on Discrete Mathematics* 23.4 (2010), pp. 2000–2034 (cit. on pp. 28, 43).

[CDM+18] Geoffroy Couteau, Aurélien Dupin, Pierrick Méaux, Mélissa Rossi, and Yann Rotella. "On the concrete security of Goldreich's pseudorandom generator". In: *ASIACRYPT (2)*. Lecture Notes in Computer Science. Springer, 2018, pp. 96–124 (cit. on p. 7).

[CG88] Benny Chor and Oded Goldreich. "Unbiased bits from sources of weak randomness and probabilistic communication complexity". In: *SIAM J. Comput.* 17.2 (1988), pp. 230–261 (cit. on p. 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: *STOC*. ACM, 2023, pp. 1058–1066 (cit. on pp. 1, 2, 8).

[CHO+22] Lijie Chen, Shuichi Hirahara, Igor C. Oliveira, Ján Pich, Ninad Rajgopal, and Rahul Santhanam. "Beyond natural proofs: hardness magnification and locality". In: *J. ACM* 69.4 (2022), 25:1–25:49 (cit. on p. 10).

[Cho11] Timothy Y. Chow. "Almost-natural proofs". In: *J. Comput. Syst. Sci.* 77.4 (2011), pp. 728–737 (cit. on p. 10).

[CHR24] Lijie Chen, Shuichi Hirahara, and Hanlin Ren. "Symmetric exponential time requires near-maximum circuit size". In: *STOC*. ACM, 2024, pp. 1990–1999. ISBN: 9798400703836 (cit. on p. 1).

[CJW19] Lijie Chen, Ce Jin, and R. Ryan Williams. "Hardness magnification for all sparse NP languages". In: *FOCS*. IEEE Computer Society, 2019, pp. 1240–1255 (cit. on p. 10).

[CJW20] Lijie Chen, Ce Jin, and R. Ryan Williams. "Sharp threshold results for computational complexity". In: *STOC*. ACM, 2020, pp. 1335–1348 (cit. on p. 10).

[CK07] Stephen A. Cook and Jan Krajíček. "Consequences of the provability of $\mathsf{NP} \subseteq \mathsf{P}/_{\mathrm{poly}}$". In: *J. Symb. Log.* 72.4 (2007), pp. 1353–1371 (cit. on p. 18).

[CL24] Yilei Chen and Jiatu Li. "Hardness of range avoidance and remote point for restricted circuits via cryptography". In: *STOC*. ACM, 2024, pp. 620–629 (cit. on pp. 3, 5, 13, 27).

[CL69] George F. Clements and Bernt Lindström. "A generalization of a combinatorial theorem of Macaulay". In: *Journal of combinatorial theory* 7.3 (1969), pp. 230–238 (cit. on p. 14).

[CLY22] Lijie Chen, Jiatu Li, and Tianqi Yang. "Extremely efficient constructions of hash functions, with applications to hardness magnification and PRFs". In: *CCC*. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 23:1–23:37 (cit. on p. 10).

[CM01] Mary Cryan and Peter Bro Miltersen. "On pseudorandom generators in NC^0". In: *MFCS*. Berlin, Heidelberg: Springer-Verlag, 2001, pp. 272–284. ISBN: 3540424962 (cit. on p. 7).

[CMMW19] Lijie Chen, Dylan M. McKay, Cody D. Murray, and R. Ryan Williams. "Relations and equivalences between circuit lower bounds and Karp–Lipton theorems". In: *CCC*. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, 30:1–30:21 (cit. on p. 10).

[Coh21] Gil Cohen. "Two-source dispersers for polylogarithmic entropy and improved Ramsey graphs". In: *SIAM Journal on Computing* 50.3 (2021), STOC16-30-STOC16–67 (cit. on p. 9).

[CR79] Stephen A. Cook and Robert A. Reckhow. "The relative efficiency of propositional proof systems". In: *J. Symb. Log.* 44.1 (1979), pp. 36–50 (cit. on pp. 4, 12).

[CWY23] Lijie Chen, R. Ryan Williams, and Tianqi Yang. "Black-box constructive proofs are unavoidable". In: *ITCS*. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, 35:1–35:24 (cit. on p. 10).

[CZ19] Eshan Chattopadhyay and David Zuckerman. "Explicit two-source extractors and resilient functions". In: *Annals of Mathematics* 189.3 (2019), pp. 653–705 (cit. on p. 1).

[DS16] Amit Daniely and Shai Shalev-Shwartz. "Complexity theoretic limitations on learning DNF's". In: *COLT*. Vol. 49. JMLR Workshop and Conference Proceedings. JMLR.org, 2016, pp. 815–830 (cit. on pp. 7, 22, 24, 25, 41).

[Erd59] Paul Erdős. "Graph theory and probability". In: *Canadian Journal of Mathematics* 11 (1959), pp. 34–38 (cit. on p. 1).

[Fei02] Uriel Feige. "Relations between average case complexity and approximation complexity". In: *STOC*. ACM, 2002, pp. 534–543 (cit. on pp. 24, 27, 28).

[Fei08] Uriel Feige. "Small linear dependencies for binary vectors of low weight". In: *Building Bridges: Between Mathematics and Computer Science*. Springer, 2008, pp. 283–307 (cit. on p. 28).

[FGHK16] Magnus Gausdal Find, Alexander Golovnev, Edward A. Hirsch, and Alexander S. Kulikov. "A better-than-3n lower bound for the circuit complexity of an explicit function". In: *FOCS*. IEEE, 2016, pp. 89–98 (cit. on p. 1).

[FKO06] Uriel Feige, Jeong Han Kim, and Eran Ofek. "Witnesses for non-satisfiability of dense random 3CNF formulas". In: *FOCS*. FOCS '06. IEEE, 2006, pp. 497–508. ISBN: 0769527205 (cit. on pp. 28, 43).

[FS11] Lance Fortnow and Rahul Santhanam. "Infeasibility of instance compression and succinct PCPs for NP". In: *J. Comput. Syst. Sci.* 77.1 (2011), pp. 91–106 (cit. on p. 3).

[GGH+16] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. "Candidate indistinguishability obfuscation and functional encryption for all circuits". In: *SIAM J. Comput.* 45.3 (2016), pp. 882–929 (cit. on p. 3).

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. "How to construct random functions". In: *J. ACM* 33.4 (1986), pp. 792–807 (cit. on p. 4).

[GGNS23] Karthik Gajulapalli, Alexander Golovnev, Satyajeet Nagargoje, and Sidhant Saraogi. "Range avoidance for constant depth circuits: hardness and algorithms". In: *APPROX/RANDOM*. Vol. 275. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, 65:1–65:18 (cit. on pp. 1, 2).

[GGRT26] Michal Garlík, Svyatoslav Gryaznov, Hanlin Ren, and Iddo Tzameret. "The weak rank principle: lower bounds and applications". In: *STOC*. ACM, 2026, pp. 138–149 (cit. on p. 5).

[Gil52] E. N. Gilbert. "A comparison of signalling alphabets". In: *The Bell System Technical Journal* 31.3 (1952), pp. 504–522 (cit. on p. 1).

[GKM22] Venkatesan Guruswami, Pravesh K. Kothari, and Peter Manohar. "Algorithms and certificates for Boolean CSP refutation: smoothed is no harder than random". In: *STOC*. ACM, 2022, pp. 678–689. ISBN: 9781450392648 (cit. on pp. 28, 43).

[GLW22] Venkatesan Guruswami, Xin Lyu, and Xiuhan Wang. "Range avoidance for low-depth circuits and connections to pseudorandomness". In: *APPROX/RANDOM*. Vol. 245. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 20:1–20:21 (cit. on p. 1).

[GLY26] Venkatesan Guruswami, Xin Lyu, and Weiqiang Yuan. "Cell-probe lower bounds via semi-random CSP refutation: simplified and the odd-locality case". In: *SODA*. SIAM, 2026, pp. 372–387 (cit. on pp. 2, 7, 19, 28).

[Gol11] Oded Goldreich. "Candidate one-way functions based on expander graphs". In: *Studies in Complexity and Cryptography*. Vol. 6650. Lecture Notes in Computer Science. Springer, 2011, pp. 76–87 (cit. on p. 21).

[GS12a] Ariel Gabizon and Ronen Shaltiel. "Increasing the output length of zero-error dispersers". In: *Random Structures & Algorithms* 40.1 (2012), pp. 74–104 (cit. on p. 9).

