Hardness of Range Avoidance and Proof Complexity Generators from Demi-Bits

Hanlin Ren
Institute for Advanced Study, Princeton, NJ, USA

Yichuan Wang
Department of EECS, University of California, Berkeley, CA, USA

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

Abstract

Given a circuit $G : \{0,1\}^n \to \{0,1\}^m$ with $m > n$, the *range avoidance* problem (Avoid) asks to output a string $y \in \{0,1\}^m$ that is not in the range of G. Besides its profound connection to circuit complexity and explicit construction problems, this problem is also related to the existence of *proof complexity generators* – circuits $G : \{0,1\}^n \to \{0,1\}^m$ where $m > n$ but for every $y \in \{0,1\}^m$, it is infeasible to prove the statement "$y \notin \mathrm{Range}(G)$" in a given propositional proof system.

This paper connects these two problems with the existence of *demi-bits generators*, a fundamental cryptographic primitive against nondeterministic adversaries introduced by Rudich (RANDOM '97).

- We show that the existence of demi-bits generators implies Avoid is hard for nondeterministic algorithms. This resolves an open problem raised by Chen and Li (STOC '24). Furthermore, assuming the demi-hardness of certain LPN-style generators or Goldreich's PRG, we prove the hardness of Avoid even when the instances are constant-degree polynomials over $\mathbb{F}_2$.
- We show that the dual weak pigeonhole principle is unprovable in Cook's theory PV_1 under the existence of demi-bits generators secure against $\mathbf{AM}/_{O(1)}$, thereby separating Jeřábek's theory APC_1 from PV_1. Previously, Ilango, Li, and Williams (STOC '23) obtained the same separation under different (and arguably stronger) cryptographic assumptions.
- We transform demi-bits generators to proof complexity generators that are *pseudo-surjective* in certain parameter regime. Pseudo-surjectivity is the strongest form of hardness considered in the literature for proof complexity generators.

Our constructions are inspired by the recent breakthroughs on the hardness of Avoid by Ilango, Li, and Williams (STOC '23) and Chen and Li (STOC '24). We use *randomness extractors* to significantly simplify the construction and the proof.

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

Keywords and phrases Range Avoidance, Proof Complexity Generators

Digital Object Identifier 10.4230/LIPIcs.ITCS.2026.111

Related Version *Full Version*: `https://arxiv.org/abs/2511.14061`

Acknowledgements We thank Yilei Chen for helpful discussions regarding the LPN assumption, Xin Li for helpful discussions about extractors, and Rahul Ilango for sending us a draft version of [32]. We are also grateful to Erfan Khaniki and Iddo Tzameret for helpful suggestions that improved the presentation of this paper.

1 Introduction

This paper makes progress on the hardness of the *range avoidance problem* and the existence of *proof complexity generators*. We begin with a brief overview of these two lines of research.

1.1 Range Avoidance

The *range avoidance* problem (Avoid) is a total search problem introduced by [43, 45, 72]. Given a Boolean circuit $G : \{0,1\}^n \to \{0,1\}^m$ with $m > n$ (usually we also require $m \leq \text{poly}(n)$), the goal is to find a string $y \in \{0,1\}^m$ such that $y \notin \text{Range}(G)$. This problem has attracted considerable interest due to its connection to central problems in complexity theory such as explicit constructions [16, 26, 30, 45, 72] and circuit lower bounds [13, 45, 47, 61]. We refer the reader to [46] for a comprehensive survey on the range avoidance problem.

The range avoidance problem admits a trivial randomized algorithm: simply output a uniformly random m-bit string, which will lie outside the range of G with high probability. On the other hand, deterministic algorithms for Avoid would imply breakthroughs in explicit constructions and circuit lower bounds [26, 30, 45, 72]. Since such breakthroughs are widely believed to be *true* (albeit *difficult to prove*), the aforementioned results only suggest that deterministic algorithms for Avoid would be difficult to obtain *unconditionally*, rather than that such algorithms are *unlikely to exist*. This raises a natural question: Is there a deterministic algorithm for Avoid?

Perhaps surprisingly, recent results suggested that the answer is likely *no* under plausible cryptographic assumptions. Ilango, Li, and Williams [33] showed that Avoid is hard for deterministic algorithms assuming the existence of subexponentially secure indistinguishability obfuscation ($i\mathcal{O}$) and that $\mathbf{NP} \neq \mathbf{coNP}$. Chen and Li [17] extended this result and showed that Avoid is hard even for *nondeterministic* algorithms, under certain assumptions regarding the nondeterministic hardness of LWE (Learning with Errors) or LPN (Learning Parity with Noise). In addition to providing compelling evidence for the hardness of Avoid, these results establish a strong separation between deterministic and randomized algorithms (recall that there exists a trivial randomized algorithm for Avoid).

The hardness results in [17, 33] open up several exciting research directions:

1. **Can the hardness of Avoid be based on weaker (or alternative) assumptions?** The assumptions used in prior work, i.e., $i\mathcal{O}$ [33] and public-key encryption [17], belong to Cryptomania in the terminology of Impagliazzo's worlds [34]. Can we base the hardness of range avoidance on assumptions of a "Minicrypt" flavor, such as one-way functions or pseudorandom generators? Additionally, both [33] and [17] rely on *subexponential* indistinguishability assumptions[1]. Are such subexponential assumptions necessary?
2. **Can we obtain hardness of Avoid for instances computed by restricted circuits?** Previously, under assumptions related to LWE, Chen and Li [17] showed that Avoid remains hard even when each output bit of G is computed by a so-called "$\mathsf{DOR} \circ \mathsf{MAJ} \circ \mathsf{AND}_{O(\log n)}$ circuit". No such results were known for other restricted circuit classes. The related *remote point* problem has been shown to be hard under LPN-style assumptions for $\mathsf{XOR} \circ \mathsf{AND}_{O(\log n)}$ circuits (i.e., $O(\log n)$-degree polynomials over $\mathbb{F}_2$) [17].

This paper makes progress on both fronts. We show that Avoid is hard for nondeterministic algorithms under the existence of *demi-bits generators* with sufficient stretch[2] [73]. A formal definition of demi-bits generators is deferred to Subsection 2.1; in Appendix B

[1] More precisely, the assumptions in [17, 33] assert subexponential indistinguishability against polynomial-time adversaries. This level of security is referred to as "JLS-security" in [33], where "JLS" comes from the strengths of the "well-founded" assumptions used to construct $i\mathcal{O}$ in [37].

[2] The stretchability of generic demi-bits generators is only partially understood. Recent work of Tzameret and Zhang [77] shows that demi-bits generator with 1-bit stretch $G : \{0,1\}^n \to \{0,1\}^{n+1}$ implies those with a *sublinear* bits of stretch $G' : \{0,1\}^n \to \{0,1\}^{n+n^c}$ for any constant $0 < c < 1$. This is the first proof that generic demi-bits generators are stretchable at all, but it still falls short of the *linear* or *polynomial* stretch assumed in our hypothesis.

of the full version of this paper, we also provide some candidate constructions supporting their existence. For the purpose of this introduction, it suffices to keep in mind that demi-bits generators are a version of cryptographic pseudorandom generators secure against nondeterministic adversaries.

We highlight three key features of our results here:

1. **Minicrypt-style assumptions against nondeterministic adversaries.**
   Roughly speaking, demi-bits generators are (cryptographic) pseudorandom generators secure against nondeterministic adversaries[3]. They are arguably a natural "Minicrypt" analog of pseudorandom generators in the context of cryptography against nondeterministic adversaries. Moreover, our results only rely on the *super-polynomial* hardness of these demi-bits generators, thereby completely getting rid of the subexponential (or "JLS"-style) assumptions used in prior work.
2. **Hardness for restricted circuit classes.**
   Under the assumption that certain concrete demi-bits generators are secure (e.g., those based on LPN or Goldreich's PRG), we show that the range avoidance problem remains hard for nondeterministic algorithms even when the underlying circuits belong to $\mathsf{XOR} \circ \mathsf{AND}_{O(1)}$, i.e., constant-degree polynomials over $\mathbb{F}_2$.
3. **Simplicity of the proof for hardness of range avoidance.**
   In contrast to [17, 33], which rely on sophisticated and delicate adaptations of high-end cryptographic assumptions, our proof of the hardness of range avoidance (Theorem 1) is based solely on the most elementary pseudorandom constructions together with the existence of demi-bits. This approach distills the arguments of [17, 33] to their essence, yielding a proof that is both conceptually cleaner and technically simpler. In fact, the proof of Theorem 1 fits in just half a page. Moreover, by isolating the core ingredients, this simplified framework opens the door to potential extensions and generalizations that may be harder to see in the more cryptographically heavy approaches.

1.2 Proof Complexity Generators

Let $G : \{0,1\}^n \to \{0,1\}^m$ be a Boolean circuit where $m > n$, and $\mathcal{P}$ be a propositional proof system. We say that G is a (secure) *proof complexity generator* [3, 49] against $\mathcal{P}$ if, for every string $y \in \{0,1\}^m$, the (properly encoded) statement "$y \notin \mathrm{Range}(G)$" does not admit short proofs in $\mathcal{P}$.[4] A comprehensive survey about proof complexity generators can be found in [57].

The study of proof complexity generators is motivated by at least the following themes:

1. **Pseudorandomness in proof complexity [3].**
   A standard *pseudorandom generator* (PRG) $G : \{0,1\}^n \to \{0,1\}^m$ [78] fools a (polynomial-time) algorithm $\mathcal{D}$ if $\mathcal{D}$ cannot distinguish the outputs of G from truly random m-bit strings; that is, $\mathcal{D}(\mathcal{U}_m) \approx \mathcal{D}(G(\mathcal{U}_n))$, where $\mathcal{U}_\ell$ denotes the uniform distribution over ℓ-bit strings. Analogously, one can say that G fools a propositional proof system $\mathcal{P}$ if $\mathcal{P}$ cannot distinguish between the outputs of G and truly random m-bit strings, and a natural way of formalizing this is to say that $\mathcal{P}$ cannot efficiently prove any string outside the range of G.

[3] In fact, Rudich [73] introduced two ways to define pseudorandomness against nondeterministic adversaries: super-bits and demi-bits. Demi-bits are weaker than super-bits.

[4] If y is in fact in the range of G, then "$y \notin \mathrm{Range}(G)$" is a false statement and hence has no proof in any sound proof system. Therefore, this requirement is equivalent to that, for every $y \notin \mathrm{Range}(G)$, the tautology "$y \notin \mathrm{Range}(G)$" is hard to prove in $\mathcal{P}$.

Following the idea of pseudorandomness in proof complexity, subsequent works [42, 53, 54, 65, 71] studied the hardness of the *Nisan–Wigderson* generator ([63]) as a proof complexity generator in various settings. An influential conjecture of Razborov asserts that the Nisan–Wigderson generator based on any "sufficiently hard" function in $\mathbf{NP} \cap \mathbf{coNP}$ is a proof complexity generator against Extended Frege [71, Conjecture 2]; that is, computational hardness can be transformed into proof complexity pseudorandomness.

