Electronic Colloquium on Computational Complexity, Report No. 3 (2021)

Inverse-Exponential Correlation Bounds and Extremely Rigid Matrices from a New Derandomized XOR Lemma

Lijie Chen
MIT
lijieche@mit.edu

Xin Lyu
Tsinghua University
lvx17@mails.tsinghua.edu.cn

January 5, 2021

Abstract

In this work we prove that there is a function $f \in \mathsf{E}^{\mathsf{NP}}$ such that, for every sufficiently large n and $d = \sqrt{n}/\log n$, f_n (f restricted to n-bit inputs) cannot be $(1/2 + 2^{-d})$-approximated by $\mathbb{F}_2$-polynomials of degree d. We also observe that a minor improvement (*e.g.*, improving d to $n^{1/2+\varepsilon}$ for any $\varepsilon > 0$) over our result will imply E^{NP} cannot be computed by depth-3 AC^0-circuits of $2^{n^{1/2+\varepsilon}}$ size, which is a notoriously hard open question in complexity theory.

Using the same proof techniques, we are also able to construct *extremely rigid* matrices over $\mathbb{F}_2$ in P^{NP}. More specifically, we show that for every constant $\varepsilon \in (0,1)$, there is a P^{NP} algorithm which on input 1^n outputs an $n \times n$ $\mathbb{F}_2$-matrix H_n satisfying $\mathcal{R}_{H_n}(2^{\log^{1-\varepsilon} n}) \geq (1/2 - \exp(-\log^{2/3 \cdot \varepsilon} n)) \cdot n^2$, for every sufficiently large n. This improves the recent P^{NP} constructions of rigid matrices in [Alman and Chen, FOCS 2019] and [Bhangale et al., FOCS 2020], which only gives $\Omega(n^2)$ rigidity.

The key ingredient in the proof of our new results is a new *derandomized XOR lemma* based on *approximate linear sums*, which roughly says that given an n-input function f which cannot be 0.99-approximated by certain linear sum of s many functions in $\mathcal{F}$ within ℓ_1-distance, one can construct a new function Amp^f with $\widetilde{O}(n)$ input bits, which cannot be $(1/2 + s^{\Omega(1)})$-approximated by $\mathcal{F}$-functions. Taking $\mathcal{F}$ to be a function collection containing low-degree $\mathbb{F}_2$-polynomials or low-rank $\mathbb{F}_2$-matrices, our results are then obtained by first using the algorithmic method to construct a function which is weakly hard against linear sums of $\mathcal{F}$ in the above sense, and then apply the derandomized XOR lemma to f.

We obtain our new derandomized XOR lemma by giving a generalization of the famous hardcore lemma by Impagliazzo. Our generalization in some sense constructs a *non-Boolean* hardcore of a weakly hard function f with respect to $\mathcal{F}$-functions, from the weak inapproximability of f by any linear sum of $\mathcal{F}$ with bounded ℓ_p-norm. This generalization recovers the original hardcore lemma by considering the ℓ_∞-norm. Surprisingly, when we switch to the ℓ_1-norm, we immediately rediscover Levin's proof of Yao's XOR Lemma. That is, these first two proofs of Yao's XOR Lemma can be unified with our new perspective. For proving the correlation bounds, our new derandomized XOR lemma indeed works with the $\ell_{4/3}$-norm.

ISSN 1433-8092

Contents

1 Introduction

1.1 Background

We consider multivariate polynomials over $\mathbb{F}_2$. In particular, every n-variable polynomial $P\colon \{0,1\}^n \to \{0,1\}$ over $\mathbb{F}_2$ can be written as

$$P(x_1,\ldots,x_n) \equiv \sum_{S\subseteq[n]} \alpha_S \cdot \prod_{i\in S} x_i. \pmod 2$$

The degree of P, denoted as $\deg(P)$, is defined as $\max\{|S| : \alpha_S \neq 0\}$.

Understanding the power and limitations of $\mathbb{F}_2$-polynomials as a model of computation is of fundamental interest in complexity theory. We will be particularly interested in exhibiting functions which are hard to be approximated by low-degree $\mathbb{F}_2$-polynomials.[1] Formally, for any two functions $f, g\colon \{0,1\}^n \to \{0,1\}$, we define their correlation as

$$\operatorname{corr}(f,g) = \left| \Pr_{x\leftarrow U_n}[f(x) = g(x)] - \Pr_{x\leftarrow U_n}[f(x) \neq g(x)] \right|.$$

Let $\mathcal{P}_d$ be the collection of $\mathbb{F}_2$-polynomials with degree $\leq d$. Slightly abuse notation, we also write $\operatorname{corr}(f,d) := \max_{P\in\mathcal{P}_d} \operatorname{corr}(f,P)$.

The correlation bound against $\mathbb{F}_2$-polynomials[2], a major question in complexity theory, asks to find an *explicit* function (*i.e.*, a function in NP) which cannot be approximated by low-degree $\mathbb{F}_2$-polynomials, that is, to prove upper bounds on $\operatorname{corr}(f,d)$ for an explicit function f and various choices of d.

Since $\mathbb{F}_2$-polynomial is such a simple and elegant computational model, this question is clearly interesting in its own right. It is nonetheless also inherently connected to a variety of other fundamental questions in complexity theory. In fact, it was shown that improvements on correlation bound against $\mathbb{F}_2$-polynomials are necessary or sufficient for progress on other major open questions. We highlight one typical example below (for more connections, we refer to the excellent exposition by Viola [Vio09]).

1.1.1 Connection to $\mathsf{AC}^0[\oplus]$-Circuits

$\mathsf{AC}^0[\oplus]$ denotes the class of constant-depth circuits consisting of AND, OR, XOR and NOT gates of unbounded fan-in. The celebrated polynomial approximation method, which shows lower bounds for $\mathsf{AC}^0[\oplus]$-circuits follow from correlation bounds against $\mathbb{F}_2$-polynomials, was first established by Razborov [Raz87], who applied this connection to prove the first super-polynomial lower bounds against $\mathsf{AC}^0[\oplus]$, which was later refined and improved by [Smo87, Smo93]. This connection has since then become the integral tool in the study of the computational power of $\mathsf{AC}^0[\oplus]$-circuits, see [RSS18, KS18, CHLT19, OSS19, LSSTV19, Vio20b, CGLLS20] for some recent work on this direction.

1.1.2 Known Progress on Correlation Bounds for Explicit Functions against $\mathbb{F}_2$-Polynomials

The renowned work by Razborov and Smolensky [Raz87, Smo87, Smo93] mentioned above proved that the majority function MAJ[3] satisfies $\operatorname{corr}(\mathsf{MAJ}_n, d) \leq O(d/\sqrt{n})$. Despite nearly 40 years

[1] We remark that it is not hard to exhibit a function which cannot be *exactly* computed by $\mathbb{F}_2$-polynomials of degree less than n. An example would be the AND function on n input bits.

[2] In the Boolean setting, correlation bounds and average-case lower bounds are equivalent, and we will use these two terms interchangeably in this paper (see Section 3.1).

[3] The function $\mathsf{MAJ}_n\colon \{0,1\}^n \to \{0,1\}$ outputs 1 if and only if there are more ones than zeros in its input.

passed, they remain the strongest correlation bound we known for any explicit functions in NP against $\log n$-degree $\mathbb{F}_2$-polynomials, and no $o(1)$ correlation bounds were known for $\sqrt{n}$-degree $\mathcal{F}_n$-polynomials. Indeed, even the following questions are still open (see [Vio09] for more in-depth discussions of these open problems).

Open Problem 1. *Is there a function $f \in \mathsf{NP}$ such that $\operatorname{corr}(f, \log n) \le o(1/\sqrt{n})$?*

Open Problem 2. *Is there a function $f \in \mathsf{NP}$ such that $\operatorname{corr}(f, \sqrt{n}) \le o(1)$?*

Recall that the Razborov-Smolensky bound showed that $\operatorname{corr}(\mathsf{MAJ}_n, \log n) \le O(\log n/\sqrt{n})$ and $\operatorname{corr}(\mathsf{MAJ}_n, o(\sqrt{n})) \le o(1)$.[4] These two open questions ask to improve the Razborov-Smolensky bound for the degree $\log n$ or $\sqrt{n}$, by *even a tiny amount*.

For lower degrees ($d \le o(\log n)$), stronger correlation bounds were known by [BNS92, Bou05, GRS05, VW08, CHHLZ20]. It was also proved that very strong inverse exponential correlation bounds against constant-degree $\mathbb{F}_2$-polynomials would improve the state-of-the-art lower bound against unrestricted Boolean circuits [GKW18].