[GS12b] Ariel Gabizon and Ronen Shaltiel. "Invertible zero-error dispersers and defective memory with stuck-at errors". In: *APPROX-RANDOM*. Springer, 2012, pp. 553–564 (cit. on p. 9).

[Har66] Lawrence H. Harper. "Optimal numberings and isoperimetric problems on graphs". In: *Journal of Combinatorial Theory* 1.3 (1966), pp. 385–393 (cit. on p. 14).

[Hir18] Shuichi Hirahara. "Non-black-box worst-case to average-case reductions within NP". In: *FOCS*. IEEE Computer Society, 2018, pp. 247–258 (cit. on p. 24).

[Hir23] Shuichi Hirahara. "Non-disjoint promise problems from meta-computational view of pseudorandom generator constructions". In: *Theory Comput.* 19 (2023), pp. 1–61 (cit. on p. 10).

[HLZ26] Shengtang Huang, Xin Li, and Yan Zhong. "Range avoidance and remote point: new algorithms and hardness". In: *ITCS*. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2026, 79:1–79:19 (cit. on pp. 1, 2, 19).

[HS17] Shuichi Hirahara and Rahul Santhanam. "On the average-case complexity of MCSP and its variants". In: *CCC*. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, 7:1–7:20 (cit. on p. 24).

[IK00] Yuval Ishai and Eyal Kushilevitz. "Randomizing polynomials: A new representation with applications to round-efficient secure computation". In: *FOCS*. IEEE Computer Society, 2000, pp. 294–304 (cit. on p. 1).

[IK02] Yuval Ishai and Eyal Kushilevitz. "Perfect constant-round secure computation via perfect randomizing polynomials". In: *ICALP*. Lecture Notes in Computer Science. Springer, 2002, pp. 244–256 (cit. on p. 1).

[Ila25] Rahul Ilango. "The oracle derandomization hypothesis is false (and more) assuming no natural proofs". In: *Electron. Colloquium Comput. Complex.* (2025) (cit. on pp. 3, 5, 6, 8, 10, 17, 18).

[ILW23] Rahul Ilango, Jiatu Li, and R. Ryan Williams. "Indistinguishability obfuscation, range avoidance, and bounded arithmetic". In: *STOC*. ACM, 2023, pp. 1076–1089 (cit. on pp. 3, 11).

[IW97] Russell Impagliazzo and Avi Wigderson. "P = BPP if E requires exponential circuits: derandomizing the XOR lemma". In: *STOC*. ACM, 1997, pp. 220–229 (cit. on p. 3).

[Jac97] Jeffrey C. Jackson. "An efficient membership-query algorithm for learning DNF with respect to the uniform distribution". In: *J. Comput. Syst. Sci.* 55.3 (1997), pp. 414–440 (cit. on p. 32).

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. "Indistinguishability obfuscation from well-founded assumptions". In: *STOC*. ACM, 2021, pp. 60–73 (cit. on p. 3).

[Kat68] Gyula Katona. "A theorem of finite sets". In: *Classic Papers in Combinatorics*. Springer, 1968, pp. 381–401 (cit. on p. 14).

[Kha22] Erfan Khaniki. "Nisan–Wigderson generators in proof complexity: new lower bounds". In: *CCC*. Vol. 234. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 17:1–17:15 (cit. on p. 5).

[KK09] Adam Tauman Kalai and Varun Kanade. "Potential-based agnostic boosting". In: *NIPS*. Curran Associates, Inc., 2009, pp. 880–888 (cit. on p. 29).

[KKMP21] Robert Kleinberg, Oliver Korten, Daniel Mitropolsky, and Christos Papadimitriou. "Total Functions in the Polynomial Hierarchy". In: *ITCS*. Vol. 185. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, 44:1–44:18. ISBN: 978-3-95977-177-1 (cit. on pp. 1, 2).

[KL18] Pravesh K. Kothari and Roi Livni. "Improper Learning by Refuting". In: *ITCS*. Vol. 94. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018, 55:1–55:10. ISBN: 978-3-95977-060-6 (cit. on pp. 28, 29).

[KMOW17] Pravesh K. Kothari, Ryuhei Mori, Ryan O'Donnell, and David Witmer. "Sum of squares lower bounds for refuting any CSP". In: *STOC*. ACM, 2017, pp. 132–145. ISBN: 9781450345286 (cit. on pp. 28, 42).

[KMV08] Adam Tauman Kalai, Yishay Mansour, and Elad Verbin. "On agnostic boosting and parity learning". In: *STOC*. 2008, pp. 629–638 (cit. on p. 47).

[Kor21] Oliver Korten. "The hardest explicit construction". In: *FOCS*. IEEE, 2021, pp. 433–444 (cit. on pp. 1, 11).

[Kor25] Oliver Korten. "Range avoidance and the complexity of explicit constructions". In: *Bull. EATCS* 145 (2025) (cit. on p. 1).

[KP24] Oliver Korten and Toniann Pitassi. "Strong vs. weak range avoidance and the linear ordering principle". In: *FOCS*. IEEE, 2024, pp. 1388–1407 (cit. on p. 1).

[KPI25] Oliver Korten, Toniann Pitassi, and Russell Impagliazzo. "Stronger cell probe lower bounds via local PRGs". In: *FOCS*. 2025, pp. 2480–2504 (cit. on pp. 2, 7, 19).

[Kra01] Jan Krajíček. "On the weak pigeonhole principle". In: *Fundamenta Mathematicae* 170 (2001), pp. 123–140 (cit. on p. 5).

[Kra04] Jan Krajíček. "Dual weak pigeonhole principle, pseudo-surjective functions, and provability of circuit lower bounds". In: *Journal of Symbolic Logic* 69.1 (2004), pp. 265–286 (cit. on p. 5).

[Kra11] Jan Krajíček. "On the proof complexity of the Nisan–Wigderson generator based on a hard NP ∩ coNP function". In: *J. Math. Log.* 11.1 (2011) (cit. on p. 5).

[Kra12] Jan Krajíček. "Pseudo-finite hard instances for a student-teacher game with a Nisan–Wigderson generator". In: *Log. Methods Comput. Sci.* 8.3 (2012) (cit. on p. 5).

[Kra19] Jan Krajíček. "Proof Complexity". Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2019 (cit. on p. 12).

[Kra25] Jan Krajíček. "Proof complexity generators". Cambridge University Press, 2025. ISBN: 9781009611664 (cit. on pp. 5, 20).

[Kra26] Jan Krajíček. "Failure of the strong feasible disjunction property". In: *CoRR* abs/2604.04830 (2026) (cit. on p. 18).

[Kra97] Jan Krajíček. "Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic". In: *J. Symb. Log.* 62.2 (1997), pp. 457–486 (cit. on p. 5).

[Kru63] Joseph B. Kruskal. "The number of simplices in a complex". In: *Mathematical optimization techniques* 10 (1963), pp. 251–278 (cit. on p. 14).

[KS04] Adam R. Klivans and Rocco A. Servedio. "Learning DNF in time $2^{\tilde{O}(n^{1/3})}$". In: *J. Comput. Syst. Sci.* 68.2 (2004), pp. 303–318 (cit. on p. 7).

[KS25a] Oliver Korten and Rahul Santhanam. "How to construct random strings". In: *CCC*. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025, 35:1–35:32 (cit. on p. 5).

[KS25b] Neha Kuntewar and Jayalal Sarma. "Avoiding range via Turan-type bounds". In: *APPROX/RANDOM*. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025, 62:1–62:21 (cit. on p. 2).

[KSS92] Michael J. Kearns, Robert E. Schapire, and Linda M. Sellie. "Toward efficient agnostic learning". In: *COLT*. ACM, 1992, pp. 341–352. ISBN: 089791497X (cit. on p. 8).

[Lau83] Clemens Lautemann. "BPP and the polynomial hierarchy". In: *Inf. Process. Lett.* 17.4 (1983), pp. 215–217 (cit. on p. 17).

[Li11] Xin Li. "A new approach to affine extractors and dispersers". In: *CCC*. IEEE, 2011, pp. 137–147. ISBN: 9780769544113 (cit. on p. 9).

[Li23] Xin Li. "Two source extractors for asymptotically optimal entropy, and (many) more." In: *FOCS*. IEEE, 2023, pp. 1271–1281 (cit. on p. 1).

[Li24] Zeyong Li. "Symmetric exponential time requires near-maximum circuit size: simplified, truly uniform". In: *STOC*. ACM, 2024, pp. 2000–2007. ISBN: 9798400703836 (cit. on p. 1).

