# The Exact Complexity of Pseudorandom Functions and Tight Barriers to Lower Bound Proofs

Zhiyuan Fan* Jiatu Li† Tianqi Yang‡

*Institute for Interdisciplinary Information Sciences*
*Tsinghua University, Beijing, China*

August 23, 2021

**Abstract**

How much computational resource do we need for cryptography? This is an important question of both theoretical and practical interests. In this paper, we study the problem on pseudorandom functions (PRFs) in the context of circuit complexity. Perhaps surprisingly, we prove extremely tight upper and lower bounds in various circuit models.

- In general $B_2$ circuits, assuming the existence of PRFs, PRFs can be constructed in $2n + o(n)$ size, simplifying and improving the $O(n)$ bound by Ishai et al. (STOC 2008). We show that such construction is almost optimal by giving an unconditional $2n - O(1)$ lower bound.
- In logarithmic depth circuits, assuming the existence of $\mathsf{NC}^1$ PRFs, PRFs can be constructed in $2n + o(n)$ size and $(1 + \varepsilon) \log n$ depth simultaneously.
- In constant depth linear threshold circuits, assuming the existence of $\mathsf{TC}^0$ PRFs, PRFs can be constructed with wire complexity $n^{1+O(1.61^{-d})}$. We also give an $n^{1+\Omega(c^{-d})}$ wire complexity lower bound for some constant $c$.

The upper bounds are proved with generalized Levin's trick and novel constructions of "almost" universal hash functions; the lower bound for general circuits is proved via a tricky but elementary wire-counting argument; and the lower bound for $\mathsf{TC}^0$ circuits is proved by extracting a "black-box" property of $\mathsf{TC}^0$ circuits from the "white-box" restriction lemma of Chen, Santhanam, and Srinivasan (Theory Comput. 2018). As a byproduct, we prove unconditional tight upper and lower bounds for "almost" universal hashing, which we believe to have independent interests.

Following Natural Proofs by Razborov and Rudich (J. Comput. Syst. Sci. 1997), our results make progress in realizing the difficulty to improve known circuit lower bounds, which recently becomes significant due to the discovery of several "bootstrapping results". In $\mathsf{TC}^0$, this reveals the limitation of the current restriction-based methods; in particular, it brings new insights in understanding the strange phenomenon of "sharp threshold results" such as the one presented by Chen and Tell (STOC 2019).

*fan-zy19@mails.tsinghua.edu.cn
†lijt19@mails.tsinghua.edu.cn
‡yangtq19@mails.tsinghua.edu.cn

# Contents


**1 Introduction** **3**
- 1.1 Our results . . . 5
- 1.2 Organization of the paper . . . 7

**2 Connections to circuit lower bound proofs** **7**
- 2.1 A case study: Chen-Tell's bootstrapping result . . . 9
- 2.2 Comparisons to previous barriers . . . 10

**3 Proof overview** **11**
- 3.1 Upper bounds: Levin's trick and hash function . . . 11
- 3.2 Lower bounds in general circuits . . . 12
- 3.3 Lower bounds in linear threshold circuits: random restriction . . . 13

**4 Preliminaries** **14**
- 4.1 Boolean circuits . . . 14
- 4.2 Threshold function and threshold circuits . . . 15
- 4.3 Restriction . . . 15
- 4.4 Pseudorandom function . . . 16
- 4.5 Hash function and error-correcting code . . . 16
- 4.6 Circuit complexity and uniformity . . . 17
- 4.7 Levin's trick for domain extension . . . 18
- 4.8 Probability theory . . . 19

**5 A $2n + o(n)$ upper bound for $B_2$ circuits** **19**
- 5.1 An $O(n)$ upper bound . . . 20
- 5.2 Constructing hash function from 1-detector . . . 20
- 5.3 A simple probabilistic construction . . . 21
- 5.4 Better 1-detector from high-girth graphs . . . 23
- 5.5 The upper bound of depth for PRF and hash . . . 24

**6 A $2n - O(1)$ lower bound for $B_2$ circuits** **26**

**7 Constant-depth linear threshold circuits** **29**
- 7.1 Upper bound via efficient ECC . . . 29
- 7.2 Extracting black-box property from white-box restriction . . . 30
- 7.3 Proof of restriction lemma . . . 33

**8 Open problems** **38**

**A The leftover lemma for Levin's trick** **44**

**B Proof of Lemma 5.6** **45**

**C An optimally sparse explicit almost universal hash in $\mathsf{CC}^0[2]$** **48**

# 1 Introduction

Pseudorandom function (PRF), capturing the indistinguishability of a set of functions from a random function, is a cornerstone of cryptography. The celebrated result of Goldreich, Goldwasser, and Micali [GGM84] revealed the power of such a notion by showing its equivalence to pseudorandom generator and therefore one-way function by later works [GL89; HILL99]. Being simple and powerful, it serves as the starting point of many constructions to useful cryptographic primitives including message authentication, "memoryless" digital signature [Gol86], better obfuscation [App14], etc (see [BR17] for an excellent survey on this topic). From both practical and theoretical perspective, it is a natural problem to study the amount of computational recourse we need to construct pseudorandom function.

We investigate this problem in the context of circuit complexity. Let $n$ be the input length. The syntax of a pseudorandom function can be modeled as a collection $F_n$ of Boolean functions in $\{0,1\}^n \to \{0,1\}$ and a sampling distribution $\mathcal{D}_n$ supported over $F_n$ (for simplicity, we only consider single output PRFs). We consider PRFs that are secure against any probabilistic polynomial-time (p.p.t.) adversary. We can naturally define the circuit complexity of a PRF as the maximum complexity of functions in $F_n$ (see Section 4.4 and 4.6 for formal definition).

In this work, we present tight upper bounds and lower bounds of pseudorandom functions in general $B_2$ circuits, $\mathsf{NC}^1$ circuits, and $\mathsf{TC}^0$ circuits[1]. We sketch the results below.

- In general $B_2$ circuits, PRFs can be constructed in size $2n + o(n)$ assuming PRFs exist, simplifying and improving the $O(n)$ upper bound by Ishai, Kushilevitz, Ostrovsky, and Sahai [IKOS08]. We also prove that any PRF would require at least $2n - O(1)$ size circuit to compute, which is the first non-trivial (and already optimal) lower bound of PRF.

- In $\mathsf{NC}^1$ circuits, PRFs can be constructed in size $2n + o(n)$ and depth $(1+\varepsilon)\log n$ simultaneously for arbitrarily small $\varepsilon > 0$ assuming $\mathsf{NC}^1$ PRFs exist. A trivial lower bound says that any PRF would require at least $\log n - O(1)$ depth.

- In $\mathsf{TC}^0$ circuits of depth $d$, PRFs can be constructed in wire complexity $n^{1+O(\phi^{-d})}$, where $\phi = \frac{1+\sqrt{5}}{2}$, assuming $\mathsf{TC}^0$ PRFs exist. We give a matching lower bound saying that any PRF in $\mathsf{TC}^0$ requires wire complexity at least $n^{1+\Omega(c^{-d})}$ for some constant $c > \phi$. Both $O(\cdot)$ and $\Omega(\cdot)$ hides absolute constants independent of $n$ and $d$. Following the paradigm of natural proofs [RR97], our result can be interpreted as a barrier for current techniques to deal with sparse $\mathsf{TC}^0$ circuits, see Section 2 for detailed discussion.

**Natural proof barriers.** Another motivation of our work is to understand why proving circuit lower bounds are hard, following the Natural Proof barriers introduced by Razborov and Rudich [RR97]. Informally, they showed that certain kinds of proofs do not seem to be strong enough to prove super polynomial lower bounds since they break commonly-believed cryptographic assumptions. However, it is not capable to refute the existence of "slightly non-trivial" improvements to explicit lower bounds using current techniques.

On the other hand, realizing the (im-)possibility to improve known explicit lower bounds and derandomization results recently becomes a significant problem in circuit complexity since the discovery of a sequence of bootstrapping results (see, e.g., [AK10; OS18; Tel18; OPS19; CT19;

[1] $B_2$ circuits refers to the circuits in which each gate can compute arbitrary fan-in 2 Boolean functions. $\mathsf{TC}^0$ is the class of circuits with constant layers of unbounded-fanin linear threshold functions (see Section 4.2).

MMW19; CJW19; Che+20; CJW20]). These works reveal a mysterious phenomenon that a minor improvement of known results (for example, $n^{1+\varepsilon}$ lower bound for $B_2$ circuits on some natural problems) would imply a breakthrough (for example, $\mathsf{NP} \not\subseteq \mathsf{P}_{/\mathsf{poly}}$). Since the bootstrapping step may not be natural [Che+20], no strong evidence hints the impossibility to obtain a breakthrough by improving current results (even with natural proofs).

In this work, we make progress in understanding the limitation of this line of works. Following the duality of lower bound techniques and cryptanalysis [RR97], we are able to show that even slight improvements to known lower bounds and derandomization results could not be expected for certain kinds of techniques. The most interesting setting would be the *wire complexity* of constant-depth linear threshold circuits $\mathsf{TC}^0$, which has been the frontier of proving circuit lower bounds for years. The first explicit lower bound was given by Impagliazzo, Paturi, and Saks [IPS93], who proved a worst-case lower bound of $n^{1+\Omega(c^{-d})}$ form. Years later, a similar average-case lower bound (with larger $c$) was given by Chen, Santhanam, and Srinivasan [CSS18] following a rather different argument. With their new technique, similar results in quantified derandomization [Tel18; CT19] and the construction of pseudorandom generators [HHTT21] can even be proved.

We briefly illustrate our barriers in the context of quantified derandomization studied by Chen and Tell [CT19] (see Section 2.1 for details). In their work, they showed that quantified derandomization is possible for $n^{1+O(c^{-d})}$ circuits for some $c > 1$, and improving it to $n^{1+\Omega(1.61^{-d})}$ would imply breakthrough. We inspect the proof techniques and conclude the following.

- The proof of quantified derandomization for $n^{1+O(c^{-d})}$ circuits utilizes a structural lemma about sparse $\mathsf{TC}^0$ circuits, which implies an $n^{1+\Omega(c^{-d})}$ lower bound on $\mathsf{TC}^0$ circuits for PRFs.
- The proof of bootstrapping results for slightly improved quantified derandomization constructs an error-correctig code in sparse $\mathsf{TC}^0$ circuits, which implies an $n^{1+O(\phi^{-d})}$ wire complexity upper bound for PRFs assuming $\mathsf{TC}^0$ PRFs exist, where $\phi = \frac{1+\sqrt{5}}{2}$.

Essentially, this means that we cannot expect to get the breakthrough by simple improvements to any side of Chen-Tell's results. We note that $\mathsf{TC}^0$ PRF follows from standard cryptographic assumptions, such as factoring, decisional Diffie-Hellman [NR04], and ring learning-with-error [BPR12].

**Related works.** To study the circuit upper bounds of PRFs, one needs to rely on specific cryptographic assumptions. We focus on the weakest assumption possible: we study the upper bound merely based on the existence of pseudorandom functions. To handle things in this setting, *Levin's domain extension trick* tells us that universal hash functions can be used to reduce the circuit complexity of any PRF (see, e.g., [BR17] or Section 4.7 for more discussion). Combining with Spielman's error-correcting code [Spi96], Ishai, Kushilevitz, Ostrovsky, and Sahai [IKOS08] showed that universal hash function can be constructed in $O(n)$ size, which further implies linear-size construction of PRF and many other primitives. However, their construction is quite complicated and the multiplicative overhead (though constant) is huge, making it impossible to be implemented for real-world applications. Due to this weakness, [IKOS08] gives little structural insights about the power of small-size general circuits in constructing pseuodorandom primitives and cannot be generalized to restricted circuit classes (for example, $\mathsf{TC}^0$ circuits).

On the other hand, proving circuit lower bounds for particular functions could be unexpectedly hard. Although most people believe that $\mathsf{NP} \not\subseteq \mathsf{P/poly}$, the best explicit circuit lower bounds

we can prove are $3.1n - o(n)$ for $B_2$ circuits [LY21] and $5n - o(n)$ for $U_2$ circuits[2] [IM02]. In more restricted setting, we know $n^{2-o(1)}$ lower bound for $B_2$ formulas [Nec66], $n^{3-o(1)}$ lower bound for De Morgon formulas [And87; IN93; PZ93; Hås98; Tal14] and $n^{1+\Omega(2.42^{-d})}$ lower bound for $\mathsf{TC}^0$ circuits of depth $d$ [IPS93]. Super-polynomial lower bounds can only be proved up to $\mathsf{ACC}^0$ [MW20], and even $\mathsf{NEXP} \not\subseteq \mathsf{TC}^0$ remains to be open (see, e.g., [Che18; CT19]). For PRFs, prior work [RR97; KL01] only refutes the existence of pseudorandom functions in $\mathsf{AC}^0$, unweighted depth-2 threshold circuits and in $\mathsf{AC}^0[p]$ against quasi-polynomial adversaries for primes $p$. To the best of our knowledge, there is no known impossibility results of PRF on general circuits or $\mathsf{TC}^0$ circuits with large depth.

### 1.1 Our results

**General circuits.** We will begin with our bounds for general circuits. The first theorem gives a linear upper bound on the gate complexity of PRFs in general circuits. This will be proved as Corollary 5.13.

**Theorem 1.1.** If PRF exists, then there exists a PRF of circuit complexity $2n + o(n)$. ♢

The proof of this theorem (together with all the other upper bounds to be presented below) is constructive, in the sense that we can explicitly give an algorithm, which takes a polynomial-size PRF as input, and outputs a PRF of size $2n + o(n)$. This means that if the original PRF is uniform, then our newly constructed one is uniform as well. Indeed, uniform PRF follows from standard cryptography, such as the existence of one-way functions [GGM84; GL89; HILL99]. So it is plausible that p.p.t. adversary can be fooled even by functions with circuit complexity only $2n + o(n)$.

We then give an almost matching lower bound, saying that this construction cannot be improved significantly. This lower bound will be proved as Corollary 6.5.

**Theorem 1.2.** The circuit complexity of any PRF must be at least $2n - O(1)$. ♢

The proof of this lower bound follows from a completely combinatorial argument, which does not require uniformity. This essentially says that under the belief that PRFs exist, they should be computed in complexity exactly around $2n$, with no more and no less. We believe that this might give us more efficient constructions of other useful cryptographic primitives.

**$\mathsf{NC}^1$ circuits.** By the same techniques of constructing efficient PRFs in general circuits, we can additionally reduce the depth almost optimally. Indeed, the following theorem is proved as Corollary 5.15.

**Theorem 1.3.** If PRF in $\mathsf{NC}^1$ exists, then for any constant $\varepsilon > 0$, there exists a PRF computable by $2n + o(n)$ size and $(1 + \varepsilon) \log n$ depth circuits. ♢

We note that the output of PRFs should depend on all of its inputs, or we can distinguish it from truly random functions by identifying the unused input bit. This gives a trivial $\log n$ depth lower bound. Hence this theorem is also close to optimal.

[2] In $U_2$ circuits, gates can compute functions except for XOR and its complement.

**$\mathsf{TC}^0$ circuits.** Another restricted model we are interested in is constant-depth linear threshold circuits (denoted by $\mathsf{TC}^0$). We emphasize that we consider the number of wires in the circuit as size complexity. Our upper bound, which is presented below and proved as Corollary 7.2, tightly matches the parity upper bound in $\mathsf{TC}^0$ circuits. This is unavoidable since our proof utilizes a linear error-correcting code by Chen and Tell [CT19]. Previously, only candidates of $n^{1+O(1/d)}$ size based on much stronger assumptions are known [MV15]. Indeed, showing better candidates was proposed as an open problem by Chen and Tell [CT19].

**Theorem 1.4.** Let $\phi = \frac{1+\sqrt{5}}{2}$ be an absolute constant. If PRF exists in $\mathsf{TC}^0$ of depth $d_0$, then for any depth $d \geq d_0 + 4$, there exists a PRF computable by depth $d$ linear threshold circuits of size $n^{1+O(\phi^{-d})}$. ♢

We are also able to prove a nearly matching lower bound, which will be formally stated as Theorem 7.3. The proof of this theorem builds upon the random restriction paradigm which has already been used to prove size-depth trade-off lower bounds and derandomization for linear threshold circuits [CSS18; Tel18; CT19; HHTT21]. In fact, the constant $c$ in our theorem tightly matches the corresponding constants in the previous line of work.

**Theorem 1.5.** There exists an absolute constant $c$ such that for any $d \geq 1$, any depth $d$ linear threshold circuits computing a PRF should have size at least $n^{1+\Omega(c^{-d})}$. ♢

**Unconditional exact complexity of "almost" universal hash functions.** As a byproduct of our analysis, we are also able to get upper and lower bounds for a weaker variant of universal hash functions. These results are quantitatively similar to those for PRFs but are completely unconditional.

Recall that a hash function $H_n$ is a collection of functions mapping $n$-bit strings to $m$-bit strings. It is called *universal* if for any inputs $x \neq y \in \{0,1\}^n$, we have $\Pr_{h \leftarrow H_n}[h(x) = h(y)] = 2^{-m}$. However, in many cases, we do not need the collision probability to be exactly $2^{-m}$. We only need the collision to be not noticeable. This motivates the definition of *almost universal hash function*, where the requirement is loosened to $\Pr_{h \leftarrow H_n}[h(x) = h(y)] = \mathsf{negl}(n)$.

For almost universal hash functions, we can get unconditional constructions from our proofs of previous theorems. The first part of the following theorem (linear size and logarithmic depth) is proved as Lemma 5.14. The second part of the theorem (slightly superlinear size for linear threshold circuits) directly follows from Lemma 7.1 and Proposition 4.9. We can see that the depth upper bound is even slightly better than the one for PRFs.

**Theorem 1.6.** Universal hash functions with output length $m = n^{\Theta(1)}$ can be constructed by general circuits of size $2n + o(n)$ and depth $(1 + o(1)) \log n$ simultaneously, or depth $d$ linear threshold circuits of wire complexity $n^{1+O(\phi^{-d})}$ for any $d \geq 4$, where $\phi = \frac{1+\sqrt{5}}{2}$ is the absolute constant in Theorem 1.4. ♢

Similarly, unconditional lower bounds for PRFs can be adapted to almost universal hash functions. The general circuit lower bound is proved as Corollary 6.6, and the linear threshold circuit lower bound is proved as Theorem 7.7.

**Theorem 1.7.** Let $c$ be the absolute constant in Theorem 1.5. Any universal hash function with output length[3] $m = n^{o(1)}$ needs general circuits of size $2n - 2m$, or linear threshold circuits of wire complexity $n^{1+\Omega(c^{-d})}$ to compute, for any depth $d \geq 1$. ♢

By the general connection between almost universal hash functions and error-correcting codes (see Proposition 4.9), this lower bound also holds for error-correcting codes.

### 1.2 Organization of the paper

We will first discuss in Section 2 how our results give tighter barriers to particular proof techniques in circuit lower bounds. We then give some intuition on how the upper bounds and lower bounds are derived in Section 3. In Section 4, we will formally define our notations. We will also prove a generalized version of Levin's trick in this section, which is the starting point of our upper bound constructions. We then give technical proofs to upper bounds in general circuits in Section 5, then lower bounds in general circuits in Section 6. We will finally prove the results on linear threshold circuits in Section 7. Some open problems will be discussed in Section 8.

**Directions for readers** We expect readers with various backgrounds to be interested in our paper. For different interests, we give a guideline here for convenience.

- Readers interested in PRF (and almost universal hashing) complexity in general circuits are referred to Section 3.1 and 3.2, Section 4, Section 5 for upper bounds, and Section 6 for lower bounds. In preliminaries, one may skip Section 4.2, which are for linear threshold circuits.
- For readers interested in linear threshold circuits, and come for the lower bounds in $\mathsf{TC}^0$, we refer them to Section 3.3, Section 4, and Section 7. In overview and preliminaries, one may skip Section 3.1, 4.5 and 4.7 if you are not interested in how error-correcting codes give us better upper bounds.
- Readers interested in barriers in circuit lower bounds are referred to Section 2. The remaining parts are for proving the tight complexity results underlying the barriers. One may refer to them for detailed proofs.

## 2 Connections to circuit lower bound proofs

We now discuss how our exact complexity results give insights of many circuit lower bound results in the current frontier. Let us first review the natural proof framework by Razborov and Rudich [RR97]. Let $\Gamma$ and $\Lambda$ be standard complexity classes. A combinatorial property $\mathcal{C} = \{C_n \subseteq \{0,1\}^n \to \{0,1\}\}_{n\geq 1}$ over functions of input lengths $n$ is $\Gamma$-*natural useful against* $\Lambda$ if the following conditions holds.

**(Constructivity)** Given the truth table of a function $f_n : \{0,1\}^n \to \{0,1\}$ as input (note that the input length is $2^n$), the language indicating whether $f_n$ is in $C_n$ is decidable in $\Gamma$.

[3]One may note that there is a small gap between the output length of our upper bounds and lower bounds. In fact, our lower bound are proved even for $m = n^{\varepsilon}$, where $\varepsilon > 0$ is some constant. We present it here in this form to keep the statement clean and easy to read. One may refer to Theorem 7.7 for the exact statement.

**(Largeness)** For sufficiently large input length $n$, let $f_n : \{0,1\}^n \to \{0,1\}$ be a uniformly random function from all possible functions, then $\Pr[f_n \in C_n] \geq 1/\mathsf{poly}(n)$.

**(Usefulness)** For any language $L = \{L_n : \{0,1\}^n \to \{0,1\}\}_{n\geq 1} \in \Lambda$, there exists infinitely many input length $n$ such that $L_n \notin C_n$.

What Razborov and Rudich [RR97] observed is that such a property induces an algorithm in $\Gamma$ that distinguishes any function family in $\Lambda$ from truly random functions given the truth tables. Hence assuming the existence of exponentially hard PRFs against $\Gamma$ computable in $\Lambda$, such combinatorial properties do not exist. For a typical setting such as $\Gamma = \Lambda = \mathsf{P}_{/\mathsf{poly}}$, the existence of desired PRF follows from the existence of exponentially hard PRG via [GGM84].

Although natural proof paradigm successfully explains why proving super-polynomial circuit lower bounds is hard, it remains open to understand the difficulty in proving much weaker lower bounds such as $\mathsf{NP} \notin \mathsf{SIZE}[10n]$. This is because there is no plausible candidates of sufficiently hard PRFs in weak classes like $\mathsf{SIZE}[10n]$. For instance, there is no PRF in $\mathsf{SIZE}[10n]$ secure against $2^{n^{1.01}}$-time adversary, because to break PRF candidates in $10n$ size, the adversary can enumerate all $10n$ size circuits in $2^{O(n \log n)}$ time and check whether one of them realizes the truly table.

The main observation leading to our barrier is that some techniques such as random restriction (e.g. switching lemma of [Hås86] or simplifier sets characterization of [Tel17]) actually imply distinguishers which are much more efficient than the requirements of natural proofs. Take random restriction method for example. If we can show that certain type of circuits becomes constant under random restrictions with nice probability, we can distinguish it from truly random function with only polynomially many oracle accesses. This means that to refute the existence of such proofs, it is sufficient to use standard cryptographic PRFs, which can be constructed in weak classes like $\mathsf{SIZE}[2n + o(n)]$, $\mathsf{Formula}[n^{1.01}]$ or depth-$d$ threshold circuits of size $n^{1+O(c^{-d})}$ by our upper bounds.