2. **Candidate hard tautologies for strong proof systems.**
   There are two difficulties in proving lower bounds for strong proof systems such as Frege and Extended Frege: the lack of techniques and the lack of candidate hard tautologies. The latter problem was highlighted by Bonet, Buss, and Pitassi [9], who demonstrated that many combinatorial tautologies can be proved efficiently in Frege, hence disqualifying them as hard candidates. This issue has been further discussed in [42, 50, 53, 74].
   Tautologies from proof complexity generators are among the few natural candidates that appear hard for strong proof systems. It seems plausible that for some mapping $G : \{0,1\}^n \to \{0,1\}^{n+1}$ and some (or even *every*) $y \in \{0,1\}^{n+1} \setminus \mathrm{Range}(G)$, the natural CNF encoding of the tautology "$y \notin \mathrm{Range}(G)$" requires super-polynomially long Extended Frege proofs.
3. **Unprovability of circuit lower bounds.**
   Given our very limited progress in circuit complexity, it is tempting to conjecture that circuit lower bounds are hard to prove in formal proof systems. For a Boolean function $f : \{0,1\}^n \to \{0,1\}$ and a size parameter s, one can write down a propositional formula $\mathsf{lb}(f, s)$ (of size $2^{O(n)}$) asserting that no circuit of size at most s computes f. The proof complexity of such formulas have been studied extensively [53, 66–71, 74], due to its implications for the metamathematics of complexity theory.
   Consider the *truth table generator* $\mathsf{TT} : \{0,1\}^{\mathrm{poly}(s)} \to \{0,1\}^{2^n}$, which maps a size-$s$ circuit C to its 2^n-bit truth table. By definition, TT is a proof complexity generator against a proof system $\mathcal{P}$ if and only if $\mathcal{P}$ cannot efficiently prove any circuit lower bound $\mathsf{lb}(f, s)$. Krajíček [53] introduced the notion of *pseudo-surjectivity* and showed that TT is the hardest pseudo-surjective generator: The existence of any generator pseudo-surjective against $\mathcal{P}$ implies that TT is pseudo-surjective against $\mathcal{P}$ (and thus that $\mathcal{P}$ cannot prove circuit lower bounds). Razborov [71] further showed unprovability of circuit lower bounds in the proof system $\mathrm{Res}(\varepsilon \log\log N)$ by exhibiting a proof complexity generator that is iterable for this system.[5]

Krajíček [51, 53, 56] conjectured that there exists a proof complexity generator that is secure against every propositional proof system. One could also consider a slightly weaker conjecture that for every propositional proof system $\mathcal{P}$, there is a proof complexity generator $C_{\mathcal{P}}$ (possibly depending on $\mathcal{P}$) that is hard against $\mathcal{P}$. At first glance, these conjectures may appear unrelated to standard hardness assumptions in complexity theory or cryptography, as proof complexity generators require "$y \notin \mathrm{Range}(G)$" to be hard to prove for *every* y (i.e., the *best-case* y), while complexity-theoretic or cryptographic hardness assumptions tend to be either *worst-case* or *average-case*. We elaborate on the notion of "best-case" proof complexity in Subsection 1.4.

[5] Iterability is a weaker notion than pseudo-surjectivity. Krajíček [53] also showed that TT is the "hardest" iterable generator.

In this paper, we give strong evidence for the weaker conjecture by showing that it follows from the existence of demi-bits generators (with sufficiently large stretch) [73]. The latter is a natural and fundamental conjecture in the study of *cryptography against nondeterministic adversaries*. Furthermore, we show that our generators are even pseudo-surjective under certain regimes.[6]

1.3 Our Results

Hardness of range avoidance. Our main result is that the existence of demi-bits generators implies that Avoid $\notin$ **SearchNP**, i.e., Avoid is hard for nondeterministic search algorithms.

▶ **Theorem 1** (Main). *If there exists a demi-bits generator* $G : \{0,1\}^n \to \{0,1\}^{10n}$*, then* Avoid $\notin$ **SearchNP**.

In fact, we show that composing the demi-bits generator with a hash function in some pairwise independent hash family would yield a hard instance for Avoid. In its full generality, our arguments hold for arbitrary *strong seeded extractors*, and the theorem below follows from the leftover hash lemma [35], which guarantees that pairwise independent hash families are such extractors; see Theorem 1 for details.

We present the version using pairwise independent hash families here due to its elegance:

▶ **Theorem 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 a nondeterministic polynomial-time algorithm. If* $N > 10m$ *and* $m > n$*, then there exists* $h \in \mathcal{H}$ *such that* $\mathcal{A}$ *fails to solve the range avoidance problem on the input* $h \circ G$*.*

As discussed before, this result improves upon [17, 33] in several key aspects. First, we only require super-polynomial hardness of the demi-bits generators, thereby completely eliminating the subexponential- or JLS-hardness assumptions. Second, our assumptions are solely based on the existence of demi-bits generators, a primitive arguably situated within "nondeterministic Minicrypt." Finally, by instantiating the extractors with pairwise independent hash functions computable by linear transformations over $\mathbb{F}_2$ and using demi-bits generators computable by constant-degree $\mathbb{F}_2$-polynomials, we establish the hardness of Avoid even for circuits where each output bit is computable in constant $\mathbb{F}_2$-degree (i.e., $\mathsf{XOR} \circ \mathsf{AND}_{O(1)}$ circuits):

▶ **Corollary 3** (Informal). *Assuming the existence of demi-bits generators computable in* $\mathsf{XOR} \circ \mathsf{AND}_{O(1)}$ *(Assumption 2.3), the range avoidance problem for* $\mathsf{XOR} \circ \mathsf{AND}_{O(1)}$ *circuits is not in* **SearchNP**.

Proof complexity generators. Building on this result, we show that for any fixed propositional proof system $\mathcal{P}$ closed under certain reductions, demi-bits generators for $\mathcal{P}$ imply proof complexity generators for $\mathcal{P}$. In particular, the existence of demi-bits generators secure against $\mathbf{NP}/_{\mathrm{poly}}$ implies the weaker version of Krajíček's conjecture, providing strong evidence that the latter conjecture is true.

Moreover, this result suggests a new approach for constructing proof complexity generators for concrete proof systems closed under certain reductions, such as $\mathsf{Res}[\oplus]$: it suffices to construct demi-bits generators secure against *the same proof system*.

[6] The parameters of our pseudo-surjectivity results *fall just short* of those required to apply Krajíček's result [53], hence they do not imply the hardness of the truth table generator. This limitation is inherent; we discuss this issue in more details after presenting Theorem 7.

▶ **Theorem 4.** *Let $\mathcal{P}$ be a proof system closed under "simple parity reductions" (Definition 3). If there exists a demi-bits generator $G : \{0,1\}^n \to \{0,1\}^{10n}$ secure against $\mathcal{P}$, then there is a (non-uniform) proof complexity generator secure against $\mathcal{P}$.*

Unprovability of dwPHP(PV) in PV from demi-bits. A central goal in bounded arithmetic is to delineate the logical power required to formalize reasoning about computational complexity. Two well-studied theories in this context are Cook's theory PV_1 [20], which corresponds to reasoning in deterministic polynomial time, and Jeřábek's theory APC_1 [38, 41][7], which extends PV_1 by adding the *Dual Weak Pigeonhole Principle* for polynomial-time functions (dwPHP(PV)), and captures aspects of randomized polynomial-time reasoning.

Despite decades of interest, it has remained open whether APC_1 and PV_1 are actually distinct theories, i.e., whether dwPHP(PV) is unprovable in PV_1. Recently, Ilango, Li, and Williams [33] provided the first evidence separating the two: they showed that dwPHP(PV) is unprovable in PV_1 under the assumptions that indistinguishability obfuscation ($i\mathcal{O}$) with JLS-security exists and that **coNP** is not infinitely often in **AM**. We remark that the same separation was also shown by Krajíček [55], albeit under an assumption that is regarded as "unlikely" (**P** admits fixed-polynomial size circuits). In this work, we establish the same separation assuming the existence of demi-bits generators against $\mathbf{AM}/_{O(1)}$.

▶ **Theorem 5.** *Assuming there exists a demi-bits generator $G : \{0,1\}^n \to \{0,1\}^{\omega(n)}$ secure against $\mathbf{AM}/_{O(1)}$, the Dual Weak Pigeonhole Principle for polynomial-time functions* (dwPHP(PV)) *is not provable in* PV*. (In particular, APC_1 is a strict extension of PV_1.)*

The only property of PV_1 used in our argument is the KPT witnessing theorem [59], which states that if PV_1 proves the dual weak pigeonhole principle for polynomial-time functions, then there exists a deterministic polynomial-time Student that wins the Student-Teacher game for solving Avoid in $O(1)$ rounds (see Subsection 2.5). Our separation result in Theorem 5 follows by showing that no such Student algorithm exists. Moreover, for any parameter $k = k(n)$, assuming the existence of demi-bits generators secure against $\mathbf{AM}/_{O(\log k)}$, we further rule out deterministic polynomial-time Students that wins the Student-Teacher game within k rounds.

▶ **Theorem 6.** *Let $m = m(n) > n$ and $k = k(n)$ be parameters. If there exists a demi-bits generator $G : \{0,1\}^n \to \{0,1\}^{100km}$ secure against $\mathbf{AM}/_{O(\log k)}$, then there is no polynomial-time deterministic algorithm for* Avoid *on circuits with n inputs and m outputs using k circuit-inversion oracle queries.*

Pseudo-surjective generators. Assuming demi-bits generators against $\mathbf{NP}/_{\mathrm{poly}}$, we show that our proof complexity generators are even pseudo-surjective against every proof system. (The precise definition of *k-round* pseudo-surjectivity is presented in Definition 14.)

▶ **Theorem 7.** *Let $m = m(n) > n$ and $k = k(n)$ be parameters. If there exists a demi-bits generator $G : \{0,1\}^n \to \{0,1\}^{100km}$ secure against $\mathbf{NP}/_{\mathrm{poly}}$, then for every non-uniform propositional proof system $\mathcal{P}$, there is a non-uniform proof complexity generator $G_{\mathcal{P},k} : \{0,1\}^n \to \{0,1\}^m$ that is k-round pseudo-surjective against $\mathcal{P}$.*

Krajíček [53] proved that, under appropriate parameter settings, the existence of pseudo-surjective proof complexity generators is equivalent to the pseudo-surjectivity of the truth table generator. As a corollary, the existence of pseudo-surjective generators against a proof system $\mathcal{P}$ implies that *every* circuit lower bound is hard to prove in $\mathcal{P}$.

[7] Note that the terminology APC_1 was first used in [11].

However, the parameters in our Theorem 7 *fall short* of applying Krajíček's result and therefore do *not* imply the pseudo-surjectivity of the truth table generator. Specifically, to apply Krajíček's results, we need a proof complexity generator that is computable by circuits of size s and is k-round pseudo-surjective for some $k \gg s$; see the proof of [53, Theorem 4.2] for detailed discussions. In contrast, Theorem 7 guarantees a generator computable by circuits of size $\mathrm{poly}(n, k)$ that is k-round pseudo-surjective, which is in the regime where $k < s$ and thus outside the reach of Krajíček's equivalence.

This limitation is inherent to the generality of our result: we construct generators secure against *all* proof systems, whereas under the assumption $\mathbf{E} \not\subseteq \mathbf{SIZE}[2^{o(n)}]$, there exists a proof system that *can* prove circuit lower bounds (e.g., by simply hardwiring an axiom that certain $\mathbf{E}$-complete language has exponential circuit complexity). Thus, under this standard hardness assumption, it is provably impossible to extend our results to the regime where $k \gg s$ and thereby obtain pseudo-surjectivity of the truth table generator. It remains an intriguing open question whether our approach can be refined to construct proof complexity generators of size s that are k-round pseudo-surjective against specific systems such as Extended Frege, for some $k \gg s$. Such a result would imply that Extended Frege cannot prove *any* circuit lower bounds.

Finally, we comment on the strength of the adversaries required in our assumptions on demi-bits generators. In the main theorem (Theorem 1), a **SearchNP** algorithm for Avoid is transformed into a nondeterministic adversary that breaks the demi-bits generator. Since **SearchNP** is a uniform class, it suffices to assume that the generator is secure against uniform nondeterministic adversaries. In contrast, Theorem 6 requires the generator to be secure against $\mathbf{AM}/_{O(\log k(n))}$ adversaries. This is because the adversary invokes the Goldwasser–Sipser protocol [29] which is in **AM**, and needs to hardwire the index of the circuit-inversion query that succeeds with good probability. In Theorem 7, we require security against $\mathbf{NP}/_{\mathrm{poly}}$ adversaries, as our adversary needs to hardwire a "good" sequence of teacher responses in the student-teacher game, thus is highly non-uniform.