[Lov07] László Lovász. "Combinatorial problems and exercises". Vol. 361. American Mathematical Soc., 2007 (cit. on p. 14).

[LP21] Yanyi Liu and Rafael Pass. "Cryptography from sublinear-time average-case hardness of time-bounded Kolmogorov complexity". In: *STOC*. ACM, 2021, pp. 722–735 (cit. on p. 10).

[LY22] Jiatu Li and Tianqi Yang. "$3.1n - o(n)$ circuit lower bounds for explicit functions". In: *STOC*. ACM, 2022, pp. 1180–1193 (cit. on p. 1).

[LZ24] Xin Li and Yan Zhong. "Explicit Directional Affine Extractors and Improved Hardness for Linear Branching Programs". In: *39th Computational Complexity Conference (CCC 2024)*. Ed. by Rahul Santhanam. Vol. 300. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024, 10:1–10:14. ISBN: 978-3-95977-331-7 (cit. on p. 9).

[MMW19] Dylan M. McKay, Cody D. Murray, and R. Ryan Williams. "Weak lower bounds on resource-bounded compression imply strong separations of complexity classes". In: *STOC*. ACM, 2019, pp. 1215–1225 (cit. on p. 10).

[MST06] Elchanan Mossel, Amir Shpilka, and Luca Trevisan. "On ε-biased generators in NC^0". In: *Random Struct. Algorithms* 29.1 (2006), pp. 56–81 (cit. on p. 7).

[NW94] Noam Nisan and Avi Wigderson. "Hardness vs randomness". In: *J. Comput. Syst. Sci.* 49.2 (1994), pp. 149–167 (cit. on pp. 3, 5).

[OPS21] Igor C. Oliveira, Ján Pich, and Rahul Santhanam. "Hardness magnification near state-of-the-art lower bounds". In: *Theory Comput.* 17 (2021), pp. 1–38 (cit. on p. 10).

[OS18] Igor C. Oliveira and Rahul Santhanam. "Hardness magnification for natural problems". In: *FOCS*. IEEE Computer Society, 2018, pp. 65–76 (cit. on p. 10).

[OST22] Igor C. Oliveira, Rahul Santhanam, and Roei Tell. "Expander-based cryptography meets natural proofs". In: *computational complexity* 31.1 (2022), p. 4 (cit. on p. 7).

[OW14] Ryan O'Donnell and David Witmer. "Goldreich's PRG: evidence for near-optimal polynomial stretch". In: *CCC*. IEEE, 2014, pp. 1–12. ISBN: 9781479936267 (cit. on p. 7).

[Paj85] Alain Pajor. "Sous espaces l_1^n des espaces de Banach". Editions Hermann, 1985 (cit. on pp. 6, 14).

[Păt08] Mihai Pătraşcu. "Succincter". In: *FOCS*. IEEE Computer Society, 2008, pp. 305–313 (cit. on p. 1).

[Pic11] Ján Pich. "Nisan–Wigderson generators in proof systems with forms of interpolation". In: *Math. Log. Q.* 57.4 (2011), pp. 379–383 (cit. on p. 5).

[Pic15] Ján Pich. "Circuit lower bounds in bounded arithmetics". In: *Ann. Pure Appl. Log.* 166.1 (2015), pp. 29–45 (cit. on p. 5).

[Pra62] Eugene Prange. "The use of information sets in decoding cyclic codes". In: *IRE Trans. Inf. Theory* 8.5 (1962), pp. 5–9 (cit. on p. 28).

[PS19] Ján Pich and Rahul Santhanam. "Why are proof complexity lower bounds hard?" In: *FOCS*. IEEE Computer Society, 2019, pp. 1305–1324 (cit. on pp. 5, 24).

[PS21] Ján Pich and Rahul Santhanam. "Strong co-nondeterministic lower bounds for NP cannot be proved feasibly". In: *STOC*. ACM, 2021, pp. 223–233 (cit. on p. 5).

[PS22] Ján Pich and Rahul Santhanam. "Learning algorithms versus automatability of Frege systems". In: *ICALP*. Vol. 229. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 101:1–101:20 (cit. on p. 5).

[Ram20] C. Ramya. "Recent progress on matrix rigidity - A survey". In: *CoRR* abs/2009.09460 (2020) (cit. on p. 1).

[Raz04a] Ran Raz. "Resolution lower bounds for the weak pigeonhole principle". In: *J. ACM* 51.2 (2004), pp. 115–138 (cit. on p. 5).

[Raz04b] Alexander A. Razborov. "Resolution lower bounds for perfect matching principles". In: *J. Comput. Syst. Sci.* 69.1 (2004), pp. 3–27 (cit. on p. 5).

[Raz15] Alexander A. Razborov. "Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution". In: *Annals of Mathematics* (2015), pp. 415–472 (cit. on p. 5).

[Raz95] Alexander A. Razborov. "Bounded arithmetic and lower bounds in Boolean complexity". In: *Feasible Mathematics II*. Birkhäuser Boston, 1995, pp. 344–386. ISBN: 978-1-4612-2566-9 (cit. on p. 5).

[Raz98] Alexander A. Razborov. "Lower bounds for the polynomial calculus". In: *Comput. Complex.* 7.4 (1998), pp. 291–324 (cit. on p. 5).

[RR97] Alexander A. Razborov and Steven Rudich. "Natural proofs". In: *J. Comput. Syst. Sci.* 55.1 (1997), pp. 24–35 (cit. on p. 4).

[RRS17] Prasad Raghavendra, Satish Rao, and Tselil Schramm. "Strongly refuting random CSPs below the spectral threshold". In: *STOC*. 2017, pp. 121–131 (cit. on p. 43).

[RSW22] Hanlin Ren, Rahul Santhanam, and Zhikun Wang. "On the range avoidance problem for circuits". In: *FOCS*. 2022, pp. 640–650 (cit. on pp. 1, 5).

[Rud97] Steven Rudich. "Super-bits, demi-bits, and $\mathsf{NP}/_{\mathrm{qpoly}}$-natural proofs". In: *RANDOM*. Vol. 1269. Lecture Notes in Computer Science. Springer, 1997, pp. 85–93 (cit. on pp. 3, 4).

[RWZ26] Hanlin Ren, Yichuan Wang, and Yan Zhong. "Hardness of range avoidance and proof complexity generators from demi-bits". In: *ITCS*. Vol. 362. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2026, 111:1–111:25 (cit. on pp. 3, 5, 6, 8–10, 16, 19–21, 27).

[Sau72] Norbert Sauer. "On the density of families of sets". In: *J. Comb. Theory A* 13.1 (1972), pp. 145–147 (cit. on pp. 6, 14).

[Sch90] Robert E. Schapire. "The strength of weak learnability". In: *Mach. Learn.* 5 (1990), pp. 197–227 (cit. on p. 32).

[Seg07] Nathan Segerlind. "The complexity of propositional proofs". In: *Bull. Symb. Log.* 13.4 (2007), pp. 417–481 (cit. on p. 12).

[Sha49] Claude E. Shannon. "The synthesis of two-terminal switching circuits". In: *Bell System technical journal* 28.1 (1949), pp. 59–98 (cit. on p. 1).

[She72] Saharon Shelah. "A combinatorial problem; stability and order for models and theories in infinitary languages". In: *Pacific Journal of Mathematics* 41.1 (1972), pp. 247–261 (cit. on pp. 6, 14).

[ST21] Rahul Santhanam and Iddo Tzameret. "Iterated lower bound formulas: a diagonalization-based approach to proof complexity". In: *STOC*. ACM, 2021, pp. 234–247 (cit. on p. 5).

[Ta-17] Amnon Ta-Shma. "Explicit, almost optimal, epsilon-balanced codes". In: *STOC*. ACM, 2017, pp. 238–251 (cit. on p. 1).

[Uma03] Christopher Umans. "Pseudo-random generators for all hardnesses". In: *J. Comput. Syst. Sci.* 67.2 (2003), pp. 419–440 (cit. on p. 3).

[Ünal23] Akın Ünal. "Worst-case subexponential attacks on PRGs of constant degree or constant locality". In: *EUROCRYPT (1)*. Springer, 2023, pp. 25–54 (cit. on p. 7).

[Urq87] Alasdair Urquhart. "Hard examples for resolution". In: *J. ACM* 34.1 (1987), pp. 209–219 (cit. on p. 21).

[VA68] V. N. Vapnik and A. Ya. Chervonenkis. "The uniform convergence of frequencies of the appearance of events to their probabilities". In: *Dokl. Akad. Nauk SSSR* 181 (4 1968), pp. 781–783 (cit. on pp. 6, 14).