Formally, we call a combinatorial property $\mathcal{C} = \{C_n\}_{n\geq 1}$ *black-box natural against* $\Lambda$ if, instead of constructivity, it has the following stronger property.

**(Black-box constructivity)** There exists an oracle p.p.t. algorithm $\mathcal{A}^O$ such that the following conditions hold.

- For any $f_n \in C_n$, we have $\Pr_{\mathcal{A}}\left[\mathcal{A}^{f_n}(1^n) = 1\right] \geq 2/3$.
- For uniformly random function $f_n$, we have $\Pr_{f_n,\mathcal{A}}\left[\mathcal{A}^{f_n}(1^n) \text{ accepts}\right] \leq 1/3$.

Note that the probability thresholds $\delta_1 = 2/3$ and $\delta_2 = 1/3$ are taken for simplicity of presentation and can be arbitrary functions of $n$ with non-negligible gap. In fact, the distinguishing algorithm induced by our PRF lower bound proofs satisfies $\delta_1 = 1$.

With this definition, we can immediately get the following impossibility result.

**Proposition 2.1.** Let $\Lambda$ be a complexity class. If (standard cryptographic) PRF can be constructed in $\Lambda$, then black-box natural properties against $\Lambda$ does not exist. ♢

- With Theorem 1.1, black-box natural properties against $\mathsf{SIZE}[2n + o(n)]$ does not exist, assuming PRF exists, which simply follows from the existence of one-way functions [HILL99]. We note that Theorem 1.2 in fact gives a black-box natural property against $\mathsf{SIZE}[2n - O(1)]$. One may check that known better-than-$2n$ explicit circuit lower bounds [Sto77; Pau77; Blu84; DK11; FGHK16; LY21], proved by *gate elimination*, highly rely on a non-black-box procedure that eliminates all gates by introducing constraints to input bits *cleverly* according to the circuit.

- With Theorem 1.3, black-box natural properties against $(1+\varepsilon)\log n$ depth circuits does not exist, assuming $\mathsf{NC}^1$ PRF exists. From the general connections between $\mathsf{NC}^1$ and formula complexity, this means that black-box natural properties against $n^{1+\varepsilon}$ size $B_2$-formulas should not exist for arbitrarily small $\varepsilon > 0$. We note that known quadratic lower bounds for $B_2$-formulas by Nechiporuk [Nec66] is natural in the sense of [RR97], but does not seem to be black-box natural because it utilizes a non-constructive counting argument.
- With Theorem 1.4, black-box natural properties against depth-$d$ $\mathsf{TC}^0$ circuits of wire complexity $n^{1+O(\phi^{-d})}$ does not exist, assuming $\mathsf{TC}^0$ PRF exists. We note that it is still open whether such an assumption can be derived from the existence of one-way functions, but it is known to follow from many standard cryptographic assumption, such as factoring, decisional Deffie-Hellman [NR04], and ring learning-with-error [BPR12]. On the other hand, the main lemma for our $\mathsf{TC}^0$ PRF lower bound (see Lemma 7.5), whose variants are also used by [CSS18; Tel18; HHTT21], can be viewed as a black-box natural property against $\mathsf{TC}^0$ circuits of wire complexity $n^{1+\Omega(c^{-d})}$.

We should note that learnability is a black-box natural property. Our PRFs refute the existence of learning algorithms for the concept classes of $2n + o(n)$ size general circuits, $n^{1+\varepsilon}$ size formulas and depth-$d$ $\mathsf{TC}^0$ circuits with wire complexity $n^{1+O(\phi^{-d})}$.

### 2.1 A case study: Chen-Tell's bootstrapping result

What makes black-box constructivity different from the standard constructivity is that the algorithm $\mathcal{A}$ no longer runs in exponential time $2^{O(n)}$, but in polynomial time $\mathsf{poly}(n)$. Still, if we rely on stronger cryptographic assumptions, we can loosen the running time requirement to super-polynomial time, say $2^{\mathsf{poly}\log n}$, which may slightly extend our barrier. This difference makes our barrier less general than natural proofs of Razborov and Rudich [RR97]: for instance, formula size lower bound proofs, including storage-access function against $B_2$-formulas [Nec66] and Andreev's function against De Morgan formulas [And87; IN93; PZ93; Hås98; Tal14], are natural but not likely to be black-box natural.

Nevertheless, our barrier explains the limitation of certain black-box techniques including random restriction method and its simple generalizations (e.g. random "affine" restriction). Most interestingly, it may give us some insights on the current frontier in linear threshold circuit, for example, the quantified derandomization results by Chen and Tell [CT19]. Recall that a linear threshold function $\mathsf{LTF}(x_1, x_2, \ldots, x_n)$ is defined as $\mathrm{sgn}(\sum_i w_i x_i - \theta_i)$. The class of constant depth linear threshold circuits $\mathsf{TC}^0$ contains circuits with constant layers of $\mathsf{LTF}$ functions. Also recall that the *quantified derandomization problem for circuit class $\mathcal{C}$ with exceptional inputs $B(n)$* is to distinguish, deterministically, the circuits in $\mathcal{C}$ which accept all but $B(n)$ of its inputs with those which reject all but $B(n)$ of its inputs. This problem was firstly introduced by Goldreich and Wigderson [GW14] and can be viewed as a generalization of standard derandomization problem (i.e. $B(n) = 2^n/3$). Under this framework, Chen-Tell's bootstrapping result [CT19] roughly says the following.

- There exists a constant $c_1 > 1$, such that the quantified derandomization problem for $\mathsf{TC}^0$ circuits of size $n^{1+O\left(c_1^{-d}\right)}$ with $B(n) = 2^{n^{1-O\left(c_1^{-d}\right)}}$, can be solved. This follows from a pseudo-random restriction lemma for linear threshold functions [CSS18; Tel18].
- Let $c_2$ be any constant smaller than $\phi = \frac{1+\sqrt{5}}{2}$, if the quantified derandomization problem for $\mathsf{TC}^0$ circuits of size $n^{1+\Omega\left(c_2^{-d}\right)}$ with $B(n) = 2^{n^{1-c_2^{-d}}}$ can be solved, then there exists an

algorithm for standard derandomization of $\mathsf{TC}^0$. This result utilizes error reduction by extractors, following the idea of Goldreich and Wigderson [GW14]. Their main contribution is an extremely efficient construction of extractors in $\mathsf{TC}^0$, by constructing an efficient error correcting code by the expander in [CRVW02] and then plugging it into Trevisan's extractor [Tre01; RRV02].

Similar to other sharp threshold results in circuit lower bounds, Chen-Tell's result can be interpreted in either optimistic or pessimistic perspective. In an optimistic point of view, this provides a new (and seemingly viable) approach to solve a long-standing open problem (which is derandomization of $\mathsf{TC}^0$ in this setting). In a pessimistic point of view, however, this suggests the inherent difficulty to improve current results. The pessimists can be further divided into two kinds with entirely different opinions: *pure pessimists* who believes (maybe with or without formal evidence) that the breakthrough itself is intractable; and *technical pessimists* who thinks the desired improvement to current results is out of reach with known techniques. In some sense, our results provide justifications for technical pessimists. Let us again take Chen and Tell [CT19] for example.

- The only known technique to deal with average-case hardness and derandomization of general $\mathsf{TC}^0$ circuits is the random (or pseudorandom) restriction framework [CSS18; Tel18; HHTT21], which essentially utilizes a variant of a restriction lemma (see Lemma 7.5). Although this lemma seems to be non-black-box, we can indeed extract a black-box property by a clever sampling (see Lemma 7.6). By Proposition 2.1, this suggests that it is hard to obtain the breakthrough by improving Chen-Tell's quantified derandomization algorithm with simple extensions of the restriction framework[4]. We will need to find some other non-black-box approaches, unless one wish to break factoring, DDH and ring learning-with-error.
- Recall that the bootstrapping step of Chen-Tell's result utilizes a linear error-correcting code. Constructing an efficient ECC is the most natural way to lower the bootstrapping threshold and hence bring us closer to the breakthrough. From the parity lower bound of [IPS93], [CT19] noted that linear codes cannot give us better results, but leave the non-linear code case as an interesting direction. From our results, we are able to argue that even non-linear error-correcting codes would not work. This is because any efficient error-correcting code can be translated into a PRF upper bound (see Section 4.5 and 4.7) and thus cannot go below known lower bounds assuming the existence of $\mathsf{TC}^0$ PRF. In fact, this can be made unconditional if we use the unconditional construction of almost universal hash from ECC (see Proposition 4.9), together with the unconditional lower bound of almost universal hash functions.

## 2.2 Comparisons to previous barriers

**Natural proofs.** Natural proofs are inevitably the most notable barrier in these decades. Our results can be seen as a modification of the standard natural proof barrier, since we are arguing the non-existence of a more black-box type of proofs basing on weaker assumptions. Indeed, we utilize this weakness in assumption to reduce down the complexity of the cryptographic primitives, and get very tight barriers, in the sense that such black-box proofs cannot be improved even

[4]The restriction lemma roughly shows that a random restriction followed by a white-box restriction can eliminate a layer of a sparse $\mathsf{TC}^0$ circuit. One may observe that the upper bound of parity function already implies that one cannot directly utilize the current random restriction technique to prove better lower bound results. However, our black-box natural barrier could be more general: even slightly extending the restriction used, the white-box restriction strategy or the hard function used, should not work.

slightly. Comparably, we are not able to get such tight barriers in the standard natural proofs context, since reducing the complexity of such strong assumptions (the exponential hardness assumed by natural proofs) would be difficult. This is indeed, explicitly noted by Chen et al. [Che+20] in the context of hardness magnification.

**Tell's barrier on quantified derandomization.** Another related barrier is the one by Tell [Tel17]. It says that it is impossible to obtain standard derandomization by combining a quantified derandomization algorithm with "black-box" random restriction and an error reduction procedure by seeded extractor, since these two techniques somewhat contradict with each other. Although his barrier is unconditional, the requirement of "black-box" random restriction is stronger than black-box natural proofs. Typically, the random restrictions used in $\mathsf{TC}^0$ circuits [CSS18; Tel18; HHTT21] are black-box natural, but they does not naturally fit into Tell's characterization of "black-box random restriction" as he explicitly noted in [Tel17].

Parallel to our work, Tell [Tel21] claimed very recently that his barrier can be generalized to the $\mathsf{TC}^0$ setting. His barrier and ours provide completely different view of Chen-Tell's bootstrapping result, and indeed the results are incomparible with each other. Also, Tell's barrier only works for quantified derandomization, while our results are general enough to apply to all tasks related to proving circuit lower bounds.

# 3 Proof overview

We now give intuition to how we prove the results. We will first show our general paradigm of reducing the circuit complexity of PRFs from efficient constructions of almost universal hash functions in Section 3.1. This is done via a generalization of the seemingly folkloric technique called Levin's trick. We then briefly show how almost universal hash functions are constructed in different models. Then we discuss the way of proving lower bounds in general circuits and linear threshold circuits in Section 3.2 and 3.3 respectively.

## 3.1 Upper bounds: Levin's trick and hash function

Our general paradigm of proving circuit upper bounds for pseudorandom functions is a generalization of the standard *domain extension* technique called *Levin's trick*. Informally, it states that we can construct PRF with input length $n$ as follows: we firstly shrink the $n$-bit input $x$ to $n^\varepsilon$-bit hash value $h(x)$ by a uniform *almost universal hash function*[5] and then feed $h(x)$ to a PRF with inputs length $n^\varepsilon$.

The reason why Levin's trick may improve the efficiency of PRF is that hash function is a combinatorial (instead of cryptographic) primitive so that its complexity can be much lower. Assume the original PRF has circuit complexity $n^c$, we can choose $\varepsilon < \frac{1}{c}$ so that the "pseudorandom kernel" has complexity $o(n)$, and therefore the complexity of resulting PRF mainly depends on the complexity of hash function. This reduces the construction of efficient PRF to the problem of designing low-complexity almost universal hash functions.

Previously, the efficient construction of almost universal hash functions is built upon efficient error-correcting codes, for example, Spielman's linear-size encodable code [Spi96]. Although this is sufficient to prove an $O(n)$ upper bound for PRF, the constant factor hidden in big-$O$ is rather

[5]Note that a hash function is called almost universal if for all distinct pair of inputs, their hash values collide with only negligible probability, see Section 4.

hard to analyze. Inspired by the constructions of efficient ECC [GDP73; Spi96; CT19], we observe that one can construct efficient hash function with a much simpler primitive, which we call it *1-detector*. Intuitively, a 1-detector is a linear function $D : \mathbb{F}_2^n \to \mathbb{F}_2^m$ such that for all inputs $x \neq 0$ with Hamming weight smaller than a threshold $r$, $D(x)$ is not identically zero. If we view $D$ as a hash function, it guarantees that the hash value of distinct pairs $(x, y)$ with small Hamming distance will not collide. In addition, assume that $r$ is appropriately large and $m$ is small, we can also avoid (with high probability) the collisions between pairs $(x, y)$ with large Hamming distance by involving a random subset of input bits in the hash value.

Following this observation, it is now sufficient to construct (uniformly constructible) 1-detectors with small circuit complexity and nice trade-off between parameters $r$ and $m$. In fact, the existence of (non-uniform) 1-detectors has been shown in [GDP73] by standard probabilistic method, which cannot be made uniform due to technical reasons. We provide two ways to overcome this issue.

1. Although 1-detector induced by [GDP73] (which has circuit complexity $3n$) is not uniform, we can still sample a 1-detector with small failure probability. By an error-reduction trick, for all integer $d > 0$, we can construct a p.p.t. algorithm $\mathcal{A}$ such that $\mathcal{A}(1^n)$ generates a 1-detector with failure probability at most $n^{-d}$. We call such 1-detector (and corresponding hash function) *weakly uniform*. We generalize Levin's trick for non-uniform PRF, and show that it can be adapted for weakly uniform hash function. This leads to a $3n + o(n)$ circuit upper bound for (non-uniform) PRF. Although this construction has a larger circuit size, its collision probability can be reduced to much smaller than the $2n$ construction below. We think that this may lead to independent interest.

2. To improve the upper bound to $2n$, we can no longer rely on the probabilistic method in [GDP73]. However, we can again slacken the requirement: to construct hash functions, we can allow the 1-detectors to be randomized. In particular, we define *randomized 1-detector* as a linear function $D : \mathbb{F}_n^2 \to \mathbb{F}_m^2$ such that for all $x \neq 0$ with small Hamming weight, for a random permutation $\rho$ of input bits, $D(\rho(x)) \neq 0$ with high probability. Perhaps surprisingly, such primitive with circuit complexity $2n$ can be uniformly constructible using *graphs with large girth*, whose explicit construction has been extensively studied in combinatorics. This leads to a $2n + o(n)$ PRF upper bound for both uniform and non-uniform setting.

### 3.2 Lower bounds in general circuits

We prove our $2n - O(1)$ circuit lower bound with the following two steps. Firstly, we define a combinatorial property $\mathcal{P}$ about $B_2$ circuits such that there exists a p.p.t. algorithm $\mathcal{A}$ that distinguishes circuits with such property and truly random functions. Then we prove by wire counting that all circuits with complexity $2n - O(1)$ have property $\mathcal{P}$, so that our algorithm $\mathcal{A}$ can be used to break PRF candidates with complexity $2n - O(1)$.

Let $C$ be a circuit and $I$ be the set of variables. For simplicity, we assume that each variable in $C$ has out-degree at least 1. We define the *critical path* of a variable $x \in I$ as the set of nodes reachable from $x$ via nodes with out-degree exactly 1. The (black-box natural) combinatorial property we will utilize is: *$C$ contains two variables with intersecting critical paths*. For a circuit $C$ with such property, there exists $x, y \in I$ and a Boolean function $G$, for all restrictions $\rho$ to $I \setminus \{x, y\}$, the type[6] of $C\restriction_\rho(x, y)$ only depends on the type of $G$. This can be easily separated from truly random

[6]We can classify Boolean functions in $\mathbb{F}_2^2 \to \mathbb{F}_2$ into four types: trivial functions that output a constant, degenerate functions that depends on only one of its input, $\oplus$-type functions that are linear over $\mathbb{F}_2$ and $\wedge$-type functions that are quadratic. See Section 4.1 for details.

functions $f$ because the type of $f\restriction_\rho$ is independently randomly chosen for each $\rho$.

Now we only need to prove a lower bound for circuits without intersecting critical paths (and no variable of out-degree zero). This is done via a standard wire-counting argument. Intuitively, the $n$ non-intersecting critical paths should give roughly $2n$ out-wires at their terminals. By analyzing the number of wires between the critical paths and the other part of the circuit, we can show that we need about $2n$ gates to handle all these out-wires.

### 3.3 Lower bounds in linear threshold circuits: random restriction

This part of the result follows the random restriction method which has already been used extensively on $\mathsf{TC}^0$ circuits to prove lower bounds and pseudorandom generators. We first review the effect of random restriction on a layer of linear threshold functions. We will show that after a random restriction and a cleverly chosen restriction (based on the former random restriction and the circuit), with nice probability, we can eliminate a full layer of linear threshold gates while keeping a good fraction of variables alive. This is done by considering variables and gates with different degrees separately.

**Large variables.** For variables with a large out-degree, we arbitrarily fix them to a constant. There will not be many such variables, since each of them contributes lots of edges.

**Small variables and small gates.** Since variables with a large out-degree has already been removed from the circuit, we can assume that all variables have out-degree not too large. In this case, we can choose a large subset of variables by a graph-theoretic argument, such that each gate *of small in-degree* is fed at at most one chosen variable. Then by fixing all others, we can make all these small gates depend on only one of its inputs, hence can be eliminated.

**Small variables and large gates.** Now only small variable and large gates remains. We do a random restriction to all the variables, then argue that each gate has a good probability of being extremely biased by anti-concentration bounds. We can hence approximate these biased gates by constants. Since there is not too much unbiased gates remain in expectation, we can remove them by fixing all of their inputs.

What [CSS18] argued is that we can carefully choose the parameters such that after the above three processes, there are still sufficiently many (say $n^{0.99}$) variables unfixed. The starting point of our argument is that the process above can be intuitively abstracted as follows.

> *Suppose that the variable set is $I$. We firstly take a random restriction to all variables. Let $I'$ denote the variables kept alive by the random restriction. With nice probability, there exists a large subset $S \subseteq I'$ such that randomly fixing all variables not in $S$ would eliminate a full layer of gates with high probability.*

The main technical difficulty in proving PRF lower bound is to extract a black-box natural property from this white-box argument: if we do not need to choose such $S$ according to the circuit, we can simply repeat the above process for $d$ times so that the circuit would be trivialized. The key to bypass this issue is to define another distinguishing procedure whose correctness is implied by the restriction lemma. We define an input $x$ to be good w.r.t. a $\mathsf{TC}^0$ circuit $c$ if flipping a bit of $x$ would not deviate the output of $C$. If each sparse $\mathsf{TC}^0$ circuit of depth $d$ has a large fraction of good inputs, we can also argue according to the restriction lemma that each $\mathsf{TC}^0$ circuit of depth $d+1$ has a large fraction of good inputs. By induction, we can conclude as follows.

*Let $C$ be any sparse $\mathsf{TC}^0$ circuit. For a uniformly random input $x$ and an input $y$ obtained by flipping a random bit of $x$, $C(x) = C(y)$ with non-negligible probability.*

This property itself does not suffice to break PRF candidates in sparse $\mathsf{TC}^0$, because a truly random function $f$ also has a large fraction of good inputs (for any $x \neq y$, $f(x) = f(y)$ with probability $1/2$). To deal with this issue, we transform the PRF candidate $F$ we want to break into a PRF $F'$ with output length $\log^2 n$ by expanding the last $2 \log \log n$ input bits, which would not increase its circuit complexity significantly. It is easy to verify that the property above still holds for multi-output functions. For truly random function, however, the probability that $f(x) = f(y)$ for $x \neq y$ reduces to $2^{-\log^2 n}$. This makes it possible to distinguish $F'$ (and therefore $F$) from truly random function.

# 4 Preliminaries

Throughout this paper we define $[n] \triangleq \{1, 2, \ldots, n\}$, $B_n \triangleq \mathbb{F}_2^n \to \mathbb{F}_2$ as the set of single-output Boolean functions with $n$ inputs and $B_{n,m} \triangleq \mathbb{F}_2^n \to \mathbb{F}_2^m$ as the set of $m$-output Boolean functions with $n$ inputs. We represent Boolean AND function with $\wedge$ and Boolean XOR function with $\oplus$. For binary strings $x$ and $y$ of length $n$, the *Hamming weight* $|x|$ is defined as the number of 1-entries in $x$, and the *Hamming distance* $\Delta(x, y) \triangleq |x \oplus y|$ is defined as the Hamming weight of point-wise XOR of $x$ and $y$. The *relative distance* $\Delta_r(x, y)$ is defined as $\Delta(x, y)/n$. We use $x \| y$ to denote the concatenation of two bit string $x$ and $y$.

All the graphs $G = (V, E)$ are undirected in default. A cycle in a graph $G$ is a subset of vertices $\{v_0, v_1, \ldots, v_{\ell-1}\}$ such that there is an edge between $v_i$ and $v_{(i+1) \bmod \ell}$ for all $0 \leq i < \ell$. We follow standard notations for probability and expectation, where $x \leftarrow \mathcal{D}$ represents that $x$ is a random variable sampled according to the distribution $\mathcal{D}$. In particular, for any finite set $S$, $x \leftarrow S$ means that $x$ is the random variable sampled according to uniform distribution supported on $S$.

Without further clarification, pseudorandom functions are meant to be secure against uniform probabilistic polynomial time (p.p.t. for short) adversary.

## 4.1 Boolean circuits

A *Boolean circuit* (or $B_2$ *circuit*) is a directed acyclic graph where each vertex is either a *variable* of in-degree 0 or a *gate* of in-degree 2. Each variable is labeled with an index identifying its corresponding input bit, and each gate has a corresponding Boolean function out of $B_2$. One or more nodes are marked as output nodes, each of which is labeled with a set of indices identifying the corresponding output bits[7]. During evaluation, we decide the output of each gate according to its corresponding function in topological order. We say a circuit $C$ computes a function $f \in B_{n,m}$ if $C$ contains exactly $n$ variables and $m$ output nodes, and it agrees with $f$ on all inputs in $\mathbb{F}_2^n$.

According to the functionality, we can classify the 16 gates out of $B_2$ into four types: *trivial gates* that compute constant functions (i.e. $f(x, y) = c_1$); *degenerate gates* that only depend on one of their inputs (i.e. $f(x, y) = x \oplus c_1$ or $f(x, y) = y \oplus c_2$); $\oplus$*-type gates* that compute linear functions (i.e. $f(x, y) = x \oplus y \oplus c$); and $\wedge$*-type gates* that compute quadratic functions (i.e. $f(x, y) = ((x \oplus c_1) \wedge (y \oplus c_2)) \oplus c_3$). It is easy to see that an optimal circuit computing any function $f$ does not contain trivial and degenerate gates, since we can always remove them and properly rewire the circuit while keeping functionality of the circuit.

[7] That is, a node can have more than one corresponding output bits.

The *size* of a circuit is defined as the number of gates involved. The *circuit complexity* of $f$, denoted by $B_2\text{-}\mathsf{CC}(f)$ or simply $\mathsf{CC}(f)$, is defined as the size of the smallest circuit computing $f$. The *depth* of a circuit is defined as the number of edges in the longest path of the graph.