1.4 Perspective: Average-Case to Best-Case Reductions in Proof Complexity

Theoretical computer science has traditionally focused on the *worst-case* complexity of problems: An algorithm $\mathcal{A}$ solves a problem if $\mathcal{A}(x)$ succeeds on every input x. Motivated by practical heuristics (where worst-case analysis tends to be overly pessimistic) and cryptography (where worst-case hardness is not sufficient for security), *average-case* complexity has emerged as an important research direction [7, 34]. In this setting, fixing a distribution $\mathcal{D}$ over inputs, an algorithm $\mathcal{A}$ solves a problem if $\mathcal{A}(x)$ succeeds with good probability over $x \sim \mathcal{D}$. Recently, average-case complexity has received attention in proof complexity as well: For example, [19, 24, 64] proved proof complexity lower bounds for Clique and Coloring for *random* graphs.

An important topic in average-case complexity is *worst-case to average-case* reductions: reductions showing that if a problem L is hard in the worst-case then a related problem L' is hard on average. Worst-case to average-case reductions are known for the Permanent [12], Discrete Logarithm [6], Quadratic Residuosity [28], certain lattice problems [2], and more recently for problems in meta-complexity [31]. On the other hand, "black-box" worst-case to average-case reductions are unlikely to exist for **NP**-complete problems [8, 25].

In contrast, the notion of *best-case* complexity has received far less attention. Perhaps one reason is that this notion is often trivial in standard computational complexity: for every language L, either the all-zero function or the all-one function can decide L on the "best"

input.[8] However, in proof complexity, best-case hardness is a meaningful and natural notion, as illustrated by proof complexity generators, which are stretching functions G such that the statement "$y \notin \mathrm{Range}(G)$" is hard to prove even for the *best* choice of y.

In this context, our results can be interpreted as an *average-case to best-case* reduction in proof complexity. Indeed, Theorem 4 transforms demi-bits generators, where statements of the form "$y \notin \mathrm{Range}(G)$" is hard to prove for an *average-case* y, to proof complexity generators, where such statements are hard for a *best-case* y.

We find the existence of such average-case to best-case reductions quite surprising. Our arguments crucially exploit the power of nondeterministic computation, and the phenomenon of average-case to best-case reductions seems unique to the setting of proof complexity and hardness against nondeterministic algorithms. We believe that further exploring the scope and limitations of average-case to best-case reductions is a promising direction for future research.

We remark that there are also worst-case to best-case reductions in proof complexity. Krajíček [52] constructed a proof complexity generator whose hardness can be based on the hardness of the pigeonhole principle, thereby reducing the best-case hardness of an entire family of tautologies to that of a single tautology. Inspired by this example, Garlik, Gryaznov, Ren, and Tzameret [27] recently showed a worst-case to best-case reduction for the *rank principles*. Let "$\mathrm{rank}(A) > r$" denote the collection of polynomial equations expressing that the rank of an $n \times n$ matrix A is greater than r. If a proof system (closed under certain algebraic reductions) cannot prove "$\mathrm{rank}(I_n) > r$" where I_n is the $n \times n$ identity matrix, then it also cannot prove "$\mathrm{rank}(A) > r$" for *every* $n \times n$ matrix A.

1.5 Concurrent Works

In an independent and concurrent work, Ilango [32] presented a different proof of Theorem 1 and Theorem 4. Ilango's proof implies the *average-case* hardness of Avoid under a natural distribution (namely, it is hard for errorless nondeterministic heuristic algorithms to output a truth table with high O-oracle circuit complexity given the truth table of a random oracle O). Based on this result, Ilango constructed a (non-uniform) non-interactive proof system that "looks" zero knowledge to every proof system. However, Ilango did not show the pseudo-surjectivity of his generators.

2 Preliminaries

2.1 Demi-Bits Generators

▶ **Definition 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 $\mathbf{NP}/_{\mathrm{poly}}$ adversary* Adv *of size s such that*

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

This paper requires demi-bits generators with large stretch and computable with small circuit complexity. In particular, we need the following assumptions:

[8] One notable exception appears in recent derandomization results [15] based on *almost-all-input* hardness assumptions. In particular, it was shown that pr$\mathbf{P} =$ pr$\mathbf{BPP}$ follows from the existence of depth-efficient multi-output functions with high best-case complexity against randomized algorithms.

▶ **Assumption 2.2** (Demi-Bits Generators with Polynomial Stretch). *For every constant $c \geq 1$, there exists a family of demi-bits generators $\{g_n : \{0,1\}^n \to \{0,1\}^{n^c}\}$ secure against $\mathbf{NP}/_{\mathrm{poly}}$.*

▶ **Assumption 2.3** (Demi-Bits Generators with $n^{1+\varepsilon}$ Stretch in Constant Degree). *There exist constants $\varepsilon > 0$, $d \geq 2$, and a (non-uniformly computable) family of demi-bits generators $\{g_n : \{0,1\}^n \to \{0,1\}^{n^{1+\varepsilon}}\}$ secure against $\mathbf{NP}/_{\mathrm{poly}}$, such that each output bit of g_n is computable by a degree-d polynomial over $\mathbb{F}_2$ (i.e., an $\mathsf{XOR} \circ \mathsf{AND}_d$ circuit).*

Our main hardness results for Avoid will be based on Assumption 2.2; we also need Assumption 2.3 to obtain hardness results for constant-degree Avoid. In Appendix B of the full version, we justify these assumptions and provide some candidate constructions.

2.2 Arthur–Merlin Protocols

An *Arthur–Merlin* protocol [5] 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. The round-collapse theorem of Babai [5] implies that every language in **AM** actually has an Arthur–Merlin protocol with two rounds: Verifier sends the first message, Prover sends a proof, and Verifier decides whether $x \in L$. Hence, **AM** can be seen as a randomized version of **NP**; indeed, one can prove that $\mathbf{AM} = \mathbf{NP}$ under circuit lower bound assumptions [44, 62, 75, 76].

Goldwasser–Sipser set lower bound protocol. We need the following well-known **AM** protocol for proving lower bounds on the size of efficiently recognizable sets.

▶ **Lemma 4** ([29], also see [4, section 8.4]). *There is an Arthur–Merlin protocol such that the following holds. Suppose that both Prover and Verifier receive a nondeterministic circuit $C : \{0,1\}^n \to \{0,1\}$ and a number $s \leq 2^n$. Let $S = \{x \in \{0,1\}^n : C(x) = 1\}$.*

- ***Specification:*** *The protocol is a two-round public-coin protocol, in which the Verifier first sends a random string r and receives a message m, then deterministically decides whether to accept based on r and m.*
- ***Completeness:*** *If $|S| \geq s$, then w.p. $\geq 2/3$ over r, there exists a proof m that makes the Verifier accept.*
- ***Soundness:*** *If $|S| \leq s/2$, then w.p. $\geq 2/3$ over r, the Verifier rejects regardless of m.*

Arthur–Merlin protocols as adversaries.

▶ **Definition 5** (Breaking demi-bits generators by **AM** adversaries). *Let $m > n$. An **AM** adversary breaks a demi-bits generator $G : \{0,1\}^n \to \{0,1\}^m$ if both Prover and Verifier receive a common input $y \in \{0,1\}^m$, and:*

- *for $\geq 1/3$ fraction of $y \in \{0,1\}^m$, there exists a Prover that makes the Verifier accept with probability $\geq 2/3$;*
- *for all $y \in \mathrm{Range}(G)$, for every Prover, the Verifier accepts with probability $\leq 1/3$.*

We also consider **AM** adversaries with advice:

▶ **Definition 6** (**AM**/$_{k(n)}$ adversaries). *An* **AM**/$_{k(n)}$ *adversary is an Arthur–Merlin protocol where the* Verifier *is a probabilistic polynomial-time machine that additionally receives a $k(n)$-bit advice string a_n (which may depend on the input length n but not on the specific input y). The interaction on input y proceeds as follows:*

1. *The* Verifier *uses a_n and randomness r to generate a message to the* Prover*;*
2. *The* Prover *replies with a message;*
3. *The* Verifier *accepts or rejects based on y, a_n, r, and the* Prover*'s response.*

The acceptance probabilities are still defined over Verifier*'s internal randomness, with the advice string fixed to a_n.*

▶ **Proposition 7.** *Let G be a demi-bits generator.*

- *If there exists an* **AM** *adversary breaking G, then there exists an* **NP**/$_{\text{poly}}$ *adversary breaking G* ([1])*.*
- *For every constant $k \geq 2$, if there exists a k-round* **AM** *adversary breaking G, then there exists a (standard) two-round* **AM** *adversary breaking G* ([5])*.*

2.3 FNP v.s. SearchNP

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

▶ **Definition 8** (**SearchNP** [17])**.** *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 9** (**FNP** [17])**.** *The class of search problems defined by a polynomial-time relation, i.e., $R \in \mathbf{P}$ is defined as* **FNP**.

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

▶ **Proposition 10** ([17])**.** *If* $\mathbf{P} \neq \mathbf{NP}$*, then there is a total search problem in* **SearchNP** $\setminus$ **FNP**.

For more knowledge about nondeterministic algorithms, readers are referred to [17, Section 2.4].

2.4 Proof Complexity

Recall that TAUT, the set of DNF tautologies, is the canonical **coNP**-complete problem. A *propositional proof system* is simply a nondeterministic algorithm for TAUT. More formally:

▶ **Definition 11** ([22])**.** *An algorithm $\mathcal{P}(\varphi, \pi)$ is called a* propositional proof system *if it satisfies the following conditions:*

- **(Completeness)** *For every $\varphi \in$ TAUT, there exists a string $\pi \in \{0,1\}^*$ such that $\mathcal{P}(\varphi, \pi)$ accepts.*
- **(Soundness)** *For every $\varphi, \pi \in \{0,1\}^*$, if $\mathcal{P}(\varphi, \pi)$ accepts, then $\varphi \in$ TAUT.*
- **(Efficiency)** *$\mathcal{P}(\varphi, \pi)$ runs in deterministic* $\text{poly}(|\varphi| + |\pi|)$ *time.*

We say that $\mathcal{P}$ is a non-uniform *propositional proof system if $\mathcal{P}$ is a polynomial-size circuit instead of a uniform algorithm (that is, $\mathcal{P}$ is equipped with non-uniform advice).*

▶ **Definition 12** (Proof Complexity Generators [3, 53])**.** *Let $s(n) < n$ be a function for seed length. A proof complexity generator is a map $C_n : \{0,1\}^s \to \{0,1\}^n$ computed by a family of polynomial-size circuits $\{C_n\}_n$. A generator is secure against a propositional proof system P if for every large enough n and every $y \in \{0,1\}^n$, P does not have a polynomial-size proof of the (properly encoded) statement*

$$\forall x \in \{0,1\}^s, C_n(x) \neq y.$$

It is easy to see that the existence of proof complexity generators is closely related to the hardness of range avoidance. In fact we have:

▶ **Theorem 13** (Informal version of [72, Theorem 6.6])**.** *The range avoidance problem with suitable stretch is in* **FNP** *if and only if there exists a propositional proof system that breaks every proof complexity generator.*

Pseudo-surjectivity. In addition to the basic notion of hardness for proof complexity generators, several stronger notions have been proposed in the literature, including freeness [50], iterability and pseudo-surjectivity [53], and $\bigvee$-hardness [51]. Pseudo-surjectivity is the strongest hardness notion among them. In this paper, we show that our proof complexity generators are pseudo-surjective in certain parameter regimes.