[Vad17] Salil P. Vadhan. "On learning vs. refutation". In: *COLT*. Vol. 65. Proceedings of Machine Learning Research. PMLR, 2017, pp. 1835–1848 (cit. on pp. 7, 22, 28, 42–44, 46).

[Val77] Leslie G. Valiant. "Graph-theoretic arguments in low-level complexity". In: *MFCS*. Lecture Notes in Computer Science. Springer, 1977, pp. 162–176 (cit. on pp. 1, 2).

[Var64] R. Varshamov. "Estimate of the number of signals in error correcting codes". In: *Avtomat. i Telemekh* 25 (11 1964), pp. 1628–1629 (cit. on p. 1).

[Wil16] R. Ryan Williams. "Natural proofs versus derandomization". In: *SIAM J. Comput.* 45.2 (2016), pp. 497–529 (cit. on p. 10).

[Yao82] Andrew C. Yao. "Theory and application of trapdoor functions". In: *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*. 1982, pp. 80–91 (cit. on pp. 5, 46).

[Yu22] Huacheng Yu. "Nearly optimal static Las Vegas succinct dictionary". In: *SIAM J. Comput.* 51.3 (2022), pp. 20–174 (cit. on p. 1).

[Zha22] Lu-Ming Zhang. "Nondeterministic variants of pseudorandomness, and learning theory". MSc thesis. Imperial College London, Sept. 2022 (cit. on pp. 47, 48).

A Omitted Proofs

A.1 Proof of Fact 5.6

Fact 5.6. *If there is an* AM *algorithm that refutes random k-SAT with $m(n)$ clauses with error ε, then there is an* $\mathsf{NP}/_{\mathrm{poly}}$ *algorithm that refutes random k-SAT with $m(n)$ clauses with error $O(\varepsilon)$.*

Proof. Let $\mathcal{A}(C, r, w)$ denote an AM algorithm that refutes random k-SAT with $m(n)$ clauses with error ε, where r denotes Arthur's randomness and w denotes Merlin's proof. We repeat this algorithm $t := \mathrm{poly}(m(n), \log(1/\varepsilon))$ times: let $\mathcal{B}(C, r_{1\ldots t}, w_{1\ldots t})$ denote another AM algorithm that accepts if the majority of $\mathcal{A}(C, r_1, w_1), \ldots, \mathcal{A}(C, r_t, w_t)$ accepts. As usual, we use "$\mathcal{B}(C)$ accepts" to denote the event (over the choice of $r_{1\ldots t}$) that there exists $w_{1\ldots t}$ such that $\mathcal{B}(C, r_{1\ldots t}, w_{1\ldots t})$ accepts. We can see that:

- If C is a satisfiable k-CNF with $m(n)$ clauses, then by a Chernoff bound, $\Pr[\mathcal{B}(C) \text{ accepts}] \leq 2^{-\Omega(t)}$.
- We say that a k-CNF is *good* if $\Pr[\mathcal{A}(C) \text{ accepts}] \geq 3/4$, then by a Markov bound, a random k-CNF is good w.p. $\geq 1-4\varepsilon$. If C is good, then by a Chernoff bound, $\Pr[\mathcal{B}(C) \text{ accepts}] \geq 1-2^{-\Omega(t)} \geq 1-\varepsilon$. It follows from a union bound that a random k-CNF is accepted by $\mathcal{B}$ with probability $\geq 1-5\varepsilon$.

Now choose $r_{1\ldots t}$ uniformly at random and hardwire it into the algorithm $\mathcal{B}$. Call the resulting algorithm $\mathcal{B}'$; clearly, $\mathcal{B}'$ is an $\mathsf{NP}/_{\text{poly}}$ algorithm. We claim that with nonzero probability, $\mathcal{B}'$ refutes random k-SAT with $m(n)$ clauses.

- Since there are only $2^{\tilde{O}(m(n))}$ many k-CNFs, it follows from a union bound that with probability at least $1-2^{-\Omega(t)} \cdot 2^{m(n)} \geq 0.99$, $\mathcal{B}'$ rejects every satisfiable k-CNF.
- By a Markov bound, at least half of random strings $r_{1\ldots t}$ satisfy that $\mathcal{B}'$ accepts a random k-CNF with probability at least $1-10\varepsilon$.

Hence, with probability at least 0.49, the algorithm $\mathcal{B}'$ is an $\mathsf{NP}/_{\text{poly}}$ algorithm that refutes random k-SAT with $m(n)$ clauses with error 10ε. □

A.2 Proof of Theorem 5.7

Theorem 5.7 ([DS16]). *Let $n, k, m \in \mathbb{N}$, $\delta > 0$, and let $t := O(2^k \log(m/\delta))$. Then the random k-SAT problem with n inputs and m clauses reduces to PAC-learning CNFs of input length $n' := 2nt$, size $s := O(nt)$, and top fan-in t, using $L := \lfloor m/t \rfloor$ samples.*

More precisely, there exists a polynomial-time randomized reduction $\mathcal{R}$ that takes a k-SAT instance φ as input and outputs a partial function $\{(x_i \mapsto b_i)\}_{i\in[L]}$, such that the following holds:

- **$\mathcal{R}$ maps satisfiable instances to easy partial functions.**

  For every φ that is satisfiable, with probability at least $1-\delta$ over the internal randomness of $\mathcal{R}$, the partial function $\{(x_i \mapsto b_i)\} \leftarrow \mathcal{R}(\varphi)$ can be computed by CNFs of size s and top fanin t.

- **$\mathcal{R}$ maps random instances to random partial functions.**

  The following holds for some input distribution $\mathcal{D}$ over $\{0,1\}^{n'}$. Let φ be a random k-SAT instance and let $\{(x_i \mapsto b_i)\} \leftarrow \mathcal{R}(\varphi)$, then each (x_i, b_i) is i.i.d. distributed according to $\mathcal{D} \times \{0,1\}$.

Random k-SAT				**Learning CNFs**			
n	k	m	δ	s	t	L	n'
#inputs	arity	#clauses	fail. prob.	CNF size	top fanin	domain size	input length
				$O(nt)$	$O(2^k \log(m/\delta))$	$\lfloor m/t \rfloor$	$2nt$

Table 1: Parameters in Theorem 5.7

Proof. Let $\varphi(x) = C_1(x) \wedge C_2(x) \wedge \cdots \wedge C_m(x)$ denote the input CNF, where C_i denotes the i-th clause of φ. Let $t := O(2^k \log(m/\delta))$, we first group the clauses into CNFs of top fanin t. That is, we define

$$C'_i(x) = C_{(i-1)t+1}(x) \wedge C_{(i-1)t+2}(x) \wedge \cdots \wedge C_{it}(x).$$

Consider the list $\{(C'_i, 1)\}_{i\in[\lfloor m/t \rfloor]}$. We say that an input x *satisfies* the list if for every $i \in [\lfloor m/t \rfloor]$, $C'_i(x) = 1$. Observe that if φ is satisfiable, then the same assignment satisfies the list; if φ is a random k-CNF, then every C'_i is a random k-CNF of top fanin t.

Then, we replace each item in the list $\{(C'_i, 1)\}$ with (random k-CNF, 0) with probability $1/2$. More precisely, we obtain a new list $\{(C''_i, b_i)\}$ where for each $i \in [\lfloor m/t \rfloor]$:

- with probability $1/2$, $C_i'' = C_i'$ and $b_i = 1$;
- otherwise, C_i'' is a random k-CNF of top fanin t, and $b_i = 0$.

Note that if φ is satisfied by an assignment x, then with probability $\geq 1-\delta$, x also satisfies the new list in the sense that $C_i''(x) = b_i$ for every $i \in [\lfloor m/t \rfloor]$. This is because the probability that x satisfies a random k-CNF is at most $(1-2^{-k})^t \leq \delta/m$; hence by a union bound, with probability at least $1-\delta$, for every $i \in [\lfloor m/t \rfloor]$, if $b_i = 0$ then $C_i''(x) = 0$. On the other hand, if φ is a random k-CNF, then every C_i'' is a random k-CNF of top fanin t and every b_i is a random bit.