1.1.3 Recent Progress on Correlation Bounds for "Semi-explicit" Functions against $\mathbb{F}_2$-Polynomials

Since the progress on resolving Open Problem 1 and Open Problem 2 for an explicit function in NP has been lacking for decades, it motivates the interest to exhibit a hard function in some larger complexity classes first.

In fact, Open Problem 1 was even open for the gigantic classes E^{NP}, until the recent independent work by Chen and Ren [CR20] and Viola [Vio20b]. In [CR20], a $(1/2 + 2^{-\operatorname{polylog}(n)})$-inapproximability bound for NQP[5] against ACC^0 was proved, which implies that the there is a function $f \in \mathsf{NQP}$ such that $\operatorname{corr}(f, \operatorname{polylog}(n)) \le 2^{-\operatorname{polylog}(n)}$. In Viola [Vio20b], it is established that there is a function $f \in \mathsf{E}^{\mathsf{NP}}$ such that $\operatorname{corr}(f, n^{o(1)}) \le n^{-1+\varepsilon}$ for any $\varepsilon > 0$.

Later, a recent work by Chen, Lyu and Williams [CLW20] proved that for every constant $\varepsilon > 0$, there is a function $f \in \mathsf{E}^{\mathsf{NP}}$ such that $\operatorname{corr}(f, n^{1/2-\varepsilon}) \le \exp(-n^{\Omega_\varepsilon(1)})$, where $\Omega_\varepsilon(1)$ is a constant approaching 0 when ε approaching 0. However, Open Problem 2 still remained open even for E^{NP}.

1.2 Our Results

1.2.1 Strong Correlation Bounds "One Epsilon" Away from a Circuit Lower Bound Breakthrough

In this work, we significantly improved upon the correlation bounds from [CLW20].

Theorem 1.1. *There is a function $f \in \mathsf{E}^{\mathsf{NP}}$ which, for every sufficiently large n and for any $d \le o(n/\log n)^{1/2}$, it holds that $\operatorname{corr}(f_n, d) \le 2^{-d}$.*

One may wonder whether Theorem 1.1 can be further improved to, say, $\operatorname{corr}(f_n, d) \le 2^{-d}$ for $d = n^{1/2+\varepsilon}$. We observe that such improvement will imply new depth-3 AC^0-circuits lower bounds.

Theorem 1.2. *For any function $d(n) \colon \mathbb{N} \to \mathbb{N}$, if there is a function f in E^{NP} such that $\operatorname{corr}(f_n, d(n)) \le 2^{-d(n)}$ for infinitely many $n \in \mathbb{N}$, then there is a function g in E^{NP} that does not admit depth-3 AC^0-circuits of size at most $2^{o(d(n))}$.*

[4] These are indeed tight for MAJ, see [Vio20a].

[5] $\mathsf{NQP} := \mathsf{NTIME}[n^{\operatorname{polylog}(n)}]$ stands for nondeterministic quasi-polynomial time.

Currently, the best known depth-3 AC^0-circuits lower bound is $2^{\Omega(\sqrt{n})}$ [Hås89]. It has been a notorious open question to prove better lower bounds against depth-3 AC^0-circuits, even for functions in complexity class as large as E^{NP}. Therefore, improving our result, even by an "epsilon amount" on the exponent, would imply a breakthrough in constant-depth circuit lower bounds.

1.2.2 Better Degree-Error Trade-Off for E^{NP} against $\mathbb{F}_2$-Polynomials

Due to some technical obstacles, the proof of Theorem 1.1 does not give any non-trivial correlation bounds for higher degrees (*i.e.*, $d \geq \sqrt{n}$). Still, using a different proof, we managed to show a trade-off between error and degree for E^{NP} against degree-d $\mathbb{F}_2$-polynomials.

Theorem 1.3. *For every $\beta \in (0,1)$, there is an E^{NP} function f such that, for every sufficiently large n, it holds that* $\operatorname{corr}(f, n^{\beta}/\log n) \leq \exp(-\Omega(n^{\frac{2}{3}(1-\beta)}))$.

Setting $\beta = 0.5$, Theorem 1.3 resolved Open Problem 2 for E^{NP} in a very strong way. It gives an inverse exponential correlation of $2^{-\widetilde{\Omega}(n^{1/3})}$ against $\sqrt{n}$-degree $\mathbb{F}_2$-polynomials. However, we remark that this does not match the correlation bound in Theorem 1.1 when β is slightly less than 0.5, and we leave it as an interesting open question to obtain a better trade-off. (We believe that $\operatorname{corr}(f,d) \leq \exp(-\Omega(n^{\frac{2}{3}(1-\beta)}/\log n))$ can be further improved to $\operatorname{corr}(f,d) \leq \exp(-\Omega(n^{(1-\beta)}/\log n))$, for all $\beta \in (0,1)$.)

1.2.3 P^{NP} Construction of Extremely Rigid Matrices

In fact, the proof of Theorem 1.3 above builds on the techniques behind the P^{NP}-construction of rigid matrices [BHPT20, AC19]. More specifically, we rely on the rectangular PCPs constructed in [BHPT20]. Fully utilizing this technique, we can also construct *extremely rigid* matrices over $\mathbb{F}_2$ in P^{NP}.

We first recall the definition of rigid matrices.

Definition 1.4. *For $r, n \in \mathbb{N}$ and a matrix $M \in \mathbb{F}_2^{n\times n}$, the r-rigidity of M, denoted as $\mathcal{R}_M(r)$, is the minimum number of entries one needs to change in M to make its rank over $\mathbb{F}_2$ at most r.*

Theorem 1.5. *For every constant $\varepsilon \in (0,1)$, there is a P^{NP} algorithm which on input 1^n outputs an $n \times n$ $\mathbb{F}_2$-matrix H_n satisfying $\mathcal{R}_{H_n}(2^{\log^{1-\varepsilon} n}) \geq (1/2 - \exp(-\log^{2/3\cdot\varepsilon} n)) \cdot n^2$, for every sufficiently large n.*

The matrix H constructed in Theorem 1.5 is *extremely rigid* in the sense that even $\mathcal{R}_{H_n}(1)$ cannot be greater than $1/2 \cdot n^2$ (either the all-zero or the all-one matrix agrees with H on at least $1/2 \cdot n^2$ entries).

Comparison with previous work. The problem of efficiently constructing *rigid matrices* is a long-standing open problem in complexity theory [Val77, Lok09, Raz89].

Alman and Chen [AC19] established a P^{NP} construction of matrices H_n satisfying $\mathcal{R}_{H_n}(2^{\log^{1/4-\varepsilon} n}) \geq \Omega(n^2)$, which was later significantly improved to $\mathcal{R}_{H_n}(2^{\log^{1-\varepsilon} n}) \geq \Omega(n^2)$ by Bhangale, Harsha, Paradise and Tal [BHPT20]. The constructions in both of [AC19] and [BHPT20] are infinitely-often constructions in the sense that they only work for infinitely many input lengths n. [CLW20] recently improved [BHPT20]'s construction so that it also works *almost everywhere*, in the sense that it now construct rigid matrices for every sufficiently large input length n.

Our construction improved upon all the previous work by further showing an almost-everywhere P^{NP} construction of extremely rigid matrices, still with respect to rank $2^{\log^{1-\varepsilon} n}$.

Independent Work. Without using our new derandomized XOR lemma, it is possible to combine the known techniques in [CLW20] and [BHPT20] in a very non-trivial way to obtain a weaker trade-off that $\mathcal{R}_{H_n}(2^{\log^{1-\varepsilon} n}) \geq (1/2 - \exp(-\log^{1/2\cdot\varepsilon} n)) \cdot n^2$, which would also resolve Open Problem 2 for E^{NP}. Such a weaker trade-off has been independently proved by Lu [Lu20] and by Huang and Viola [HV20].[6]

The common insight in our work and both [Lu20] and [HV20] is that we can apply the classic XOR Lemma even in the setting of constructing rigid matrices.

1.2.4 Nondeterministic PRGs with Near-Logarithmic Seed-Length from Non-Trivial Algorithms

In [CLW20], it was shown that for a typical circuit class $\mathcal{C}$, a $2^{n-n^{\varepsilon}}$-time CAPP algorithm[7] for $2^{n^{\varepsilon}}$-size $\mathcal{C}$-circuits implies an infinite-often non-deterministic PRGs (i.o.-NPRGs)[8] for polynomial-size $\mathcal{C}$ with $\mathrm{polylog}(n)$ seed length. Combing with the corresponding algorithm for polynomial-size ACC^0 [Wil14], this immediately implies an i.o.-NPRG for ACC^0 with $\mathrm{polylog}(n)$ seed length.

Naturally, one may wonder whether that seed length can be improved to the optimal $O(\log n)$ if one starts with a $2^{n-\varepsilon n}$-time algorithm instead. We show that this is the case, by proving the following theorem.