To motivate the definition of pseudo-surjectivity [53, Definition 3.1], it is helpful to consider *Student-Teacher games* for solving Avoid. Let $G : \{0,1\}^n \to \{0,1\}^m$ be a mapping where $m > n$. A polynomial-time *Student* attempts to find a string $y \in \{0,1\}^m \setminus \text{Range}(G)$ with the help of a *Teacher* who has unbounded computational power. The game proceeds in rounds. In each round i, the Student proposes a candidate string $y_i \in \{0,1\}^m$, and if $y_i \in \text{Range}(G)$, the Teacher returns a preimage $q_i \in \{0,1\}^n$ such that $G(q_i) = y_i$. If the Student ever proposes a string outside the range of G, they win the game.

A Student who attempts to solve Avoid in k rounds can be represented as k circuits $B_1, B_2, \ldots, B_k$, where each B_i uses the Teacher's responses from previous rounds to generate the next query. Specifically, B_1 outputs a fixed string $y_1 \in \{0,1\}^m$, and each subsequent circuit B_i $(i > 1)$ takes the previous responses $q_1, q_2, \ldots, q_{i-1} \in \{0,1\}^n$ as inputs and outputs $y_i \in \{0,1\}^m$. The game proceeds as follows:

- The Student proposes $y_1 := B_1 \in \{0,1\}^m$. If $y_1 \notin \text{Range}(G)$, then the Student wins the game; otherwise, the Teacher returns some $q_1 \in \{0,1\}^n$ such that $G(q_1) = y_1$.
- The Student then proposes $y_2 := B_2(q_1) \in \{0,1\}^m$. If $y_2 \notin \text{Range}(G)$ then the Student wins the game; otherwise, the Teacher returns $q_2 \in \{0,1\}^n$ such that $G(q_2) = y_2$.
- ...
- This continues until round k, where the Student proposes $y_k := B_k(q_1, \ldots, q_{k-1}) \in \{0,1\}^m$. If $y_k \notin \text{Range}(G)$ then the Student wins the game; otherwise the Student loses the game.

To formally express whether the Student succeeds in the game, we define a formula stating that at least one of the Student's queries is outside the range of G. Let $B : \{0,1\}^{n'} \to \{0,1\}^m$ be a circuit, $z \in \{0,1\}^{n'}$ and $x \in \{0,1\}^n$ be disjoint variables, we define $\tau(G)_{B(z)}(x)$ to be the (properly encoded) statement that $B(z) \neq G(x)$. Then, using $\vec{q}_1, \ldots, \vec{q}_{k-1} \in \{0,1\}^n$ to represent the Teacher's responses, the Student wins if and only if

$$\bigvee_{i=1}^{k} \tau(G)_{B_i(q_1,\ldots,q_{i-1})}(q_i). \tag{1}$$

Roughly speaking, a generator G is *pseudo-surjective* for a proof system $\mathcal{P}$ if $\mathcal{P}$ cannot prove any Student wins the game, no matter how the Student is constructed. In other words, a generator G is pseudo-surjective for $\mathcal{P}$ if, for every sequence of Student circuits $(B_1, \ldots, B_k)$, the formula (1) is hard to prove in $\mathcal{P}$.

Note that pseudo-surjectivity is indeed a stronger notion than ordinary hardness for proof complexity generators. Indeed, for every $y \in \{0,1\}^m \setminus \mathrm{Range}(G)$, the trivial one-round Student with $B_1 = y$ clearly wins the game – yet pseudo-surjectivity implies that this fact is hard to prove in $\mathcal{P}$.

We proceed to the formal definition. We also introduce the notion of k-*round* pseudo-surjectivity, where the unprovability of (1) only holds for k-round Students for some fixed $k = k(n)$.

▶ **Definition 14** (k-round pseudo-surjectivity [53]). *Let $\mathcal{P}$ be any proof system, $G : \{0,1\}^n \to \{0,1\}^m$ be a circuit where $m > n$, and s be a size parameter.*

- *We say that G is* s-pseudo-surjective *for $\mathcal{P}$ if for every k and every sequence of Student circuits $(B_1, B_2, \ldots, B_k)$, (1) requires $\mathcal{P}$-proof of size at least s.*
- *Fixing a parameter $k = k(n)$, we say that G is* k-round s-pseudo-surjective *for $\mathcal{P}$ if for every sequence of Student circuits $(B_1, B_2, \ldots, B_k)$, (1) requires $\mathcal{P}$-proof of size $\geq s$.*

When $s = n^{\omega(1)}$, we omit the parameter s and simply say that G is (k-round) pseudo-surjective for $\mathcal{P}$.

2.5 Bounded Arithmetic

Roughly speaking, PV_1 is a theory of bounded arithmetic capturing "polynomial-time" reasoning. The language of PV_1, $\mathcal{L}(\mathsf{PV})$, contains a function symbol for every polynomial-time algorithm $f : \mathbb{N}^k \to \mathbb{N}$, defined using Cobham's characterization of polynomial-time functions [18]. Although Cook's PV [20] was originally defined as an equational theory (i.e., the only relation in PV is equality and there are no quantifiers), one can define a first-order theory PV_1 by adding suitable induction schemes [20, 59]. In the literature, the notation PV is often used to refer to the set of polynomial-time computable functions as well. The precise definition of PV_1 is somewhat involved, and we refer the reader to the textbooks [21, 48, 58] and references [14, 20, 40, 60].

To capture reasoning in *randomized* polynomial time, Jeřábek [38, 39, 41] defined a theory APC_1 by extending PV_1 with the *dual weak Pigeonhole Principle* for PV functions (i.e., polynomial-time functions). Let $\mathsf{Eval}(G, x) := G(x)$ be the circuit evaluation function. For a function $\ell(n) > n$, define $\mathrm{dwPHP}_\ell(\mathsf{Eval})$ to be the following sentence

$$\mathrm{dwPHP}_\ell(\mathsf{Eval}) :=$$
$$\forall n \in \mathsf{Log}\ \forall \text{circuit } G : \{0,1\}^n \to \{0,1\}^{\ell(n)}\ \exists y \in \{0,1\}^{\ell(n)}\ \forall x \in \{0,1\}^n\ [\mathsf{Eval}(G, x) \neq y].$$

Here, "$n \in \mathsf{Log}$" is the standard notation in bounded arithmetic, which means that n is the bit-length of some object; this notation allows us to reason about objects of size $\mathrm{poly}(n)$ instead of merely size $\mathrm{polylog}(n)$. The above sentence can be interpreted as the totality of Avoid: every input $G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ has at least one solution y.

For this paper, it suffices to think of $\ell(n)$ as a large polynomial in n; in fact, under suitable hardness assumptions, we will be able to show that PV_1 cannot prove dwPHP_ℓ for every polynomial $\ell(n)$. This suffices to separate APC_1 from PV_1.

KPT witnessing and Student-Teacher games. The only property of bounded arithmetic theories that we need in this paper is the *KPT witnessing theorem*:

▶ **Theorem 15** (KPT Witnessing Theorem for PV_1 [59]). *For every quantifier-free formula $\varphi(\vec{x}, y, z)$ in the language $\mathcal{L}(\mathsf{PV})$, if $\mathsf{PV}_1 \vdash \forall \vec{x}\ \exists y\ \forall z\ \varphi(\vec{x}, y, z)$, then there is a $k \in \mathbb{N}$ and $\mathcal{L}(\mathsf{PV})$-terms $t_1, t_2, \ldots, t_k$ such that*

$$\mathsf{PV}_1 \vdash \forall \vec{x}\ \forall z_1\ \forall z_2 \ldots \forall z_k \bigvee_{i=1}^{k} \varphi(\vec{x}, t_i(\vec{x}, z_1, \ldots, z_{i-1}), z_i). \tag{2}$$

In particular, Theorem 15 implies that if $\mathsf{PV}_1 \vdash \mathrm{dwPHP}_\ell(\mathsf{Eval})$, then there exists a constant k and a polynomial-time Student that wins the Student-Teacher game for the range avoidance problem. (Note that here the Student is computed by a *uniform* algorithm that gets $(1^n, G)$ as inputs, as opposed to a family of non-uniform circuits in Subsection 2.4). To see this, let $\varphi((1^n, G), y, x) = 1$ iff $\mathsf{Eval}(G, x) \neq y$ and apply Theorem 15. We obtain a constant $k \in \mathbb{N}$ and $\mathcal{L}(\mathsf{PV})$-terms $t_1, t_2, \ldots, t_k$ such that:

$$\mathsf{PV}_1 \vdash \forall n \in \mathsf{Log}\ \forall G\ \forall z_1\ \forall z_2 \ldots \forall z_k \bigvee_{i=1}^{k} \mathsf{Eval}(G, z_i) \neq t_i(G, z_1, \ldots, z_{i-1}). \tag{3}$$

This implies that the following Student wins the Student-Teacher game in k rounds:

1. The Student and the Teacher are given a circuit $G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$ as input.
2. In the first round, the Student proposes $y_1 := t_1(G)$ as a candidate non-output. If y_1 is correct (i.e., $\forall z_1\ \varphi(G, y_1, z_1)$ is true), then the Student wins the game. Otherwise, the Teacher provides a counterexample z_1 such that $\mathsf{Eval}(G, z_1) = y_1$, i.e., a preimage $z_1 \in G^{-1}(y_1)$.
3. Then, the Student proposes a new candidate $y_2 := t_2(G, z_1)$ based on the counterexample given in the first round. If y_2 is a correct non-output, then the Student wins the game. Otherwise, the Teacher again provides a counterexample z_2 such that $\mathsf{Eval}(G, z_2) = y_2$, i.e., a preimage $z_2 \in G^{-1}(y_2)$.
4. The game proceeds until the Student provides a correct witness y.

2.6 Extractors

▶ **Definition 16** (k-Source). *A random variable X is a k-*source *if for every $x \in \mathrm{Supp}(X)$, $\Pr[X = x] \leq 2^{-k}$.*

▶ **Definition 17** (Strong Seeded Extractors). *A polynomial-time computable function $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k, ε)-*strong seeded extractor *if for every k-source $\mathcal{X}$ over $\{0,1\}^n$, the statistical distance of $(\mathcal{U}_d, \mathsf{Ext}(X, \mathcal{U}_d))$ and $(\mathcal{U}_d, \mathcal{U}_m)$ is at most ε.*

Below is the only property of strong seeded extractors that we will use:

▶ **Fact 2.18.** *Suppose $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k, ε)-strong seeded extractor. Then for every (possibly unbounded) adversary $\mathcal{A} : \{0,1\}^{d+m} \to \{0,1\}$, the number of strings $x \in \{0,1\}^n$ such that*

$$\Pr_{r \sim \{0,1\}^d}[\mathcal{A}(r, \mathsf{Ext}(x, r)) = 1] < \Pr_{r \sim \{0,1\}^d, z \sim \{0,1\}^m}[\mathcal{A}(r, z) = 1] - \varepsilon \tag{4}$$

is at most 2^k.

Proof Sketch. Fix an adversary $\mathcal{A}$, let $\mathcal{X}$ be the set of strings $x \in \{0,1\}^n$ such that (4) holds. (We abuse notation and also use $\mathcal{X}$ to denote the uniform distribution over itself.) Note that $\mathcal{A}$ distinguishes $\mathsf{Ext}(\mathcal{X}, r)$ from the uniform distribution with advantage ε. If $|\mathcal{X}| \geq 2^k$, then the min-entropy of $\mathcal{X}$ is at least k, contradicting the extractor properties of Ext. Hence $|\mathcal{X}| < 2^k$. ◀

This work requires extractors with exponentially small ε, which can be constructed from any family of pairwise independent hash functions.