## 4.2 Threshold function and threshold circuits

Another computation model we are interested in this paper is *linear threshold circuit*. For notational convenience, we represent Boolean value by $\{1, -1\}$ instead of $\{0, 1\}$ when talking about linear threshold circuits (i.e. we represent `true` by $-1$ and `false` by 1, so that XOR is simply multiplication). We will also often omit "linear" since we will not consider any non-linear threshold functions in this paper.

**Definition 4.1 (Linear threshold function).** Let $w \in \mathbb{R}^m$ and $\theta \in \mathbb{R}$, a *linear threshold function* (or simply *threshold function*) corresponding to weights $w$ and threshold $\theta$ is defined as $\mathsf{LTF}_{w,\theta}(x) \triangleq \mathrm{sgn}(\langle w, x\rangle - \theta)$, where $\langle \cdot, \cdot \rangle$ is the standard inner product of real vectors and $\mathrm{sgn}(x)$ is the sign function. ♢

A *linear threshold circuit* (or simply *threshold circuit*) is a direct acyclic graph where each vertex is either a variable corresponding to an input bit, or a gate of arbitrary in-degree labeled with a threshold function. Similar to $B_2$ circuits, one or more nodes of a threshold circuit are marked as output nodes.

The *depth* of a vertex in a threshold circuit is defined as the number of edges in the longest path from any variable to it. The *depth* of the threshold circuit is the maximum depth of all vertices. The *size* of a threshold circuit is defined as the number of *wires* (i.e. *edges*) in it. As a convention, we use $\mathsf{TC}^0_d$ to denote the class of depth-$d$ circuits, and the $\mathsf{TC}^0$-circuit complexity $\mathsf{TC}^0_d\text{-}\mathsf{CC}(f)$ represents the minimum size of $\mathsf{TC}^0_d$ circuits to compute $f$. To compute a function $f \in B_n$ with threshold circuit, there is usually a trade-off between the depth and the size. For example, Paturi and Saks [PS94] shows that the parity function $\bigoplus_n(x_1, \ldots, x_n) \triangleq x_1 \oplus \cdots \oplus x_n$ can be computed by depth $d$ threshold circuits of size $n^{1+O(1)^d}$. A matching $n^{1+\Omega(1)^d}$ lower bound is also given in Impagliazzo, Paturi, and Saks [IPS93].

**Theorem 4.2 ([PS94; IPS93]).** Let $\phi_1 = 1 + \sqrt{2}$ and $\phi_2 = (1 + \sqrt{5})/2$. There exist absolute constants $c_1$ and $c_2$, such that for sufficiently large $n$, $\mathsf{TC}^0_d\text{-}\mathsf{CC}(\bigoplus_n) \in [n^{1+c_1\phi_1^{-d}}, n^{1+c_2\phi_2^{-d}}]$. ♢

## 4.3 Restriction

To prove PRF lower bounds against $B_2$ and $\mathsf{TC}^0_d$ circuits, we need to define the notation of *restriction*. A *restriction* $\rho$ is a mapping from input bits to $\{0, 1, \star\}$[8], where those bits mapped to $\star$, denoted by $\rho^{-1}(\star)$, are called *unfixed* or *free* bits. Let $f \in B_{n,m}$ be a Boolean function and $\rho : [n] \to \{0, 1, \star\}$ be a restriction. We can then define the restricted function $f\restriction_\rho \in B_{|\rho^{-1}(\star)|,m}$ as the function over unfixed bits obtained by fixing the $i^{\text{th}}$ bit as $\rho(i)$ for each $i \in \rho^{-1}(\{0, 1\})$.

**Definition 4.3 (Random restriction).** Let $n$ be the number of input bits. A *random $p$-restriction* (or *$p$-restriction*) is the following distribution $\mathcal{R}^n_p$ over all restrictions: independently for each input bit, we set it to $\star$ with probability $p$ and to 0 and 1 with probability $(1 - p)/2$ each. ♢

[8] Or $\{1, -1, \star\}$ when we are working with threshold circuits.

During probabilistic arguments, it may be convenient to view a random $p$-restriction as a pair $(S, y)$ of random variables, where $S$ denotes the set of fixed bits and $y \in \mathbb{F}_2^{|S|}$ refers to the assignment to the fixed bits. Condition on a particular $S$, the distribution of the assignment $y$ is uniformly chosen from $\mathbb{F}_2^{|S|}$.

## 4.4 Pseudorandom function

The syntax of pseudorandom functions consists of a collection of functions $F_n \subseteq B_n$ and a distribution $\mathcal{D}_n$ supported over $F_n$, both of which are labeled by the input length $n$. In this work, we assume that PRF is defined for all input length. For convenience, we represent a PRF as $\mathcal{F} = \{F_n \subseteq B_n\}_{n\geq 1}$, implicitly keep the distribution $\mathcal{D}_n$ in mind and denote the sampling procedure simply by $f \leftarrow F_n$.

**Definition 4.4 (Negligible function).** A function $\varepsilon(n)$ is called *negligible* if for all $c > 0$ and sufficiently large $n$, we have $\varepsilon(n) < n^{-c}$. We use the notation $\mathsf{negl}(n)$ to mean an arbitrary negligible functions w.r.t. $n$ if there is no ambiguity. ♢

**Definition 4.5 (Indistinguishability).** Two function families $\mathcal{F} = \{F_n \subseteq B_n\}_{n\geq 1}$ and $\mathcal{G} = \{G_n \subseteq B_n\}_{n\geq 1}$ are *indistinguishable*, denoted by $\mathcal{F} \approx_c \mathcal{G}$, if for all p.p.t. adversary $\mathcal{A}^O$ with oracle access to $O$, there exists a negligible function $\varepsilon(\cdot)$ such that

$$\left| \Pr_{f\leftarrow F_n, \mathcal{A}}[\mathcal{A}^f(1^n) = 1] - \Pr_{g\leftarrow G_n, \mathcal{A}}[\mathcal{A}^g(1^n) = 1] \right| \leq \varepsilon(n).$$

♢

One can easily verify that the indistinguishability relation is an equivalence relation, i.e. it is transitive, symmetric and reflexive.

**Definition 4.6 (Pseudorandom functions).** A *pseudorandom function* (PRF) is a family $\mathcal{F} = \{F_n \subseteq B_n\}_{n\geq 1}$ that is indistinguishable from truly random function $\mathcal{B} = \{B_n\}_{n\geq 1}$. ♢

## 4.5 Hash function and error-correcting code

Let $n$ be the input length. Similar to PRF, the syntax of a hash function is defined by a family of functions $\mathcal{H} = \{H_n \subseteq B_{n,m}\}$ and a family of distribution $\mathcal{D}_n$ supported over $H_n$. Again, we will simply omit the distribution and denote the sampling procedure by $h \leftarrow H_n$.

**Definition 4.7 (Hash functions).** Let $m = m(n)$ be a function, a *hash function* is a family $\mathcal{H} = \{H_n \subseteq B_{n,m}\}_{n\geq 1}$. It is called *universal* if for all $n$ and $x \neq y \in \mathbb{F}_2^n$,

$$\Pr_{h\leftarrow H_n}[h(x) = h(y)] = 2^{-m}.$$

It is called *almost universal* if there exists a negligible function $\varepsilon(\cdot)$ such that for all $n$ and distinct inputs $x, y \in \mathbb{F}_2^n$,

$$\Pr_{h\leftarrow H_n}[h(x) = h(y)] \leq \varepsilon(n).$$

♢

**Definition 4.8 (Error-correcting code).** Let $m = m(n) > n$ be a function. A function family $E = \{\mathrm{Enc}_n \in B_{n,m}\}_{n\geq 1}$ is called an *error-correcting code* with relative distance $\delta \in (0, 1)$ if for sufficiently large $n$, for all $x \neq y \in \mathbb{F}_2^n$, the Hamming distance $\Delta(\mathrm{Enc}_n(x), \mathrm{Enc}_n(y))$ is at least $\delta$. Moreover, $E$ is called *systematic* if the encoding function can be interpreted as $\mathrm{Enc}_n(x) = x \| \mathrm{Par}_n(x)$, where the last $m - n$ bits generated by $\mathrm{Par}_n$ is called the *parity-checking bits*. ♢

There is a simple construction of almost universal hash function from error-correcting code, which is probably folklore. Let $0 < \varepsilon < 1$, $m = \Theta(n^\varepsilon)$ be the desired output length and $\mathrm{Enc}_n \in B_{n,m'}$ be an encoding function with relative distance $\delta$. The hash function $H_n^{\mathrm{Enc}}$ is defined as the collection of all functions $h_S$ indexed by subsets $S = \{i_1, i_2, \ldots, i_m\} \subseteq [m']$ of size exactly $m$, such that

$$h_S(x) \triangleq \mathrm{Enc}_n(x)_{i_1} \| \mathrm{Enc}_n(x)_{i_2} \| \ldots \| \mathrm{Enc}_n(x)_{i_m}.$$

That is, the hash function $\mathcal{H}^{\mathrm{Enc}}$ selects a random $m$-subset of the output of $\mathrm{Enc}_n(x)$. Clearly, this construction does not increase the circuit complexity of the function since we only need to relabel the output nodes.

**Proposition 4.9.** $\mathcal{H}^{\mathrm{Enc}} = \{H_n^{\mathrm{Enc}}\}_{n \geq 1}$ is almost universal. ♢

**Proof.** Let $x \neq y$ be distinct inputs of length $n$. By the distance property of error-correcting code, $\mathrm{Enc}_n(x)$ and $\mathrm{Enc}_n(y)$ have Hamming distance at least $\delta m'$. The probability that $h(x) = h(y)$ given $h \leftarrow H_n^{\mathrm{Enc}}$ can be bounded by

$$\Pr_{h \leftarrow H_n^{\mathrm{Enc}}}[h(x) = h(y)] \leq \frac{\binom{(1-\delta)m'}{m}}{\binom{m'}{m}} = \prod_{0 \leq i < m} \frac{(1-\delta)m' - i}{m' - i} \leq (1-\delta)^m.$$

Recall that we take $m = \Theta(n^\varepsilon)$, so this would be negligible whenever the relative distance $\delta$ of the error-correcting code is constant. □

## 4.6 Circuit complexity and uniformity

Both pseudorandom functions and hash functions are defined as families of function collections $\mathcal{F} = \{F_n \subseteq B_n\}_{n \geq 1}$, hence there are several ways to define the complexity of them. For example, one can define the complexity of $\mathcal{F}$ as the complexity to sample the distribution $\mathcal{D}_n$. In this paper, however, we define the complexity of $\mathcal{F}$ just as the maximum circuit complexity of $f \in F_n$.

**Definition 4.10 (Circuit complexity of a function collection).** Let $\mathcal{C}$ be a circuit class (for example, $B_2$ or $\mathsf{TC}^0_d$). For a family of function collection $F = \{F_n \subseteq B_n\}_{n \geq 1}$, the $\mathcal{C}$-circuit complexity[9] of $\mathcal{F}$, denoted by $\mathcal{C}\text{-}\mathsf{CC}(\mathcal{F})$, is defined as the size function $s(n) \triangleq \max_{f \in F_n} \mathcal{C}\text{-}\mathsf{CC}(f)$. ♢

In default, pseudorandom functions and hash functions can be *non-uniform*, i.e. there is no requirement on the complexity of generating the circuits for a function $f \in F_n$ given the corresponding key. We can also define *uniform* counterparts of these primitives.

**Definition 4.11 (Uniformity of collection).** Let $\mathcal{C}$ be a circuit class. A family $\mathcal{F} = \{F_n\}_{n \in \mathbb{N}}$ (with sampling distribution $\mathcal{D}_n$ over $F_n$) is called a *uniform-$\mathcal{C}$* family if there exists a p.p.t. algorithm $\mathcal{G}$, such that for all $n$ and $f \in F_n$,

$$\mathcal{D}(f) = \Pr_{\mathcal{G}}[\mathcal{G}(1^n) \text{ outputs a } \mathcal{C}\text{-circuit computing } f].$$

Similarly, it is called a *weakly uniform-$\mathcal{C}$* family if for all $d \in \mathbb{N}$, there exists a p.p.t. algorithm $\mathcal{G}$ and an event $\mathcal{E}$ (denoting whether $\mathcal{G}$ successes) such that for all $n$,

$$\Pr_{\mathcal{G}(1^n)}[\mathcal{E}] \geq 1 - \frac{1}{n^d},$$

[9] For simplicity of presentation, when we say $\mathcal{C}\text{-}\mathsf{CC}(\mathcal{F}) \leq s_1(n)$ (or $\mathcal{C}\text{-}\mathsf{CC}(\mathcal{F}) \geq s_2(n)$) in the rest of the paper, we always mean that the inequality holds for sufficiently large $n$. This means that both our upper bounds and lower bounds are "almost everywhere" versions.

and for all $f \in F_n$,

$$\mathcal{D}(f) = \Pr_{\mathcal{G}}[\mathcal{G}(1^n) \text{ outputs a } \mathcal{C}\text{-circuit computing } f \mid \mathcal{E}].$$

We say a collection has *uniform complexity* (or *weakly-uniform complexity*) $s(n)$, if it is uniform (or weakly uniform), and the generated circuits is of size at most $s(n)$ for sufficiently large $n$. ♢

We introduce the notion of *weak uniformity* mainly due to technical reasons. We require our primitives to have a certain degree of uniformity even when we consider non-uniform PRF, since the adversary is uniform. Intuitively, a family is weakly uniform if one can sample from the family with error probability $n^{-d}$ for arbitrarily large $d$.

For simplicity, we may omit the circuit class if $\mathcal{C}$ is the class of $B_2$ circuits.

### 4.7 Levin's trick for domain extension

One of the key tools to prove circuit upper bounds for PRF is the Levin's trick for domain extension. It shows that one can construct a PRF $F_n \subseteq B_n$ with a PRF $F'_m \subseteq B_m$ for $m = \Omega(n^\varepsilon)$ by hiding $F'$ behind a uniform universal hash function. We will need a generalized version of Levin's trick by allowing the hash function to be almost universal and weakly uniform.

**Lemma 4.12 (Levin's trick, generalized).** Let $\mathcal{F} = \{F_n \subseteq B_n\}_{n\geq 1}$ be a PRF and $\mathcal{H} = \{H_n \subseteq B_{n,m}\}_{n\geq 1}$ be an almost universal hash function of polynomial weakly-uniform complexity with $m = m(n) = \Theta(n^\varepsilon)$ for some absolute constant $0 < \varepsilon < 1$. The composition of $\mathcal{F}$ and $\mathcal{H}$, i.e. $\mathcal{F}' = \{F'_n\}_{n\geq 1}$ for $F'_n = \{f \circ h \mid f \in F_m, h \in H_n\}$, is also a PRF. ♢

**Proof.** Let $\mathcal{B}'$ be the composition of $\mathcal{B} = \{B_m\}_{m\geq 1}$ and $\mathcal{H}$, i.e., $B'_n = \{f \circ h \mid f \in B_m, h \in H_n\}$. It is known that $\mathcal{B}'$ is indistinguishable from truly random functions $\mathcal{B} = \{B_n\}_{n\geq 1}$ (for completeness, we present a complete proof in Appendix A). By transitivity, it suffices to show that $\mathcal{F}'$ is indistinguishable from $\mathcal{B}'$.

Towards a contradiction assume that $\mathcal{F}'$ and $\mathcal{B}'$ are distinguishable, then there exists a p.p.t. adversary $\mathcal{A}$ such that there exists a constant $c$ and infinitely many bad input length, say $n \in \{n_1, n_2, \dots\}$, such that

$$\left| \Pr_{f\leftarrow F'_n,\mathcal{A}}[\mathcal{A}^f(1^n) = 1] - \Pr_{f\leftarrow B'_n,\mathcal{A}}[\mathcal{A}^f(1^n) = 1] \right| > n^{-c}.$$

Now we construct a p.p.t. adversary that breaks the original PRF on inputs $m \in \{m(n_1), m(n_2), \dots\}$. By the (weakly) uniformity of $\mathcal{H}$, there exists a p.p.t. generator $\mathcal{G}$ and an event $\mathcal{E}$ (denoting whether $\mathcal{G}$ successes) such that $\mathcal{G}(1^n)$ successfully samples a hash function conditioning on $\mathcal{E}$, and $\Pr[\mathcal{E}] \geq 1 - n^{-(c+1)}$. Our adversary $\mathcal{A}'^f(1^m)$ samples such a circuit $C \leftarrow \mathcal{G}(1^n)$ and then simulates $\mathcal{A}^{f\circ C}(1^n)$. That is, whenever $\mathcal{A}$ performs an oracle call for $x$, it evaluate $C$ on $x$ and perform an oracle call for $C(x)$. Note that

$$\begin{aligned}
&\Pr_{f'\leftarrow F'_n,\mathcal{A}}[\mathcal{A}^{f'}(1^n) = 1] \\
=\ &\Pr_{\substack{f\leftarrow F_m \\ h\leftarrow H_n,\mathcal{A}}}[\mathcal{A}^{f\circ h}(1^n) = 1] \\
=\ &\Pr_{\substack{f\leftarrow F_m \\ \mathcal{A},C\leftarrow\mathcal{G}(1^n)}}[\mathcal{A}^{f\circ C}(1^n) = 1 \mid \mathcal{E}]
\end{aligned}$$

$$= \Pr_{f \leftarrow F_m, \mathcal{A}'}[\mathcal{A}'^f(1^n) = 1 \mid \mathcal{E}].$$

Similarly, we have

$$\Pr_{f' \leftarrow B'_n, \mathcal{A}}[\mathcal{A}^{f'}(1^n) = 1] = \Pr_{f \leftarrow B_m, \mathcal{A}'}[\mathcal{A}'^f(1^n) = 1 \mid \mathcal{E}].$$

Together with the fact that $\Pr[\mathcal{E}] \geq 1 - n^{-(c+1)}$, we can see that for large $n$,

$$\begin{aligned}
&\left| \Pr_{f \leftarrow F_m, \mathcal{A}'}[\mathcal{A}'^f(1^m) = 1] - \Pr_{f \leftarrow B_m, \mathcal{A}'}[\mathcal{A}'^f(1^m) = 1] \right| \\
>& n^{-c} \Pr[\mathcal{E}] - \left| \Pr_{f \leftarrow F_m, \mathcal{A}'}[\mathcal{A}'^f(1^m) = 1 \mid \neg\mathcal{E}] - \Pr_{f \leftarrow B_m, \mathcal{A}'}[\mathcal{A}'^f(1^m) = 1 \mid \neg\mathcal{E}] \right| \Pr[\neg\mathcal{E}] \\
\geq& n^{-c}(1 - n^{-(c+1)}) - n^{-(c+1)} \\
\geq& n^{-(c+1)} \\
\geq& \Theta(m^{-\varepsilon^{-1}(c+1)}),
\end{aligned}$$

which is non-negligible. Hence $\mathcal{A}'$ breaks the original PRF and a contradiction arises. □

### 4.8 Probability theory

We need to use standard Hoeffding's inequality and Chernoff bound.

**Lemma 4.13 (Hoeffding's inequality).** Assume that $X_1, X_2, \ldots, X_n$ are independent random variables such that $X_i \in [a_i, b_i]$. Let $X = X_1 + X_2 + \cdots + X_n$ and $\mu = \mathbb{E}[X]$, then for any $t > 0$,

$$\Pr[|X - \mu| \geq \varepsilon n] \leq 2 \exp\left( -\frac{2n^2\varepsilon^2}{\sum_{i=1}^n (b_i - a_i)^2} \right). \qquad \diamondsuit$$

**Lemma 4.14 (Chernoff bound).** Assume that $X_1, X_2, \ldots, X_n$ are independent random variables such that $X_i \in [0, 1]$. Let $X = X_1 + X_2 + \cdots + X_n$ and $\mu = \mathbb{E}[X]$, then for any $0 \leq \delta \leq 1$,

$$\Pr[|X - \mu| > \delta\mu] \leq 2 \exp\left( -\frac{\delta^2\mu}{3} \right). \qquad \diamondsuit$$

## 5 A $2n + o(n)$ upper bound for $B_2$ circuits

In this section, we will present a construction of PRF of $2n + o(n)$ size assuming the existence of PRF, and a construction of PRF of both $2n + o(n)$ size and $(1 + \varepsilon) \log n$ depth for any constant $\varepsilon > 0$ assuming the existence of $\mathsf{NC}^1$ PRF. Both of our constructions preserve the uniformity of the original PRF. The key ingredient is a uniform construction of almost universal hash function in $2n + o(n)$ size and $(1 + o(1)) \log n$ depth simultaneously.

To gain more intuition on our construction, we will firstly demonstrate a direct $O(n)$ upper bound using linear-size encodable error-correcting code [Spi96] in Section 5.1. Then in Section 5.2, we show that it is sufficient to construct a primitive called *1-detector* that is much simpler then error-correcting code, and prove a $3n + o(n)$ construction of it. We improve the upper bound to $2n + o(n)$ using a novel construction of almost universal hash function based on graphs with large girth in Section 5.4, and consider the circuit depth in Section 5.5.

## 5.1 An $O(n)$ upper bound

We present a simplified version of $O(n)$ upper bound from Ishai, Kushilevitz, Ostrovsky, and Sahai [IKOS08] based on the following explicit error-correcting code with linear circuit complexity.

**Lemma 5.1 (Spielman [Spi96]).** There exists a uniform error-correcting code $\{\mathrm{Enc}_n \in B_{n,m}\}_{n\geq 1}$ with constant distance such that $m = O(n)$ and $\mathsf{CC}(\mathrm{Enc}_n) = O(n)$. ♢

**Theorem 5.2.** There exists a PRF (resp. uniform PRF) $\{F_n \subseteq B_n\}_{n\geq 1}$ with $\mathsf{CC}(F_n) = O(n)$ if PRF (resp. uniform PRF) of polynomial circuit complexity exists. ♢

**Proof.** Assume that there exists a PRF $\mathcal{F} = \{F_n \subseteq B_n\}$ of circuit complexity $n^c$. By Lemma 5.1, there exists a uniform error-correcting code $\{\mathrm{Enc}_n \in B_{n,m}\}_{n\geq 1}$ with distance $\delta \in (0,1)$ of complexity $O(n)$. Using Proposition 4.9, we can construct a linear-size uniform almost universal hash function $\mathcal{H} = \{H_n \subseteq B_{n,m}\}_{n\geq 1}$ for $m = \lceil n^{1/2c} \rceil$. By Levin's trick (see Lemma 4.12), the concatenation of $\mathcal{F}$ and $\mathcal{H}$, say $\mathcal{F} \circ \mathcal{H}$, is still a PRF. The circuit complexity of $\mathcal{F} \circ \mathcal{H}$ is at most

$$\max_{h\in H_n, f\in F_m} \mathsf{CC}(f \circ h) \leq \mathsf{CC}(H_n) + \mathsf{CC}(F_m) \leq O(n) + O(m^c) = O(n).$$

This first inequality holds since to compute $f \circ h$, it is sufficient to identify the outputs of the circuit computing $h$ as the inputs of the circuit computing $f$. In addition, it is easy to see that $\mathcal{F} \circ \mathcal{H}$ is uniform if $\mathcal{F}$ is uniform. □

## 5.2 Constructing hash function from 1-detector

As seen from the above proof, if we have an almost universal hash function $\{H_n \in B_{n,m}\}_{n\geq 1}$ of size $cn$ for $m = o(n)$, we can composite the hash function and the $O(n)$ PRF in Theorem 5.2 to construct a PRF of size $cn + o(n)$. We note that for the construction above, the exact constant hidden in $O(\cdot)$ is hard to analyze. This is because the explicit construction in [Spi96] utilizes bipartite expander and brute-force searched good codes, whose exact parameters are not well-studied.