Finally, it is well known that one can encode a k-CNF C over n input bits with top fanin t into $2n \cdot t$ bits $\langle C \rangle \in \{0,1\}^{2n \cdot t}$, and transform an input $x \in \{0,1\}^n$ into a CNF ϕ_x over $2n \cdot t$ input bits with size $s = O(nt)$ and top fanin t, such that $C(x) = \phi_x(\langle C \rangle)$. (See, e.g., [Vad17, Proposition 10].) Our reduction $\mathcal{R}(\varphi)$ outputs the partial function

$$\{(\langle C_i'' \rangle, b_i)\}_{i \in [\lfloor m/t \rfloor]}.$$

Suppose that φ is satisfied by an assignment x. Then with probability $\geq 1-\delta$, the CNF ϕ_x computes the above partial function. On the other hand, let $\mathcal{D}$ denote the encoding of a random k-CNF with top fanin t. If φ is a random k-CNF, then every $(\langle C_i'' \rangle, b_i)$ is an i.i.d. sample from $\mathcal{D} \times \{0,1\}$. □

B Literature on Refuting CSP

The table below summarizes known results on CSP refutation, following the presentation in Table 1 of [AOW15]. Throughout, the arity k is fixed and $m = \text{poly}(n)$. Unless otherwise specified, the instances are fully random.

The "CSP" column specifies the predicate and instance model. The "Strength" column describes the certified upper bound on the fraction of simultaneously satisfiable constraints: a "refutation" certifies unsatisfiability; a "$1/n^{\Omega(1)}$-refutation" certifies an upper bound of $1 - 1/n^{\Omega(1)}$; and a "strong" refutation, as used here, certifies an upper bound of $\rho(P) + \eta$ for any fixed $\eta > 0$, where $\rho(P) := |P^{-1}(1)|/2^k$. The last guarantee is near-optimal, since a uniformly random assignment satisfies exactly a $\rho(P)$ fraction of the constraints in expectation.

The "Eff./Exist." column distinguishes *efficient* refutation by a deterministic polynomial-time algorithm from *existential* refutation by polynomial-size certificates verifiable in deterministic polynomial time (i.e., a *nondeterministic* polynomial-time refutation algorithm). All certificates are sound on every input instance, while their existence or efficient discovery is guaranteed with high probability over the randomness of the specified instance model.

In the bounds below, implicit constants may depend on k and, for strong refutation, on η. The notation $\widetilde{\Omega}$ suppresses polylogarithmic factors in n. For the smoothed model, all literal-perturbation rates are fixed positive constants.

The lower bounds of [KMOW17] show that polynomial-time sum-of-squares refutation of fully random k-XOR requires $\widetilde{\Omega}(n^{k/2})$ constraints, essentially matching the corresponding upper bound. These results provide evidence for the optimality of the spectral threshold for strong refutation, but do not establish a lower bound for unrestricted polynomial-time or nondeterministic algorithms.

C Equivalence between Agnostic PAC-learning $\mathscr{C}$ and Strong RRHS-refutation of $\mathscr{C}^*$

We prove the equivalence between agnostic PAC-learning and strong RRHS-refutation, extending [Vad17, Theorem 5] (which establishes PAC-learning $\iff$ RRHS-refutation in the realizable

CSP	Regime for m	Strength	Eff./Exist.	Reference
3-SAT	$\Omega(n^{7/5})$	Refutation	Existential	[FKO06]
3-XOR	$m \geq 2n$	$1/n^{\Omega(1)}$-refutation	Efficient	Gaussian elimination
3-SAT	$\Omega(n^{3/2})$	Refutation	Efficient	Claimed in [FKO06]
Boolean k-CSP	$\widetilde{\Omega}(n^{\lceil k/2 \rceil})$	Strong	Efficient	[CCF10]
k-XOR	$\widetilde{\Omega}(n^{k/2})$	Strong	Efficient	[BM16; AOW15]
Random k-CSP	$\widetilde{\Omega}(n^{k/2})$	Strong	Efficient	[AOW15; RRS17]
Semi-random k-XOR/CSP	$\widetilde{\Omega}(n^{k/2})$	Strong	Efficient	[AGK21]
Smoothed k-CSP	$\widetilde{\Omega}(n^{\frac{k}{2}-\frac{k-2}{2(k+8)}})$	Refutation	Existential	[GKM22]

Table 2: Known results on deterministic and nondeterministic CSP refutation. For smoothed 3-CSPs with constant smoothing rates, [GKM22] improves the general bound to $\widetilde{O}(n^{7/5})$ constraints. Note that as instance models, random $\subset$ semi-random $\subset$ smoothed, so results for more general models subsume those for more restricted ones.

setting) to the agnostic setting. Throughout, we use the notation in [Vad17]: an evaluation function $\mathsf{Eval}\colon \{0,1\}^s \times \{0,1\}^t \to \{0,1\}$ gives rise to a circuit class $\mathscr{C} = \{C_x\}_{x\in\{0,1\}^s}$ and its dual $\mathscr{C}^* = \{C^*_y\}_{y\in\{0,1\}^t}$, where $C_x(y) = C^*_y(x) = \mathsf{Eval}(x,y)$. We suppress unary encodings of the accuracy parameters from the displayed algorithm inputs, and all probabilities include the algorithms' internal randomness.

We begin by recalling the relevant definitions.

Definition C.1 (Agnostic PAC-learning, prediction version)**.** $\mathscr{C}$ is *agnostically PAC-learnable* with sample complexity $m = m(s,t,1/\varepsilon)$ if there is a polynomial-time algorithm $\mathcal{A}$ such that for every $0 < \varepsilon \leq 1/2$, every $s,t \in \mathbb{N}$, and every distribution $\mathcal{D}$ on $\{0,1\}^t \times \{0,1\}$, given $m+1$ i.i.d. samples $(y_1,b_1),\ldots,(y_{m+1},b_{m+1})$ from $\mathcal{D}$, it holds that

$$\Pr\Big[\mathcal{A}\big(1^s, 1^t, (y_1,b_1),\ldots,(y_m,b_m),\ y_{m+1}\big) = b_{m+1}\Big] \ \geq \ \max_{x\in\{0,1\}^s} \Pr_{(y,b)\leftarrow\mathcal{D}}\big[C_x(y) = b\big] - \varepsilon.$$

We say $\mathscr{C}$ is agnostically PAC-learnable if $m(s,t,1/\varepsilon) \leq \mathrm{poly}(s,t,1/\varepsilon)$.

Remark C.2. In the realizable case (i.e., $b = C_{x^*}(y)$ for some x^*), the maximum equals 1 and the definition reduces to requiring prediction probability $\geq 1-\varepsilon$, recovering [Vad17, Definition 1] up to the parametrization of the constant.

Definition C.3 (Strong RRHS-refutation)**.** For every $0 < \gamma \leq 1/2$, let $N = N(s,t,1/\gamma)$ satisfy $N \geq \lceil 2/\gamma^2 \rceil$. We say that $\mathscr{C}^*$ is *strongly RRHS-refutable* with N equations if there is a polynomial-time algorithm $\mathcal{B}$ such that for every such γ and every $y_1,\ldots,y_N \in \{0,1\}^t$:

- *(Soundness)* For every $b_1,\ldots,b_N \in \{0,1\}$, if there exists $x \in \{0,1\}^s$ with

$$\frac{1}{N}\big|\{i \in [N] : C^*_{y_i}(x) = b_i\}\big| \ \geq \ \tfrac{1}{2} + \gamma,$$

  then $\Pr\big[\mathcal{B}(1^s,1^t,(y_1,b_1),\ldots,(y_N,b_N)) = 1\big] \leq 1/3$.

- *(Completeness)* If we sample i.i.d. $b_1,\ldots,b_N \leftarrow \{0,1\}$, then $\Pr\big[\mathcal{B}(1^s,1^t,(y_1,b_1),\ldots,(y_N,b_N)) = 1\big] \geq 2/3$.

We say $\mathscr{C}^*$ is strongly RRHS-refutable if $N(s,t,1/\gamma) \leq \mathrm{poly}(s,t,1/\gamma)$.

Remark C.4. At $\gamma = 1/2$, the soundness condition reduces to that of standard RRHS-refutation [Vad17, Definition 4]: it applies only when the system is *fully satisfiable*. Strong refutation extends this condition to partial satisfiability.

The main result of this appendix shows that these two notions are equivalent.