Theorem 1.6 (Informal)**.** *Let $\mathcal{C}$ be a nice circuit class. If there is an $\varepsilon > 0$ such that, the $\#\mathsf{SAT}$ (or CAPP) problem of $\mathcal{C} \circ \mathsf{Junta}_{2^{\varepsilon n}}$-circuit of size $2^{\varepsilon n}$ can be solved in $2^{(1-\varepsilon)n}$ time, then there exists an infinitely often NPRG which takes $O(\log n \log\log^2 n)$ bits seeds, runs in time $\mathrm{poly}(n)$ and fools $\mathscr{C}$-circuits of size n.*

1.3 Our Techniques

Perhaps more interestingly, our new results are all proved by a new derandomized XOR lemma based on approximate linear sums, and as far as we know, this is the first application of hardness amplification in the context of constructing rigid matrices. Before formally stating and discussing our new derandomized XOR lemma, it is instructive to review the XOR lemma in [CLW20], and why it cannot be used to prove the strong correlation bounds as in Theorem 1.1.

Notation. From now on, we will always use Boolean functions to denote a function from $\{0,1\}^*$ to $\{-1,1\}$, where -1 and 1 are interpreted as True and False, respectively. This choice will be particularly convenient for studying and stating correlation bounds or average-case lower bounds. For two functions $f, g\colon \{0,1\}^n \to \{-1,1\}$, we will use $\langle f, g\rangle$ to denote their inner product $\mathbb{E}_{x\in\{0,1\}^n}[f(x)\cdot g(x)]$.[9]

For a collection of functions $\mathcal{F}$, we always use $\mathcal{F}_n$ to denote the subset of $\mathcal{F}$ consisting of n-bit functions from $\mathcal{F}$. We will also need to define $\mathsf{Sum}\circ\mathcal{F}$-functions: a $\mathsf{Sum}\circ\mathcal{F}$-function $C\colon \{0,1\}^n \to \mathbb{R}$ can be written as $C(x) = \sum_{i=1}^{\ell} \alpha_i \cdot C_i(x)$, where each α_i is a real, and each $C_i(x)$ is an $\mathcal{F}_n$-function. Here ℓ is called the *sparsity* of C. We also use $\mathsf{complexity}(C)$ to denote $\max(\ell, \sum_{i=1}^{\ell} |\alpha_i|)$.

[6] More precisely, [HV20] and [Lu20] stated a more fine-grained result that $\mathcal{R}_{H_n}(\rho) \geq (1/2 - 2^{-k}) \cdot n^2$ for $\log\rho \leq \delta\log n/k(\log\log n + k)$ for a sufficiently small $\delta > 0$. Our results imply the same but for $\log\rho \leq \delta\log n/\sqrt{k}(\log\log n + k)$. See Section 7 for more details.

[7] CAPP stands for Circuit Acceptance Probability Problem, which is complete for promise-BPP and is the canonical derandomization problem. See Definition 8.2 for its definition.

[8] See Definition 8.1 for a formal definition.

[9] See Section 3.1 for more details on notation used in this paper.

1.3.1 The XOR lemma in [CLW20] and Its Disadvantage

Formally, following Levin's proof of Yao's XOR Lemma [Lev87, GNW11], [CLW20] proved the following lemma.

Lemma 1.7 ([Lev87] and [CLW20, Lemma 3.8])**.** *Let $\mathcal{F}$ be a collection of functions closed under negation and restriction. For $n \in \mathbb{N}_{\ge 1}$, $\delta, \varepsilon \in (0,1)$ and every function $f\colon \{0,1\}^n \to \{-1,1\}$, if*

$$\langle f, C\rangle < (1-\delta)$$

for every $\mathsf{Sum} \circ \mathcal{F}_n$-function C such that $\mathsf{complexity}(C) \le 10 \cdot n/\varepsilon^2$ and $\|C\|_\infty \le 1$, then $\langle f^{\oplus k}, C\rangle \le (1-\delta)^k + \varepsilon/\delta$ for any $f \in \mathcal{F}$.[10]

That is, given a function f which cannot be weakly approximated (say, 0.99-approximated) by $\mathsf{Sum} \circ \mathcal{F}_n$-functions, one can show that $f^{\oplus k}$ is strongly average-case hard for $\mathcal{F}$. The advantage of Lemma 1.7 above over other versions of XOR lemmas [Yao82, Imp95, GNW11, IW97] is that it adds *minimal* computational overhead from the target class $\mathcal{F}$ to the starting class $\mathsf{Sum} \circ \mathcal{F}$, which enables [CR20, CLW20] to apply Williams' algorithmic method to obtain the required hardness against $\mathsf{Sum} \circ \mathcal{F}$ *using algorithms only for $\mathcal{F}$*.

Still, to obtain a $2^{-\Omega(\sqrt{n})}$ correlation bound using Lemma 1.7 with a constant δ (say, $\delta = 0.01$), one has to set $\varepsilon \approx 2^{-\sqrt{n}}$ and $k \approx \sqrt{n}$. Applying the algorithmic method, [CLW20] indeed managed to prove that there is an E^{NP}-computable function $f\colon \{0,1\}^n \to \{-1,1\}$ which cannot be $(1-\delta)$-approximated by linear sums of $2^{\sqrt{n}}$ many $\mathbb{F}_2$-polynomials of degree at most $\sqrt{n}$. Applying Lemma 1.7, one can obtain a function $\mathsf{Amp}^f := f^{\oplus k}$ such that $\mathrm{corr}(\mathsf{Amp}^f, \sqrt{n}) \le 2^{-\Omega(\sqrt{n})}$. However, this is not enough, since Amp^f in fact takes $m = \Theta(n^{1.5})$ bits of input, the correlation bounds *deteriorate* to $\mathrm{corr}(\mathsf{Amp}^f, m^{1/3}) \le 2^{-\Omega(m^{1/3})}$.

1.3.2 A New Derandomized XOR Lemma

To further improve the correlation bounds in [CLW20], we managed to prove a derandomized XOR lemma based on approximate linear sums. Roughly speaking, we construct a *pseudorandom instances generator* $\mathcal{G}\colon \{0,1\}^m \to \{0,1\}^{nk}$ that takes a seed of length $m = \widetilde{O}(n)$, and produces k instances to the function f. We can show that our generator $\mathcal{G}$ is pseudorandom enough to *fool the proof of XOR lemma*, and establish the following new derandomized XOR lemma based on approximate linear sums.

Lemma 1.8 ((Informal))**.** *Let $n \in \mathbb{N}_{\ge 1}$, $\varepsilon \in (0,1)$, $k = \Theta(\log \varepsilon^{-1})$ and $\mathcal{F} = \bigcup_{n \in \mathbb{N}_{\ge 1}} \mathcal{F}_n$ be a function collection satisfying some technical conditions*[11]*. There is a polynomial-time generator $\mathcal{G}\colon \{0,1\}^m \to \{0,1\}^{nk}$ with $m = \widetilde{O}(n)$ such that, for every function $f\colon \{0,1\}^n \to \{-1,1\}$ that cannot be weakly approximated by $\mathsf{Sum} \circ \mathcal{F}$-functions of complexity at most $O(n/\varepsilon^2)$, then $\langle f^{\oplus k} \circ G, C\rangle \le \varepsilon^{\Omega(1)}$ holds for every $C \in \mathcal{F}_m$.*

The proof of our new derandomized XOR lemma is based on a "non-Boolean" generalization of the concept of hardcore sets, which is thoroughly discussed in Section 2.1.1. Such a generalization can also be used to give a completely different and duality-based proof of Lemma 1.7.

Combing Lemma 1.8 with the algorithmic method developed in [Wil13, CW19, CR20, CLW20] for proving hardness against linear sums of collection of functions which admit efficient circuit-analysis algorithms, we can then obtain our improved correlation bounds against $\mathbb{F}_2$-polynomials and construction of extremely rigid matrices.

[10] The function $f^{\oplus k}\colon (\{0,1\}^n)^k \to \{-1,1\}$ is defined as $f^{\oplus k}(x_1, \dots, x_k) = \prod_{i=1}^{k} f(x_i)$.

[11] See Lemma 4.1 for the details.

2 Technical Overview

In this section we give an overview of the proof ideas behind our new results. In Section 2.1 we define and dicuss the key concept, $(\varepsilon, \delta)_{\ell_p}$-witnesses, behind our proofs. In Section 2.2 we give a duality-based new proof of Lemma 1.7, which can be seen as a warm-up for later proofs. In Section 2.3 we discuss the intuitions behind our proof of the new derandomization XOR lemma.

Recall that we use Boolean functions to denote functions from $\{0,1\}^*$ to $\{-1,1\}$, where -1 and 1 are interpreted as True and False respectively. We also introduce the concept of Hölder conjugates below.