This hints us to explore a "low-level" primitive with smaller complexity for our application instead of directly use an ECC. Such a notion indeed exists, and we call it *1-detector*. We first formally define this concept, and show that it indeed implies almost universal hash functions in this section. In the following subsections, we will show how to construct them efficiently. In particular, we will first give a $3n + o(n)$ construction and then give a novel construction of $2n + o(n)$ from graphs with large girth.

**Definition 5.3 (1-detector).** Let $m = m(n)$ and $r = r(n)$. An $(n, r, m)$ *1-detector* is a linear function $L_n \in B_{n,m}$ such that for all $x \in \mathbb{F}_2^n$ with Hamming weight $|x| \leq r$, $L_n(x) \neq 0$. A family of linear functions $\mathcal{L} = \{L_n \in B_{n,m}\}_{n\geq 1}$ is called a $(r, m)$ 1-detector if $L_n$ is $(n, r, m)$ 1-detector for all $n$. The output bits $L_n(x)$ are called the *parity-checking bits* of $x$. ♢

We emphasize that our definition requires 1-detector to be *linear*, in the sense that every output bit of $L_n(x)$ is the parity function of some of the input bits. For our application of constructing hash function and PRF, even a *randomized 1-detector* is sufficient.

**Definition 5.4 (Randomized 1-detector).** Let $m = m(n)$, $r = r(n)$ and $\varepsilon = \varepsilon(n)$. An $(n, r, m, \varepsilon)$ *randomized 1-detector* is a linear function $L_n \in B_{n,m}$ such that for all $x \in \mathbb{F}_2^n$ with Hamming weight $|x| \leq r$, $\Pr_\rho[L_n(\rho(x)) = 0] < \varepsilon(n)$ for a random permutation $\rho$ over input bits. A family of

linear functions $\mathcal{L} = \{L_n \in B_{n,m}\}_{n\geq 1}$ is called a $(r, m, \varepsilon)$ randomized 1-detector if $L_n$ is $(n, r, m, \varepsilon)$ randomized 1-detector for all $n$. For simplicity, we may omit $\varepsilon$ if it is a negligible function and simply call it $(r, m)$ randomized 1-detector. The output bits $L_n(\rho(x))$ are called the *parity-checking bits* of $x$. ♢

It is easy to see that a $(n, r, m)$ (deterministic) 1-detector is also an $(n, r, m, \varepsilon)$ randomized 1-detector with $\varepsilon = 0$.

Let $\mathcal{L} = \{L_n \in B_{n,m}\}_{n\geq 1}$ be an $(r, m)$ randomized 1-detector. By the linearity, we can see that for all $x, y \in \mathbb{F}_2^n$ such that $1 \leq \Delta(x, y) \leq r$, $L_n(\rho(x)) = L_n(\rho(y))$ with only negligible probability. If $m = o(n)$ and $r$ is moderately large, say $r = \Theta(n^\varepsilon)$, we can construct an almost universal hash function as follows. Let $\rho \in \mathcal{S}_n$ be a permutation over input bits and let $S \subseteq [n]$ be a subset of size $|S| = s = \Theta(n^{1-\varepsilon/2})$. For $S = \{i_1, i_2, \ldots, i_s\}$ we can define

$$h_{\rho,S}(x) \triangleq x_{i_1} \| x_{i_2} \| \ldots \| x_{i_s} \| L_n(\rho(x)),$$

that is the concatenation of $m$ random bits from $x$ and the parity-checking bits.

**Lemma 5.5.** Assume that $s = \omega(n \log n)/r$. If $\mathcal{L}$ is an $(r, m)$ randomized 1-detector, the collection $\mathcal{H}^{\mathcal{L}} = \{H_n = \{h_{\rho,S} \mid \rho \in \mathcal{S}_n, |S| = s, S \subseteq [n]\}\}_{n\geq 1}$ is almost universal. ♢

**Proof.** Let $x, y \in \mathbb{F}_2^n$ be an arbitrary pair of distinct inputs. If $\Delta(x, y) \leq r$, the probability that the last $m$ bits of $h_S(x)$ and $h_S(y)$ coincide is less than $\varepsilon(n)$ according to randomized 1-detector. If $\Delta(x, y) > r$, the probability that the first $s$ bits of $h_S(x)$ and $h_S(y)$ are the same is at most

$$\frac{\binom{n-r}{s}}{\binom{n}{s}} \leq \prod_{i=0}^{s-1} \frac{n-r-i}{n-i} \leq \left(1 - \frac{r}{n}\right)^s \leq \exp\left(-\frac{rs}{n}\right),$$

which is negligible since $rs/n = \omega(\log n)$. □

The existence of (deterministic) 1-detector with nice parameter has been known for a long time. Gelfand, Dobrushin, and Pinsker [GDP73] presents a construction of ECC using 1-detector (although they did not name it) with slightly different parameter. A primitive called *range detector* used by Gál, Hansen, Koucký, Pudlák, and Viola [GHKPV13] is a generalized version of 1-detector. The intermediate primitive called *error-reduction code* of Spielman's ECC [Spi96] also has the property of 1-detection. However, these constructions are either not constructive or of circuit complexity larger than $3n$, so that they are insufficient for our use.

### 5.3 A simple probabilistic construction

In this section, we will present a construction of 1-detectors inspired by standard existence proof with probabilistic method. Concretely speaking, for each positive integer $k$, we will give a p.p.t. algorithm $\mathcal{G}_k$ that outputs a $3n$ size circuit computing an $(n, m, r)$ (deterministic) 1-detector with probability[10] at least $1 - n^{-\Omega(k)}$. One may see that it is sufficient for constructing PRF using Lemma 5.5 and Levin's trick.

[10] On a fail execution, our algorithm may output an arbitrary circuit (or simply $\perp$) without any additional information, so there is no obvious way to amplify success probability. We note that it is similar (but not precisely equivalent) to the concept of weak uniformity: our algorithm will output a 1-detector on a successful execution, but it is not guaranteed to output a particular one for all successful executions.

The evaluation circuit generated by $\mathcal{G}_k$ is a depth-1 circuit containing $m$ XOR gates of unbounded fan-in while each variable is of out-degree exactly $d \geq 3$, which can be transformed into a standard $B_2$ circuit with $d \cdot n$ gates. The underlying topology between variables and gates are defined by a bipartite graph $G = (V_1 \cup V_2, E \subseteq V_1 \times V_2)$, where $|V_1| = n$ and $|V_2| = m = m(n)$. The evaluation circuit $C_G$ corresponding to a graph $G = (V_1 \cup V_2, E)$ is a depth-1 linear circuit: each vertex in $V_1$ corresponds to an input variable, each vertex in $V_2$ corresponds to an XOR gate, and the connection between gates and variables follows the edges of the graph. A bipartite graph $G$ is called *good* if $C_G$ computes an $(n, r, m)$ 1-detector. Equivalently, a graph is good if for any subset $S \subseteq V_1$ of size at most $r$, at least one of variable $v' \in V_2$ connects to odd number of variables in $S$. For a typical choice of parameters, we assume that $r = \Theta(n^\varepsilon)$ and $m = \Theta(n^{1-\varepsilon/2})$ for some constant $\varepsilon \in (0, 1)$.

To complete our algorithm $\mathcal{G}_k$, it suffices to describe an algorithm that generates a good graph with nice probability. By adopting the random procedure defined by [GDP73] together with an error reduction trick, we can actually design Algorithm 1 running in time $n^{O(k)}$ that generates a good graph with probability $n^{-0.1k}$, for any positive integer $k$.

**Algorithm 1:** Generating good graphs

```
for i = 1, 2, ..., t do
    Let G ← (V_1 ∪ V_2, ∅) be an empty graph;
    for v ∈ V_1, j = 1, 2, ..., d do
        Link a random edge e_{v,j} = (v, v') with v' ← V_2;
    end
    if ∀S ⊆ V_1 of size ≤ k, there exists v' ∈ V_2 connecting to odd number of vertices in S then
        return G;
    end
end
return ⊥
```

**Lemma 5.6.** Let $t = \omega(\log n)$ and $d \geq 3$. There exists a constant $\varepsilon \in (0, 1)$, such that for $r = \Theta(n^\varepsilon)$ and $m = \Theta(n^{1-\varepsilon/2})$, with probability at least $1 - n^{-0.1k}$, Algorithm 1 generates a good graph (and therefore a 1-detector) for sufficiently large $n$. ♢

**Corollary 5.7.** For some constant $\varepsilon \in (0, 1)$, let $m = m(n) = \Theta(n^\varepsilon)$, there exists an almost universal hash function $\mathcal{H} = \{H_n \subseteq B_{n,m}\}_{n \geq 1}$ with weakly uniform complexity $3n$. ♢

Since the proofs of Lemma 5.6 and Corollary 5.7 are technical and the construction in Section 5.4 will have better circuit complexity[11], we defer them to the Appendix B. The intuition of the corollary is that, if we take $d = 3$ in Lemma 5.6, the generated depth-1 XOR circuit can be transformed into a $B_2$ circuit of size $3n$ by expanding each XOR gate independently. Then we can construct a desired hash function using Lemma 5.5. By this corollary, a $3n + o(n)$ upper bound directly follows using Levin's trick (Lemma 4.12).

**Theorem 5.8.** There exists a PRF of circuit complexity $3n + o(n)$ assuming PRF exists. ♢

[11] We should also note, however, that the collision probability of this $3n$ size hash function could be much better than the construction in Section 5.4. This may make it of independent interests.

## 5.4 Better 1-detector from high-girth graphs

We will now present a randomized 1-detector which is realizable by depth-1 unbounded fan-in XOR circuit $C$ where each variable is of out-degree exactly 2. Let $G = (V_1 \cup V_2, E \subseteq V_1 \times V_2)$ be the graph indicating the topology of $C$. To further improve the construction in the previous subsection, we would like to inspect graphs with fine structures instead of picking random graphs. Since each input variable is of out-degree 2, graph $G$ can be viewed as the edge-vertex incidence relation of another undirected graph $D$, such that each vertex $u \in V_2$ corresponds to a vertex $\widetilde{u}$ in the new graph $D$ and each vertex in $v \in V_1$ adjacent to $u_1, u_2 \in V_2$ corresponds to an edge connecting $\widetilde{u_1}$ and $\widetilde{u_2}$. In such case, an assignment $x \in \mathbb{F}_2^n$ of the input bits such that $C(x) = 0$ corresponds to a subset $X$ of edges in $D$ such that each vertex is incident to an even number of edges in $X$. Thus, $X$ contains an Eulerian cycle and therefore a cycle. This means that, intuitively, if we want to construct a nice 1-detector, the graph $D$ cannot contain small cycles.

By convention, the *girth* of a graph is defined as the length of its shortest cycle. The following lemma shows the connection between good graphs for randomized 1-detector and the girth of graph.

**Lemma 5.9.** Let $D = (V, E)$ be a graph of girth $g \geq 5$ and let $S \subseteq E$ be a random subset of size $k$. For all $1 \leq k < |E|/2$, with probability at most $(|E|/g)^{-g/3}$, every vertex is incident to an even number of edges in $S$. ♢

**Proof.** Let $D = (V, E)$ be a graph and $S \subseteq E$ be a subset of size $k$. If all vertices connect to even number of edges in $S$, each vertex in the subgraph $D' = (V, S)$ is of even degree. In such case, any connected component of $D'$ consists an Eulerian cycle, which can only happen if $G$ contains a cycle of length no more than $|S|$. This means that if $k < g$, there always exists a vertex that is incident to an odd number of edges in $S$.

Now we consider the case when $k \geq g$. For a random subset $S$ of size $k$, we can view it as first taking a random subset of size $k - \lceil g/3 \rceil$, then another random subset of size $\lceil g/3 \rceil$ in the remaining edges. Let $\mathcal{L}(S)$ be the event that every vertex is incident to even number of edges in $S$, then

$$\begin{aligned}\Pr_{S \subseteq E, |S|=k}[\mathcal{L}(S)] &= \mathop{\mathbb{E}}_{S_1 \subseteq E, |S_1| = k - \lceil g/3 \rceil} \left[ \Pr_{S_2 \subseteq E \setminus S_1, |S_2| = \lceil g/3 \rceil} [\mathcal{L}(S_1 \cup S_2)] \right] \\ &= \mathop{\mathbb{E}}_{S_1 \subseteq E, |S_1| = k - \lceil g/3 \rceil} \left[ \binom{|E \setminus S_1|}{\lceil g/3 \rceil}^{-1} \sum_{S_2 \subseteq E \setminus S_1, |S_2| = \lceil g/3 \rceil} [\mathcal{L}(S_1 \cup S_2)] \right].\end{aligned}$$

We will show the summation in the above equation does not exceed 1, that is for any fixed $S_1 \subseteq E$ with $|S_1| = k - \lceil g/3 \rceil$, there exists at most one $S_2$ makes $\mathcal{L}(S_1 \cup S_2)$ happens. If this is true, then we can clearly bound the above probability by

$$\Pr_{S \subseteq E, |S|=k}[\mathcal{L}(S)] \leq \binom{|E| - (k - \lceil g/3 \rceil)}{\lceil g/3 \rceil}^{-1} \leq \left( \frac{|E|}{g} \right)^{-g/3}.$$

What remains is the claim above. Towards a contradiction, assume $\mathcal{L}(S_1 \cup S_2)$ and $\mathcal{L}(S_1 \cup S_2')$ holds simultaneously for $S_2 \neq S_2'$. Consider the symmetric difference of the two sets $S_2 \oplus S_2' = (S_1 \cup S_2) \oplus (S_1 \cup S_2')$. By our definition of the event $\mathcal{L}$, each vertex is incident to even number of edges in $S_2 \oplus S_2'$, resulting in a cycle of length no more than $|S_2 \oplus S_2'| \leq |S_2| + |S_2'| = 2\lceil g/3 \rceil < g$. This contradicts to the fact that the girth of $G$ is $g$. □

This lemma shows that if we construct a circuit $C$ based on the edge-vertex incident graph of an undirected graph $D = (V, E)$ with $|V| = m$, $|E| = n$ and girth $g = \omega(1)$, for an assignment $x$ with Hamming weight $|x| \leq n/2$ and a random permutation $\rho$ of input bits, $C(\rho(x)) = 0$ with only negligible probability. If the graph $D$ is moderately dense, say $m = \Theta(n/\log n)$, we can obtain a depth-1 XOR circuit computing an $(n, n/2, m)$ randomized 1-detector and therefore an almost universal hash function by Lemma 5.5.

What remains is the explicit construction of graphs with large girth. This has been studied in combinatorics since 1960s motivated by both theoretical interests and the practical issue to construct efficient low-density parity-checking (LDPC) codes. For our application, it is sufficient to use the simple construction given by Chandran [Cha03].

**Lemma 5.10 ([Cha03]).** There exists a polynomial time algorithm such that given any $m$ and $k < m/3$, constructs a graph $G = (V, E)$ of $m$ vertices with $|E| = \lfloor mk/2 \rfloor$ such that the girth of the graph is at least $g > \log_k m + O(1)$. Moreover, the degree of each vertex is $k-1, k$ or $k+1$. ♢

**Corollary 5.11.** For every $n$ and $m$ such that $\lceil 2n/m \rceil < m/3$, there exists a graph $D_{m,n}$ with $m$ vertices and $n$ edges where the girth $g > \frac{\log m}{\log(\lceil 2n/m \rceil)} + O(1)$. The degree of every vertex is at most $2n/m + O(1)$. Moreover, there exists a deterministic polynomial time algorithm construct $D_{m,n}$ taking $m, n$. ♢

**Proof.** Assume that we are given $n$ and $m$. Let $k = \lceil 2n/m \rceil$, our algorithm firstly calls the algorithm in Lemma 5.10 to generate a graph $D = (V, E)$ with $|V| = m$, $|E| = \lfloor mk/2 \rfloor \geq n$ and girth $g > \log_k(m) + O(1)$. The degree of each vertex is at most $2n/m + 2$. We can arbitrarily remove $|E| - n$ edges to obtain a desired graph. □

Now we formally describe the construction of our uniform randomized 1-detector and hash function. Take $m = \Theta(n/\log n)$. We firstly construct a graph $D_{m,n}$ with girth $g = \Omega\left(\frac{\log m}{\log(2n/m)}\right) = \Omega\left(\frac{\log n}{\log\log n}\right)$. Then, we construct bipartite graph $G$ using $D_{m,n}$ and then generate depth-1 XOR circuit $C$ according to it. By Lemma 5.9 and previous discussion, one can easily see that $C$ is an $(n, n/2, n/\log n)$ randomized 1-detector and can be transformed into $B_2$ circuit of size $2n - m$. Finally, we take $s = \Theta(\log^2 n)$ and construct an almost universal hash function with sufficient shrinkage using the sampling trick in Lemma 5.5.

**Theorem 5.12.** There exists $m = m(n) = o(n)$ and a uniform almost universal hash function $\mathcal{H} = \{H_n \in B_{n,m}\}_{n \geq 1}$ of size $2n - m$. ♢

We can then compose this hash function with the simple $O(n)$ PRF in Theorem 5.2 by Levin's trick (see Lemma 4.12) and give the following upper bound of PRF.

**Corollary 5.13.** There exists a PRF (resp. uniform PRF) of circuit complexity $2n + o(n)$ if PRF (resp. uniform PRF) exists. ♢

## 5.5 The upper bound of depth for PRF and hash

Besides circuit size (or running time in uniform case), circuit depth (or parallel time) is also an important measure with both theoretical and practical interests. Many efforts have been made on studying the existence of PRF in low-depth circuit classes such as $\mathsf{NC}^1$ and even $\mathsf{TC}^0$. Indeed, it turns out that $\mathsf{NC}^1$ and $\mathsf{TC}^0$ PRFs can be based on quite standard cryptographic assumptions, such

as Decisional Diffie-Hellman [NR04] and Ring Learning with Errors [BPR12]. In this part, we will show how to reduce the circuit depth of PRF together with the circuit size in previous subsections using similar techniques.

As before, we start with the uniform construction of an almost universal hash function of size $2n$ and depth $(1+o(1))\log n$ that shrinks $n$-bit input to $n^\varepsilon$-bit output for arbitrary $\varepsilon > 0$. This is achieved by stacking the hash functions constructed before for super-constant number of times, under the output length is reduced down to $n^\varepsilon$ bits.

**Lemma 5.14.** For any $\varepsilon > 0$, there exists a uniform construction of almost universal hash function $\mathcal{H} = \{H_n \in B_{n,c}\}_{n\geq 1}$ of size $2n + o(n)$ and depth $(1+o(1))\log n$ for some $c = c(n) < n^\varepsilon$. ♢

**Proof.** Let $m = m(n)$ be the function in Theorem 5.12. In Section 5.4 we have present a uniform construction of $(n, n/2, m)$ randomized 1-detector based on graph with large girth. Such 1-detector can be evaluated by a depth-1 XOR circuit $C_n$ whose topology is induced from the vertex-edge incident graph of $D_{m,n} = (V, E)$ given by Corollary 5.11. Note that the degree of each vertex in $D_{m,n}$ is at most $2n/m + O(1)$, so that each XOR gate in $C$ is of fan-in $\Theta(n/m)$. This means that $C_n$ can be realized by a $B_2$ circuit of size at most $2n$ and depth $\log(n/m) + O(1)$.

Now we will construct a hash function $\mathcal{H} = \{H_n \subseteq B_{n,c}\}_{n\geq 1}$ for arbitrary $c < n^\varepsilon$ by stacking the hash function $\widetilde{\mathcal{H}} = \{\widetilde{H}_n \in B_{n,m}\}_{n\geq 1}$ in Theorem 5.12. Define $n_0, n_1, \ldots, n_\ell$ by $n_0 = n$ and $n_{i+1} = m(n_i)$ for all $0 \leq i < \ell$, where $\ell$ is the smallest integer such that $n_\ell < n^\varepsilon$. It is easy to see that $\ell = O(\log n)$. Then we define

$$H_n \triangleq \{h_{\ell-1} \circ h_{\ell-2} \circ \cdots \circ h_0 \mid h_i \in \widetilde{H}_{n_i}\},$$

where each $h_i$ is sampled independently from $\widetilde{H}_{n_i}$.

Since $h_i$ can be computed by a circuit of size $2n_i$ size and $\log(n_i/n_{i+1}) + O(1)$ depth, it is easy to verify that $\mathcal{H}$ can be computed by a circuit of size $2n + o(n)$ and depth $(1+o(1))\log n$. Now it is sufficient to show that $\mathcal{H}$ is indeed an almost universal hash function. Because $\widetilde{\mathcal{H}}$ is almost universal, there exists a negligible function $\varepsilon(n)$ such that for all $n$ and $x \neq y$ of length $n$, $\Pr[h(x) = h(y)] \leq \varepsilon(n)$ for $h \leftarrow \widetilde{H}_n$. Then we can see that for all $x \neq y$ of length $n$,

$$\begin{aligned}
&\Pr_{h\leftarrow H_n}[h(x) = h(y)] \\
&= \Pr_{h=(h_0,\ldots,h_\ell)\leftarrow H_n}[\exists i, h_i \text{ is the first layer with same output for } x \text{ and } y] \\
&\leq \sum_{i=1}^{\ell-1} \varepsilon(n_i),
\end{aligned}$$

which is also negligible. This completes the proof. □

Then the depth upper bound of PRF follows directly from Levin's trick (Lemma 4.12).

**Corollary 5.15.** For any $\varepsilon > 0$, there exists a PRF (resp. uniform PRF) of circuit size $2n + o(n)$ and depth $(1+\varepsilon)\log n$ assuming $\mathsf{NC}^1$ PRF (resp. uniform $\mathsf{NC}^1$ PRF) exists. ♢

**Proof.** Suppose that there exists a PRF computable by a circuit of depth $d\log n$ and size $n^d$. By Lemma 4.12, we can compose it with a hash function that shrinks $n$-bit input to $n^{\varepsilon/(2d)}$ bits to obtain a PRF with $2n + o(n)$ size and $(1+\varepsilon)\log n$, where $\varepsilon$ can be arbitrary constants. □

**Remark.** Although our "stacking" construction in Lemma 5.14 is natural, it fails to provide a hash function with *constant depth*, *polynomial shrinkage* and *low wire complexity* in unbounded fan-in circuit models. Note that weakly uniform almost universal hash with depth 1, polynomial shrinkage and wire complexity $3n$ can be obtained using the simple probabilistic construction in Section 5.3.

In fact, uniform construction in constant depth 2 size circuit can be done by combining our high-girth based 1-detector with range-detector in [CT19]. In more details, it is an almost universal hash function in $\mathsf{CC}^0[2]$ (constant depth circuits with unbounded fan-in XOR gates) with depth 2 and wire complexity $2n + o(n)$. By looking into the construction, we can implement it in $B_2$ circuits with $\log n + O(1)$ depth (this is a slight improvement comparing with the stacking upper bound $(1 + o(1)) \log n$). We give a complete discussion of this construction in Appendix C.

# 6 A $2n - O(1)$ lower bound for $B_2$ circuits

In this section, we will prove a matching circuit lower bound, showing that any circuit family of size less than $2n - O(1)$ cannot be pseudorandom function. Our proof first studies a combinatorial structure on circuits, which we call *critical path*. We present an efficient algorithm that distinguishes circuits with intersecting critical paths and truly random functions via oracle access. Then we do a standard wire counting argument to show that $2n - O(1)$ gates are required to avoid intersecting critical paths, which leads to a circuit lower bound for PRF. By noticing that our distinguisher is non-adaptive, we can also prove a tight unconditional lower bound for universal hash function with super-linear shrinkage.