Theorem C.5. *Let $\mathscr{C} = (\mathscr{C}^*)^*$ be a circuit class given by* $\mathsf{Eval}\colon \{0,1\}^s \times \{0,1\}^t \to \{0,1\}$. *Then:*

- *If $\mathscr{C}$ is agnostically PAC-learnable with sample complexity $m(s,t,1/\varepsilon)$, then $\mathscr{C}^*$ is strongly RRHS-refutable using $N = O(m(s,t,3/\gamma)/\gamma^3)$ equations.*
- *If $\mathscr{C}^*$ is strongly RRHS-refutable using $N(s,t,1/\gamma)$ equations, then $\mathscr{C}$ is agnostically PAC-learnable with sample complexity $m = \mathrm{poly}(N(s,t,3/\varepsilon))$.*

In particular, $\mathscr{C}$ is agnostically PAC-learnable if and only if $\mathscr{C}^*$ is strongly RRHS-refutable.

Proof. We prove each direction in turn.

Agnostic PAC-learning $\implies$ Strong RRHS-refutation. Let $\mathcal{A}$ be the agnostic PAC-learner with sample complexity $m(s,t,1/\varepsilon)$. We may assume that $m(s,t,1/\varepsilon) \geq 1$, since the learner can ignore an additional sample. Given the refutation threshold $1/2 \geq \gamma > 0$, set $\varepsilon = \gamma/3$, $m_0 = m(s,t,3/\gamma)$, and $q = \lceil 10m_0/\gamma \rceil$.

Define the following one-block procedure $\mathcal{B}_0$ on input $(1^s, 1^t, (y_1, b_1), \ldots, (y_q, b_q))$ as follows:

- Sample i.i.d. $i_1, \ldots, i_{m_0+1} \leftarrow [q]$.
- Output 1 if and only if $\mathcal{A}(1^s, 1^t, (y_{i_1}, b_{i_1}), \ldots, (y_{i_{m_0}}, b_{i_{m_0}}), y_{i_{m_0+1}}) \neq b_{i_{m_0+1}}$ or $i_{m_0+1} \in \{i_1, \ldots, i_{m_0}\}$.

Claim C.6. *Let E be the event of collision: $E := \{i_{m_0+1} \in \{i_1, \ldots, i_{m_0}\}\}$, then it holds that $\gamma/11 \leq \Pr[E] \leq \gamma/10$.*

Proof. Let $p := \Pr[E] = 1 - (1 - 1/q)^{m_0}$. By the union bound,

$$p \leq \frac{m_0}{q} \leq \frac{\gamma}{10}.$$

For the other direction, writing $u = m_0/q$, we have

$$u \geq \frac{\gamma m_0}{10m_0 + \gamma} \geq \frac{\gamma}{10 + \gamma}.$$

Hence

$$p \geq 1 - e^{-u} \geq \frac{u}{1+u} \geq \frac{\gamma}{10 + 2\gamma} \geq \frac{\gamma}{11},$$

where the last inequality uses $\gamma \leq 1/2$. □

Soundness of $\mathcal{B}_0$: Suppose there exists $x^* \in \{0,1\}^s$ with $\frac{1}{q}|\{i : C^*_{y_i}(x^*) = b_i\}| \geq \frac{1}{2} + \gamma$. Define the empirical distribution $\mathcal{D}$ as uniform on the multiset $\{(y_i, b_i) : i \in [q]\}$. Then

$$\max_x \Pr_{(y,b)\leftarrow\mathcal{D}}\big[C_x(y) = b\big] \;\geq\; \Pr_{\mathcal{D}}\big[C_{x^*}(y) = b\big] = \frac{1}{q}\big|\{i : C^*_{y_i}(x^*) = b_i\}\big| \;\geq\; \frac{1}{2} + \gamma.$$

Since $i_1, \ldots, i_{m_0+1}$ are sampled independently with replacement, the samples fed to $\mathcal{A}$ are i.i.d. from $\mathcal{D}$, and the agnostic learning guarantee gives

$$\Pr[\mathcal{A} \text{ correct}] \;\geq\; \left(\frac{1}{2} + \gamma\right) - \varepsilon = \frac{1}{2} + \gamma - \frac{\gamma}{3} = \frac{1}{2} + \frac{2\gamma}{3}.$$

Accounting for collisions:

$$\Pr[\mathcal{B}_0 = 1] = \Pr[\mathcal{A} \text{ incorrect} \vee E] \;\leq\; \left(\frac{1}{2} - \frac{2\gamma}{3}\right) + \frac{\gamma}{10} = \frac{1}{2} - \frac{17\gamma}{30}.$$

Completeness of $\mathcal{B}_0$: Suppose $b_1, \ldots, b_q \leftarrow \{0,1\}$ are i.i.d. samples. Conditioned on $i_{m_0+1} \notin \{i_1, \ldots, i_{m_0}\}$, the bit $b_{i_{m_0+1}}$ is uniform and independent of the view of $\mathcal{A}$, so $\mathcal{A}$ predicts it correctly with probability exactly $1/2$. Therefore

$$\Pr[\mathcal{B}_0 = 1] = \frac{1}{2}\Pr[i_{m_0+1} \notin \{i_1, \ldots, i_{m_0}\}] + \Pr[i_{m_0+1} \in \{i_1, \ldots, i_{m_0}\}] \;\geq\; \frac{1}{2} + \frac{\gamma}{22}.$$

Amplification. Let R be the smallest odd integer satisfying $R \geq 242 \ln 3/\gamma^2$, and set $N = Rq$. Define the final refuter $\mathcal{B}$ on N equations by partitioning them into R disjoint blocks of size q, independently running $\mathcal{B}_0$ on each block to obtain bits $X_1, \ldots, X_R$, and outputting their majority.

For soundness, suppose some $x^* \in \{0,1\}^s$ satisfies at least a $1/2 + \gamma$ fraction of all N equations. Let a_r be the fraction of equations in block r satisfied by x^*. Then

$$\frac{1}{R}\sum_{r=1}^{R} a_r \geq \frac{1}{2} + \gamma.$$

Writing $p_r = \Pr[X_r = 1]$, the preceding argument gives

$$p_r \leq 1 - a_r + \frac{\gamma}{3} + \frac{\gamma}{10}.$$

Therefore

$$\frac{1}{R}\sum_{r=1}^{R} p_r \leq 1 - \frac{1}{R}\sum_{r=1}^{R} a_r + \frac{\gamma}{3} + \frac{\gamma}{10} \leq \frac{1}{2} - \frac{17\gamma}{30}.$$

For a fixed soundness instance, the variables $X_1, \ldots, X_R$ are independent because independent internal randomness is used across the blocks. Hence Hoeffding's inequality gives

$$\Pr[\mathcal{B} = 1] \leq \exp\left(-2R\left(\frac{17\gamma}{30}\right)^2\right) \leq \exp\left(-\frac{R\gamma^2}{242}\right) \leq \frac{1}{3}.$$

For completeness, suppose all right-hand sides are independent uniform bits. Since the blocks are disjoint and use independent internal randomness, $X_1, \ldots, X_R$ are independent. Moreover, for every $r \in [R]$,

$$\Pr[X_r = 1] \geq \frac{1}{2} + \frac{\gamma}{22}.$$

Thus Hoeffding's inequality gives

$$\Pr[\mathcal{B} = 0] \leq \exp\left(-2R\left(\frac{\gamma}{22}\right)^2\right) = \exp\left(-\frac{R\gamma^2}{242}\right) \leq \frac{1}{3}.$$

Therefore $\mathcal{B}$ has soundness at most $1/3$ and completeness at least $2/3$. The total number of equations is

$$N = Rq = O\left(\frac{m_0}{\gamma^3}\right) = \mathrm{poly}(s, t, 1/\gamma).$$

Moreover, $N = Rq \geq R \geq 2/\gamma^2$, as required by Definition C.3.

Strong RRHS-refutation $\Longrightarrow$ Agnostic PAC-learning. This is the more interesting direction. We follow the strategy in [Vad17] — applying Yao's next-bit prediction [Vad17, Lemma 6] — but the analysis must account for the fact that the labels b_i need not come from any concept C_x.

Let $\mathcal{B}$ be the strong RRHS-refuter for $\mathscr{C}^*$ using $N = N(s,t,1/\gamma)$ equations with threshold γ. It suffices to consider $0 < \gamma < 1/4$. We construct an agnostic PAC-learner for $\mathscr{C}$ with error 3γ.