For every real $p \geq 1$, recall that the ℓ_p-norm of f is defined as $\|f\|_p := (\mathbb{E}_{x \leftarrow U_n} |f(x)|^p)^{1/p}$, and the ℓ_∞-norm of f is defined as $\|f\|_\infty := \max_{x \in \{0,1\}^n} |f(x)|$. For $p, q \in \mathbb{R}_{\geq 1} \cup \{\infty\}$, we say that p and q are *Hölder conjugates* of each other, if it holds that $1/p + 1/q = 1$.[12]

2.1 The Dual Witness to Inapproximability by Linear Sums

The first ingredient of our proof is a dual witness for inapproximability of a function f by $\mathsf{Sum} \circ \mathcal{F}_n$-functions.

Definition 2.1. *Let $f \colon \{0,1\}^n \to \{-1,1\}$ be a function, let $p \in \mathbb{R}_{\geq 1} \cup \{\infty\}$, and let $\delta > 0, \varepsilon > 0$ be two reals. We say that a function $h \colon \{0,1\}^n \to \mathbb{R}$ is a $(\delta, \varepsilon)_{\ell_p}$-witness for f against $\mathcal{F}_n$-functions, if $\|h\|_p \leq 1$, $\|h\|_1 \leq 1 - \delta$ and $|\langle C, f - h \rangle| \leq \varepsilon$ for every $C \in \mathcal{F}_n$.*

We will often consider the setting where ε is very small and δ is a small constant (*e.g.*, $\varepsilon \leq n^{-\omega(1)}$ and $\delta = 0.01$). That is, a $(\delta, \varepsilon)_{\ell_p}$-witness h for f against $\mathcal{F}_n$-functions can be used to *perturb* f so that the resulting function $f - h$ is *extremely hard* for $\mathcal{F}_n$-functions.

If additionally we can also make $1 - \delta$ very small (instead of being a constant), then we would immediately obtain strong average-case lower bounds for f against $\mathcal{F}_n$. Formally, we have the following remark.

Remark 2.2. *If there is a $(\delta, \varepsilon)_{\ell_p}$-witness for f against $\mathcal{F}_n$-functions, then $|\langle C, f \rangle| \leq \varepsilon + (1 - \delta)$ for every $C \in \mathcal{F}_n$.*

By Remark 2.2, to show that a function f is strongly average-case hard against $\mathcal{F}_n$-functions, it suffices to construct a witness h with very small ε and ℓ_1-norm. This will be the approach adopted in the proofs of this section.

2.1.1 $(\delta, \varepsilon)_{\ell_p}$-witnesses and Hardcore Sets

One may notice that the witness defined in Definition 2.1 appears to be very similar to the concept of *hardcore sets*[13], which are studied extensively in the complexity theory [Imp95, Hol05, RTTV08, TTV09, BHK09]. The following discussions in this subsection are aimed to provide more intuitions on Definition 2.1 and its connections to hardcore sets, and may be skipped without affecting the understanding of the proofs in this paper.

[12] We use the convention that $1/\infty = 0$, so it can be the case that $p = 1$ and $q = \infty$ and vice versa.

[13] Roughly speaking, a hardcore set H for a function f is a subset of $\{0,1\}^n$ with at least $\delta \cdot 2^n$ elements such that f is strongly average-case hard to compute by a certain class of functions with respect to the uniform distribution over H.

From $(\delta,\varepsilon)_{\ell_\infty}$-witnesses to (Boolean) hardcore sets. We remark quickly that when $p = \infty$, a $(\delta,\varepsilon)_{\ell_\infty}$-witness h is *essentially equivalent* to an $\Omega(\delta)$-dense hardcore set of f. Let $P_{\mathsf{core}} := f - h$, since $\|h\|_\infty \le 1$ and f is Boolean, it immediately implies that $P_{\mathsf{core}}(x)$ either has the same sign with $f(x)$ or is zero. We can then construct a function f_{core} by setting each $f_{\mathsf{core}}(x) = f(x)$ independently with probability $|P_{\mathsf{core}}(x)|$, and 0 otherwise.[14] Note that for every x, $\mathbb{E}[f_{\mathsf{core}}(x)] = P_{\mathsf{core}}(x)$.

We can then obtain a hardcore set of f from f_{core} since with high probability: (1) Since $\|P_{\mathsf{core}}\|_1 \ge \|f\|_1 - \|h\|_1 \ge \delta$, at least an $\Omega(\delta)$-fraction of $f_{\mathsf{core}}(x)$ are non-zero (either -1 or 1), those are the inputs in our hardcore set. (2) $\langle f_{\mathsf{core}}, C\rangle$ will be very close to $\langle P_{\mathsf{core}}, C\rangle$ (by a Chernoff bound), which is at most ε. This means f is extremely hard on this hardcore set.

$(\delta,\varepsilon)_{\ell_p}$-witnesses as "non-Boolean" hardcore sets. When $p \ne \infty$, a $(\delta,\varepsilon)_{\ell_p}$-witness h (or more accurately, the function $P^{\ell_p}_{\mathsf{core}} := f - h$ obtained by perturbing f with h) can be thought of as a non-Boolean hardcore set, in the following sense: (1) $P^{\ell_p}_{\mathsf{core}}$ has ℓ_1-norm at least $\|f\|_1 - \|h\|_1 \ge \delta$, so $P^{\ell_p}_{\mathsf{core}}$ is still of "δ-density" and (2) $P^{\ell_p}_{\mathsf{core}}$ is still extremely hard against for $\mathcal{F}$-functions.

We cannot construct from $P^{\ell_p}_{\mathsf{core}}$ a Boolean hardcore $f^{\ell_p}_{\mathsf{core}}$ anymore, since many points in $P^{\ell_p}_{\mathsf{core}}$ can have very large absolute values. Indeed, the norm p controls the *Booleanness* of $f^{\ell_p}_{\mathsf{core}}$: the larger the p is, the closer the $f^{\ell_p}_{\mathsf{core}}$ is to Boolean functions.

We also remark that another way to interpret $P^{\ell_p}_{\mathsf{core}}$ is that it corresponds to a certain hardcore pseudodistribution instead of a hardcore distribution.[15]

2.1.2 From Inapproximability by Linear sums to $(\delta,\varepsilon)_{\ell_p}$-Witnesses

The following "inapproximability-to-witness" lemma shows that we can construct a non-trivial witness for f against $\mathcal{F}_n$-function from the weak inapproximability of f by $\mathsf{Sum} \circ \mathcal{F}_n$-function. This lemma serves as the starting point for our derandomized XOR lemma. Its proof can be found in Section 4.1.

Lemma 2.3. *Let $n \in \mathbb{N}_{\ge 1}$, and let $\mathcal{F}_n$ be a collection of n-input functions that is closed under negation. Let $p, q \in \mathbb{R}_{\ge 1} \cup \{\infty\}$ be such that p and q are Hölder conjugates of each other. For every function $f\colon \{0,1\}^n \to \{-1,1\}$ and $\delta, \varepsilon > 0$, if we have*

$$\langle f, C\rangle < (1-\delta)$$

for every $\mathsf{Sum} \circ \mathcal{F}_n$-function C such that $\mathsf{complexity}(C) \le 10 \cdot n/\varepsilon^2$ and $\|C\|_q \le 1$, then there is a $(\delta,\varepsilon)_{\ell_p}$-witness h for f against $\mathcal{F}_n$-functions.

Moreover, for the case $p = \infty$ and $q = 1$, the condition can be replaced by that for every $\mathsf{MAJ} \circ \mathcal{F}$-function C with top-sparsity bounded by $10n/\varepsilon^2$, it holds that $\langle f, C\rangle < 1 - 2\delta$.

Remark 2.4. *By the discussions in Section 2.1.1, when $(p,q) = (\infty, 1)$, a $(\delta,\varepsilon)_{\ell_p}$-witness immediately implies the existence of an $\Omega(\delta)$-dense hardcore set of f against $\mathcal{F}_n$-functions. Thus, the moreover part of Lemma 2.3 is equivalent to Impagliazzo's Hardcore Lemma.*

2.2 A New Proof of the Original XOR Lemma

As a warm-up, in this section we will first give a new proof of Levin's XOR Lemma [Lev87], reformulated by [CLW20, Lemma 3.8].

[14] If $\|P_{\mathsf{core}}\|_\infty > 1$, we can scale P_{core} by $1/2$, this only reduces its density by a factor of 2.

[15] see [BCG20, CL20] for more discussions on recent works in derandomization using pseudodistributions.