We begin by defining the combinatorial structure to be interested.

**Definition 6.1 (Critical path).** Let $C$ be a circuit, and $x$ be one of its inputs. The *critical path of* $x$ in $C$ is a sequence of vertices $v_0, v_1, \ldots, v_k$ satisfying the following conditions:

1. $v_0 = x$, and $v_i$ is a descendent of $v_{i-1}$ for all $i \geq 1$, and
2. out-degree$(v_i) = 1$ for all $0 \leq i < k$, and out-degree$(v_k) \neq 1$. ♢

Without loss of generality, we can only deal with circuits without obvious redundancy. Formally, a circuit $C$ is called *normalized* if each gate of out-degree 0 is an output gate. Since a non-output gate of out-degree 0 can be removed, any optimal circuit computing a function $f$ must be normalized.

Informally speaking, the *critical path of* $x$ *in* $C$ is the maximal path starting from $x$ such that all but the last vertices have out degree exactly 1. The last vertex may have out-degree 0 or more than 2. It is obvious that the critical path is unique for each input $x$, so we denote it by $\mathcal{L}_C(x)$. We will be interested in the intersection of critical paths. Two critical paths are called intersecting, if they share a common vertex. We emphasize here that sharing the last vertex of out-degree not 1 is also called intersecting.

Our key observation is that if a circuit has an *isolated variable* (i.e. of out-degree 0 and is not an output node, recall that we are considering single-output functions) or *intersecting critical paths*, then it can be distinguished from truly random functions. This is presented in the following two lemmas.

**Lemma 6.2.** There exists a p.p.t. oracle algorithm $\mathcal{A}$ which always accepts if it is given oracle access to a circuit $C$ with intersecting critical path, and rejects with high probability if it is given the truly random function. ♢

**Proof.** Suppose that inputs $x$ and $y$ have intersecting critical paths. Let $G$ be the gate on their first (i.e. closest to input) intersection. Under any restriction $\rho$ to all the variables except $x$ and $y$, we can see that the circuit can be viewed as computing

$$C\restriction_\rho(x,y) = f_\rho(G(g_\rho(x), h_\rho(y)))$$

for some unary functions $f_\rho, g_\rho, h_\rho$. By trying all possible truth tables, one can easily check that no matter what function $G$ is, $C\restriction_\rho(x,y)$ can never compute an $\oplus$-type function for some restriction $\rho_1$, and compute an $\wedge$-type function for another restriction $\rho_2$. Indeed, when $G$ is of $\wedge$-type, then $C\restriction_\rho(x,y)$ cannot be an $\oplus$-type function; when $G$ is of $\oplus$-type, $C\restriction_\rho(x,y)$ cannot be an $\wedge$-type function; and when $G$ is degenerate or trivial, $C\restriction_\rho(x,y)$ is also degenerate or trivial.

This motivates us to do the following test for each pair of inputs $(x,y)$. Randomly sample $n$ restrictions for all inputs except $x$ and $y$. For each sampled restriction $\rho$, compute the truth table of $C\restriction_\rho(x,y)$, and check whether truth tables of $\oplus$-type and $\wedge$-type appear both. We accept if there exists a pair $(x,y)$, such that either $\oplus$-type or $\wedge$-type does not appear in the truth tables of $C\restriction_\rho(x,y)$ for our $n$ samples of $\rho$.

If our algorithm is given a circuit with intersecting critical paths, there always exists a pair $(x,y)$ such that the two kinds of truth table does not appear simultaneously, so that our algorithm always accepts. Assume otherwise our algorithm is given a truly random function. Since for each pair $(x,y)$, the $n$ samples of restriction $\rho$ are drawn independently, we can show by union bound that the sampled restrictions $\rho$ are pairwisely distinct with high probability. In such case, the oracle returns independent random bits for our queries, hence both $\oplus$-type and $\wedge$-type truth tables appear for truly random functions with probability $1 - \exp(O(n))$. By union bound, our algorithm accepts with high probability. □

**Lemma 6.3.** There exists a p.p.t. oracle algorithm $\mathcal{B}$ which accepts if it is given oracle access to a circuit $C$ with isolated variables, and rejects with high probability if it is given the truly random function. ♢

**Proof.** Consider the following algorithm. For each variable $x$, we sample $n$ restrictions for all variables except $x$. We accept if there exists a variable such that $n$ sampled restrictions give the same output, and reject otherwise. The correctness of our algorithm is easy to verify. □

By combining these two tests, we can distinguish a circuit with either intersecting critical paths or isolated variables. To complete the lower bound, it is sufficient to show that small circuits contain either intersecting critical paths or isolated variables. We prove this by a standard wire counting technique.

**Lemma 6.4.** For any *normalized* $n$-input $m$-output circuit $C$ with no intersecting critical paths and isolated variable, the number of gates in the circuit should be at least $2n - 2m$. ♢

**Proof.** We divide all nodes (including variables and gates) in the circuit into two types: on the critical path of some variable, or outside of all critical paths. In particular, all variables fall into the first type. Suppose that there are $c_1$ nodes of the first type, and $c_2$ nodes of the second. Let $l$ be the number of wires connecting two nodes of the first type, and $o$ be the number of output nodes belonging to the first type.

Since there is no isolated variables, the endpoints of critical paths must be gates or variables of out-degree at least 2. By the non-intersection property of critical paths, there should be exactly $n$ nodes of the first type having out-degree not equal to 1, hence $(c_1 - n)$ of them have out-degree

1. Since the circuit is normalized, only output gates can have out-degree 0, so that at least $(n-o)$ nodes of first type have out-degree at least 2.

We now count different types of wires. Type $i \to$ Type $j$ denotes the number of wires from Type $i$ nodes to Type $j$ nodes.

**(Type 1 $\to$ Type 1)** By definition this is $l$.

**(Type 1 $\to$ Type 2)** There are at least $2(n-o)+(c_1-n)$ wires going out of Type 1 nodes, among which $l$ wires go to Type 1, hence there should be at least $c_1+n-2o-l$ wires going to Type 2.

**(Type 2 $\to$ Type 1)** Among the $c_1$ nodes of Type 1, $(c_1-n)$ of them are gates. Since each of these gates takes two wires as inputs, the number of wires going from Type 2 to Type 1 is exactly $2(c_1-n)-l$.

**(Type 2 $\to$ Type 2)** Since all gates except for $(m-o)$ output nodes of Type 2 have out-degree at least 1, there is at least $c_2-(m-o)$ wires going out of Type 2. Because $2(c_1-n)-l$ of them goes to Type 1, there are at least $(c_2-(m-o))-2(c_1-n)+l$ remains.

Now, notice that the total number of wires going into Type 2 gates is exactly $2c_2$, so we should have the inequality

$$(c_1+n-2o-l)+((c_2-(m-o))-2(c_1-n)+l) \le 2c_2,$$

which gives us $c_1+c_2 \ge 3n-m-o \ge 3n-2m$. Subtracting the $n$ input nodes from it completes the proof of the lemma. □

Combining these three lemmas, the lower bound is immediate.

**Corollary 6.5.** If $\mathcal{F}=\{F_n \subseteq B_n\}_{n\ge 1}$ is a PRF, then $\mathsf{CC}(\mathcal{F}) \ge 2n-2$. ♢

Since we can construct efficient PRF with almost universal hash function, this lower bound also yields a lower bound for almost universal hash function assuming PRF exists. By modifying the proof a little bit, we are also able to make this lower bound unconditional.

**Corollary 6.6.** If $\mathcal{H}=\{H_n \subseteq B_{n,m}\}_{n\ge 1}$ is an almost universal hash function, then $\mathsf{CC}(\mathcal{H}) \ge 2n-2m$. In particular, if $m=o(n)$, then $\mathsf{CC}(\mathcal{H}) \ge 2n-o(n)$. ♢

**Proof.** Towards a contradiction, we assume that for infinitely many bad $n$, $\mathsf{CC}(H_n) < 2n-2m$. By Lemma 6.4, for each $h \in H_n$ and any normalized circuit $C$ computing $h$, $C$ contains either isolated variable or intersecting critical paths.

Let $n$ be an arbitrary "bad" input length, we will construct a set $T_n$ of distinct pairs of inputs such that for all $h \in H_n$, there exists a pair $(x,y) \in T_n$ with $h(x)=h(y)$.

We first consider intersecting critical paths. Suppose that $x$ and $y$ have their critical paths intersecting, and the intersection starts from the gate $G$. Then by our definition of critical paths, the outputs of the circuit should be completed determined by the output of $G$ if all other inputs are arbitrarily fixed (e.g., fixed to be zeros). Hence there must exist a pair of assignments to $x$ and $y$ such that their outputs collide with all other variables fixed. We can always fix the other variables to 0, and let $S_n$ be the set of all possible colliding pairs of inputs from above. Formally, define

$$\rho_{x,y,a,b}(v)=\begin{cases} a, & v=x \\ b, & v=y \\ 0, & \text{otherwise} \end{cases},$$

then the set $S_n$ is

$$S_n = \{(\rho_{x,y,a,b}, \rho_{x,y,c,d}) \mid x, y \text{ are input variables, } a,b,c,d \in \{0,1\}, (a,b) \neq (c,d)\}.$$

Hence for any circuit containing an intersecting critical path, there must exist a pair in $S_n$ making their outputs collide.

Finding collision for circuits with an isolated variable is rather simple. Let $I_n$ be the set of pairs $(x,y)$ such that $x = 0\|0\|\ldots\|0$ and $y$ contains exactly one non-zero index. All circuits with an isolated variable collides on one of the pairs in $I_n$.

Let $T_n \triangleq I_n \cup S_n$. It is easy to see that $|T_n| = O(n^2)$. Since $n$ is an bad input length, each $h \in H_n$ has circuit complexity smaller than $2n - 2m$ so that contains either intersecting critical paths or isolated variables, hence indeed has a collision in $T_n$. By previous discussion we know that

$$\Pr_{h \leftarrow H_n, (x,y) \leftarrow T_n}[h(x) = h(y)] \geq \frac{1}{|T_n|} = \Omega(n^{-2}).$$

By averaging argument, there exists a particular pair in $T_n$ with collision probability $\Omega(n^{-2})$. This is clearly not an almost universal hash function. □

Note that this lower bound is (almost) tight for $m = o(n)$ by the $2n$ upper bound (see Lemma 5.14). Since we can construct almost universal hash function from good ECC (see Proposition 4.9), we can also obtain an unconditional (almost) tight lower bound for ECC of the same complexity.

# 7 Constant-depth linear threshold circuits

In this section, we consider the regime of constant-depth linear threshold circuits, and prove a slightly super-linear lower bound for pseudorandom function. Our lower bound follows from a structural result for threshold circuits used in developing average-case hardness [CSS18], quantified derandomization [Tel18] and pseudorandomness [HHTT21]. Our result can be interpreted as a "fine-grained" barrier (although weaker than Natural Proofs [RR97]) on circuit lower bound techniques that explains why proving $\mathsf{TC}^0_d$ lower bounds beyond $n^{1+c^{-d}}$ is not easy.

## 7.1 Upper bound via efficient ECC

We first show the easy part: an upper bound by Levin's trick. To do so, it is necessary to have a construction for PRF with polynomial size threshold circuits. Although it is unknown whether the existence of $\mathsf{TC}^0$ PRF can be based on the elementary primitive such as one-way functions, there are several constructions under standard cryptographic assumption like factoring (of Blum-integers) or decisional Diffie-Hellman [NR04], as well as ring learning-with-error [BPR12].

For completeness, we sketch the construction of Naor and Reingold [NR04] based on decisional Diffie-Hellman assumption. Let $n$ be desired input length. The key of the PRF is a tuple $(p,q,g,a)$, where $p$ is an $n$-bit prime, $q$ is a prime dividing $p-1$, $g$ is an element in $\mathbb{Z}_p^\times$ with order $q > 2^n$ and $a \in \mathbb{Z}_q^{n+1}$. Note that $(p,q,g)$ is chosen over some polynomially samplable distribution and $a$ is chosen uniformly. Let $x = x_1\|x_2\|\ldots\|x_n$. The output of the PRF is defined as

$$f_{p,q,g,a}(x) \triangleq (g^{a_0})^{\prod_{i=1}^n a_i^{x_i}}.$$

By the efficient *multiple product* circuit given by Reif and Tate [RT92], with suitable preprocessing, this PRF can be evaluated by polynomial size $\mathsf{TC}^0$ circuits.

To obtain an upper bound for $\mathsf{TC}^0$ PRF, we only need to construct a low complexity almost universal hash function with nice shrinkage. For instance, we can use the error-correcting code given by Chen and Tell [CT19].

**Lemma 7.1 ([CT19], Proposition 10).** Let $\phi = \frac{1+\sqrt{5}}{2}$. For every $d \geq 4$ there exists a family of linear threshold circuits of size $n^{1+O(\phi^{-d})}$ and depth $d$ that encodes an linear error correcting code of constant relative distance. Moreover, this circuit family is uniformly constructible in polynomial time. ♢

This immediately shows that for any $d \geq 4$, there exists a uniform almost universal hash function realizable by $n^{1+O(\phi^{-d})}$ size $\mathsf{TC}^0_d$ circuits that shrinks $n$-bit input to $m = \Theta(n^\varepsilon)$ bits for arbitrarily small $\varepsilon > 0$ (see Proposition 4.9). Then by Levin's trick (see Lemma 4.12), we have the following upper bound.

**Corollary 7.2.** Let $\phi = \frac{1+\sqrt{5}}{2}$. For any constant $d_0$, assume that there exists a PRF (resp. uniform PRF) computable in $\mathsf{TC}^0_{d_0}$, then there exists a PRF (resp. uniform PRF) computable by threshold circuits of size $n^{1+O(\phi^{-d})}$ and depth $d + d_0$ for any $d \geq 4$. ♢

**Remark.** The construction of Chen and Tell [CT19] is based on the framework of Gál, Hansen, Koucký, Pudlák, and Viola [GHKPV13], the lossless expander constructed in Capalbo, Reingold, Vadhan, and Wigderson [CRVW02] and the parity upper bound in $\mathsf{TC}^0_d$ circuits given by Paturi and Saks [PS94]. Combining the lossless expander approach and the high-girth graph from Section 5.4, we are able to get an explicit construction of almost universal hash function in depth 2 $\mathsf{CC}^0[2]$ of size $2n + o(n)$. We give a self-contained description for this construction in Appendix C.

## 7.2 Extracting black-box property from white-box restriction

In this section, we will show that computing PRF with depth-$d$ threshold circuits require size at least $n^{1+\Omega(1)^d}$.

**Theorem 7.3.** There exists universal constants $\theta > 0$ and $c > 1$, such that for any PRF $\mathcal{F}$, $\mathsf{TC}^0_d\text{-}\mathsf{CC}(\mathcal{F}) \geq n^{1+\theta c^{-d}}$. ♢

The technique we will use is the "white-box" random restriction method developed in previous works of average-case hardness and pseudorandomness for $\mathsf{TC}^0$ circuits [CSS18; Tel18; HHTT21]. Although their results are presented in different forms, all of them essentially use the following fact about threshold circuits: for a random restriction (or pseudorandom restriction) $\rho$ applied to a small threshold circuit $C$ of depth $d$, with nice probability, there exists a properly large subset $S_\rho$ of free variables such that for a random assignments $\sigma$ to all variables but $S_\rho$, again with nice probability, the circuit $C$ restricted by $\rho\sigma$ can be approxmable by a small threshold circuit of depth $d - 1$. For our purpose, we formalize this fact as Lemma 7.5 and generalize it to multi-output case. We defer its proof to Section 7.3.

**Definition 7.4.** Let $n, m \geq 1$ and $0 \leq \varepsilon \leq 1$ be a parameter. A function $f \in B_{n,m}$ is said to be $\varepsilon$-approximable by a $\mathsf{TC}^0_d$ circuit $C$ if for a uniformly random input $x$, $C(x) \neq f(x)$ with probability at most $\varepsilon$. A function $f \in B_{n,m}$ is said to be *transparent* if each of its output bits is a constant or only depends on exact one input variable. ♢

**Lemma 7.5 (Restriction lemma for $\mathsf{TC}^0$).** There exists absolute constants $c > 1$ and some constant $0 < \varepsilon_0 < 1$ such that following holds. For all $\varepsilon < \varepsilon_0$ and function $\alpha_1(n)$, there exists some $\delta > 0$, such that for all depth $d \geq 1$, output length $m = m(n)$ and sufficiently large $n$, for any $f \in B_{n,m}$ that is $\alpha_1(n)$-approximable by a multi-output depth $d$ threshold circuit of size $n^{1+\varepsilon}$, if we apply the restriction $\rho \leftarrow \mathcal{R}_{n^{-\delta}}$ to the function, with probability at least $1/8$ there exists a set of unfixed variables $S_\rho$ (which depends on $\rho$) of size at least $n^{0.99}$, such that at least a half of the restrictions $\sigma$ to *all* variables not fixed by $\rho$ and not in $S_\rho$ would make $f\restriction_{\rho\sigma}$ $4(\alpha_1(n) + \alpha_2(n))$-approximable by a multi-output threshold circuit of depth $d-1$ and size $|S_\rho|^{1+c\varepsilon}$ (or a transparent function if $d = 1$), where $\alpha_2(n) \triangleq 2n^{1+\varepsilon-\delta}\exp(-2n^{\delta c_1})$. $\diamondsuit$

Lemma 7.5 and its counterparts in [CSS18; Tel18; HHTT21] are considered as white-box techniques because the subset $S_\rho$ of unfixed variables is chosen according to the circuit approximating $f$ and the result of the random restriction $\rho$. To prove circuit lower bound for pseudorandom functions, we need to specify a black-box property of sparse $\mathsf{TC}^0_d$ circuits to distinguish it from truly random function with only oracle accesses. The key observation of our lower bound is that the following black-box property can be extracted from the white-box restriction lemma.

**Lemma 7.6.** Let $c$ be the constant in Lemma 7.5. There exists constants $\theta > 0$ so that the following holds. For all depth $d \geq 1$, output length $m = m(n) < n^{0.99^d}$, and sufficiently large $n$, for any $f \in B_{n,m}$ that is $1/8^{d+1}$-approximable by a multi-output depth $d$ threshold circuit of size less than $n^{1+\theta c^{-d}}$, for a random assignment $x \in \mathbb{F}_2^n$ to all input variables, with probability at least $1/16^{d+1}$, there exists another assignment $y$ different from $x$ by exact one bit such that $f(x) = f(y)$ holds. $\diamondsuit$

**Proof.** Assume that $\varepsilon_0, \delta, p, S_\rho, \alpha_2(n)$ follows Lemma 7.5. Take $\theta < \varepsilon_0$ (therefore $\theta c^{-d} < \varepsilon_0$ for any $d$). We will prove the statement by doing induction over $d$ using Lemma 7.5. Assume that $n$ is sufficiently large[12] and $f \in B_{n,m}$ is $1/8^{d+1}$-approximable by a multi-output $\mathsf{TC}^0_d$ circuit of size $n^{1+\theta c^{-d}}$. Since $\alpha_2(n) = \exp(-\mathrm{poly}(n))$, we have $\alpha_2(n) < 1/8^{d+1}$ for sufficiently large $n$.

For simplicity, we call an assignment $x$ *good* if there exists another assignment $y$ that differs on exactly one bit from $x$ such that $f(x) = f(y)$. What we want to show is that $\Pr[x \text{ is good}] \geq 1/16^{d+1}$ for uniformly random $x$. For a random assignment to all input variables, we consider it as a pair of a random restriction $\rho \leftarrow \mathcal{R}_p$ and a random assignment $\beta$ to the leftover variables. Clearly, it is sufficient to show that

$$\Pr_{\rho\leftarrow\mathcal{R}_p,\beta\leftarrow\{-1,1\}^{|\rho^{-1}(\star)|}}[(\rho,\beta) \text{ is good}] \geq \frac{1}{16^{d+1}}. \tag{1}$$

We call a random restriction $\rho \leftarrow \mathcal{R}_p$ *good* if $S_\rho$ in Lemma 7.5 exists. By Lemma 7.5, $\Pr[\rho \text{ is good}] \geq 1/8$. Then it is sufficient to show that for any fixed good $\rho$,

$$\Pr_{\beta\leftarrow\{-1,1\}^{|\rho^{-1}(\star)|}}[(\rho,\beta) \text{ is good}] \geq \frac{8}{16^{d+1}}. \tag{2}$$

Now we fix any good $\rho$ (so that there exists $S_\rho$ satisfying Lemma 7.5) and prove Equation (2). In such case, the assignment $\beta$ to the leftover variables can be further considered as the composition of a random restriction $\sigma$ to variables not in $S_\rho$ and another random assignment $\tau$ to the remaining variables $S_\rho$. Identify $\beta = (\sigma, \tau)$ as discussed above. We call an assignment $\sigma$ *good* if $f\restriction_{\rho\sigma}$ can

[12] Rigorously speaking, we will take $n > n_0$, where $n_0$ satisfies the following three constraints: (1) Lemma 7.5 holds for depth $d$; (2) induction hypothesis holds for depth $d-1$; and (3) the inequality $\alpha_2(n) < 1/8^{d+1}$ holds.

be approximable by a $\mathsf{TC}^0_{d-1}$ circuit of size $|S_\rho|^{1+\theta c^{-(d-1)}}$ (or a transparent function) with error $4(1/8^{d+1} + \alpha_2(n)) < 1/8^d$. By Lemma 7.5, at least a half of the assignments $\sigma$ are good. This means to prove Equation (2), it is sufficient to show that for any fixed good $\rho$ and good $\sigma$,

$$\Pr_{\tau \leftarrow \{-1,1\}^{|S_\rho|}}[(\rho, \sigma, \tau) \text{ is good}] \geq \frac{1}{16^d}. \tag{3}$$

Firstly we consider $d = 1$. In this case, $f\restriction_{\rho\sigma}$ can be $1/8$ approximated by a transparent function such that each output bit depends on at most one input variable. Since the output length $m < n^{0.99}$ is smaller than the input length $|S_\rho| \geq n^{0.99}$, the transparent function is independent from some input variable $v$. By union bound, for uniformly chosen $\tau$ and the corresponding $\tau'$ only differ on input variable $v$, $\Pr[f\restriction_{\rho\sigma}(\tau) = f\restriction_{\rho\sigma}(\tau')] \geq 3/4$. By averaging argument, for at least $1/2$ fraction of $\tau$, there exists some $\tau'$ such that $f\restriction_{\rho\sigma}(\tau) = f\restriction_{\rho\sigma}(\tau')$. This immediately shows that

$$\Pr_{\tau \leftarrow \{-1,1\}^{|S_\rho|}}[(\rho, \sigma, \tau) \text{ is good}] \geq \frac{1}{2} \geq \frac{1}{16^d}.$$

When $d > 1$, there exists a multi-output threshold circuit of depth $d-1$ and size less than $|S_\rho|^{1+\theta c^{-(d-1)}}$ that $1/8^d$-approximates $f\restriction_{\rho\sigma}$, where $|S_\rho|$ is the input length of the new circuit. Since $|S_\rho| \geq n^{0.99}$, this circuit is of output length $m < n^{0.99^d} \leq |S_\rho|^{0.99^{d-1}}$ which satisfies the induction condition. According to induction hypothesis, for at least $1/16^d$ fraction of $\tau$, there exists $\tau'$ such that $f\restriction_{\rho\sigma}(\tau) = f\restriction_{\rho\sigma}(\tau')$. Then Equation (3) immediately follows since for each of such $\tau$, $(\rho, \sigma, \tau)$ is good. This completes the induction. □

This lemma shows that finding a collision is easy for multi-output functions approximable by sparse threshold circuits. Since it is hard to find a collision for truly random functions, we can then distinguish sparse threshold circuits from truly random functions.