▶ **Theorem 19** (Leftover Hash Lemma [35]). *Let $h : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a family of pairwise independent hash functions, where the first component (length n) is its input and the second component (length d) is its key. Then for every k, ε such that $m = k - 2\log(1/\varepsilon)$, h is a (k,ε)-strong seeded extractor.*

In particular, if $n \geq m$ and $d \geq 2n$, there exists a family of pairwise independent hash functions h that is $\mathbb{F}_2$-linear.[9] *If we set $n \geq 3m + 3$, $d \geq 2n$, $k := n - 1$, $\varepsilon := 2^{-m-1}$, then h is an $(n-1,\varepsilon)$-strong seeded extractor.*

3 Hardness of Range Avoidance

▶ **Theorem 1.** *Assume that for some $m > n$, there exists a demi-bits generator $G : \{0,1\}^n \to \{0,1\}^N$ and $\mathsf{Ext} : \{0,1\}^N \times \{0,1\}^d \to \{0,1\}^m$ is an $(N-1, 2^{-m-1})$-strong seeded extractor. ($N, d \leq \mathrm{poly}(m)$.) Then* Avoid *for polynomial-size circuits of stretch $n \to m$ is not in* **SearchNP**.

Proof. Let $r \in \{0,1\}^d$, define the circuit $C_r : \{0,1\}^n \to \{0,1\}^m$ with r hardwired:

$$C_r(s) = \mathsf{Ext}(G(s), r).$$

Assume towards contradiction that there is a nondeterministic polynomial-time algorithm $\mathcal{A}$ solving Avoid. We construct the following nondeterministic adversary $\mathcal{B}(y)$ that breaks the demi-bits generator G. Given an input $y \in \{0,1\}^N$, the adversary $\mathcal{B}$ accepts y if and only if there exists $r \in \{0,1\}^d$ such that some nondeterministic branch of $\mathcal{A}(C_r)$ outputs $\mathsf{Ext}(y, r)$.

It is easy to see that $\mathcal{B}$ rejects every string $y \in \mathrm{Range}(G)$. To see this, suppose that $y = G(s)$ for some $s \in \{0,1\}^n$. Then

$$C_r(s) = \mathsf{Ext}(G(s), r) = \mathsf{Ext}(y, r),$$

hence $\mathcal{A}(C_r)$ will never output $\mathsf{Ext}(y, r)$.

It remains to show that $\mathcal{B}$ accepts at least $1/2$ fraction of strings $y \in \{0,1\}^N$. For $r \in \{0,1\}^d, z \in \{0,1\}^m$, let $\mathcal{A}'(r, z)$ be the adversary that outputs 1 if there is a nondeterministic branch of $\mathcal{A}(C_r)$ that outputs z, and outputs 0 otherwise. Since Ext is a $(N-1, 2^{-m-1})$-strong seeded extractor, the following is true for at least $1/2$ fraction of $y \in \{0,1\}^N$:

$$\Pr_{r \sim \{0,1\}^d}[\mathcal{A}'(r, \mathsf{Ext}(y,r)) = 1] \geq \Pr_{\substack{r \sim \{0,1\}^d \\ z \sim \{0,1\}^m}}[\mathcal{A}'(r,z) = 1] - 2^{-m-1} \geq 2^{-m-1}.$$

For such y, there exists $r^\star \in \{0,1\}^d$ and a nondeterministic branch of $\mathcal{A}(C_{r^\star})$ that outputs $\mathsf{Ext}(y, r^\star)$.

It follows that $\mathcal{B}$ rejects every string in $\mathrm{Range}(G)$ but accepts at least $1/2$ fraction of strings $y \in \{0,1\}^N$. This contradicts the security of G. ◀

[9] That is, for each fixed $r \in \{0,1\}^d$, the function $h(-, r)$ is an $\mathbb{F}_2$-linear function over its inputs.

▶ **Theorem 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 a nondeterministic polynomial-time algorithm. If $N > 10m$ and $m > n$, then there exists $h \in \mathcal{H}$ such that $\mathcal{A}$ fails to solve the range avoidance problem on the input $h \circ G$.*

Proof. We construct an extractor $\mathsf{Ext}(y, h) := h(y)$ $(y \in \{0,1\}^N, h \in \mathcal{H})$. According to Theorem 19, $\mathsf{Ext}\colon \{0,1\}^N \times \mathcal{H} \to \{0,1\}^m$ is an $(N - 1, 2^{-m-1})$-strong seeded extractor. Therefore, by Theorem 1, there exists $h \in \mathcal{H}$ such that $\mathcal{A}$ fails to solve Avoid on $\mathsf{Ext}(G(\cdot), h)$, i.e., $h \circ G$. ◀

▶ **Corollary 2.** *Under Assumption 2.3, there are constants $\varepsilon > 0$ and $d \geq 2$ such that* Avoid *for $\mathsf{XOR} \circ \mathsf{AND}_d$ circuits (i.e., degree-d polynomials over $\mathbb{F}_2$) of stretch $n \mapsto n^{1+\varepsilon}$ is not in* **SearchNP**.

Proof. Assumption 2.3 implies a demi-bits generator $G : \{0,1\}^n \to \{0,1\}^{n^{1+\delta}}$ and each output bit of G can be computed by a degree-d polynomial over $\mathbb{F}_2$, where $\delta > 0$ and $d \geq 2$ are constants. Let $\varepsilon := \delta/2$, $\mathsf{Ext} : \{0,1\}^{n^{1+\delta}} \times \{0,1\}^{2n^{1+\delta}} \to \{0,1\}^{n^{1+\varepsilon}}$ be a $(n^{1+\delta} - 1, 2^{-n^{1+\varepsilon}-1})$-strong linear seeded extractor guaranteed by Theorem 19. Then, for every nondeterministic adversary $\mathcal{A}$, there exists $r \in \{0,1\}^{2n^{1+\delta}}$ such that $\mathcal{A}$ fails to solve Avoid on the instance

$$C_r(s) := \mathsf{Ext}(G(s), r).$$

Since Ext is multi-linear and G is a degree-d polynomial over $\mathbb{F}_2$, C_r is an $\mathsf{XOR} \circ \mathsf{AND}_d$ circuit. ◀

3.1 From Demi-Bits Generators to Proof Complexity Generators

Let $\mathcal{P}$ be a proof system and $G : \{0,1\}^n \to \{0,1\}^\ell$ be a function computable in polynomial size where $\ell > n$. (We allow G to take non-uniform advice.) Let $b \in \{0,1\}^\ell$, denote as $\tau_b(G)$ the propositional formula encoding that b is not in the range of G. We say G is a:

- *demi-bits generator* against $\mathcal{P}$, if for at least a $1/3$ fraction of $b \in \{0,1\}^\ell$, $\mathcal{P}$ does not have polynomial-size proof of $\tau_b(G)$; and G is a
- *proof complexity generator* against $\mathcal{P}$, if for *every* $b \in \{0,1\}^\ell$, $\mathcal{P}$ does not have polynomial-size proof of $\tau_b(G)$.

The precise definition of $\tau_b(G)$ as a 3-CNF is as follows. The variables of $\tau_b(G)$ consist of $x \in \{0,1\}^n$ and $\mathsf{hist} \in \{0,1\}^s$, where s is the number of internal gates in G (including the output gates but not including the input gates). The intended meaning is that $G(x) = b$ and hist represents the values of internal gates of G during the computation of $G(x)$. Each gate in G corresponds to a bit v_g; if g is an input (internal) gate then v_g refers to some x_i (hist_i). For each internal gate $g \in G$ labeled by an operation $\circ_g$ (such as $\wedge$, $\vee$, or $\oplus$) and two children gates g_l, g_r, we have a constraint

$$v_g = v_{g_l} \circ_g v_{g_r}$$

in $\tau_b(G)$. Similarly, for each $i \in [\ell]$ representing an output gate g_i, we have a constraint

$$v_{g_i} = b_i$$

in $\tau_b(G)$. Note that since every constraint only depends on at most 3 variables, it can be written as a 3-CNF of size at most $2^3 = 8$, and we can add every clause in this 3-CNF into $\tau_b(G)$. We assume that the $\oplus$ gate of fan-in 2 is included in our basis (looking ahead, it will be used to implement the extractor). The 3-CNF $\tau_b(G)$ is simply the union of (3-CNFs generated from) these constraints over every internal and output gate $g \in G$.

Now we define the notion of *simple parity reductions* between two CNFs. This is a technical notion that we need in Claim 4.

▶ **Definition 3.** *Let $F(x)$ and $G(y)$ be CNF formulas over variables $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$. We say that there is a* simple parity *reduction from F to G, denoted as $F \leq^{\oplus} G$, if:*

- **Variables.** *The reduction is computed by a* GF(2)*-linear mapping* $\mathsf{redu} : \{0,1\}^n \to \{0,1\}^m$ *(that is, every output bit of* redu *is the* XOR *of a subset of its input bits).*
- **Axioms.** *For any clause $g \in G$, one of the following happens:*
  - $g \circ \mathsf{redu} \equiv \mathsf{True}$*;*
  - $g \circ \mathsf{redu}$ *is equal to some axiom in F; or*
  - *g is a width-*1 *clause (i.e., one that consists of a* single *literal) and $g \circ \mathsf{redu}$ is the* XOR *of a subset of axioms in F (in which case these axioms in F are also width-*1 *clauses).*

We say a proof system $\mathcal{P}$ is *closed under simple parity reductions* if there is a polynomial p such that the following holds. For every CNF F and G, if there is a simple parity reduction from F to G and there is a length-ℓ $\mathcal{P}$-proof of G, then there is a length-$p(\ell)$ $\mathcal{P}$-proof of F.

We note that this notion is weaker than that of (degree-1) algebraic reductions in [10, 23]. It follows from [23, Lemma 8.3] that many algebraic proof systems (such as Nullstellensatz and Polynomial Calculus) over GF(2) are closed under simple parity reductions when the complexity measure is degree. While we do not know if $\mathsf{Res}[\oplus]$ (resolution over linear equations modulo 2 [36]) is closed under low-degree algebraic reductions, it is straightforward to prove that $\mathsf{Res}[\oplus]$ is closed under simple parity reductions (see Appendix C of the full version).

Recall that $G : \{0,1\}^n \to \{0,1\}^N$ is a purported demi-bits generator, $\mathsf{Ext} : \{0,1\}^N \times \{0,1\}^d \to \{0,1\}^m$ is an extractor, and for a fixed $r \in \{0,1\}^d$ we define the circuit $C_r : \{0,1\}^n \to \{0,1\}^m$ as

$$C_r(s) := \mathsf{Ext}(G(s), r).$$

We say that Ext is *linear* if for every fixed randomness r, the function $\mathsf{Ext}(\cdot, r) : \mathrm{GF}(2)^N \to \mathrm{GF}(2)^m$ is GF(2)-linear. For every r we fix a circuit Ext_r for computing $\mathsf{Ext}(\cdot, r)$ using $\oplus$ gates of fan-in 2 only.

▷ Claim 4. Suppose that Ext is a linear extractor. For every $y \in \{0,1\}^N$ and $r \in \{0,1\}^d$, there is a simple parity reduction from $\tau_y(G)$ to $\tau_z(C_r)$, where $z := \mathsf{Ext}(y, r)$.

First, as a sanity check, we show that $\tau_y(G)$ follows from $\tau_z(C_r)$ logically: Suppose that $\tau_y(G)$ is false and that $y = G(s)$ for some $s \in \{0,1\}^n$, then

$$C_r(s) = \mathsf{Ext}(G(s), r) = \mathsf{Ext}(y, r) = z,$$

meaning that $\tau_z(C_r)$ is also false. Now we show that if Ext is a linear extractor, then the above deduction is actually a simple parity reduction under our formalization of $\tau_b(G)$:

Proof of Claim 4. Recall that the variables of $\tau_y(G)$ consist of $s \in \{0,1\}^n$ and $\mathsf{hist}_G \in \{0,1\}^{|G|}$, where $|G|$ denotes the number of internal gates in G. Also, recall the variables of $\tau_z(C_r)$ consist of $s \in \{0,1\}^n$ and $\mathsf{hist}_{C_r} \in \{0,1\}^{|C_r|}$. Since $C_r(s) = \mathsf{Ext}(G(s), r)$, hist_{C_r} consists of hist_G as well as the internal gates of $\mathsf{Ext}(\cdot, r)$. Since Ext is linear, each internal gate in $\mathsf{Ext}(\cdot, r)$ is an XOR of variables in hist_G. Therefore, one can compute a GF(2)-linear map $\mathsf{redu} : \{0,1\}^{n+|G|} \to \{0,1\}^{n+|C_r|}$ that maps (s, hist_G) to (s, hist_{C_r}).