Reminder of Lemma 1.7. *Let $\mathcal{F}$ be a collection of functions closed under negation and restriction. For $n \in \mathbb{N}_{\geq 1}$, $\delta, \varepsilon \in (0,1)$ and every function $f\colon \{0,1\}^n \to \{-1,1\}$, if*

$$\langle f, C\rangle < (1-\delta)$$

for every $\mathsf{Sum} \circ \mathcal{F}_n$*-function C such that* $\mathsf{complexity}(C) \leq 10 \cdot n/\varepsilon^2$ *and* $\|C\|_\infty \leq 1$*, then* $\langle f^{\oplus k}, C\rangle \leq (1-\delta)^k + \varepsilon/\delta$ *for any* $f \in \mathcal{F}$*.*

ε-Indistinguishability. For two functions $f, g\colon \{0,1\}^n \to \mathbb{R}$ and a parameter $\varepsilon > 0$, we say that f and g are *ε-indistinguishable* by $\mathcal{F}_n$-functions if $|\langle f - g, C\rangle| \leq \varepsilon$ for every $C \in \mathcal{F}_n$.

Proof of Lemma 1.7. Applying Lemma 2.3 with $(p,q) = (1,\infty)$, the condition in the lemma implies that there is a $(\delta,\varepsilon)_{\ell_1}$-witness h for f against $\mathcal{F}_n$-functions. That is:

1. f and h are ε-indistinguishable by $\mathcal{F}$-functions.
2. h has ℓ_1-norm at most $(1-\delta)$, which is slightly less than 1.

Proof plan. Our proof will be duality-based. That is, to show $f^{\oplus k}(x)$ is strongly average-case hard, we will show that the function $h^{\oplus k}$ is a sufficient witness to apply Remark 2.2. That is, we want to show the following:

1. **(Indistinguishability.)** $f^{\oplus k}$ is (ε/δ)-indistinguishable from $h^{\oplus k}$ by $\mathcal{F}$-functions.
2. **(Bounded ℓ_1-norm.)** $h^{\oplus k}$ has ℓ_1-norm bounded by $(1-\delta)^k$.

The second item above is easy to establish, since $\|h^{\oplus k}\|_1 = \|h\|_1^k \leq (1-\delta)^k$. Hence it only remains to show the first item.

A hybrid argument. We will show the indistinguishability between $f^{\oplus k}$ and $h^{\oplus k}$ by a hybrid argument. For every $i \in \{0,\ldots,k\}$, we define a hybrid function $H_i^{h,f} := h^{\oplus i} \otimes f^{\oplus k-i}$. That is, for every $r = (r_1,\ldots,r_k) \in (\{0,1\}^n)^k$, we have

$$H_i^{h,f}(r) = \prod_{j=1}^{i} h(r_j) \cdot \prod_{j=i+1}^{k} f(r_j).$$

Note that $H_0^{h,f}$ and $H_k^{h,f}$ are just $f^{\oplus k}$ and $h^{\oplus k}$, respectively. We will show that $H_0^{h,f}$ and $H_k^{h,f}$ are indistinguishable by showing that for every $i \in \{0,\ldots,k-1\}$, the two consecutive functions $H_i^{h,f}$ and $H_{i+1}^{h,f}$ are indistinguishable. Formally, we have the following claim.

Claim 2.5. *For every $i \in [k]$ and every $C \in \mathcal{F}_{nk}$, $|\langle H_{i-1}^{h,f} - H_i^{h,f}, C\rangle| \leq \varepsilon \cdot (1-\delta)^{i-1}$.*

We will prove Claim 2.5 later, but assuming it for now, for every $C \in \mathcal{F}_{nk}$, we have

$$\begin{aligned}
|\langle f^{\oplus k}, C\rangle| &\leq |\langle h^{\oplus k}, C\rangle| + |\langle f^{\oplus k} - h^{\oplus k}, C\rangle| \\
&\leq \|h\|_1^k + \sum_{i=1}^{k} \left|\langle H_{i-1}^{h,f} - H_i^{h,f}, C\rangle\right| && (\|C\|_\infty = 1) \\
&\leq (1-\delta)^k + \varepsilon \cdot \sum_{i=0}^{k-1} (1-\delta)^i && (\text{Claim 2.5 and } \|h\|_1 \leq 1-\delta) \\
&\leq (1-\delta)^k + \varepsilon/\delta. && \square
\end{aligned}$$

Finally, we prove Claim 2.5.

Proof of Claim 2.5. From the definition of $H_{i-1}^{h,f}$ and $H_i^{h,f}$, we have

$$\langle H_{i-1}^{h,f} - H_i^{h,f}, C\rangle = \mathop{\mathbb{E}}_{r\leftarrow U_{nk}}\left[C(r_1,\ldots,r_k)\cdot\prod_{j=1}^{i-1}h(r_j)\cdot\prod_{j=i}^{k}f(r_j)\right] - \mathop{\mathbb{E}}_{r\leftarrow U_{nk}}\left[C(r_1,\ldots,r_k)\cdot\prod_{j=1}^{i}h(r_j)\cdot\prod_{j=i+1}^{k}f(r_j)\right]. \tag{1}$$

We use r_{-i} to denote $(r_1,\ldots,r_{i-1},r_{i+1},\ldots,r_k) \in (\{0,1\}^n)^{k-1}$, so that $r \in \{0,1\}^{nk}$ can be decomposed into r_i and r_{-i}. Organizing the right side of (1), we have

$$\langle H_{i-1}^{h,f} - H_i^{h,f}, C\rangle = \mathop{\mathbb{E}}_{r_{-i}\leftarrow U_{n(k-1)}}\prod_{j=1}^{i-1}h(r_j)\cdot\prod_{j=i+1}^{k}f(r_j)\cdot\mathop{\mathbb{E}}_{r_i\leftarrow U_n}[C(r_1,\ldots,r_k)\cdot(f_i(r_i)-h_i(r_i))]. \tag{2}$$

To further bound (2), for each $r_{-i} \in (\{0,1\}^n)^{k-1}$, we define a function $D^{r_{-i}}\colon \{0,1\}^n \to \{-1,1\}$ as

$$D^{r_{-i}}(x) := C(r_1,\ldots,r_{i-1},x,r_{i+1},\ldots,r_k).$$

It follows that $D^{r_{-i}} \in \mathcal{F}_n$ since $\mathcal{F}$ is closed under restriction. Therefore, since h is a $(\delta,\varepsilon)_{\ell_1}$-witness for f against $\mathcal{F}_n$-functions, we have

$$|\langle f-h, D^{r_{-i}}\rangle| \le \varepsilon. \tag{3}$$

Plugging in (2), we have

$$\begin{aligned}
|\langle H_{i-1}^{h,f} - H_i^{h,f}, C\rangle| &= \left|\mathop{\mathbb{E}}_{r_{-i}\leftarrow U_{n(k-1)}}\prod_{j=1}^{i-1}h(r_j)\cdot\prod_{j=i+1}^{k}f(r_j)\cdot\langle f-h, D^{r_{-i}}\rangle\right| && (4)\\
&\le \varepsilon\cdot\mathop{\mathbb{E}}_{r_{-i}\leftarrow U_{n(k-1)}}\prod_{j=1}^{i-1}|h(r_j)| && (\text{by (3) and } \|f\|_\infty = 1)\\
&\le \varepsilon\cdot\prod_{j=1}^{i-1}\mathop{\mathbb{E}}_{r_j\leftarrow U_n}|h(r_j)| && (5)\\
&\le \varepsilon\cdot(1-\delta)^{i-1}. && (\|h\|_1 \le 1-\delta)
\end{aligned}$$

□

2.3 New Derandomized XOR Lemma

Now we turn to the proof intuitions behind the proof of our new derandomized XOR lemma. We begin by introducing the concept of pseudorandom instance generator and some useful notation.

Pseudorandom instance generators and notation. For convenience, let $\mathcal{F} = \bigcup_{n\in\mathbb{N}_{\ge 1}}\mathcal{F}_n$ be a collection of functions closed under negation and restriction. We will always use $f\colon \{0,1\}^n \to \{-1,1\}$ to denote a weakly average-case hard function on which we will apply the hardness amplification, and we will always use n to denote the input length of f.

The idea is to use a *pseudorandom instance generator* $\mathcal{G}\colon \{0,1\}^m \to (\{0,1\}^n)^k$ (m is much less than nk) to generate inputs to the function $f^{\oplus k}$, similar to the original derandomized XOR lemma in [IW97]. That is, by directly composing the generator $\mathcal{G}$ and $f^{\oplus k}$, one obtain a function $\mathsf{Amp}^f := f^{\oplus k}\circ\mathcal{G}$, which has input length m instead of nk.