**Proof of Theorem 7.3.** We will prove this result by presenting a distinguisher which separates functions computable by sparse $\mathsf{TC}^0_d$ circuits and truly random functions. Suppose that the set of input variables is $I$, and the algorithm is given oracle access to the function $f$. Our algorithm is quite simple.

- Let $S$ be the subset of first $\ell(n) = 2\lceil \log \log n \rceil$ input variables.
- Randomly choose an assignment $x \leftarrow \mathbb{F}_2^{|I \setminus S|}$ for other variables.
- Randomly flip an input variable of $x$ to obtain $y$.
- Accept if $f(z\|x) = f(z\|y)$ holds for all assignment $z \in \mathbb{F}_2^{|S|}$.

Since there are in total $2^{\ell(n)} = \Theta(\log^2 n)$ different assignments to $S$, the algorithm queries $\Theta(\log^2 n)$ separated pairs of assignments which can be done in polynomial time. For truly random functions, each pair of assignments outputs the same result with probability $1/2$ independently. Thus, the algorithm would accept with probability $2^{-\Theta(\log^2 n)}$, which is negligible.

We then show the algorithm would accept functions computable by sparse $\mathsf{TC}^0$ circuits with non-negligible probability. Firstly, $f$ can be viewed as a multi-output function $g \in B_{n-\ell(n),2^{\ell(n)}}$ such that $z$-th output bit of $g(x)$ is $f(z\|x)$. Clearly, our algorithm accept if and only if $g(x) = g(y)$ holds.

Since each output bit of $g$ compute $f$ under some restriction, $g$ can be computed by a $\mathsf{TC}^0_d$ circuit of output length $\Theta(\log^2(n))$ and wire complexity less than $\Theta(\log^2(n)) \cdot n^{1+\theta c^{-d}}$. Let $\theta$ be some constant moderately smaller than the parameter required in Lemma 7.6. When $n$ is sufficiently large, both the output length and the wire complexity would satisfy the requirement in Lemma 7.6. Thus, there exists some $y$ such that $g(x) = g(y)$ with probability at least $1/16^{d+1}$. Since there are only $n$ different ways to flip one bit, the algorithm can hit the proper $y$ with probability at least $1/n$. This shows the algorithm can accept the original $\mathsf{TC}^0$ circuit with probability at least $1/(16^{d+1}n)$, which is non-negligible.

Notice that we can easily boost up the gap between the acceptance probability of the two cases by repeating the test polynomial times. So the algorithm can effectively distinguish two cases, which concludes that any PRF cannot be computed by threshold circuits with less than $n^{1+\theta c^{-d}}$ wires if the depths of the circuits are bounded by $d$. □

Similar to Corollary 6.6, we can modify the proof for PRF to obtain an unconditional circuit lower bound for almost universal hash function (and also for ECC with exactly the same wire complexity).

**Theorem 7.7.** Let $c$ be the constant in Lemma 7.5. There exists constants $\theta > 0$ so that the following holds. Let $m = m(n) < n^{0.99^d}$. If $\mathcal{H} = \{H_n \subseteq B_{n,m}\}$ is an almost universal hash function, then for all depth $d \geq 1$, $\mathsf{TC}^0_d\text{-}\mathsf{CC}(\mathcal{H}) \geq n^{1+\theta c^{-d}}$. ♢

**Proof.** Let $\theta$ be the constant in Lemma 7.6. Towards a contradiction assume that $\mathcal{H}$ can be computed by $\mathsf{TC}^0_d$ circuits of size $n^{1+\theta c^{-d}}$ for infinitely many length $n$. Consider the following distribution $\mathcal{D}_n$ supported over pairs of distinct $n$-bit inputs: we first choose an input $x$ uniformly, choose an index $i \leftarrow [n]$, and generate the pair $(x, y)$ where $y$ differs from $x$ only on the $i^{\text{th}}$ bit. By Lemma 7.6, for any $\mathsf{TC}^0_d$ circuit $C$ of size $n^{1+\theta c^{-d}}$, we have

$$\Pr_{(x,y)\leftarrow\mathcal{D}_n}[C(x) = C(y)] \geq \frac{1}{16^{d+1}n}.$$

This means that for those bad $n$,

$$\Pr_{(x,y)\leftarrow\mathcal{D}_n, h\leftarrow H_n}[h(x) = h(y)] \geq \frac{1}{16^{d+1}n}.$$

By averaging argument, for each of the bad $n$, there exists a pair $(x, y)$ of distinct inputs such that $h(x) = h(y)$ with non-negligible probability. This contradicts the almost universality of $\mathcal{H}$. □

### 7.3 Proof of restriction lemma

The technique we use to prove the restriction lemma follows the standard method based on anti-concentration of threshold functions, which has been used to prove average-case hardness [CSS18], quantified derandomization [Tel18] and pseudorandom generator [HHTT21] for sparse $\mathsf{TC}^0$ circuits. There are essentially nothing new in the proof, but we need to carefully check that the previous proofs can be adapted to our statement. The proof mainly involves three steps.

1. We notice that after a random restriction, most gates in depth-1 that are connected to many variables become highly "imbalanced" so that can be approximable by constants. This is done by a structural lemma from [CSS18].

2. Then we count the number of variables feeding a gate that is both "balanced" and of large fan-in. It can be shown that with nice probability, the number of such variables is $o(n)$. We will not include these variables in $S_\rho$, so that fixing all variables outside $S_\rho$ will make all such gates (i.e. both balanced and of large fan-in) become constants.

3. Finally, we only need to consider gates with small fan-in. It is easy to see that it is able to select sufficiently many variables into $S_\rho$, such that for each small gate, at most one of its input variables is in $S_\rho$. By fixing all variables but $S_\rho$, all gates in depth-1 can be approximable by constants.

We start by defining what "balance" means to a threshold function.

**Definition 7.8 ($t$-balance).** A function $\mathsf{LTF}_{w,\theta}(x)$ is called *$t$-balanced* if $|\theta| \le t \cdot \|w\|_2$. Otherwise, it is called *$t$-imbalanced*. ♢

By Hoeffding's inequality (Lemma 4.13), a $t$-imbalanced LTF function has a large fraction of its input being constant.

**Proposition 7.9.** Let $\mathsf{LTF}_{w,\theta}(x)$ be a $t$-imbalanced function, then

$$\Pr_{x\in\{-1,1\}^n}[\mathsf{LTF}_{w,\theta}(x) = \mathrm{sgn}(\theta)] \le 2\exp(-2t^2). \qquad ♢$$

**Proof.** We suppose, without loss of generality, that $\theta > 0$. Then,

$$\begin{aligned}
\Pr_{x\in\{-1,1\}^n}[\mathsf{LTF}_{w,\theta}(x) = 1] &= \Pr_{x\in\{-1,1\}^n}[\langle w, x\rangle \ge t\|w\|_2] && (t\text{-imbalance})\\
&\le 2\exp\left(-\frac{2t^2\|w\|_2^2}{\sum_{i=1}^n w_i^2}\right) && (\text{Lemma 4.13})\\
&= 2\exp(-2t^2). && \square
\end{aligned}$$

We will make use of the following anti-concentration lemma for linear threshold functions.

**Lemma 7.10 ([CSS18], Lemma 4.4; or [Tel18], Proposition 5.8).** There exists absolute constants $p_0 < 1, c_1, c_2 > 0$, such that for any $p \in [0, p_0]$, any LTF function $\Phi$ over $n$ input variables satisfies

$$\Pr_{\rho\leftarrow\mathcal{R}_p}\left[\Phi\restriction_\rho \text{ is } p^{-c_1}\text{-balance}\right] \le O(p^{c_2}). \qquad ♢$$

This lemma tells us that after a random $p$-restriction, any LTF function will become extremely biased with nice probability. If a function is indeed imbalanced after random restriction, we can then approximate it using a constant by Proposition 7.9.

We are now ready to prove the main restriction lemma, following the intuition above.

**Remainder of Lemma 7.5.** There exists absolute constants $c > 1$ and some constant $0 < \varepsilon_0 < 1$ such that following holds. For all $\varepsilon < \varepsilon_0$ and function $\alpha_1(n)$, there exists some $\delta > 0$, such that for all depth $d \ge 1$, output length $m = m(n)$, and sufficiently large $n$, for any $f \in B_{n,m}$ that is $\alpha_1(n)$-approximable by a multi-output depth $d$ threshold circuit of size $n^{1+\varepsilon}$, if we apply the restriction $\rho \leftarrow \mathcal{R}_{n^{-\delta}}$ to the function, with probability at least $1/8$ there exists a set of unfixed variables $S_\rho$ (which depends on $\rho$) of size at least $n^{0.99}$, such that at least a half of the restrictions $\sigma$ to *all* variables not fixed by $\rho$ and not in $S_\rho$ would make $f\restriction_{\rho\sigma}$ $4(\alpha_1(n) + \alpha_2(n))$-approximable by a multi-output threshold circuit of depth $d-1$ and size $|S_\rho|^{1+c\varepsilon}$ (or a transparent function if $d = 1$), where $\alpha_2(n) \triangleq 2n^{1+\varepsilon-\delta}\exp(-2n^{\delta c_1})$. ♢

**Proof.** Let $f \in B_{n,m}$ be a function that can be $\alpha_1(n)$-approximated by a depth $d$ threshold circuit $C$ of size $n^{1+\varepsilon}$. Assume that $\phi_1, \phi_2, \ldots, \phi_t$ are the gates of depth 1 (i.e. directly fed by inputs). A gate $\phi_i$ is called *small* if its in-degree is at most $n^\delta$, where $\delta$ is a constant to be chosen later; and is called *large* otherwise. We use $\Phi_s$ and $\Phi_l$ to denote the set of small and large gates of depth 1, respectively. A variable $x_i$ is called *small* if its out-degree is at most $n^{2\varepsilon}$, and is called *large* otherwise. We use $X_s$ and $X_l$ to denote the set of small and large variables, respectively. Since the circuit has only $n^{1+\varepsilon}$ wires, $|\Phi_l| \le n^{1+\varepsilon-\delta}$ and $|X_l| \le n^{1-\varepsilon}$.

Assume that the circuit $C$ is randomly restricted with $\rho \leftarrow \mathcal{R}_p$ for $p = n^{-\delta/2}$. For simplicity, we identify the restriction $\rho$ as a pair $(I, y)$ where $I \subseteq [n]$ represents the fixed variables and $y \in \{-1,1\}^{|I|}$ represents the assignment. A restriction $\rho = (I, y)$ is called *generic for a large gate* $\phi_i \in \Phi_l$ of in-degree $k$ if the number of free variables after the restriction is in the range $[kp/2, 3kp/2]$, and a restriction $\rho = (I, y)$ (or simply an $I$) is called *generic* if $|X_s \cap \rho^{-1}(\star)| \ge pn/2$, and it is generic for all large gates. Let $\mathcal{G}$ be the event that $\rho$ is generic. Note that $|X_s| \ge n - n^{1-\varepsilon} = n - o(n)$. Clearly,

$$
\begin{aligned}
&\Pr_{\rho \leftarrow \mathcal{R}_p}[\neg \mathcal{G}] \\
\le &\Pr_{\rho \leftarrow \mathcal{R}_p}[|X_s \cap \rho^{-1}(\star)| < pn/2] + \sum_{\phi \in \Phi_l} [\rho \text{ is not generic for } \phi] && \text{(Union bound)} \\
\le &\exp(-\Omega(pn)) + \sum_{\phi \in \Phi_l} [\rho \text{ is not generic for } \phi] && \text{(Chernoff bound)} \\
\le &\exp(-\Omega(pn)) + |\Phi_l| \cdot \exp(-\Omega(p \cdot \text{in-degree}(\phi))) && \text{(Chernoff bound)} \\
\le &\exp(-\Omega(n^{1-\delta/2})) + n^{1+\varepsilon-\delta} \exp(-\Omega(n^{\delta/2})) \\
\le &\mathsf{negl}(n).
\end{aligned}
$$

Hence we can stick to the case when the restriction is generic from now on.

Recall that our goal is to find a large subset $S_\rho$ of unfixed variables such that at least a half of the restrictions that fixes all variables not in $S_\rho$ would make $f$ approximable by a small $\mathsf{TC}^0_{d-1}$ circuit. Let $F_\rho$ be the set of unfixed variables not in $S_\rho$. Now will we define the set $F_\rho$ and $S_\rho$ by the following three-phase procedure.

**Large variables**. Let $n_1 = |X_s \cap I|$, which is the number of unfixed small variables. If $\rho$ is generic, we know that $n_1 \ge pn/2 \ge n^{1-\delta/2}/2$, which is sufficiently large for small $\delta$. In this phase, we simply put all unfixed large variables into $F_\rho$.

**Large gates**. Let $\phi_i \in \Phi_l$ be a large gate. By Lemma 7.10, we know that $\Pr_{\rho \leftarrow \mathcal{R}_p}[\phi_i \text{ is } p^{-c_1}\text{-balanced}] \le p^{c_2}$ for absolute constants $c_1$ and $c_2$, which means that for large $n$,

$$
\Pr_{\rho \leftarrow \mathcal{R}_p}[\phi_i \text{ is } p^{-c_1}\text{-balanced} \mid \mathcal{G}] \le \frac{p^{c_2}}{\Pr_{\rho \leftarrow \mathcal{R}_p}[\mathcal{G}]} \le \frac{p^{c_2}}{1 - \mathsf{negl}(n)} \le 2p^{c_2}.
$$

By Proposition 7.9, the large gates that become imbalanced would output a constant value with high probability, so that they will not bother us. Hence in this phase, we will put all the inputs of $p^{-c_1}$-*balanced* large gates into $F_\rho$.

To analyze the number of inputs that will be put into $F_\rho$, we define a random variable $Y_i$ for each $\phi_i \in \Phi_l$, which is 0 if $\phi_i$ is $p^{-c_1}$-imbalanced, and is the in-degree of $\phi_i$ after the restriction if $\phi_i$ is $p^{-c_1}$-balanced. Let $Y = \sum_{\phi_i \in \Phi_l} Y_i$ be an upper bound of the number of variables to be put into $F_\rho$. Clearly,

$$
\mathop{\mathbb{E}}_{\rho \leftarrow \mathcal{R}_p}[Y_i \mid \mathcal{G}] \le \frac{3p}{2} \cdot \text{in-degree}(\phi_i) \Pr_{\rho \leftarrow \mathcal{R}_p}[Y_i \mid \mathcal{G}] \le 3p^{1+c_2} \cdot \text{in-degree}(\phi_i),
$$

hence

$$\mathop{\mathbb{E}}_{\rho\leftarrow\mathcal{R}_p}[Y \mid \mathcal{G}] = \sum_{\phi_i\in\Phi_l} \mathop{\mathbb{E}}_{\rho\leftarrow\mathcal{R}_p}[Y_i \mid \mathcal{G}] \le 3p^{1+c_2} \cdot \sum_{\phi_i\in\Phi_l} \text{in-degree}(\phi_i) \le 3p^{1+c_2}n^{1+\varepsilon}.$$

Let $\mu \triangleq 3p^{1+c_2}n^{1+\varepsilon}$. By Markov's inequality, we know that

$$\Pr_{\rho\leftarrow\mathcal{R}_p}[Y > 4\mu \mid \mathcal{G}] \le \frac{1}{4}.$$

Let $\mathcal{L}$ be the event that $Y \le 4\mu$, i.e. only $4\mu$ variables are put into $F_\rho$ in this phase. If $\rho$ is generic and we set $\delta$ to be moderately large, say $\delta = 3\varepsilon/c_2$, with probability at least $3/4$, $\mathcal{L}$ will happen, in which case we will put only $O(n^{1+\varepsilon-(1+c_2)\delta/2}) = o(n_1)$ variables into $F_\rho$.

**Small variables feeding small gates**. Now we deal with small gates in depth 1. We will put (roughly) all but $n_1^{1-(2\varepsilon+\delta)}$ variables into $F_\rho$, so that for each small gate $\phi_j \in \Phi_s$, at most one of its input variables is free. Consider the undirected graph $G = (V, E)$ where each node represents a variables that is neither fixed nor put in $F_\rho$ in the above two cases, and two nodes are connected if both of them feed a small gate $\phi_j \in \Phi_s$. Since the out-degree of small variables and in-degree of small gates are all bounded, each node in $G$ is of degree at most $n^{2\varepsilon+\delta}$, hence there exists an independent set $S$ of size $|V|/n^{2\varepsilon+\delta}$. We define $S_\rho$ to be all the vertices inside the independent set, and put all other variables into $F_\rho$. Note that condition on $\mathcal{G}$ and $\mathcal{L}$, the size of $S_\rho$ is at least $\beta \triangleq \frac{np/2-4\mu}{n^{2\varepsilon+\delta}}$.

**Analysis**. Now it is sufficient to show that under the random restriction $\rho$, with nice probability, $|S_\rho| \ge \beta = \frac{np/2-4\mu}{n^{2\varepsilon+\delta}}$ and for at least a half of the assignments $\sigma$ to the unfixed variables not in $S_\rho$, $f\restriction_{\rho\sigma}$ can be approximated by a circuit of size $|S_\rho|^{1+c\varepsilon}$ for an absolute constant $c$. Let $\rho = (I_\rho, y_\rho)$ and $\sigma = (I_\sigma, y_\sigma)$, the circuit $C'$ that approximates $f\restriction_{\rho\sigma}$ is defined as follows. We start with the circuit $C$ that approximates $f$.

1. For a small gate $\rho_j \in \Phi_s$, since at most one of its input variables is in $S_\rho$, it is of in-degree at most 1 after the restrictions $\rho$ and $\sigma$. Hence we can make its descendants directly fed by its input nodes and modify the functions computed by its descendants accordingly without deviating the functionality of the circuit.

2. By the phase dealing with large gates, all $p^{-c_1}$-balanced large gates become in-degree 0. We can replace them by constants and simplify the circuit accordingly. For a $p^{-c_1}$-imbalanced large gate $\phi_i = \mathsf{LTF}_{w,\theta}$ after the restriction, we simply replace it by the most probable constant following Proposition 7.9.

For $d > 1$, the depth of $C'$ becomes $d-1$ since all the gates in depth 1 are eliminated. When $d = 1$, this circuit becomes transparent, i.e. each output node depends on at most one input bit. By the preceding discussion, we have already known that $|S_\rho| \ge \beta$ with certainty if $\mathcal{G}$ and $\mathcal{L}$ holds simultaneously. Now we set the parameters $\varepsilon_0 \triangleq 0.01\min\{c_2, 1/(2+4.5/c_2)\}$ and $\delta \triangleq 3\varepsilon/c_2$, such that for large $n$,

$$\mu = o(np), \beta = \left(\frac{1}{2} - o(1)\right) n^{1-2\varepsilon-3\delta/2} \ge \left(\frac{1}{2} - o(1)\right) n^{1-(2+4.5/c_2)\varepsilon} > n^{0.99}.$$

Since the size of $C'$ is at most the size of $C$, we can see that for large constant $c$ independent of $\varepsilon, n$ and $d$,

$$\text{size}(C') \le \text{size}(C) \le n^{1+\varepsilon} \le \beta^{\frac{1+\varepsilon}{1-(2+4.5/c_2)\varepsilon}} \le |S_\rho|^{1+c\varepsilon}$$

for sufficently large $n$ conditioning on $\mathcal{G}$ and $\mathcal{L}$. Let $\mathcal{S}$ be the event that $|S_\rho| \geq \beta$ and $\text{size}(C') \leq |S_\rho|^{1+c\varepsilon}$. So we can see that

$$\Pr_{\rho\leftarrow\mathcal{R}_p, y_\sigma\leftarrow\mathbb{F}_2^{|I_\sigma|}}[\mathcal{S} \mid \mathcal{G}] = \Pr_{\rho\leftarrow\mathcal{R}_p, y_\sigma\leftarrow\mathbb{F}_2^{|I_\sigma|}}[\mathcal{L}] \cdot \Pr_{\rho\leftarrow\mathcal{R}_p, y_\sigma\leftarrow\mathbb{F}_2^{|I_\sigma|}}[\mathcal{S} \mid \mathcal{G}, \mathcal{L}] = \Pr_{\rho\leftarrow\mathcal{R}_p, y_\sigma\leftarrow\mathbb{F}_2^{|I_\sigma|}}[\mathcal{L}] \geq \frac{3}{4}. \tag{4}$$

Now we show that $C'$ approximates $f\restriction_{\rho\sigma}$. Fix an arbitrary generic $I_\rho$. Firstly, we can see that

$$\Pr_{y_\rho\leftarrow\mathbb{F}_2^{|I_\rho|}, y_\sigma\leftarrow\mathbb{F}_2^{|I_\sigma|}, z\leftarrow\mathbb{F}_2^{|S_\rho|}}[f\restriction_{\rho\sigma}(z) \neq C(y_\rho, y_\sigma, z)] = \Pr_{z\leftarrow\mathbb{F}_2^n}[f(z) \neq C(z)] \leq \alpha_1(n),$$

since $f$ is $\alpha_1(n)$-approximated by $C$. Notice that replacing imbalanced gates is the only place that introduces additional errors in the procedure defining $S_\rho$. Let $\phi\restriction_\rho$ denote the gate $\phi$ under restriction $\rho$ (so that its input is restricted to unfixed variables and its internal function is modified). We can see that

$$\begin{aligned}
&\Pr_{y_\rho\leftarrow\mathbb{F}_2^{|I_\rho|}, y_\sigma\leftarrow\mathbb{F}_2^{|I_\sigma|}, z\leftarrow\mathbb{F}_2^{|S_\rho|}}[C(y_\rho, y_\sigma, z) \neq C'(z)] \\
\leq &\Pr_{y_\rho\leftarrow\mathbb{F}_2^{|I_\rho|}, y_\sigma\leftarrow\mathbb{F}_2^{|I_\sigma|}, z\leftarrow\mathbb{F}_2^{|S_\rho|}}[\exists \phi = \mathsf{LTF}_{w,\theta} \in \Phi_l, \phi\restriction_\rho \text{ is } p^{-c_1}\text{-imbalanced} \wedge \phi(y_\rho, y_\sigma, z) = \text{sgn}(\phi\restriction_\rho)] \\
\leq &\sum_{\phi=\mathsf{LTF}_{w,\theta}\in\Phi_l} \left[\Pr_{y_\rho\leftarrow\mathbb{F}_2^{|I_\rho|}, y_\sigma\leftarrow\mathbb{F}_2^{|I_\sigma|}, z\leftarrow\mathbb{F}_2^{|S_\rho|}}[\phi \text{ is } p^{-c_1}\text{-imbalanced} \wedge \phi = \text{sgn}(\phi\restriction_\rho)]\right] && \text{(Union bound)} \\
\leq &\sum_{\phi=\mathsf{LTF}_{w,\theta}\in\Phi_l} \left[\Pr_{y_\rho\leftarrow\mathbb{F}_2^{|I_\rho|}, y_\sigma\leftarrow\mathbb{F}_2^{|I_\sigma|}, z\leftarrow\mathbb{F}_2^{|S_\rho|}}[\phi = \text{sgn}(\phi\restriction_\rho) \mid \phi \text{ is } p^{-c_1}\text{-imbalanced}]\right] \\
\leq &2|\Phi_l| \cdot \exp(-2p^{-c_1}) && \text{(Proposition 7.9)} \\
\leq &2n^{1+\varepsilon-\delta}\exp(-2n^{\delta c_1}).
\end{aligned}$$