Fix any distribution $\mathcal{D}$ on $\{0,1\}^t \times \{0,1\}$ with $\max_x \Pr_{(y,b)\leftarrow\mathcal{D}}[C_x(y) = b] \geq 1/2 + 2\gamma$, and let $(Y_1, B_1), \ldots, (Y_N, B_N) \leftarrow \mathcal{D}$ be i.i.d. samples. We first observe that the refuter $\mathcal{B}$ can be turned into a distinguisher between real and random labels. Define

$$T\big((y_1,\ldots,y_N),\,(b_1,\ldots,b_N)\big) \;:=\; 1 - \mathcal{B}\big(1^s, 1^t, (y_1,b_1),\ldots,(y_N,b_N)\big),$$

so that T outputs 1 when $\mathcal{B}$ does not refute. Write $\mathbf{Y} = (Y_1,\ldots,Y_N)$, $\mathbf{B} = (B_1,\ldots,B_N)$, and let $\mathbf{R} = (R_1,\ldots,R_N)$ be independent uniform bits, also independent of $(\mathbf{Y},\mathbf{B})$.

Claim C.7. *T distinguishes $(\mathbf{Y},\mathbf{B})$ from $(\mathbf{Y},\mathbf{R})$ with advantage $\geq 1/4$.*

Proof of Claim C.7. When bits are real (i.e., $\mathbf{B}$): Let x^* achieve $\Pr_{(y,b)\leftarrow\mathcal{D}}[C_{x^*}(y) = b] \geq 1/2 + 2\gamma$. By Hoeffding's inequality and the lower bound on N in Definition C.3,

$$\Pr\left[\frac{1}{N}\big|\{i : C^*_{Y_i}(x^*) = B_i\}\big| \geq \frac{1}{2} + \gamma\right] \;\geq\; 1 - e^{-2N\gamma^2} \;\geq\; \frac{11}{12}.$$

When this event holds, the soundness of the strong RRHS-refuter gives $\Pr[\mathcal{B} = 1] \leq 1/3$, hence $\Pr[T = 1] \geq 2/3$. Therefore

$$\Pr\big[T(\mathbf{Y},\mathbf{B}) = 1\big] \;\geq\; \frac{11}{12}\cdot\frac{2}{3} \;=\; \frac{11}{18}.$$

When bits are random (i.e., $\mathbf{R}$): By completeness, $\Pr[\mathcal{B} = 1] \geq 2/3$, so $\Pr[T(\mathbf{Y},\mathbf{R}) = 1] \leq 1/3$.

Thus the distinguishing advantage is

$$\Pr\big[T(\mathbf{Y},\mathbf{B}) = 1\big] - \Pr\big[T(\mathbf{Y},\mathbf{R}) = 1\big] \;\geq\; \frac{11}{18} - \frac{1}{3} = \frac{5}{18} > \frac{1}{4}. \qquad \square$$

By Yao's lemma [Yao82] (see [Vad17, Lemma 6]), this distinguisher with advantage $\alpha \geq 1/4$ yields a next-bit predictor T' satisfying

$$\Pr\big[T'(\mathbf{Y}, B_1, \ldots, B_{I-1}) = B_I\big] \;\geq\; \frac{1 + \alpha/N}{2} \;\geq\; \frac{1}{2} + \frac{1}{8N},$$

where $I \leftarrow [N]$ is uniform. Concretely, T' on input $((y_1,\ldots,y_N), b_1, \ldots, b_{i-1})$ draws independent uniform bits $r_i, r_{i+1}, \ldots, r_N$, runs $T\big((y_1,\ldots,y_N),\,(b_1,\ldots,b_{i-1},r_i,\ldots,r_N)\big)$, and outputs r_i if T accepts and $1 - r_i$ otherwise. Crucially, T' is constructed purely from $\mathcal{B}$ and does not depend on the unknown distribution $\mathcal{D}$ or any particular concept C_{x^*}.

We now reinterpret T' as a weak agnostic prediction algorithm $\mathcal{A}_{\text{weak}}$ using $N-1$ labeled examples. Given $(y_1,b_1),\ldots,(y_{N-1},b_{N-1})$ and a challenge point y, choose $I \leftarrow [N]$ uniformly, insert y as the Ith point in $(y_1,\ldots,y_{N-1})$, and give T' the labels of the points preceding y. The labels of the remaining points are simply ignored. By exchangeability, the resulting sequence of N points is distributed exactly as $\mathbf{Y}$. Thus

$$\Pr\big[\mathcal{A}_{\text{weak}} \text{ predicts correctly}\big] \;\geq\; \frac{1}{2} + \frac{1}{8N}$$

whenever

$$\min_x \Pr_{(y,b)\leftarrow\mathcal{D}}[C_x(y) \neq b] \leq \frac{1}{2} - 2\gamma.$$

Set $\eta = 1/(8N)$; note that $0 < \eta/4 < 2\gamma < 1/2$ by the lower bound on N. In the terminology of [KMV08, Definition 4], $\mathcal{A}_{\text{weak}}$ is an $(N-1, 2\gamma, \eta)$-expected weak agnostic guesser. By [KMV08, Lemma 4], it yields an $(M, 2\gamma, \eta/4)$-weak agnostic learner (with the guesser's internal randomness fixed in the resulting hypotheses), where

$$M = \frac{N-1}{\eta} + \frac{32}{\eta^2} \log \frac{20}{\eta} = O(N^2 \log N).$$

Applying the agnostic boosting theorem [KMV08, Theorem 3] with accuracy parameter $\gamma/2$ and failure probability $\gamma/2$, we obtain a hypothesis h such that, except with probability at most $\gamma/2$,

$$\Pr_{(y,b)\leftarrow\mathcal{D}}[h(y) \neq b] \leq \min_{x} \Pr_{(y,b)\leftarrow\mathcal{D}}[C_x(y) \neq b] + \frac{5\gamma}{2}.$$

Using the trivial error bound 1 on the failure event, the resulting prediction algorithm has expected error at most the optimum plus 3γ. The sample complexity is polynomial in N; here we use again that $N = \Omega(1/\gamma^2)$. Taking $\gamma = \varepsilon/3$ proves the result. □

D Refuter Implies Learner

In this appendix, we justify our definition of nondeterministic PAC learning (Definition 5.1) by showing that our nondeterministic refuter notion (Definition 5.1) yields PAC learning with nondeterministic hypotheses in the sense of [Zha22]. The learner construction and hybrid argument below are the same as in [Zha22, Lemma 4.2.1], which reformulates [BFKL93, Theorem 7]. We include the proof to record the parameters in our setting; averaging directly over the random hybrid index gives confidence p/L and accuracy $1/2 + p/(2L)$. We begin by restating the two definitions for the reader's convenience.

Definition D.1 (PAC Learning with Nondeterministic Hypotheses [Zha22, Definition 4.1.1])**.** Let $\mathscr{C}$ be a class of Boolean functions and let $\mathcal{D} = (\mathcal{D}_n)_{n\geq 1}$ be a distribution ensemble, where $\mathcal{D}_n$ is supported on $\{0,1\}^n$. A randomized P/poly algorithm $\mathfrak{L}$ with oracle access to f and sample access to $\mathcal{D}_n$ is a *learner with nondeterministic hypothesis* for $(\mathscr{C}, \mathcal{D})$ at confidence δ and accuracy $1-\varepsilon$ if for every $f \in \mathscr{C}$ and all sufficiently large n:

- $\mathfrak{L}^f(1^n, w)$ outputs a string $\langle h \rangle$ which is a description of a nondeterministic (or co-nondeterministic) circuit $h_w : \{0,1\}^n \to \{0,1\}$ of size $\mathrm{poly}(n)$;
- with probability $\geq \delta$ over the random tape w, the produced hypothesis satisfies $\Pr_{x\leftarrow\mathcal{D}_n}[h_w(x) = f(x)] \geq 1-\varepsilon$.

We require $h_w = \mathfrak{L}^f(1^n, w)$ to be evaluable[10] by an algorithm of matching nondeterministic type (this is automatic when h_w is itself a circuit of polynomial size).

Definition D.2 (Nondeterministic PAC Learning; restatement of Definition 5.1)**.** Let $\mathscr{C}$ be a class of Boolean functions, $L \geq 1$, $p \in (0,1)$, and let $\mathcal{D} = (\mathcal{D}_n)_{n\geq 1}$ be a distribution ensemble, where $\mathcal{D}_n$ is a distribution over $\{0,1\}^n$. A nondeterministic algorithm $\mathcal{A}$ is said to *PAC-learn* $\mathscr{C}$ over input distribution $\mathcal{D}_n$ using L samples with success probability p if:

- *(Soundness)* for every $x_1, \ldots, x_L \in \{0,1\}^n$ and labels $b_1, \ldots, b_L \in \{0,1\}$, if there exists a function $C \in \mathscr{C}$ such that $C(x_i) = b_i$ for every $i \in [L]$, then $\mathcal{A}((x_1, b_1), \ldots, (x_L, b_L))$ rejects.