Now we show that for every clause $c \in \tau_z(C_r)$, one of the three cases in Definition 3 happens. Note that c comes from an internal gate or an output gate of C_r.

- If c comes from an internal gate of G, then $c \circ \mathsf{redu}$ (which is equal to c itself) is an axiom in $\tau_y(G)$.
- If c comes from an internal gate in $\mathsf{Ext}(\cdot, r)$, then $c \circ \mathsf{redu} \equiv \mathsf{True}$ by the definition of redu.
- The only remaining case is that c comes from an output gate. Suppose this is the i-th output gate of C_r (where $i \in [m]$), and let v'_i denote the variable (of $\tau_z(C_r)$) representing the i-th output of C_r. Note that c is a width-1 axiom stating that $v'_i = z_i$.
  Let $S_i \subseteq [N]$ be such that $\mathsf{Ext}(y, r)_i = \bigoplus_{j \in S_i} y_j$. Then redu maps v'_i to $\bigoplus_{j \in S_i} v^G_j$, where v^G_j is the variable in $\tau_y(G)$ that represents the j-th output gate of G. We also have that $z_i = \bigoplus_{j \in S_i} y_j$. Hence $c \circ \mathsf{redu}$ is the XOR of the axioms $v^G_j = y_j$ over all $j \in S_i$. Since each $v^G_j = y_j$ is an axiom in $\tau_y(G)$, this concludes the proof. ◁

▶ **Theorem 5.** *Let $\mathcal{P}$ be a proof system closed under parity reductions. Let $G : \{0,1\}^n \to \{0,1\}^N$ be a demi-bits generator secure against $\mathcal{P}$, and $\mathsf{Ext} : \{0,1\}^N \times \{0,1\}^d \to \{0,1\}^m$ be an $(N - 2, 2^{-m-1})$-strong linear seeded extractor. Then there is a non-uniform proof complexity generator secure against $\mathcal{P}$.*

Proof. Suppose for contradiction that for every $r \in \{0,1\}^d$, there exists a string $z(r) \in \{0,1\}^m$ such that $\mathcal{P}$ admits a length-ℓ proof of $\tau_{z(r)}(C_r)$, where $\ell \leq \mathrm{poly}(|G|)$. For $r \in \{0,1\}^d$ and $z \in \{0,1\}^m$, let $\mathcal{A}'(r, z)$ be the adversary that outputs 1 if $\mathcal{P}$ admits a length-ℓ proof of $\tau_z(C_r)$ and outputs 0 otherwise. Since for every r, $\mathcal{A}'(r, z(r)) = 1$, we have

$$\Pr_{\substack{r \sim \{0,1\}^d \\ z \sim \{0,1\}^m}} [\mathcal{A}'(r, z) = 1] \geq 2^{-m}.$$

Since Ext is a $(n - 2, 2^{-m-1})$-strong extractor, for at least a $3/4$ fraction of $y \in \{0,1\}^N$, we have

$$\Pr_{r \sim \{0,1\}^d} [\mathcal{A}'(r, \mathsf{Ext}(y, r)) = 1] \geq \Pr_{\substack{r \sim \{0,1\}^d \\ z \sim \{0,1\}^m}} [\mathcal{A}'(r, z) = 1] - 2^{-m-1} > 0.$$

Hence, for such $y \in \{0,1\}^N$, there exists some $r := r(y)$ such that $\mathcal{P}$ admits a length-ℓ proof of $\tau_z(C_r)$ where $z := \mathsf{Ext}(y, r)$. Since there is a parity reduction from $\tau_y(G)$ to $\tau_z(C_r)$ and $\mathcal{P}$ is closed under parity reductions, it follows that $\mathcal{P}$ admits a length-$\mathrm{poly}(\ell)$ proof of $\tau_y(G)$ as well, contradicting the security of G as a demi-bits generator against $\mathcal{P}$. ◀

Although super-polynomial lower bounds for $\mathsf{Res}[\oplus]$ remain open, it seems conceivable that we will eventually prove such lower bounds sooner or later. Our results suggest a potential approach for designing proof complexity generators against $\mathsf{Res}[\oplus]$: it suffices to design a *demi-bits generator* against $\mathsf{Res}[\oplus]$ (which might be an easier task) and then apply Theorem 5.

4 Lower Bounds for Student-Teacher Games

In this section, we show that Avoid is hard for Student-Teacher games. In Subsection 4.1 we prove lower bounds against uniform, polynomial-time Students, which implies a conditional separation between bounded arithmetic theories PV_1 and APC_1. In Subsection 4.2, we show that demi-bits generators can be transformed into proof complexity generators that are pseudo-surjective.

4.1 Separating PV_1 from APC_1

As discussed in Subsection 2.5, to separate PV_1 from APC_1, it suffices to show that there is no polynomial-time Student that wins the Student-Teacher game for Avoid in $O(1)$ rounds. In fact, we will show something stronger: Let $k = k(n)$ be a parameter, assuming the existence of demi-bits generators secure against $\mathbf{AM}/_{O(\log k)}$, there is no polynomial-time Student that wins the Student-Teacher game for Avoid in $k(n)$ rounds.

▶ **Theorem 1.** *Let m, n, k be parameters such that $m > n$. Assume there exists a demi-bits generator $G : \{0,1\}^n \to \{0,1\}^N$ secure against $\mathbf{AM}/_{O(\log k)}$. Let $\mathsf{Ext} : \{0,1\}^N \times \{0,1\}^d \to \{0,1\}^m$ be an $(N - 1, 2^{-10km})$-strong extractor. Then for every deterministic polynomial-time Student A, there is a string $r \in \{0,1\}^d$ and a Teacher such that A fails to solve* Avoid *on C_r in k rounds, where $C_r : \{0,1\}^n \to \{0,1\}^m$ is the circuit*

$$C_r(s) := \mathsf{Ext}(G(s), r).$$

Proof. Let A denote the Student algorithm where $A(i, C, z_1, \ldots, z_{i-1})$ outputs the i-th candidate solution. For strings $s_1, \ldots, s_j \in \{0,1\}^n$ (where $j \le k$), we say that $(s_1, \ldots, s_j)$ is a *valid trace* for A on the input C_r if all of the following are true:

- $C_r(s_1) = A(1, C_r)$ (that is, s_1 is a valid counterexample for $A(1, -)$);
- $C_r(s_2) = A(2, C_r, s_1)$ (that is, s_2 is a valid counterexample for $A(2, -)$);
- ...
- and $C_r(s_j) = A(j, C_r, s_1, s_2, \ldots, s_{j-1})$ (that is, s_j is a valid counterexample for $A(j, -)$).

We prove the following stronger claim that implies Theorem 1:

▷ Claim 2. For every $j \le k$, there exist $s_1, s_2, \ldots, s_j \in \{0,1\}^n$ such that

$$\Pr_{r \sim \{0,1\}^d}[(s_1, \ldots, s_j) \text{ is a valid trace for } A \text{ on the input } C_r] \ge 2^{-2jm}.$$

Clearly, Claim 2 implies Theorem 1 by setting $j := k$ and noticing that $2^{-2jm} > 0$.

We prove Claim 2 by induction on j. The base case $j = 0$ is trivially true. Now we assume the claim is true for $j - 1$, which gives strings $s_1, \ldots, s_{j-1}$ such that

$$\Pr_{r \sim \{0,1\}^d}[(s_1, \ldots, s_{j-1}) \text{ is a valid trace for } A \text{ on the input } C_r] \ge 2^{-2(j-1)m}.$$

Consider the following $\mathbf{AM}/_{O(\log k)}$ protocol that attempts to break the demi-bits generator G. This protocol has the index j hardwired as advice but is otherwise uniform.

Algorithm 4.1 The $\mathbf{AM}/_{O(\log k)}$ protocol $\mathcal{P}$ breaking demi-bits generator G.

Input: A string $y \in \{0,1\}^N$.

1 Prover sends $s_1, \ldots, s_{j-1}$;

2 Prover and Verifier run the Goldwasser–Sipser protocol to estimate

$$p := \Pr_{r \sim \{0,1\}^d}\left[\begin{array}{c}(s_1, \ldots, s_{j-1}) \text{ is a valid trace for } A \text{ on the input } C_r \\ \text{and } \mathsf{Ext}(y, r) = A(j, C_r, s_1, \ldots, s_{j-1}).\end{array}\right];$$

3 If $p \ge 2^{-(2j-1)m-1}$ then Verifier accepts; if $p \le 2^{-(2j-1)m-2}$ then Verifier rejects;

Completeness of $\mathcal{P}$. We show that for at least a $1/2$ fraction of y, there is a Prover such that the Verifier accepts w.p. $\geq 2/3$ in $\mathcal{P}$. In the first round, the honest Prover sends $(s_1, \ldots, s_{j-1})$ as guaranteed by the induction hypothesis. Recall that this means

$$\Pr_{r\sim\{0,1\}^d}[(s_1, \ldots, s_{j-1}) \text{ is a valid trace for } A \text{ on the input } C_r] \geq 2^{-2(j-1)m}.$$

Let $\mathsf{Test}(r, z) = 1$ if $(s_1, \ldots, s_{j-1})$ is a valid trace for A on the input C_r and $z = A(j, C_r, s_1, \ldots, s_{j-1})$, and $\mathsf{Test}(r, z) = 0$ otherwise. Clearly, we have

$$\Pr_{r\sim\{0,1\}^d, z\sim\{0,1\}^m}[\mathsf{Test}(r, z) = 1] \geq 2^{-2(j-1)m}/2^m = 2^{-(2j-1)m}.$$

Since Ext is an $(N - 1, 2^{-10km})$-strong extractor, for $\geq 1/2$ fraction of y's, it holds that

$$\Pr_{r\sim\{0,1\}^d}[\mathsf{Test}(r, \mathsf{Ext}(y, r)) = 1] \geq \Pr_{r\sim\{0,1\}^d, z\sim\{0,1\}^m}[\mathsf{Test}(r, z) = 1] - 2^{-10km} \geq 2^{-(2j-1)m-1}.$$

It follows that there is a Prover for the Goldwasser–Sipser protocol in Line 2 of Algorithm 4.1 such that the verifier accepts with probability at least $2/3$.

Employing the lack of soundness. Since G is a demi-bits generator secure against $\mathbf{AM}/_{O(\log k)}$ adversaries, $\mathcal{P}$ does not have the soundness for all sufficiently large n. In other words, there is a Prover^* that makes the Verifier accepts some $y \in \mathrm{Range}(G)$ with probability $> 1/3$. Fix such a string y, let s_j be any n-bit string such that $G(s_j) = y$, and let $s_1, \ldots, s_{j-1} \in \{0,1\}^n$ be the message sent in Line 1 (of Algorithm 4.1) by Prover^* on the input y.

Since Verifier accepts with probability $> 1/3$, by the soundness of the Goldwasser–Sipser Protocol (Lemma 4), we have that

$$\Pr_{r\sim\{0,1\}^d}\left[\begin{array}{c}(s_1, \ldots, s_{j-1}) \text{ is a valid trace for } A \text{ on the input } C_r \\ \text{and } \mathsf{Ext}(y, r) = A(j, C_r, s_1, \ldots, s_{j-1}).\end{array}\right] \geq 2^{-j(2m+1)-2}.$$

Recall that $\mathsf{Ext}(y, r) = C_r(s_j)$, hence the above condition inside $\Pr_{r\sim\{0,1\}^d}[\cdot]$ means exactly that $(s_1, \ldots, s_j)$ is a valid trace for $A(C_r)$. This implies Claim 2 for j. ◀

We remark that the parameters we obtained in Theorem 1 are (almost) tight in the following sense. Theorem 1 showed that (under plausible assumptions) for every parameter $k \leq \mathrm{poly}(n)$, there is no deterministic polynomial-time Student that wins the Student-Teacher game for Avoid in k rounds, when given an Avoid instance of size $s = \mathrm{poly}(k, n) > k$. On the other hand, under plausible derandomization assumptions, for every size parameter s, there exists a deterministic polynomial-time Student that wins the game on size-s circuits within $k = \mathrm{poly}(s, n) > s$ rounds [33, Appendix A].