Let $\alpha_2(n) \triangleq 2n^{1+\varepsilon-\delta}\exp(-2n^{\delta c_1})$. Again by union bound,

$$\Pr_{y_\rho\leftarrow\mathbb{F}_2^{|I_\rho|}\ y_\sigma\leftarrow\mathbb{F}_2^{|I_\sigma|}\ z\leftarrow\mathbb{F}_2^{|S_\rho|}}[f\restriction_{\rho\sigma}(z) \neq C'(z)] \leq \alpha_1(n) + \alpha_2(n).$$

By averaging argument, for at least $1/2$ fraction of assignments $y_\rho$ to $I_\rho$, there exists $1/2$ fraction of assignments $y_\sigma$ to $I_\rho$ such that

$$\Pr_{z\leftarrow\mathbb{F}_2^{|S_\rho|}}[f\restriction_{\rho\sigma}(z) \neq C'(z)] \leq 4(\alpha_1(n) + \alpha_2(n)).$$

Let $\mathcal{A}$ be the event that for at least $1/2$ fraction of assignments $y_\sigma$ to the variables in $I_\sigma$, $f\restriction_{\rho\sigma}$ is $4(\alpha_1(n) + \alpha_2(n))$-approximated by $C'$. Then

$$\Pr_{\rho\leftarrow\mathcal{R}_p}[\mathcal{A} \mid \mathcal{G}] \geq \frac{1}{2}. \tag{5}$$

Combining (4) and (5), we can see that for large $n$,

$$\Pr_{\rho\leftarrow\mathcal{R}_\rho}[\mathcal{S} \wedge \mathcal{A}]$$

$$
\begin{aligned}
&\geq \Pr_{\rho \leftarrow \mathcal{R}_\rho}[\mathcal{S} \wedge \mathcal{A} \mid \mathcal{G}] \Pr_{\rho \leftarrow \mathcal{R}_p}[\mathcal{G}] \\
&\geq \left(1 - \Pr_{\rho \leftarrow \mathcal{R}_\rho}[\neg\mathcal{S} \mid \mathcal{G}] - \Pr_{\rho \leftarrow \mathcal{R}_\rho}[\neg\mathcal{A} \mid \mathcal{G}]\right) \Pr_{\rho \leftarrow \mathcal{R}_p}[\mathcal{G}] \\
&\geq \left(1 - \frac{1}{4} - \frac{1}{2}\right)(1 - \mathsf{negl}(n)) \\
&\geq \frac{1}{8}.
\end{aligned}
$$

This means that under a random restriction $\rho \leftarrow \mathcal{R}_p$, with probability $1/8$, there exists a subset $S_\rho$ of unfixed variables such that $|S_\rho| \geq \beta > n^{0.99}$ and for at least a half of the assignments $\sigma$ to unfixed variables not in $S_\rho$, the function $f\restriction_{\rho\sigma}$ can be $4(\alpha_1(n) + \alpha_2(n))$-approximated by a circuit of size $|S_\rho|^{1+c\varepsilon}$, which completes the proof. □

# 8 Open problems

**Other complexity measures.** Besides size or wire complexity, upper bounds and lower bounds about other complexity measures may give us new insights about pseudorandomness. For instance, multiplicative complexity, which is the number of $\wedge$-type gates in the circuit, is closely related to the error propagation in fully-homomorphic encryption. Proving an upper bound better than $n^\varepsilon$ (i.e. Levin's trick) or a lower bound beyond $\log\log n$ (i.e. brute-force low-degree test) is quite interesting. Also, it is open whether there exists a circuit depth lower bound slightly better than $\log n$.

**Improving exact security.** Since our $2n + o(n)$ size hash function has quasi-polynomial collision probability, it cannot be used to create exponentially secure PRFs. Our $3n$ size hash function has collision probability $\exp(-n^\varepsilon)$ but is not uniform. It would be interesting to prove upper and lower bounds for exponentially secure PRFs and uniform almost universal hash functions with exponentially small collision probability.

**PRF and hash functions.** By Levin's trick, a PRF lower bound only implies a conditional lower bound for almost universal hash functions. However, we can even derive *unconditional* lower bounds for hash functions from our techniques. Can we always extract such unconditional lower bounds from PRF lower bound proofs, or proofs under restrictions (for example, non-adaptive, constructive, etc)?

**Impossibility results.** Can we refute the existence of PRF in certain circuit models (for example, $\mathsf{AC}^0[2]$) by proving conflicting upper and lower bounds? Or is there formal evidence that our method cannot be used to prove powerful impossibility results? This can be considered together with the previous one: can we obtain a general connection from PRF lower bounds to hash function lower bounds, to show that this kind of proof cannot be used to refute the existence of PRFs in certain models, since they will refute the existence of hash functions at the same time.

**Practical issues.** Is our PRF construction of size $2n + o(1)$ and depth $(1 + \varepsilon)\log n$ practical? To the best of our knowledge, this is the first construction of pseudorandom function (even in practice) with both small size (related to energy consumption) and low depth (related to efficiency). Also,

our construction is strongly uniform: our hash function only requires a random permutation over input bits and a random sampling to inputs, which seems relatively easy to be implemented in physical hardwares.

### Acknowledgement

We are greatly thankful to Yilei Chen for his support throughout this project and providing lots of useful comments. We also thank Lijie Chen and Ryan Williams for useful discussion.

## References

[AK10] Eric Allender and Michal Koucký. "Amplifying lower bounds by means of self-reducibility". In: *J. ACM* 57.3 (2010), 14:1–14:36. DOI: 10.1145/1706591.1706594. URL: https://doi.org/10.1145/1706591.1706594 (cit. on p. 3).

[And87] A. E. Andreev. "On a method for obtaining more than quadratic effective lower bounds for the complexity of $\pi$-schemes". In: *Vestnik Moskov. Univ. Ser. 1. Mat. Mekh.* (1 1987), pp. 70–73 (cit. on pp. 5, 9).

[App14] Benny Applebaum. "Bootstrapping Obfuscators via Fast Pseudorandom Functions". In: *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014, Proceedings, Part II*. Ed. by Palash Sarkar and Tetsu Iwata. Vol. 8874. Lecture Notes in Computer Science. Springer, 2014, pp. 162–172. DOI: 10.1007/978-3-662-45608-8_9. URL: https://doi.org/10.1007/978-3-662-45608-8%5C_9 (cit. on p. 3).

[AHIKV17] Benny Applebaum, Naama Haramaty, Yuval Ishai, Eyal Kushilevitz, and Vinod Vaikuntanathan. "Low-Complexity Cryptographic Hash Functions". In: *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*. Ed. by Christos H. Papadimitriou. Vol. 67. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, 7:1–7:31. DOI: 10.4230/LIPIcs.ITCS.2017.7. URL: https://doi.org/10.4230/LIPIcs.ITCS.2017.7 (cit. on p. 47).

[BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. "Pseudorandom Functions and Lattices". In: *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*. Ed. by David Pointcheval and Thomas Johansson. Vol. 7237. Lecture Notes in Computer Science. Springer, 2012, pp. 719–737. DOI: 10.1007/978-3-642-29011-4_42. URL: https://doi.org/10.1007/978-3-642-29011-4%5C_42 (cit. on pp. 4, 9, 25, 29).

[Blu84] Norbert Blum. "A Boolean Function Requiring 3n Network Size". In: *Theor. Comput. Sci.* 28 (1984), pp. 337–345. DOI: 10.1016/0304-3975(83)90029-4. URL: https://doi.org/10.1016/0304-3975(83)90029-4 (cit. on p. 8).

[BR17] Andrej Bogdanov and Alon Rosen. "Pseudorandom Functions: Three Decades Later". In: *Tutorials on the Foundations of Cryptography*. Ed. by Yehuda Lindell. Springer International Publishing, 2017, pp. 79–158. DOI: 10.1007/978-3-319-57048-8_3. URL: https://doi.org/10.1007/978-3-319-57048-8%5C_3 (cit. on pp. 3, 4).

[CRVW02] Michael R. Capalbo, Omer Reingold, Salil P. Vadhan, and Avi Wigderson. "Randomness conductors and constant-degree lossless expanders". In: *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*. Ed. by John H. Reif. ACM, 2002, pp. 659–668. DOI: 10.1145/509907.510003. URL: https://doi.org/10.1145/509907.510003 (cit. on pp. 10, 30, 48).

[Cha03] L. Sunil Chandran. "A High Girth Graph Construction". In: *SIAM J. Discret. Math.* 16.3 (2003), pp. 366–370. DOI: 10.1137/S0895480101387893. URL: https://doi.org/10.1137/S0895480101387893 (cit. on p. 24).

[Che18] Lijie Chen. "Toward Super-Polynomial Size Lower Bounds for Depth-Two Threshold Circuits". In: *CoRR* abs/1805.10698 (2018). arXiv: 1805.10698. URL: http://arxiv.org/abs/1805.10698 (cit. on p. 5).

[CJW19] Lijie Chen, Ce Jin, and R. Ryan Williams. "Hardness Magnification for all Sparse NP Languages". In: *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*. Ed. by David Zuckerman. IEEE Computer Society, 2019, pp. 1240–1255. DOI: 10.1109/FOCS.2019.00077. URL: https://doi.org/10.1109/FOCS.2019.00077 (cit. on p. 4).

[CJW20] Lijie Chen, Ce Jin, and R. Ryan Williams. "Sharp threshold results for computational complexity". In: *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*. Ed. by Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy. ACM, 2020, pp. 1335–1348. DOI: 10.1145/3357713.3384283. URL: https://doi.org/10.1145/3357713.3384283 (cit. on p. 4).

[CT19] Lijie Chen and Roei Tell. "Bootstrapping results for threshold circuits "just beyond" known lower bounds". In: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*. Ed. by Moses Charikar and Edith Cohen. ACM, 2019, pp. 34–41. DOI: 10.1145/3313276.3316333. URL: https://doi.org/10.1145/3313276.3316333 (cit. on pp. 3, 4, 5, 6, 9, 10, 12, 26, 30, 48).

[Che+20] Lijie Chen et al. "Beyond Natural Proofs: Hardness Magnification and Locality". In: *11th Innovations in Theoretical Computer Science Conference, ITCS 2020, January 12-14, 2020, Seattle, Washington, USA*. Ed. by Thomas Vidick. Vol. 151. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 70:1–70:48. DOI: 10.4230/LIPIcs.ITCS.2020.70. URL: https://doi.org/10.4230/LIPIcs.ITCS.2020.70 (cit. on pp. 4, 11).

[CSS18] Ruiwen Chen, Rahul Santhanam, and Srikanth Srinivasan. "Average-Case Lower Bounds and Satisfiability Algorithms for Small Threshold Circuits". In: *Theory Comput.* 14.1 (2018), pp. 1–55. DOI: 10.4086/toc.2018.v014a009. URL: https://doi.org/10.4086/toc.2018.v014a009 (cit. on pp. 4, 6, 9, 10, 11, 13, 29, 30, 31, 33, 34).

[DK11] Evgeny Demenkov and Alexander S. Kulikov. "An Elementary Proof of a 3n - o(n) Lower Bound on the Circuit Complexity of Affine Dispersers". In: *Mathematical Foundations of Computer Science 2011 - 36th International Symposium, MFCS 2011, Warsaw, Poland, August 22-26, 2011. Proceedings*. Ed. by Filip Murlak and Piotr Sankowski. Vol. 6907. Lecture Notes in Computer Science. Springer, 2011, pp. 256–265. DOI: 10.1007/978-3-642-22993-0_25. URL: https://doi.org/10.1007/978-3-642-22993-0_25 (cit. on p. 8).

[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: *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*. Ed. by Irit Dinur. IEEE Computer Society, 2016, pp. 89–98. DOI: 10.1109/FOCS.2016.19. URL: https://doi.org/10.1109/FOCS.2016.19 (cit. on p. 8).

[GHKPV13] Anna Gál, Kristoffer Arnsfelt Hansen, Michal Koucký, Pavel Pudlák, and Emanuele Viola. "Tight Bounds on Computing Error-Correcting Codes by Bounded-Depth Circuits With Arbitrary Gates". In: *IEEE Trans. Inf. Theory* 59.10 (2013), pp. 6611–6627. DOI: 10.1109/TIT.2013.2270275. URL: https://doi.org/10.1109/TIT.2013.2270275 (cit. on pp. 21, 30).

[GDP73] S. I. Gelfand, R. L. Dobrushin, and M. S. Pinsker. "On the Complexity of Coding". In: *Second International Symposium on Information Theory*. 1973, pp. 177–184 (cit. on pp. 12, 21, 22).

[Gol86] Oded Goldreich. "Two Remarks Concerning the Goldwasser-Micali-Rivest Signature Scheme". In: *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*. Ed. by Andrew M. Odlyzko. Vol. 263. Lecture Notes in Computer Science. Springer, 1986, pp. 104–110. DOI: 10.1007/3-540-47721-7_8. URL: https://doi.org/10.1007/3-540-47721-7%5C_8 (cit. on p. 3).

[GGM84] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. "How to Construct Random Functions (Extended Abstract)". In: *25th Annual Symposium on Foundations of Computer Science, West Palm Beach, Florida, USA, 24-26 October 1984*. IEEE Computer Society, 1984, pp. 464–479. DOI: 10.1109/SFCS.1984.715949. URL: https://doi.org/10.1109/SFCS.1984.715949 (cit. on pp. 3, 5, 8).

[GL89] Oded Goldreich and Leonid A. Levin. "A Hard-Core Predicate for all One-Way Functions". In: *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*. Ed. by David S. Johnson. ACM, 1989, pp. 25–32. DOI: 10.1145/73007.73010. URL: https://doi.org/10.1145/73007.73010 (cit. on pp. 3, 5).

[GW14] Oded Goldreich and Avi Wigderson. "On derandomizing algorithms that err extremely rarely". In: *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*. Ed. by David B. Shmoys. ACM, 2014, pp. 109–118. DOI: 10.1145/2591796.2591808. URL: https://doi.org/10.1145/2591796.2591808 (cit. on pp. 9, 10).

[Hås86] Johan Håstad. "Almost Optimal Lower Bounds for Small Depth Circuits". In: *Proceedings of the 18th Annual ACM Symposium on Theory of Computing, May 28-30, 1986, Berkeley, California, USA*. Ed. by Juris Hartmanis. ACM, 1986, pp. 6–20. DOI: 10.1145/12130.12132. URL: https://doi.org/10.1145/12130.12132 (cit. on p. 8).

[Hås98] Johan Håstad. "The Shrinkage Exponent of de Morgan Formulas is 2". In: *SIAM J. Comput.* 27.1 (1998), pp. 48–64. DOI: 10.1137/S0097539794261556. URL: https://doi.org/10.1137/S0097539794261556 (cit. on pp. 5, 9).

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. "A Pseudorandom Generator from any One-way Function". In: *SIAM J. Comput.* 28.4 (1999), pp. 1364–1396. DOI: 10.1137/S0097539793244708. URL: https://doi.org/10.1137/S0097539793244708 (cit. on pp. 3, 5, 8).

[HHTT21] Pooya Hatami, William Hoza, Avishay Tal, and Roei Tell. "Fooling Constant-Depth Threshold Circuits". In: *Electron. Colloquium Comput. Complex.* 28 (2021), p. 2. URL: https://eccc.weizmann.ac.il/report/2021/002 (cit. on pp. 4, 6, 9, 10, 11, 29, 30, 31, 33).

[IN93] Russell Impagliazzo and Noam Nisan. "The Effect of Random Restrictions on Formula Size". In: *Random Struct. Algorithms* 4.2 (1993), pp. 121–134. DOI: 10.1002/rsa.3240040202. URL: https://doi.org/10.1002/rsa.3240040202 (cit. on pp. 5, 9).

[IPS93] Russell Impagliazzo, Ramamohan Paturi, and Michael E. Saks. "Size-depth tradeoffs for threshold circuits". In: *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, May 16-18, 1993, San Diego, CA, USA*. Ed. by S. Rao Kosaraju, David S. Johnson, and Alok Aggarwal. ACM, 1993, pp. 541–550. DOI: 10.1145/167088.167233. URL: https://doi.org/10.1145/167088.167233 (cit. on pp. 4, 5, 10, 15).

[IKOS08] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. "Cryptography with constant computational overhead". In: *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*. Ed. by Cynthia Dwork. ACM, 2008, pp. 433–442. DOI: 10.1145/1374376.1374438. URL: https://doi.org/10.1145/1374376.1374438 (cit. on pp. 3, 4, 20).

[IM02] Kazuo Iwama and Hiroki Morizumi. "An Explicit Lower Bound of 5n - o(n) for Boolean Circuits". In: *Mathematical Foundations of Computer Science 2002, 27th International Symposium, MFCS 2002, Warsaw, Poland, August 26-30, 2002, Proceedings*. Ed. by Krzysztof Diks and Wojciech Rytter. Vol. 2420. Lecture Notes in Computer Science. Springer, 2002, pp. 353–364. DOI: 10.1007/3-540-45687-2_29. URL: https://doi.org/10.1007/3-540-45687-2%5C_29 (cit. on p. 5).

[KL01] Matthias Krause and Stefan Lucks. "On the Minimal Hardware Complexity of Pseudorandom Function Generators". In: *STACS 2001, 18th Annual Symposium on Theoretical Aspects of Computer Science, Dresden, Germany, February 15-17, 2001, Proceedings*. Ed. by Afonso Ferreira and Horst Reichel. Vol. 2010. Lecture Notes in Computer Science. Springer, 2001, pp. 419–430. DOI: 10.1007/3-540-44693-1_37. URL: https://doi.org/10.1007/3-540-44693-1%5C_37 (cit. on p. 5).

[LY21] Jiatu Li and Tianqi Yang. "3.1n - o(n) Circuit Lower Bounds for Explicit Functions". In: *Electron. Colloquium Comput. Complex.* 28 (2021), p. 23. URL: https://eccc.weizmann.ac.il/report/2021/023 (cit. on pp. 5, 8).

[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: *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*. Ed. by Moses Charikar and Edith Cohen. ACM, 2019, pp. 1215–1225. DOI: 10.1145/3313276.3316396. URL: https://doi.org/10.1145/3313276.3316396 (cit. on p. 4).

[MV15] Eric Miles and Emanuele Viola. "Substitution-Permutation Networks, Pseudorandom Functions, and Natural Proofs". In: *J. ACM* 62.6 (2015), 46:1–46:29. DOI: 10.1145/2792978. URL: https://doi.org/10.1145/2792978 (cit. on p. 6).

[MW20] Cody D. Murray and R. Ryan Williams. "Circuit Lower Bounds for Nondeterministic Quasi-polytime from a New Easy Witness Lemma". In: *SIAM J. Comput.* 49.5 (2020). DOI: 10.1137/18M1195887. URL: https://doi.org/10.1137/18M1195887 (cit. on p. 5).

[NR04] Moni Naor and Omer Reingold. "Number-theoretic constructions of efficient pseudo-random functions". In: *J. ACM* 51.2 (2004), pp. 231–262. DOI: 10.1145/972639.972643. URL: https://doi.org/10.1145/972639.972643 (cit. on pp. 4, 9, 25, 29).

[Nec66] E.I. Nechiporuk. "On a Boolean function". In: *Soviet Math. Dokl.* (4 1966), pp. 999–1000 (cit. on pp. 5, 9).

[OPS19] Igor Carboni Oliveira, Ján Pich, and Rahul Santhanam. "Hardness Magnification near State-Of-The-Art Lower Bounds". In: *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA*. Ed. by Amir Shpilka. Vol. 137. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, 27:1–27:29. DOI: 10.4230/LIPIcs.CCC.2019.27. URL: https://doi.org/10.4230/LIPIcs.CCC.2019.27 (cit. on p. 3).

[OS18] Igor Carboni Oliveira and Rahul Santhanam. "Hardness Magnification for Natural Problems". In: *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*. Ed. by Mikkel Thorup. IEEE Computer Society, 2018, pp. 65–76. DOI: 10.1109/FOCS.2018.00016. URL: https://doi.org/10.1109/FOCS.2018.00016 (cit. on p. 3).

[PZ93] Mike Paterson and Uri Zwick. "Shrinkage of de Morgan Formulae under Restriction". In: *Random Struct. Algorithms* 4.2 (1993), pp. 135–150. DOI: 10.1002/rsa.3240040203. URL: https://doi.org/10.1002/rsa.3240040203 (cit. on pp. 5, 9).

[PS94] Ramamohan Paturi and Michael E. Saks. "Approximating Threshold Circuits by Rational Functions". In: *Inf. Comput.* 112.2 (1994), pp. 257–272. DOI: 10.1006/inco.1994.1059. URL: https://doi.org/10.1006/inco.1994.1059 (cit. on pp. 15, 30, 48).

[Pau77] Wolfgang J. Paul. "A 2.5 n-Lower Bound on the Combinational Complexity of Boolean Functions". In: *SIAM J. Comput.* 6.3 (1977), pp. 427–443. DOI: 10.1137/0206030. URL: https://doi.org/10.1137/0206030 (cit. on p. 8).

[RRV02] Ran Raz, Omer Reingold, and Salil P. Vadhan. "Extracting all the Randomness and Reducing the Error in Trevisan's Extractors". In: *J. Comput. Syst. Sci.* 65.1 (2002), pp. 97–128. DOI: 10.1006/jcss.2002.1824. URL: https://doi.org/10.1006/jcss.2002.1824 (cit. on p. 10).

[RR97] Alexander A. Razborov and Steven Rudich. "Natural Proofs". In: *J. Comput. Syst. Sci.* 55.1 (1997), pp. 24–35. DOI: 10.1006/jcss.1997.1494. URL: https://doi.org/10.1006/jcss.1997.1494 (cit. on pp. 3, 4, 5, 7, 8, 9, 29).

[RT92] John H. Reif and Stephen R. Tate. "On Threshold Circuits and Polynomial Computation". In: *SIAM J. Comput.* 21.5 (1992), pp. 896–908. DOI: 10.1137/0221053. URL: https://doi.org/10.1137/0221053 (cit. on p. 29).

[Spi96] Daniel A. Spielman. "Linear-time encodable and decodable error-correcting codes". In: *IEEE Trans. Inf. Theory* 42.6 (1996), pp. 1723–1731. DOI: 10.1109/18.556668. URL: https://doi.org/10.1109/18.556668 (cit. on pp. 4, 11, 12, 19, 20, 21).

[Sto77] Larry J. Stockmeyer. "On the Combinational Complexity of Certain Symmetric Boolean Functions". In: *Math. Syst. Theory* 10 (1977), pp. 323–336. DOI: 10.1007/BF01683282. URL: https://doi.org/10.1007/BF01683282 (cit. on p. 8).

[Tal14] Avishay Tal. "Shrinkage of De Morgan Formulae by Spectral Techniques". In: *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*. IEEE Computer Society, 2014, pp. 551–560. DOI: 10.1109/FOCS.2014.65. URL: https://doi.org/10.1109/FOCS.2014.65 (cit. on pp. 5, 9).