[10] For every f and w, $h_w = \mathfrak{L}^f(1^n, w)$ is P/poly-evaluable if there is another circuit family $\mathsf{Eval} \in \mathsf{P}/\mathrm{poly}$ such that for every possible output $h_w = \mathfrak{L}^f(1^n, w)$ of $\mathfrak{L}$ and every $x \in \{0,1\}^n$ given as input to Eval, Eval computes $h_w(x)$. When h_w is (co-)nondeterministic or randomized, Eval is allowed to be (co-)nondeterministic or randomized, respectively.

- *(Completeness)* If we sample i.i.d. $x_i \leftarrow \mathcal{D}_n$ and $b_i \leftarrow \{0,1\}$, then

$$\Pr[\mathcal{A}((x_1,b_1),\dots,(x_L,b_L)) \text{ accepts}] \geq p.$$

We also say that a nondeterministic algorithm $\mathcal{A}$ *distribution-freely* PAC-learns $\mathscr{C}$ with success probability p if it also satisfies the following stronger (Completeness) condition:

- *(Completeness')* For every $x_1,\dots,x_L \in \{0,1\}^n$, if we choose i.i.d. labels $b_1,\dots,b_L \leftarrow \{0,1\}$, then

$$\Pr[\mathcal{A}((x_1,b_1),\dots,(x_L,b_L)) \text{ accepts}] \geq p.$$

Theorem D.3 (Refuter Implies Learner, adapted from [BFKL93, Theorem 7] and [Zha22, Lemma 4.2.1])**.** *Let $L = L(n)$ be polynomially bounded and let $p = p(n) \in (0,1)$. Let $\mathcal{A}$ be a nondeterministic polynomial-time refuter parameterized by $(\mathscr{C}, \mathcal{D}_n, L, p)$ as in Definition D.2. Then there is a randomized P/poly learner $\mathfrak{L}$ for $(\mathscr{C},\mathcal{D})$ with confidence $\delta \geq p/L$ and accuracy $1-\varepsilon \geq 1/2 + p/(2L)$, where each output hypothesis is either a nondeterministic or co-nondeterministic circuit, and $\mathfrak{L}$ makes at most $L-1$ queries to f.*

Proof. We will construct the learner with access to the refuter.

Algorithm D.1: The randomized oracle learner $\mathfrak{L}^f(1^n)$

Input: 1^n and oracle access to f and sample access to $\mathcal{D}_n$
Output: A hypothesis h

1 Sample $i \leftarrow [L]$ uniformly at random
2 Sample $r_1,\dots,r_L \leftarrow \{0,1\}$ independently and uniformly
3 Sample $x_1,\dots,x_{i-1},x_{i+1},\dots,x_L \leftarrow \mathcal{D}_n$ independently
4 Query the oracle to obtain $f(x_1),\dots,f(x_{i-1})$
5 **return** the hypothesis h defined, for each $x \in \{0,1\}^n$, by

$$h(x) = r_i \oplus \mathcal{A}\big((x_1,f(x_1)),\dots,(x_{i-1},f(x_{i-1})),(x,r_i),\\ (x_{i+1},r_{i+1}),\dots,(x_L,r_L)\big).$$

Note that for nondeterministic $\mathcal{A}$, "$\mathcal{A}(\cdot)=1$" is an $\exists$-statement over the witness.

- If $r_i = 0$, then $h(x) = \mathcal{A}(\cdots)$, so $h(x)=1$ iff $\mathcal{A}$ has an accepting path, i.e., h is nondeterministic.
- If $r_i = 1$, then $h(x) = 1 \oplus \mathcal{A}(\cdots)$, so $h(x)=1$ iff $\mathcal{A}$ has no accepting path, i.e., h is co-nondeterministic.

Since L is polynomially bounded and $\mathcal{A}$ runs in nondeterministic polynomial time, hardwiring the sampled points, labels, and oracle answers into a circuit for $\mathcal{A}$ gives a polynomial-size hypothesis, and its description can be produced by a randomized P/poly learner.

Now we define $(L+1)$ hybrid distributions. For $j \in \{0,1,\dots,L\}$, define the random variable

$$H_j := ((x_1,f(x_1)),\dots,(x_j,f(x_j)),(x_{j+1},r_{j+1}),\dots,(x_L,r_L)),$$

where $x_k \leftarrow \mathcal{D}_n$ i.i.d. and $r_k \leftarrow \{0,1\}$ uniform i.i.d.

Let

$$\alpha_j := \Pr[\mathcal{A}(H_j) \text{ accepts}].$$

Note that for H_0, all labels are uniformly at random and are independent of x_k's, so $\alpha_0 \geq p$ by completeness. Also, since H_L is realizable by $f \in \mathscr{C}$, $\alpha_L = 0$ by soundness. Hence, by telescoping, we have

$$\sum_{i=0}^{L-1}(\alpha_i - \alpha_{i+1}) = \alpha_0 - \alpha_L \geq p.$$

We now proceed to bound the accuracy of the hypothesis at a fixed i.

Fix $i \in [L]$ chosen by the learner. Consider the experiment in which $x \leftarrow \mathcal{D}_n$ is the evaluation point and r_i, x_k $(k \neq i)$, r_k $(k > i)$ are sampled as in the construction. The string fed to $\mathcal{A}$ when computing $h(x)$ is

$$Z := ((x_1, f(x_1)), \ldots, (x_{i-1}, f(x_{i-1})), (x, r_i), (x_{i+1}, r_{i+1}), \ldots, (x_L, r_L)).$$

Conditioning on $r_i = f(x)$, Z is distributed as H_i (recall that $x \leftarrow \mathcal{D}_n$ in our experiment). Hence

$$\Pr[\mathcal{A}(Z) = 1 \mid r_i = f(x)] = \alpha_i.$$

On the other hand, let $\widetilde{H}_i$ denote the distribution of Z conditioning on $r_i \neq f(x)$. That is, $\widetilde{H}_i$ is H_i with the i-th label flipped. The distribution of H_{i-1} is the equal mixture of H_i and $\widetilde{H}_i$. Hence, we have

$$\alpha_{i-1} = \alpha_i/2 + \Pr_{\widetilde{H}_i}[\mathcal{A} = 1]/2,$$

or in other words,

$$\Pr_{\widetilde{H}_i}[\mathcal{A} = 1] = 2\alpha_{i-1} - \alpha_i.$$

Since $h(x) = f(x)$ iff $\mathcal{A}(Z) = r_i \oplus f(x)$, we have

$$\begin{aligned}\Pr[h(x) = f(x)] &= \Pr[\mathcal{A}(Z) = 0 \mid r_i = f(x)]/2 + \Pr[\mathcal{A}(Z) = 1 \mid r_i \neq f(x)]/2 \\ &= (1 - \alpha_i)/2 + (2\alpha_{i-1} - \alpha_i)/2 \\ &= 1/2 + (\alpha_{i-1} - \alpha_i).\end{aligned}$$

Let w denote all the randomness used by the learner, including the choice of i. Averaging also over an independent $x \leftarrow \mathcal{D}_n$, we obtain

$$\mathop{\mathbb{E}}_{w,x}[\mathbf{1}[h(x) = f(x)]] = \frac{1}{L}\sum_{i=1}^{L}\left(\frac{1}{2} + \alpha_{i-1} - \alpha_i\right) = \frac{1}{2} + \frac{\alpha_0 - \alpha_L}{L} \geq \frac{1}{2} + \frac{p}{L}.$$

Let $X = X(w) := \Pr_{x \leftarrow \mathcal{D}_n}[h_w(x) = f(x)]$ be a random variable in the learner's randomness w. We have $\mathbb{E}[X] \geq 1/2 + p/L$. Writing

$$\delta := \Pr\left[X \geq \frac{1}{2} + \frac{p}{2L}\right],$$

since $0 \leq X \leq 1$, we have

$$\frac{1}{2} + \frac{p}{L} \leq \mathbb{E}[X] \leq \delta \cdot 1 + (1 - \delta)\left(\frac{1}{2} + \frac{p}{2L}\right).$$

Therefore,

$$\delta \geq \frac{p/(2L)}{1/2 - p/(2L)} \geq \frac{p}{L}.$$

Hence, with probability $\geq p/L$ over the learner's randomness, the produced hypothesis agrees with f on a $(1/2 + p/(2L))$ fraction of inputs. □

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