Finally, setting $k = O(1)$ in Theorem 1, we obtain the following separation:

▶ **Theorem 5.** *Assuming there exists a demi-bits generator $G : \{0,1\}^n \to \{0,1\}^{\omega(n)}$ secure against $\mathbf{AM}/_{O(1)}$, the Dual Weak Pigeonhole Principle for polynomial-time functions* (dwPHP(PV)) *is not provable in* PV. *(In particular,* APC_1 *is a strict extension of* PV_1*.)*

4.2 From Demi-Bits to Pseudo-Surjectivity

▶ **Theorem 3.** *Let $G : \{0,1\}^n \to \{0,1\}^N$ be a demi-bits generator secure against $\mathbf{NP}/_{\mathrm{poly}}$, $k \in \mathbb{N}$, and $\mathsf{Ext} : \{0,1\}^N \times \{0,1\}^d \to \{0,1\}^m$ be an $(N - 1, \varepsilon)$-strong linear extractor for $\varepsilon := 2^{-10km}$. $(k, d, N \leq \mathrm{poly}(m).)$ Then for every non-uniform propositional proof system $\mathcal{P}$, there is a string $r \in \{0,1\}^d$ such that the circuit $C_r : \{0,1\}^n \to \{0,1\}^m$ defined as*

$$C_r(s) := \mathsf{Ext}(G(s), r)$$

is a non-uniform k-round pseudo-surjective proof complexity generator secure against $\mathcal{P}$.

Proof. Suppose, for contradiction, that such an $r \in \{0,1\}^d$ does not exist. Then, for any $r \in \{0,1\}^d$, there exist Student circuits

$$B^{(r)} = \left\{ B_i^{(r)} \colon \{0,1\}^{n(i-1)} \to \{0,1\}^m \right\}_{i \in [k+1]}$$

such that $\mathcal{P}$ admits a proof of

$$\bigvee_{i=1}^{k} \tau(C_r)_{B_i(q_1,q_2,\ldots,q_{i-1})}(q_i).$$

(i.e., $\mathcal{P}$ can prove that $B^{(r)}$ wins the Student-Teacher game on C_r.)

Now we attempt to break the demi-bits generator G. For $j = 0, 1, \ldots, k$, define

$$\Phi_j := \max_{(s_1,s_2,\ldots,s_j) \in \{0,1\}^{nj}} \Pr_{r \sim \{0,1\}^m} \begin{bmatrix} \exists \text{ Student } B \text{ such that:} \\ (1)\ \mathcal{P} \text{ can prove that } B \text{ wins the Student-Teacher game on } C_r; \\ (2)\ B_i(s_1, \ldots, s_{i-1}) = C_r(s_i) \text{ for all } i \in [j]. \end{bmatrix}.$$

(Item (2) says that the history of the Student-Teacher game in the first i rounds is exactly $s_1, \ldots, s_j$.) We make the following claims on the values of Φ_0 and Φ_k:

- $\Phi_0 = 1$: When $j = 0$, item (2) obviously holds, and item (1) holds by our assumption that C_r is not a pseudo-surjective proof complexity generator;
- $\Phi_k = 0$: When $j = k$, for any r, B items (1) and (2) cannot hold simultaneously. This is because (2) implies that B loses the Student-Teacher game, which contradicts (1).

Simple calculations show that there exists $j \in \{0, 1, \ldots, k-1\}$ such that

$$\Phi_j \cdot 2^{-m} - \varepsilon > 2 \cdot \Phi_{j+1}.$$

We use such j to break the demi-bits generator G. Let $(s_1^*, \ldots, s_j^*)$ be the tuple such that the maximum is achieved in the definition of Φ_j, i.e.,

$$\Phi_j = \Pr_{r \sim \{0,1\}^m} \begin{bmatrix} \exists \text{ Student } B \text{ such that:} \\ (1)\ \mathcal{P} \text{ can prove that } B \text{ wins the Student-Teacher game on } C_r; \\ (2)\ B_i(s_1^*, \ldots, s_{i-1}^*) = C_r(s_i^*) \text{ for all } i \in [j]. \end{bmatrix}.$$

Consider the following algorithm: on an input $y \in \{0,1\}^n$, let

$$p(y) := \Pr_{r \sim \{0,1\}^m} \begin{bmatrix} \exists \text{ Student } B \text{ such that:} \\ (1)\ \mathcal{P} \text{ can prove that } B \text{ wins the Student-Teacher game on } C_r; \\ (2)\ B_i(s_1^*, \ldots, s_{i-1}^*) = C_r(s_i^*) = \mathsf{Ext}(G(s_i^*), r) \text{ for all } i \in [j], \\ \quad \text{and } B_{j+1}(s_1^*, \ldots, s_j^*) = \mathsf{Ext}(y, r). \end{bmatrix}.$$

Our algorithm accepts y if $p(y) \geq \Phi_j \cdot 2^{-m} - \varepsilon$, and rejects if $p(y) \leq \Phi_{j+1}$. This can be implemented by the Goldwasser–Sipser set lower bound protocol since $\Phi_j \cdot 2^{-m} - \varepsilon > 2 \cdot \Phi_{j+1}$, and the condition inside $\Pr_{r \sim \{0,1\}^m}[\cdot]$ is certifiable in polynomial time with the help of a prover. Finally, we prove that this algorithm breaks demi-bits generator G:

- **For any $y \in \mathrm{Range}(G)$, we have $p(y) \leq \Phi_{j+1}$:**
  Suppose $y = G(s)$. Then

$$p(y) = \Pr_{r\sim\{0,1\}^m}\left[\begin{array}{l}\exists \text{ student } B \text{ such that:}\\ (1)\ \mathcal{P} \text{ can prove that } B \text{ wins the Student-Teacher game on } \mathsf{Ext}(G(\cdot),r);\\ (2)\ B_i(s_1^*,\ldots,s_{i-1}^*) = \mathsf{Ext}(G(s_i^*),r) \text{ for all } i\in[j],\\ \quad \text{and } B_{j+1}(s_1^*,\ldots,s_j^*) = \mathsf{Ext}(G(s),r).\end{array}\right]$$
$$\leq \Phi_{j+1}.$$

Where the $\leq$ in the second line follows from the definition of Φ_{j+1}.

- **For half of $y \in \{0,1\}^n$, we have $p(y) \geq \Phi_j \cdot 2^{-m} - \varepsilon$:**
  For simplicity, we use "$\mathsf{Test}(r, \mathsf{Ext}(y,r))$" to denote the condition inside $\Pr_{r\sim\{0,1\}^m}[\cdot]$ in the definition of $p(y)$. We have:

$$\Pr_{\substack{r\sim\{0,1\}^m\\ z\sim\{0,1\}^m}}[\mathsf{Test}(r,z)] = \Phi_j \cdot 2^{-m}.$$

  By the definition of strong extractors, for half of $y \in \{0,1\}^n$, we have

$$\Pr_{r\sim\{0,1\}^m}[\mathsf{Test}(r,\mathsf{Ext}(y,r))] \geq \Pr_{\substack{r\sim\{0,1\}^m\\ z\sim\{0,1\}^m}}[\mathsf{Test}(r,z)] - \varepsilon = \Phi_j \cdot 2^{-m} - \varepsilon. \quad \blacktriangleleft$$

▶ **Corollary 4.** *Suppose for any parameter $N \leq \mathrm{poly}(n)$, there exists a demi-bits generator $G : \{0,1\}^n \to \{0,1\}^N$ secure against $\mathbf{NP}/_{\mathrm{poly}}$. Then for every non-uniform propositional proof system $\mathcal{P}$ and any parameters $k \leq \mathrm{poly}(n)$ and $n < m < \mathrm{poly}(n)$, there is a circuit $C : \{0,1\}^n \to \{0,1\}^m$ of size $\mathrm{poly}(n)$ such that C is a k-round pseudo-surjective proof complexity generator secure against $\mathcal{P}$.*

Proof. In Theorem 3, let $N := 100km$ and $\mathsf{Ext}\colon \{0,1\}^N \times \{0,1\}^{O(km)} \to \{0,1\}^m$ be an $(N-1, 2^{-10km})$-strong seeded extractor guaranteed by Theorem 19. Then there exists $r \in \{0,1\}^{O(km)}$ such that $C_r := \mathsf{Ext}(G(\cdot), r)$ is a k-round pseudo-surjective proof complexity generator secure against $\mathcal{P}$. ◀

References

1 Leonard M. Adleman. Two theorems on random polynomial time. In *FOCS*, pages 75–83. IEEE Computer Society, 1978. doi:10.1109/SFCS.1978.37.

2 Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108. ACM, 1996. doi:10.1145/237814.237838.

3 Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Pseudorandom generators in propositional proof complexity. *SIAM Journal on Computing*, 34(1):67–88, 2004. doi:10.1137/S0097539701389944.

4 Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, USA, 1st edition, 2009. doi:10.1017/CBO9780511804090.

5 László Babai. Trading group theory for randomness. In *STOC*, pages 421–429. ACM, 1985. doi:10.1145/22145.22192.

6 Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. Comput.*, 13(4):850–864, 1984. doi:10.1137/0213053.

7 Andrej Bogdanov and Luca Trevisan. Average-case complexity. *Found. Trends Theor. Comput. Sci.*, 2(1), 2006. doi:10.1561/0400000004.

8 Andrej Bogdanov and Luca Trevisan. On worst-case to average-case reductions for **NP** problems. *SIAM J. Comput.*, 36(4):1119–1159, 2006. doi:10.1137/S0097539705446974.

9 Maria Luisa Bonet, Samuel R. Buss, and Toniann Pitassi. Are there hard examples for Frege systems? In *Feasible Mathematics II*, pages 30–56, Boston, MA, 1995. Birkhäuser Boston. doi:10.1007/978-1-4612-2566-9_3.

10 Samuel R. Buss, Dima Grigoriev, Russell Impagliazzo, and Toniann Pitassi. Linear gaps between degrees for the polynomial calculus modulo distinct primes. *J. Comput. Syst. Sci.*, 62(2):267–289, 2001. doi:10.1006/JCSS.2000.1726.

11 Samuel R. Buss, Leszek Aleksander Kolodziejczyk, and Neil Thapen. Fragments of approximate counting. *J. Symb. Log.*, 79(2):496–525, 2014. doi:10.1017/JSL.2013.37.

12 Jin-yi Cai, Aduri Pavan, and D. Sivakumar. On the hardness of permanent. In *STACS*, volume 1563 of *Lecture Notes in Computer Science*, pages 90–99. Springer, 1999. doi:10.1007/3-540-49116-3_8.

13 Lijie Chen, Shuichi Hirahara, and Hanlin Ren. Symmetric exponential time requires near-maximum circuit size. In *STOC*, pages 1990–1999. ACM, 2024. doi:10.1145/3618260.3649624.

14 Lijie Chen, Jiatu Li, and Igor C. Oliveira. Reverse mathematics of complexity lower bounds. In *FOCS*, pages 505–527. IEEE, 2024. doi:10.1109/FOCS61266.2024.00040.

15 Lijie Chen and Roei Tell. Hardness vs randomness, revised: Uniform, non-black-box, and instance-wise. In *FOCS*, pages 125–136. IEEE, 2021. doi:10.1109/FOCS52979.2021.00021.