High-level idea. Our goal would be to construct the desired pseudorandom instance generator $\mathcal{G}$ such that a similar argument as in the proof of Lemma 1.7 still goes through. That is, we wish to show that $\mathsf{Amp}^h := h^{\oplus k} \circ \mathcal{G}$ is a dual-witness showing that $\mathsf{Amp}^f = f^{\oplus k} \circ \mathcal{G}$ is strongly average-case hard against $\mathcal{F}$-functions (see Remark 2.2). Therefore, we need to establish the following two statements:

1. **(Indistinguishability.)** Amp^f and Amp^h are $(\varepsilon^{\Omega(1)})$-indistinguishable by $\mathcal{F}_m$-functions.
2. **(Bounded ℓ_1-norm.)** Amp^h has ℓ_1-norm at most $(1-\delta)^k$.

In the following, we show how to construct a generator meeting the two requirements above. We will omit some technical details and focus on the key insights in our approach.

2.3.1 Establishing the Indistinguishability

First, our constructed $\mathcal{G}$ needs to ensure that $\mathsf{Amp}^f = f^{\oplus k} \circ \mathcal{G}$ and $\mathsf{Amp}^h = h^{\oplus k} \circ \mathcal{G}$ are indistinguishable. In the following we will try to adapt the proof of Lemma 1.7, and figure out along the way that which properties $\mathcal{G}$ has to satisfy for the adaption to go through.

A new hybrid argument and the difficulty. Again we will try to apply a hybrid argument, recall that we have defined the hybrid functions $H_i^{h,f} = h^{\oplus i} \otimes f^{\oplus k-i}$ in the proof of Lemma 1.7. To simplify notation, we let $\mathcal{G}_i^{h,f} = H_i^{h,f} \circ \mathcal{G}$ to denote our new hybrid functions. Note that $\mathcal{G}_0^{h,f} = \mathsf{Amp}^f$ and $\mathcal{G}_k^{h,f} = \mathsf{Amp}^h$.

Fix $i \in [k]$, our goal is to show that $|\langle \mathcal{G}_{i-1}^{h,f} - \mathcal{G}_i^{h,f}, C\rangle|$ is small for every $C \in \mathcal{F}_m$. Recall in the proof of Lemma 1.7, an analogous bound (Claim 2.5) is proved by considering the following equalities:

$$\begin{aligned}|\langle H_{i-1}^{h,f} - H_i^{h,f}, C\rangle| &= \left| \mathop{\mathbb{E}}_{r \leftarrow U_{nk}} \prod_{j=1}^{i-1} h(r_j) \cdot \prod_{j=i+1}^{k} f(r_j) \cdot [C(r_1,\ldots,r_k)\cdot(f_i(r_i) - h_i(r_i))] \right| \\ &= \left| \mathop{\mathbb{E}}_{r_{-i} \leftarrow U_{n(k-1)}} \prod_{j=1}^{i-1} h(r_j) \cdot \prod_{j=i+1}^{k} f(r_j) \cdot \langle f-h, D^{r_{-i}}\rangle \right|, \end{aligned} \tag{6}$$

where $D^{r_{-i}} : \{0,1\}^n \to \{-1,1\}$ is obtained by restricting the inputs $r_1,\ldots,r_{i-1},\ldots,r_{i+1},\ldots,r_k$ to C by r_{-i}. Since $\mathcal{F}$ is closed under restriction, it follows that $D^{r_i} \in \mathcal{F}$, and we can then apply the bound on $|\langle f-h, D^{r_{-i}}\rangle|$, since f and h are indistinguishable by $\mathcal{F}_n$-functions.

The key intuition in the proof above is that, since the i-th input r_i in $r = (r_1,\ldots,r_k)$ is *completely independent* to the other $k-1$ inputs, one can *fix the other inputs* first and then apply the indistinguishability between f and h to replace f by h on r_i.

Switching to our new setting. For a seed $r \in \{0,1\}^m$, we use $\tilde{r}_i$ to denote $\mathcal{G}(r)_i$ for simplicity. Then we can still write

$$|\langle \mathcal{G}_{i-1}^{h,f} - \mathcal{G}_i^{h,f}, C\rangle| = \mathop{\mathbb{E}}_{r \leftarrow U_m} \prod_{j=1}^{i-1} h(\tilde{r}_j) \cdot \prod_{j=i+1}^{k} f(\tilde{r}_j) \cdot [C(r)\cdot(f_i(\tilde{r}_i) - h_i(\tilde{r}_i))].$$

But since now all the $\tilde{r}_i$ are no longer independent (since they are generated from a seed r with length m much less than nk). We cannot proceed as (6) anymore. That is, if we try to fix $\tilde{r}_{-i}$ first, then it *may even completely fix* the value of $\tilde{r}_i$, and we can no longer obtain a similar function $D^{\tilde{r}_{-i}}$ on $\tilde{r}_i$.

Partial independence and $\mathcal{F}$-restrictable generators. Inspired by the famous Nisan-wigderson generator [NW94], and similar to the proof of the original derandomized XOR lemma in [IW97]. Our idea to resolve the issue above is to design the generator $\mathcal{G}$ in a way that, for each i, some part of the seed r directly corresponds to $\tilde{r}_i$, yet for all other bits, they are *almost independent* to $\tilde{r}_i$.

More formally, we want a mapping $T_i\colon \{0,1\}^n \times \{0,1\}^{m-n} \to \{0,1\}^m$, such that: (1) T_i is a bijection and (2) $\mathcal{G}(T_i(x,\alpha))_i = x$ for all $(x,\alpha) \in \{0,1\}^n \times \{0,1\}^{m-n}$. That is, the first condition says that T_i is just a "reorganization" of the input space $\{0,1\}^m$ while the second condition says that x corresponds directly to $\tilde{r}_i$.

Using the mapping T_i, we can write

$$|\langle \mathcal{G}^{h,f}_{i-1} - \mathcal{G}^{h,f}_{i}, C\rangle| = \left| \mathop{\mathbb{E}}_{\substack{(x,\alpha)\leftarrow U_m \\ r=T_i(x,\alpha)}} \prod_{j=1}^{i-1} h(\tilde{r}_j) \cdot \prod_{j=i+1}^{k} f(\tilde{r}_j) \cdot [C(r)\cdot(f_i(x) - h_i(x))] \right|. \tag{7}$$

For $\alpha \in \{0,1\}^{m-n}$ and $x \in \{0,1\}^n$, we let

$$D^\alpha(x) := \prod_{j=1}^{i-1} h(\tilde{r}_j) \cdot \prod_{j=i+1}^{k} f(\tilde{r}_j) \cdot C(T_i(x,\alpha)),$$

where the $\tilde{r}_j$ above corresponds to $\mathcal{G}(T_i(x,\alpha))_j$ ($\mathcal{G}(r)_j$ if $r = T_i(x,\alpha)$).

Plugging in the above into (7), it follows that

$$|\langle \mathcal{G}^{h,f}_{i-1} - \mathcal{G}^{h,f}_{i}, C\rangle| = \mathop{\mathbb{E}}_{\alpha\leftarrow U_{m-n}} \langle f-h, D^\alpha\rangle. \tag{8}$$

Therefore, if the function D^α above still belongs to $\mathcal{F}_n$, we can then apply the indistinguishability between f and h by $\mathcal{F}_n$-functions, and proceed just as in the proof of Lemma 1.7. This motivates our definition of $\mathcal{F}$-restrictable generator as follows.

Definition 2.6 ($\mathcal{F}$-restrictable generators)**.** *Given a function collection $\mathcal{F}$ and $n \in \mathbb{N}_{\geq 1}$, a generator $\mathcal{G}\colon \{0,1\}^m \to \{0,1\}^{nk}$ is called $\mathcal{F}$-restrictable, if there are k embedding functions $T_1,\dots,T_k\colon \{0,1\}^n \times \{0,1\}^{m-n} \to \{0,1\}^m$ such that the following hold:*

1. *All the T_i are bijections.*
2. *For every $i \in [k]$ and $(x,\alpha) \in \{0,1\}^n \times \{0,1\}^{m-n}$, $\mathcal{G}(T_i(x,\alpha))_i = x$. That is, $T_i(x,\alpha) \in \{0,1\}^m$ is a seed to $\mathcal{G}$ which fixes the i-th instance of $\mathcal{G}(T_i(x,\alpha))$ to be x.*
3. *For every $\mathcal{F}_m$-function $C\colon \{0,1\}^m \to \{-1,1\}$, $i \in [k]$, $\alpha \in \{0,1\}^{m-n}$ and functions $u_1,\dots,u_k\colon \{0,1\}^n \to \{1,-1\}$, the function $D(x) := C(T_i(x,\alpha)) \cdot \prod_{j\in[k]\setminus\{i\}} u_j(\mathcal{G}(T_i(x,\alpha))_j)$ belongs to $\mathcal{F}_n$.*

In the proof of the original derandomized XOR Lemma by [IW97], a restrictable generator for small circuits was constructed by directly adapting the Nisan-Wigderson generator [NW94], which is unfortunately not enough for our applications. So instead, we design two restrictable generators which are tailored to $\mathbb{F}_2$-polynomials and low-rank matrices.