[Tel17] Roei Tell. "A Note on the Limitations of Two Black-Box Techniques in Quantified Derandomization". In: *Electron. Colloquium Comput. Complex.* 24 (2017), p. 187. URL: https://eccc.weizmann.ac.il/report/2017/187 (cit. on pp. 8, 11).

[Tel18] Roei Tell. "Quantified derandomization of linear threshold circuits". In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*. Ed. by Ilias Diakonikolas, David Kempe, and Monika Henzinger. ACM, 2018, pp. 855–865. DOI: 10.1145/3188745.3188822. URL: https://doi.org/10.1145/3188745.3188822 (cit. on pp. 3, 4, 6, 9, 10, 11, 29, 30, 31, 33, 34).

[Tel21] Roei Tell. "How to Find Water in the Ocean: A Survey on Quantified Derandomization". In: *Electron. Colloquium Comput. Complex.* 28 (2021), p. 120. URL: https://eccc.weizmann.ac.il/report/2021/120 (cit. on p. 11).

[Tre01] Luca Trevisan. "Extractors and pseudorandom generators". In: *J. ACM* 48.4 (2001), pp. 860–879. DOI: 10.1145/502090.502099. URL: https://doi.org/10.1145/502090.502099 (cit. on p. 10).

# A The leftover lemma for Levin's trick

**Lemma A.1.** Let $m = m(n) = \Theta(n^\varepsilon)$ for $0 < \varepsilon < 1$. Assume that $\mathcal{H} = \{H_n \subseteq B_{n,m}\}_{n\geq 1}$ is an almost universal hash function, then $\mathcal{B} = \{B_n\}_{n\geq 1}$ and $\mathcal{B}' = \{B'_n = \{f \circ h \mid f \in B_m, h \in H_n\}\}_{n\geq 1}$ are indistinguishable. ♢

**Proof.** Towards a contradiction we assume that $\mathcal{B}$ and $\mathcal{B}'$ are distinguishable, then there exists a p.p.t. adversary $\mathcal{A}$ distinguishing $B$ and $B'$. Without loss of generality, we assume that $\mathcal{A}$ queries the oracle on exactly $t = n^d$ distinct points for some constant $d$.

Let $X_1, X_2, \ldots, X_t$ be random variables such that $X_i$ denotes the $i^{\text{th}}$ query point, $h$ be the random variable denoting the hash function involved in $\mathcal{B}'$, and $\mathcal{E}_i$ be the event that $h(X_1), h(X_2), \ldots, h(X_i)$ are pairwisely distinct during the execution of $\mathcal{A}$. Clearly, we know that

$$\Pr_{f\leftarrow B_n,\mathcal{A}}[\mathcal{A}^f(1^n) = 1] = \Pr_{f'\leftarrow B'_n,\mathcal{A}}[\mathcal{A}^{f'}(1^n) = 1 \mid \mathcal{E}_t],$$

since under $\mathcal{E}_t$, $f$ and $f'$ are identically distributed. To obtain a contradiction, it is sufficient to show that

$$\Pr_{f'\leftarrow B',\mathcal{A}}[\neg\mathcal{E}_t] < \mathsf{negl}(n)$$

by elementary probability calculation. By union bound, we can see that

$$\Pr_{f'\leftarrow B',\mathcal{A}}[\neg\mathcal{E}_t] = \Pr_{f'\leftarrow B',\mathcal{A}}[\exists i, \neg\mathcal{E}_i \wedge \mathcal{E}_{i-1}] \leq \sum_{1\leq i\leq t} \Pr_{f'\leftarrow B',\mathcal{A}}[\neg\mathcal{E}_i \mid \mathcal{E}_{i-1}],$$

hence it is sufficient to show that $\Pr_{f' \leftarrow B', \mathcal{A}}[\neg \mathcal{E}_i \mid \mathcal{E}_{i-1}] \leq \mathsf{negl}(n)$.

Conditioning on the fact that $\mathcal{E}_{i-1}$ happens (the hash values of the first $i-1$ queries to the oracle are pairwisely distinct), the oracle returns independent random values. This means that the adaptive adversary does not gain any advantage from the oracle queries, hence the probability that $\neg \mathcal{E}_i$ happens is exactly the collision probability of the hash function, i.e.

$$\begin{aligned}
\Pr[\neg \mathcal{E}_i \mid \mathcal{E}_{i-1}] &= \Pr_{h \leftarrow H_n}[\exists j < i, h(x_i) = h(x_j) \mid \mathcal{E}_{i-1}] \\
&\leq \sum_{1 \leq j < i} \Pr[h(x_i) = h(x_j) \mid \mathcal{E}_{i-1}] \\
&= \sum_{1 \leq j < i} \Pr[h(x_i) = h(x_j) \mid x_i \neq x_j] \\
&\leq \sum_{1 \leq j < i} \mathsf{negl}(n) \\
&\leq \mathsf{negl}(n),
\end{aligned}$$

which completes the proof. □

**Remark.** From the proof one can see that $\mathcal{B}$ and $\mathcal{B}'$ are indistinguishable even if the adversary is not computational bounded, as long as it can perform oracle query for only polynomially many times.

# B Proof of Lemma 5.6

**Remainder of Lemma 5.6.** Let $t = \omega(\log n)$ and $d \geq 3$. There exists a constant $\varepsilon \in (0,1)$, such that for $r = \Theta(n^{\varepsilon})$ and $m = \Theta(n^{1-\varepsilon/2})$, with probability at least $1 - n^{-0.1k}$, Algorithm 1 generates a good graph (and therefore a 1-detector) for sufficiently large $n$. ♢

**Remainder of Algorithm 1:** Generating good graphs

```
for i = 1, 2, ..., t do
    Let G ← (V_1 ∪ V_2, ∅) be an empty graph;
    for v ∈ V_1, j = 1, 2, ..., d do
        Link a random edge e_{v,j} = (v, v') for v' ← V_2;
    end
    if ∀S ⊆ V_1 of size ≤ k, there exists v' ∈ V_2 connects to odd number of vertices in S then
        return G;
    end
end
return ⊥;
```

To analyze this algorithm, we will separately bound the probability that it returns $\perp$ and it returns a graph that is not good. In both of the case, our analysis is similar to the standard probabilistic argument while proving the existence of good graphs.

**Proposition B.1.** Let $d \geq 3$ and $k \geq 1$ be constants, $t = \omega(\log n)$. For any $0 < \varepsilon < 2 - 4/d$ and $m = \Theta(n^{1-\varepsilon/2})$, the algorithm returns $\perp$ with negligible probability. ♢

**Proof.** We bound the probability of refusing to return the graph $G$ in each iteration of the main loop. This will happen when there exists a subset $S \subseteq V_1$ of size at most $k$, such that each $v' \in V_2$ connects to even number of vertices in $S$. Since there are at most $d|S|$ wires connecting to vertices in $S$, at most $d|S|/2$ vertices in $V_2$ connect to more than 2 vertices in $S$. This means that if some subset $S \subseteq V_1$ of size at most $k$ spans more than $d|S|/2$ vertices in $V_2$, then there must exist some $v' \in V_2$ with exactly 1 incidence in $S$. Hence we can calculate the probability at follows.

$$
\begin{aligned}
&\Pr[\exists |S| \le k, \forall v' \in V_2, v' \text{ connects to even \# of vertices in } S] \\
\le &\sum_{S \subseteq V_1, |S| \le k} \left[ \sum_{T \subseteq V_2, |T| = \lfloor d|S|/2 \rfloor} \Pr[S \text{ only connects to } T] \right] \\
\le &\sum_{i=1}^{k} \binom{n}{i} \binom{m}{\lfloor di/2 \rfloor} \left( \frac{\lfloor di/2 \rfloor}{m} \right)^{di} \\
\le &\sum_{i=1}^{k} \left( \frac{ne}{i} \right)^{i} \left( \frac{2me}{di} \right)^{di/2} \left( \frac{di}{2m} \right)^{di} \\
= &\sum_{i=1}^{k} \left( i^{d/2-1} t \right)^{i}. \qquad \left( t \triangleq (ne) \left( \tfrac{2me}{d} \right)^{d/2} \left( \tfrac{d}{2m} \right)^{d} \right)
\end{aligned}
$$

Since both $d$ and $k$ are constants, clearly $t = \Theta(nm^{-d/2})$. If $\varepsilon < \min\{2 - 4/d, 1\}$, then $t = o(1)$. In such case, the probability that the algorithm refuses to return $G$ in each iteration is at most $1/2$ for sufficiently large $n$, which is reduced to negligible for $t = \omega(\log n)$ repetitions. □

**Proposition B.2.** Let $d \ge 3$ be a constant, $0 < \varepsilon < \frac{2d-5}{3d-4}$, $r = \Theta(n^{\varepsilon})$ and $m = \Theta(n^{1-\varepsilon/2})$. Conditioned on the fact that the algorithm does not return $\perp$, the probability that it outputs a good graph is at least $1 - n^{-0.2k}$ for sufficiently large $n$. ♢

**Proof.** Note that a graph $G = (V_1 \cup V_2, E)$ is not good if and only if there exists a set $S \subseteq V_1$ of size $\le r$ such that every vertex in $V_2$ connects to even number of vertices in $S$. Let $\mathcal{E}$ be the event that the algorithm outputs $\perp$. Condition on $\neg\mathcal{E}$, a graph is not good if there exists such a set $S$ with $k < |S| \le r$ (see Line 6 to 8). Similar to Proposition B.1, we can calculate the probability as follows.

$$
\begin{aligned}
&\Pr[G \text{ is not good} \mid \mathcal{E}] \\
\le &\sum_{i=k+1}^{r} \left( \frac{ne}{i} \right)^{i} \left( \frac{2me}{di} \right)^{di/2} \left( \frac{di}{2m} \right)^{di} \\
= &\sum_{i=k+1}^{r} \left( i^{d/2-1} t \right)^{i}. \qquad \left( t \triangleq (ne) \left( \tfrac{2me}{d} \right)^{d/2} \left( \tfrac{d}{2m} \right)^{d} \right)
\end{aligned}
$$

As before, we have $t = \Theta(nm^{-d/2})$. Note that $i \le r = \Theta(n^{\varepsilon})$, if

$$
1 - \frac{d(1-\varepsilon/2)}{2} + \varepsilon \left( \frac{d}{2} - 1 \right) < -\frac{1}{4},
$$

that is $\varepsilon < \frac{2d-5}{3d-4}$, then $i^{d/2-1} t = o(n^{-0.2})$ . In such case, for sufficiently large $n$, the probability that $G$ is not good condition on $\mathcal{E}$ is at most

$$
\sum_{i=k+1}^{\infty} \frac{1}{n^{0.2i}} \le n^{-0.2k}.
$$

□

Then our proof for Lemma 5.6 follows directly.

**Proof of Lemma 5.6.** Let $\varepsilon \triangleq \frac{1}{2}\min\{2-4/d, (2d-5)/(3d-4)\}$. Assume that $\mathcal{G}$ is the event that the algorithm outputs a good graph and $\mathcal{E}$ is the event that the algorithm outputs $\perp$, then

$$\Pr[\neg\mathcal{G}] \leq \Pr[\mathcal{E}] + \Pr[\neg\mathcal{G} \mid \neg\mathcal{E}] \leq \mathsf{negl}(n) + n^{-0.2k} \leq n^{-0.1k}$$

according to Proposition B.1 and B.2. □

**Remark.** From the proof we can clearly see that the *if* clause in Line 6 to 8 is the key to amplify the success probability. A natural question is whether there exists a p.p.t. algorithm checking if a graph is good or not, since such algorithm would further reduce the error probability to negligible instead of polynomial. Unfortunately, such algorithm may not exist, if solving binary analogy of shortest vector problem (binarySVP) is hard for random sparse matrix. Such assumption is used to construct low complexity collision resistant hash function by Applebaum, Haramaty, Ishai, Kushilevitz, and Vaikuntanathan [AHIKV17]. Intuitively, Line 6 to 8 of our algorithm solves binarySVP with width $\leq k = O(1)$, which is sufficient since the error probability is mainly contributed by small width terms.

**Remainder of Corollary 5.7.** For some constant $\varepsilon \in (0,1)$, let $m = m(n) = \Theta(n^{1-\varepsilon/2})$, there exists an almost universal hash function $\mathcal{H} = \{H_n \subseteq B_{n,m}\}_{n\geq 1}$ with weakly uniform complexity $3n$. ♢

**Proof.** Let $d = 3$, $\varepsilon \in (0,1)$ be a constant given by Lemma 5.6 and $r = \Theta(n^{\varepsilon})$. We firstly define a p.p.t. sampling algorithm $\mathcal{G}$ for the hash function and then write down the explicit form $\{H_n \subseteq B_{n,m}\}_{n\geq 1}$ (and the distribution $\mathcal{D}_n$ over $H_n$). Given parameters $n$ and $c$ (where the desired success probability is $n^{-c}$), our algorithm firstly runs Algorithm 1 with $k = 10c$ and obtain a depth-1 XOR circuit $C$ with unbounded fan-in (if Algorithm 1 returns $\perp$, we immediately output $\perp$). We expand $C$ to a $B_2$ circuit of size $3n$ by realizing each XOR gates as a tree of fan-in 2 XOR gates. Then, similar to Lemma 5.5, we randomly choose a subset $S \subseteq [n]$ of size $s = \Theta(n^{1-\varepsilon/2})$, say $S = \{i_1, i_2, \ldots, i_s\}$, and labels $x_{i_1}, x_{i_2}, \ldots, x_{i_s}$ to be output bits. The resulting circuit is the output of our algorithm.

Now we write down the explicit form. Let $\mathcal{E}$ be the event that Algorithm 1 generates a good graph. Let $C_n$ be the set of circuits that are generated by $\mathcal{G}(1^n, k=0)$ with non-zero probability condition on $\mathcal{E}$, and $H_n \triangleq \{h \mid \exists C \in C_n, C \text{ computes } h\}$. The distribution $\mathcal{D}_n$ over $H_n$ is defined as

$$\mathcal{D}_n(h) \triangleq \Pr_{\mathcal{G}}[\mathcal{G}(1^n, c=0) \text{ outputs a circuit computing } h \mid \mathcal{E}]$$

for all $h \in H_n$. By Lemma 5.6, we know that $\Pr[\mathcal{E}] \geq 1 - n^{-0.1k} = 1 - n^{-c}$. We can easily see that the parameter $k$ does not influence the output distribution of Algorithm 1 condition on $\mathcal{E}$, hence for all positive integer $c$,

$$\mathcal{D}_n(h) = \Pr_{\mathcal{G}}[\mathcal{G}(1^n, c) \text{ outputs a circuit computing } h \mid \mathcal{E}].$$

This means that the family $\mathcal{H} = \{H_n\}_{n\geq 1}$ is weakly uniform of complexity $3n$.

Finally it is sufficient to show that $\mathcal{H}$ is actually a hash function. According to the definition, one can see that a random function $h \leftarrow H_n$ can be represented by a pair $(C, S)$ of random variables, where $C$ is a $(n, r, m)$ 1-detector and $S$ is a subset of $[n]$ of size $s$. Similar to Lemma 5.5, we can see that for each fixed $C$ and any $x \neq y$, $\Pr[h(x) = h(y)] < \mathsf{negl}(n)$. Hence it directly implies that $\Pr_{h=(C,S)\leftarrow H_n}[h(x) = h(y)] < \mathsf{negl}(n)$, which completes the proof. □

# C An optimally sparse explicit almost universal hash in $\mathsf{CC}^0[2]$

As we have discussed above, both of our constructions in Section 5.3 and Section 5.4 have its own weakness: the $3n$ construction based on counting is only weakly uniform, and the $2n$ construction based on high-girth graph has only logarithmic shrinkage. Although we do not know how to construct a depth-1 linear-wire-complexity sparse almost universal hash function with both uniformity and polynomial shrinkage, we can construct a linear-wire-complexity almost universal hash with both uniformity and polynomial shrinkage in depth-2 XOR circuit. In this section, we will present such a construction by composing the hash function in Section 5.4 with an explicit almost universal hash of polynomial shrinkage and with slightly super-linear wire complexity.

**Definition C.1.** Let $n$ be the input length and $m$ be the output length. The circuit class $\mathsf{CC}^0[2]$ contains the set of circuits with only unbounded fan-in XOR gates of constant depth. ♢

**Theorem C.2.** Let $0 < c < 1$ be a constant and $m = m(n) = O(n^c)$ be the output length, there exists an almost universal hash function $\mathcal{H} = \{H_n \subseteq B_{n,m}\}_{n\geq 1}$ with uniform wire complexity $2n + o(n)$ and of depth 2 in $\mathsf{CC}^0[2]$. ♢

By introducing explicit lossless expander, it is possible to construct explicit 1-detector with polynomial shrinkage.

**Theorem C.3 ([CRVW02; CT19]).** There exists some function $p(n) \in \mathsf{poly}(n)$ such that the following holds. Let $\varepsilon > 0$ be some arbitrarily small constant. Let $m = m(n)$ be a function. For all $n, m$ that are both power of 2, there exists some $G = (V_1 \cup V_2, E \subseteq V_1 \times V_2)$ where $|V_1| = n, |V_2| = m$, the degree of each vertex in $V_1$ is $d = 2^{p(\log\log(n/m))}$, and every subset $S \subseteq V_1$ of size at most $\Theta(m/d)$ vertices is connected to at least $(1-\varepsilon)d|S|$ neighbors. Such graph is called a lossless expander. Moreover, there exists an algorithm computes $G$ in deterministic polynomial time. ♢

Pick $\varepsilon = 1/3$. Let $S \subseteq V_1$ be an arbitrary subset of size at most $\Theta(m/d)$. Since there are at most $d|S|$ wires connecting to vertices in $S$, at most $d|S|/2$ neighbors connects to more than 2 vertices in $S$. From the expansion of the graph, there must exist some vertex in $V_2$ connects to exactly one vertex in $S$. This shows the depth-1 circuit with such topological structure is a $(n, \Theta(m/2^{p(\log\log(n/m))}), m)$ 1-detector with $n \cdot 2^{p(\log\log(n/m))}$ wires. According to Lemma 5.5, there exists an explicit almost universal hash function of the same wire complexity.

**Lemma C.4.** There exists some function $p(n) \in \mathsf{poly}(n)$. For any constant $\varepsilon \in (0,1)$, there exists an almost universal hash function $\mathcal{H}^1 = \{H^1_{n,m} \subseteq B_{n,m}\}_{n\geq 1}$ for $m = m(n) = \Theta(n^\varepsilon)$, with uniform wire complexity $n \cdot 2^{p(\log\log(n/m))}$ and of depth 1 in $\mathsf{CC}^0[2]$. ♢

It is important to mention that this hash function also give us an explicit almost universal hash function in extremely sparse $\mathsf{TC}^0_d$. According to Paturi and Saks [PS94], the parity function over $n$ input variables can be realized by a $\mathsf{TC}^0_d$ circuit of wire complexity $n^{1+O(\phi^{-d})}$. By replacing gates by such structure, it is possible to transform a depth-1 circuit of wire complexity $m$ into a $\mathsf{TC}^0_d$ circuit of wire complexity $n^{O(\phi^{-d})}$. This shows the above almost universal hash function can be transformed into a $\mathsf{TC}^0_d$ circuit of wire complexity $n \cdot 2^{p(\log\log(n/m))} \cdot n^{O(\phi^{-d})} = n^{1+O(\phi^{-d})}$.

**Theorem C.5.** For any constant $\varepsilon \in (0,1)$ and $d \geq 2$, let $m = m(n) = \Theta(n^\varepsilon)$, there exists an almost universal hash function $\mathcal{H} = \{H_n \subseteq B_{n,m}\}_{n\geq 1}$ with wire complexity $n^{1+O(\phi^{-d})}$ and in $\mathsf{TC}^0_d$. ♢

To construct a linear-size almost universal hash function, we would hide the logarithmic factor in Lemma C.4 by composing it with a variant of construction in Section 5.4. Let $\hat{p}(x) \triangleq 2 \cdot p(x) \in \mathsf{poly}(n)$ and $m = n/2^{\hat{p}(\log\log(n))}$. The generated graph $D_{m,n}$ according to Corollary 5.11 is of girth $g = \Omega\left(\frac{\log m}{\log(2n/m)}\right) = \Omega\left(\frac{\log n}{\hat{p}(\log\log n)}\right)$. From Lemma 5.9, it can be transformed into a $(n, n/2, n/2^{\hat{p}(\log\log(n))})$ randomized 1-detector of $2n$ wires. According to Lemma 5.5, we can construct an explicit almost universal hash function with the same wire complexity.

**Lemma C.6.** For some $\hat{p}(n) \in \mathsf{poly}(n)$, let $m = m(n) = n/2^{\hat{p}(\log\log(n))}$, there exists an almost universal hash function $\mathcal{H}^2 = \{H^2_{n,m} \subseteq B_{n,m}\}_{n\geq 1}$ with uniform wire complexity $2n$ and of depth 1 in $\mathsf{CC}^0[2]$. ♢

By composing the above hash function in Lemma C.6 together with the hash function in Lemma C.4, it is possible to construct an almost universal hash function with all desired properties.

**Theorem C.7.** For any constant $\varepsilon \in (0,1)$, let $m = m(n) = \Theta(n^\varepsilon)$, there exists an almost universal hash function $\mathcal{H} = \{H_n \subseteq B_{n,m}\}_{n\geq 1}$ with uniform wire complexity $2n + o(n)$ and of depth 2 in $\mathsf{CC}^0[2]$. ♢

**Proof.** Let $H^1_{n,k}$ be the hash function in Lemma C.4 and let $H^2_{n,k}$ be the hash function in Lemma C.6, where $k = n/2^{\hat{p}(\log\log n)}$. We construct a hash function $\mathcal{H} = \{H_n \subseteq B_{n,m}\}_{n\geq 1}$ where

$$H_n \triangleq \{h_2 \circ h_1 | h_2 \in H^2_{k,m}, h_1 \in H^1_{n,k}\}.$$

This is clearly almost universal.

Now we consider the wire complexity. The first layer of the circuit requires $2n$ wires and the second layer of the circuit requires

$$k \cdot 2^{p(\log\log(k/m))} = n \cdot 2^{p(\log\log(k/m)) - \hat{p}(\log\log(n))} = o(n)$$

wires. Therefore, $\mathcal{H}$ can be computed by a $\mathsf{CC}^0[2]$ circuit of $2n + o(n)$ wires and depth 2. □

Since a fan-in-$n$ XOR gates can be realized by $n-1$ fan-in-2 XOR circuits in depth $\log n + O(1)$, we can transform the $\mathsf{CC}^0[2]$ circuit into a sparse and shallow $B_2$ circuit. This gives us an explicit linear-size almost universal hash function in $B_2$.

**Corollary C.8.** For any constant $\varepsilon \in (0,1)$, let $m = m(n) = \Theta(n^\varepsilon)$, there exists an almost universal hash function $\mathcal{H} = \{H_n \subseteq B_{n,m}\}_{n\geq 1}$ of size $2n + o(n)$ and depth $\log n + O(1)$ in $B_2$. ♢

**Proof.** Look into the construction of Theorem C.7. The first layer of the circuit can be transformed into a $B_2$ circuit of size $2n$ and depth $\log(n/k) + O(1)$. The second layer of the circuit can be transformed into a $B_2$ circuit of size $o(n)$ and depth $\log(k) + O(1)$. In general, it is a $B_2$ circuit of size $2n + o(n)$ and depth $\log(n) + O(1)$. □