16 Yeyuan Chen, Yizhi Huang, Jiatu Li, and Hanlin Ren. Range avoidance, remote point, and hard partial truth table via satisfying-pairs algorithms. In *STOC*, pages 1058–1066. ACM, 2023. doi:10.1145/3564246.3585147.

17 Yilei Chen and Jiatu Li. Hardness of range avoidance and remote point for restricted circuits via cryptography. In *STOC*, pages 620–629. ACM, 2024. doi:10.1145/3618260.3649602.

18 Alan Cobham. The intrinsic computational difficulty of functions. In *Proc. Logic, Methodology, and the Philosophy of Science*, pages 24–30, 1964.

19 Jonas Conneryd, Susanna F. de Rezende, Jakob Nordström, Shuo Pang, and Kilian Risse. Graph colouring is hard on average for Polynomial Calculus and Nullstellensatz. In *FOCS*, pages 1–11. IEEE Computer Society, 2023. doi:10.1109/FOCS57990.2023.00007.

20 Stephen A. Cook. Feasibly constructive proofs and the propositional calculus (preliminary version). In *STOC*, pages 83–97. ACM, 1975. doi:10.1145/800116.803756.

21 Stephen A. Cook and Phuong Nguyen. *Logical Foundations of Proof Complexity*, volume 11. Cambridge University Press, 2010. doi:10.1017/CBO9780511676277.

22 Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *J. Symb. Log.*, 44(1):36–50, 1979. doi:10.2307/2273702.

23 Susanna F. de Rezende, Mika Göös, Jakob Nordström, Toniann Pitassi, Robert Robere, and Dmitry Sokolov. Automating algebraic proof systems is **NP**-hard. In *STOC*, pages 209–222. ACM, 2021. doi:10.1145/3406325.3451080.

24 Susanna F. de Rezende, Aaron Potechin, and Kilian Risse. Clique is hard on average for unary Sherali-Adams. In *FOCS*, pages 12–25. IEEE, November 2023. doi:10.1109/FOCS57990.2023.00008.

25 Joan Feigenbaum and Lance Fortnow. Random-self-reducibility of complete sets. *SIAM J. Comput.*, 22(5):994–1005, 1993. doi:10.1137/0222061.

26 Karthik Gajulapalli, Alexander Golovnev, Satyajeet Nagargoje, and Sidhant Saraogi. Range avoidance for constant depth circuits: Hardness and algorithms. In *APPROX/RANDOM*, volume 275 of *LIPIcs*, pages 65:1–65:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.APPROX/RANDOM.2023.65.

27 Michal Garlik, Svyatoslav Gryaznov, Hanlin Ren, and Iddo Tzameret. The weak rank principle: Lower bounds and applications, 2025. Manuscript.

28 Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984. doi:10.1016/0022-0000(84)90070-9.

29 Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In *STOC*, pages 59–68. ACM, 1986. doi:10.1145/12130.12137.

30 Venkatesan Guruswami, Xin Lyu, and Xiuhan Wang. Range avoidance for low-depth circuits and connections to pseudorandomness. In *APPROX/RANDOM*, volume 245 of *LIPIcs*, pages 20:1–20:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.APPROX/RANDOM.2022.20.

31 Shuichi Hirahara. Non-black-box worst-case to average-case reductions within **NP**. In *FOCS*, pages 247–258. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00032.

32 Rahul Ilango. The oracle derandomization hypothesis is false (and more) assuming no natural proofs, 2025. Manuscript.

33 Rahul Ilango, Jiatu Li, and R. Ryan Williams. Indistinguishability obfuscation, range avoidance, and bounded arithmetic. In *STOC*, pages 1076–1089. ACM, 2023. doi:10.1145/3564246.3585187.

34 Russell Impagliazzo. A personal view of average-case complexity. In *SCT*, pages 134–147. IEEE Computer Society, 1995. doi:10.1109/SCT.1995.514853.

35 Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions. In *STOC*, pages 12–24. ACM, 1989. doi:10.1145/73007.73009.

36 Dmitry Itsykson and Dmitry Sokolov. Resolution over linear equations modulo two. *Ann. Pure Appl. Log.*, 171(1), 2020. doi:10.1016/J.APAL.2019.102722.

37 Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In *STOC*, pages 60–73. ACM, 2021. doi:10.1145/3406325.3451093.

38 Emil Jeřábek. Dual weak pigeonhole principle, Boolean complexity, and derandomization. *Ann. Pure Appl. Log.*, 129(1-3):1–37, 2004. doi:10.1016/J.APAL.2003.12.003.

39 Emil Jeřábek. *Weak pigeonhole principle and randomized computation*. PhD thesis, Charles University in Prague, 2005.

40 Emil Jeřábek. The strength of sharply bounded induction. *Math. Log. Q.*, 52(6):613–624, 2006. doi:10.1002/MALQ.200610019.

41 Emil Jeřábek. Approximate counting in bounded arithmetic. *J. Symb. Log.*, 72(3):959–993, 2007. doi:10.2178/JSL/1191333850.

42 Erfan Khaniki. Nisan–Wigderson generators in proof complexity: New lower bounds. In *CCC*, volume 234 of *LIPIcs*, pages 17:1–17:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.CCC.2022.17.

43 Robert Kleinberg, Oliver Korten, Daniel Mitropolsky, and Christos Papadimitriou. Total Functions in the Polynomial Hierarchy. In *ITCS*, volume 185 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 44:1–44:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ITCS.2021.44.

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

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

46 Oliver Korten. Range avoidance and the complexity of explicit constructions. *Bull. EATCS*, 145, 2025. URL: http://bulletin.eatcs.org/index.php/beatcs/article/view/825.

47 Oliver Korten and Toniann Pitassi. Strong vs. weak range avoidance and the linear ordering principle. In *FOCS*, pages 1388–1407. IEEE, 2024. doi:10.1109/FOCS61266.2024.00089.

48 Jan Krajíček. *Bounded Arithmetic, Propositional Logic, and Complexity Theory*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1995. doi:10.1017/CBO9780511529948.

49 Jan Krajíček. On the weak pigeonhole principle. *Fundamenta Mathematicae*, 170:123–140, 2001. doi:10.4064/fm170-1-8.

50 Jan Krajíček. Tautologies from pseudo-random generators. *Bulletin of Symbolic Logic*, 7(2):197–212, 2001. doi:10.2307/2687774.

51 Jan Krajíček. On the existence of strong proof complexity generators. *Bulletin of Symbolic Logic*, 30(1):20–40, 2024. doi:10.1017/bsl.2023.40.

52 Jan Krajíček. A proof complexity generator. In *Proc. from the 13th International Congress of Logic, Methodology and Philosophy of Science (Beijing, August 2007)*, Studies in Logic and the Foundations of Mathematics. King's College Publications, London, 2009.

53 Jan Krajíček. Dual weak pigeonhole principle, pseudo-surjective functions, and provability of circuit lower bounds. *Journal of Symbolic Logic*, 69(1):265–286, 2004. doi:10.2178/jsl/1080938841.

54 Jan Krajíček. On the proof complexity of the Nisan–Wigderson generator based on a hard $\mathbf{NP} \cap \mathbf{coNP}$ function. *J. Math. Log.*, 11(1), 2011. doi:10.1142/S0219061311000979.

55 Jan Krajíček. Small circuits and dual weak PHP in the universal theory of p-time algorithms. *ACM Trans. Comput. Log.*, 22(2):11:1–11:4, 2021. doi:10.1145/3446207.

56 Jan Krajíček. A proof complexity conjecture and the Incompleteness theorem. *The Journal of Symbolic Logic*, pages 1–5, 2023. doi:10.1017/jsl.2023.69.

57 Jan Krajíček. *Proof complexity generators*. Cambridge University Press, 2025. doi:10.1017/9781009611664.

58 Jan Krajíček. *Proof Complexity*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2019. doi:10.1017/9781108242066.

59 Jan Krajíček, Pavel Pudlák, and Gaisi Takeuti. Bounded arithmetic and the polynomial hierarchy. *Annals of Pure and Applied Logic*, 52(1):143–153, 1991. doi:10.1016/0168-0072(91)90043-L.

60 Jiatu Li. An introduction to feasible mathematics and bounded arithmetic for computer scientists. *Electron. Colloquium Comput. Complex.*, TR25-086, 2025. URL: https://eccc.weizmann.ac.il/report/2025/086.

61 Zeyong Li. Symmetric exponential time requires near-maximum circuit size: Simplified, truly uniform. In *STOC*, STOC 2024, pages 2000–2007. ACM, 2024. doi:10.1145/3618260.3649615.

62 Peter Bro Miltersen and N. V. Vinodchandran. Derandomizing Arthur-Merlin games using hitting sets. *Comput. Complex.*, 14(3):256–279, 2005. doi:10.1007/S00037-005-0197-7.

63 Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994. doi:10.1016/S0022-0000(05)80043-1.

64 Shuo Pang. Large clique is hard on average for resolution. In *CSR*, volume 12730 of *Lecture Notes in Computer Science*, pages 361–380. Springer, 2021. doi:10.1007/978-3-030-79416-3_22.

65 Ján Pich. Nisan-Wigderson generators in proof systems with forms of interpolation. *Math. Log. Q.*, 57(4):379–383, 2011. doi:10.1002/MALQ.201010012.

66 Ján Pich. Circuit lower bounds in bounded arithmetics. *Ann. Pure Appl. Log.*, 166(1):29–45, 2015. doi:10.1016/J.APAL.2014.08.004.

67 Ján Pich and Rahul Santhanam. Why are proof complexity lower bounds hard? In *FOCS*, pages 1305–1324. IEEE Computer Society, 2019. doi:10.1109/FOCS.2019.00080.

68 Ján Pich and Rahul Santhanam. Learning algorithms versus automatability of Frege systems. In *ICALP*, volume 229 of *LIPIcs*, pages 101:1–101:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ICALP.2022.101.

69 Alexander A. Razborov. Lower bounds for the polynomial calculus. *Comput. Complex.*, 7(4):291–324, 1998. doi:10.1007/s000370050013.

70 Alexander A. Razborov. Resolution lower bounds for perfect matching principles. *J. Comput. Syst. Sci.*, 69(1):3–27, 2004. doi:10.1016/J.JCSS.2004.01.004.

71 Alexander A. Razborov. Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution. *Annals of Mathematics*, pages 415–472, 2015. doi:10.4007/annals.2015.181.2.1.

72 Hanlin Ren, Rahul Santhanam, and Zhikun Wang. On the range avoidance problem for circuits. In *FOCS*, pages 640–650, 2022. doi:10.1109/FOCS54457.2022.00067.

73 Steven Rudich. Super-bits, demi-bits, and $\mathbf{NP}/_{\text{qpoly}}$-natural proofs. In *RANDOM*, volume 1269 of *Lecture Notes in Computer Science*, pages 85–93. Springer, 1997. doi:10.1007/3-540-63248-4_8.

74 Rahul Santhanam and Iddo Tzameret. Iterated lower bound formulas: a diagonalization-based approach to proof complexity. In *STOC*, pages 234–247. ACM, 2021. doi:10.1145/3406325.3451010.

75 Ronen Shaltiel and Christopher Umans. Simple extractors for all min-entropies and a new pseudorandom generator. *J. ACM*, 52(2):172–216, 2005. doi:10.1145/1059513.1059516.

76 Ronen Shaltiel and Christopher Umans. Pseudorandomness for approximate counting and sampling. *Comput. Complex.*, 15(4):298–341, 2006. doi:10.1007/S00037-007-0218-9.

77 Iddo Tzameret and Lu-Ming Zhang. Stretching Demi-Bits and Nondeterministic-Secure Pseudorandomness. In *ITCS*, volume 287 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 95:1–95:22, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2024.95.

78 Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *FOCS*, pages 80–91. IEEE Computer Society, 1982. doi:10.1109/SFCS.1982.45.