To prove Theorem 1.1, we carefully construct a "star-like" $\mathcal{F}$-restrictable generator for a function collection $\mathcal{F}$ which contains low-degree polynomials as a subset. And similarly, we design a "bi-coloring" restrictable generator for low-rank matrices to prove Theorem 1.5. We will overview the high-level ideas behind these constructions in Section 2.4.

If the function h is Boolean as well, then Item (3) of Definition 2.6 tells us $D^\alpha \in \mathcal{F}_n$ and we can then bound (3). However, the function h may be non-Boolean, and in fact, it may be even unbounded. This causes D^α to also be a non-Boolean function, and we can not directly apply the third condition in Definition 2.6.

Smooth witnesses come to help. We first observe that the aforementioned issue can be resolved if h is smooth in a certain sense. Suppose h is $[-1,1]$-valued (that is, $\|h\|_\infty \le 1$), we can view $D^\alpha(x)$ as a probabilistic $\mathcal{F}$-function and apply a similar argument.

In more details, we sample $i-1$ independent functions $u_1 \dots u_{i-1} \colon \{0,1\}^n \to \{-1,1\}$ from certain distributions, in a way that for every $x \in \{0,1\}^n$, letting $r = T_i(x,\alpha)$, we have

$$\mathop{\mathbb{E}}_{u_j}[u_j(\tilde{r}_j)] = h(\tilde{r}_j) \quad \text{for every } j \in [i-1]. \tag{9}$$

We then set

$$D^{\alpha;u_1,\dots,u_{i-1}}(x) := \prod_{j=1}^{i-1} u_j(\tilde{r}_j) \cdot \prod_{j=i+1}^{k} f(\tilde{r}_j) \cdot C(T_i(x,\alpha)).$$

Recall that we have set $r = T_i(x,\alpha)$, and hence $\tilde{r}_j$ above corresponds to $\mathcal{G}(T_i(x,\alpha))_j$.

By Item (3) of Definition 2.6, $D^{\alpha;u_1,\dots,u_{i-1}}$ is an $\mathcal{F}_n$-function for every possible $(i-1)$-tuples $(u_1,\dots,u_{i-1})$. Hence, we have

$$\begin{aligned} |\langle \mathcal{G}^{h,f}_{i-1} - \mathcal{G}^{h,f}_{i}, C\rangle| &= \mathop{\mathbb{E}}_{\alpha \leftarrow U_{m-n}} \langle f - h, D^\alpha \rangle. \\ &= \mathop{\mathbb{E}}_{\alpha \leftarrow U_{m-n}} \mathop{\mathbb{E}}_{u_1 \dots u_{i-1}} \left\langle f - h, D^{\alpha;u_1\dots u_{i-1}} \right\rangle && \text{(by (9))} \\ &\le \varepsilon. \end{aligned}$$

When $\|h\|_\infty \le M$, a simple scaling argument (replace h by h/M) can be used to show that $|\langle \mathcal{G}^{h,f}_{i-1} - \mathcal{G}^{h,f}_{i}, C\rangle| \le M^{i-1} \cdot \varepsilon$. We refer to Lemma 4.7 for a formal (and more general) proof of the argument above.

Norms and Smoothness of the witnesses. Setting $(p,q) = (\infty,1)$, Lemma 2.3 shows that if we can show that f is weakly inapproximable by $\mathsf{Sum} \circ \mathcal{F}_n$-functions of unit ℓ_1-norm, then we would obtain a $(\delta,\varepsilon)_{\ell_\infty}$-witness h. And one can then proceed to prove our derandomized XOR lemma.

Unfortunately, due to some inherent limitation of the polynomial method, using the algorithmic method, it seems very hard to prove there is a function $f \in \mathsf{E}^{\mathsf{NP}}$ which is weakly inapproximable by $\mathsf{Sum} \circ \mathcal{F}_n$-functions of unit ℓ_1-norm.[16]

By carefully analyzing the approaches in [CW19, CR20, CLW20], we adapt the algorithmic method to show that f is weakly inapproximable by $\mathsf{Sum} \circ \mathcal{F}_n$-functions (think of $\mathcal{F}_n$ as low-degree $\mathbb{F}_2$-polynomials) of unit ℓ_4-norm. By Lemma 2.3, this gives us a $(\delta,\varepsilon)_{\ell_{4/3}}$-witness h.

The bound on the $\ell_{4/3}$-norm of h imposes a smoothness condition on h. Formally, since $\mathbb{E}_{x \leftarrow U_n}[|h(x)|^{4/3}] \le 1$, for every $t \ge 1$, it holds that

$$\mathop{\mathbb{E}}_{x \leftarrow U_n}\left[|h(x)| \cdot \mathbb{1}_{h(x)|\ge t}\right] \le t^{-1/3} \mathop{\mathbb{E}}_{x \leftarrow U_n}\left[|h(x)|^{4/3}\right] \le t^{-1/3}.$$

That is, the total mass of "heavy points" in h is very small. This inspires us to decompose the h as the sum of two functions h_{light} and h_{heavy}, where $h_{\mathsf{light}}(x) := h(x) \cdot \mathbb{1}_{|h(x)|<t}$ and $h_{\mathsf{heavy}}(x) := h(x) \cdot \mathbb{1}_{|h(x)|\ge t}$. Now, since $\|h_{\mathsf{light}}\|_\infty$ is a most t, one can deal with it with the approach discussed above. For h_{heavy}, since $\|h_{\mathsf{heavy}}\|_1$ is small, one may try to simply ignore this part.

[16] Indeed, this is *impossible* if we relax the condition on the total sum of absolute values of coefficients in $C \in \mathsf{Sum} \circ \mathcal{F}_n$: Fixing a function $f \colon \{0,1\}^n \to \{-1,1\}$, one can always construct a $\mathsf{Sum} \circ \mathcal{F}_n$-function C such that $C(0^n) = 2^n \cdot f(0^n)$ while $C(x) = 0$ for every $x \ne 0^n$, by summing only two functions. This C is of sparsity 2, and satisfies $\|C\|_1 = 1$ and $\langle C, f\rangle = 2^{-n} \cdot 2^n = 1$. On the other hands, the algorithmic method can still be used to show weak-inapproximability by $\mathsf{Sum} \circ \mathcal{F}_n$ with unit ℓ_4-norm, even allowing the coefficients to be $2^{O(n)}$.

The real execution of the above plan is, however, much more complicated than the above sounds. For one, after decomposing h into 2 parts, the function

$$D^{\alpha}(x) := \prod_{j=1}^{i-1} h(\tilde{r}_j) \cdot \prod_{j=i+1}^{k} f(\tilde{r}_j) \cdot C(T_i(x, \alpha))$$

actually breaks into 2^{i-1} parts. We have to carefully make sure that this exponential blow-up does not cancel any advantage we gain in the decomposition.

Indeed, a simple two-way decomposition seems not sufficient, and in the real proof (see the proof of Lemma 4.5), we will actually decompose h into many levels, where the k-th level is defined as $h_k(x) := h(x) \cdot \mathbb{1}_{|h(x)| \in (2^{k-1}, 2^k]}$ (together with $h_0(x) := h(x) \cdot \mathbb{1}_{|h(x)| \le 1}$), and apply a novel way to partition the exponential parts of $D^{\alpha}(\tilde{r}_i)$ into only polynomially many groups, and bound each of them separately. Our decomposition is somewhat similar to the analysis of the "bounded independence plus noise" framework for constructing PRGs developed by Haramaty, Lee, and Viola in [HLV18, LV17], which is later used by Forbes and Kelley to construct PRGs for unordered branching programs [FK18].[17]

2.3.2 Bounding the ℓ_1-Norm

Finally, let us turn to the second condition we wish to establish for Amp^h, which requires us to bound its ℓ_1-norm.

Mixing Amp^h by introducing fresh randomness. Since we essentially have no control over Amp^h, if the generator $\mathcal{G}$ always "hits" the parts of h with large magnitude, then Amp^h could have ℓ_1-norm even larger than 1. For example, if for all $r \in \{0,1\}^m$ and $i \in [k]$ it holds $h(\tilde{r}_i) \ge 1$, then clearly $\|\mathsf{Amp}^h\|_1 \ge 1$ as well.

We resolve this issue by introducing some new fresh randomness to "mix" different parts of h. To see the idea, let $\mathcal{G}$ be an arbitrary generator. Suppose that we sample k uniformly random strings from $\{0,1\}^n$, denoted by $w = (w_1, \dots, w_k) \in (\{0,1\}^n)^k$. We then consider the following generator

$$\mathcal{G}^w(r) := (\mathcal{G}(r)_1 \oplus w_1, \dots, \mathcal{G}(r)_k \oplus w_k).$$

We can similarly define

$$\mathsf{Amp}^{f;w} := f^{\oplus k} \circ \mathcal{G}^w \quad \text{and} \quad \mathsf{Amp}^{h;w} := h^{\oplus k} \circ \mathcal{G}^w.$$

For any fixed $r \in \{0,1\}^m$, we have

$$\mathop{\mathbb{E}}_{(w_1,\dots,w_k) \leftarrow U_{nk}} |\mathsf{Amp}^{h;w}(r)| = \mathop{\mathbb{E}}_{(w_1,\dots,w_k) \leftarrow U_{nk}} \prod_{i=1}^{k} h(\mathcal{G}(r)_i \oplus w_i) = \|h\|_1^k \le (1-\delta)^k.$$

The second equality above holds since all the w_i are i.i.d., which means the strings $\{\mathcal{G}(r)_i \oplus w_i\}_{i \in [k]}$ are i.i.d. as well.

Hence, taking an average over all $r \in \{0,1\}^m$, we have

$$\mathop{\mathbb{E}}_{(w_1,\dots,w_k) \leftarrow U_{nk}} \|\mathsf{Amp}^{h;w}\|_1 = \mathop{\mathbb{E}}_{(w_1,\dots,w_k) \leftarrow U_{nk}} \mathop{\mathbb{E}}_{r \leftarrow U_m} |\mathsf{Amp}^{h;w}(r)| \le (1-\delta)^k.$$

[17] In more details, in the analysis of [FK18], they partition all monomials of a polynomial into roughly n groups depending on when the monomials become "heavy" (see [FK18, Proposition 6.1]). For the exponentially many terms resulting from decomposing h, we also partition them into roughly n groups depending on when they become "heavy". The definitions of "heavy" in our work and [FK18] differ since we are in very different settings.

With some complications, we will still be able to show that $\mathsf{Amp}^{h;w}$ and $\mathsf{Amp}^{f;w}$ are indistinguishable.[18] This is not surprising at all: for every fixed $w = (w_1, \dots, w_k)$ the overall effect of w to the generator is simply flipping some input bits to the functions f and h.

Full derandomization by PRGs for space-bounded computation. However, sampling w still requires nk bits, so it may seem we did not gain anything. Our final proof ingredient is to show that we can in fact generate "good enough" w by PRGs for space-bounded computation ([Nis92]), which only require seeds of length $O(n \log k)$. We denote this generator as $\mathcal{G}_{\mathsf{Nisan}} : \{0,1\}^{O(n \log k)} \to \{0,1\}^{nk}$, and take our final generator $\mathcal{G}_{\mathsf{final}}$ as $\mathcal{G}_{\mathsf{final}}(r_1, r_2) := \mathcal{G}(r_1) \oplus \mathcal{G}_{\mathsf{Nisan}}(r_2)$, where $\oplus$ denotes the bit-wise XOR. We remark that PRGs for space-bounded computation is also used in the proof of hardness amplification for NP [HVV06, Lemma 5.7], although the usage there is quite different from our usage.

2.4 Specific Restriction Generators for $\mathbb{F}_2$-Polynomials and Low-Rank Matrices

In this subsection, we give a high-level overview of our restrictable generators for $\mathbb{F}_2$-Polynomials and low-rank matrices. Recall the definition of a $\mathcal{F}$-restrictable generator $\mathcal{G} : \{0,1\}^m \to \{0,1\}^{nk}$: for every $i \in [k]$, advice $\alpha \in \{0,1\}^{m-n}$, functions $u_1, u_2, \dots, u_k \colon \{0,1\}^n \to \{-1,1\}$ and $C \in \mathcal{F}_m$, the function:

$$D(x) := C(T_i(x,\alpha)) \cdot \prod_{j \in [k] \setminus \{i\}} u_j(\mathcal{G}(T_i(x,\alpha))_j). \tag{10}$$

is an $\mathcal{F}_n$-function.

2.4.1 The Star-Like Generator for Correlation Bounds

Roughly speaking, the smaller the function class $\mathcal{F}$, the harder it is to get a restrictable generator for $\mathcal{F}$. Since low-degree $\mathbb{F}_2$-polynomials are not very expressive, it seems extremely hard to obtain a restrictable generator for them directly. On the other hand, exactly due to the fact that low-degree $\mathbb{F}_2$-polynomials are simple enough to be analyzed non-trivially by algorithms, we can utilize the algorithmic method to prove lower bounds for them.

We will consider a larger function collection $\mathcal{F}$ containing low-degree $\mathbb{F}_2$-polynomials as a subset, such that the following hold: (1) $\mathcal{F}$ is still simple enough to be analyzed non-trivially by algorithms and (2) it is expressive enough so that one can design a near-optimal $\mathcal{F}$-restrictable generator. Therefore, we can then apply our derandomized XOR Lemma to prove strong average-case lower bounds against $\mathcal{F}$, which immediately implies correlation bounds against $\mathbb{F}_2$-polynomials.

The larger function collection $\mathcal{F}$. Formally, fixing an integer $n \in \mathbb{N}$ and let $d = \sqrt{n}$. We define $\mathcal{F}$ as the function collection such that $\mathcal{F}$ consists of all functions f which has at least n input bits, and can be written as $f(x) = (-1)^{P(x)} \cdot g(x)$ for $x \in \{0,1\}^m$, where P is an $\mathbb{F}_2$-polynomial with degree bounded by d and $g \colon \{0,1\}^m \to \{-1,1\}$ is a function that only depends on the $(n-d)$-length prefix of its input.

[18] To be more precise, we will show that $\mathbb{E}_{w \in \{0,1\}^{nk}} \operatorname{corr}(\mathsf{Amp}^{f;w} - \mathsf{Amp}^{h;w}, \mathcal{F}_m)$ is small. See the proof of Lemma 4.5 for details.

$\mathcal{F}$ is algorithmic friendly. We observe that the fast #SAT algorithm for low-degree polynomials can be extended to $\mathcal{F}_m$-functions naturally. In fact, given a degree-d polynomial $P\colon \mathbb{F}_2^m \to \mathbb{F}_2$, we set $\ell = n/d$. For each $(x_1,\dots,x_{m-\ell}) \in \{0,1\}^{m-\ell}$, we define

$$s(x_1,\dots,x_{m-\ell}) = \sum_{(y_1,\dots,y_\ell)\in\{0,1\}^\ell} (-1)^{P(x_1,\dots,x_{m-\ell},y_1,\dots,y_\ell)},$$

where the above sum is over $\mathbb{Z}$ instead of over $\mathbb{F}_2$. Then applying the modulus-amplifying polynomials (see Lemma 6.2 for details), there is an algorithm which can compute the list

$$\left(s(x_1,\dots,x_{m-\ell})\right)_{(x_1,\dots,x_{m-\ell})\in\{0,1\}^{m-\ell}}$$

in $O(2^{m-\Omega(\ell)})$ time. Finally, taking a sum over the list, one can then compute $\sum_{x\in\{0,1\}^m}(-1)^{P(x)}$ in $2^{m-\Omega(\ell)}$ time.

Now, observing that we have set $\ell = d = \sqrt{n}$ and noting that $g(x)$ only depends on the first $n-d = n-\ell \le m-\ell$ bits of x (recall that $m \ge n$), we have

$$\sum_{x\in\{0,1\}^m} f(x) = \sum_{(x_1,\dots,x_{m-\ell})\in\{0,1\}^{m-\ell}} s(x_1,\dots,x_{m-\ell}) \cdot g(1,\dots,x_{n-\ell},0,\dots,0),$$

which allows us to compute $\sum_{x\in\{0,1\}^m} f(x)$ in $2^{m-\Omega(\ell)}$ time as well.

The star-like generator for $\mathcal{F}$. Since we aim to prove Theorem 1.1, in the following we fix the number of instances generated by the generator to be $k = \sqrt{n}$. The $\mathcal{F}$-restrictble generator $\mathcal{G}$ is then designed as follows: It has seed length $m = (n-d) + kd \le 2n$ (recall that $d = \sqrt{n}$). For a seed $r \in \{0,1\}^m$, we write $r = \alpha \circ x_1 \circ \cdots \circ x_k$ where $\alpha \in \{0,1\}^{n-d}$ and $x_1,\dots,x_k \in \{0,1\}^d$. Then $\mathcal{G}$ is defined as follows:

$$\mathcal{G}(r) := (\alpha \circ x_1, \dots, \alpha \circ x_k).$$

Now we can justify why we call it the star-like generator: the k instances generated by $\mathcal{G}$ form a star with their common intersection α as the center. For a string $\alpha \in \{0,1\}^{m-n}$, we write $\alpha = \alpha_1 \circ \cdots \circ \alpha_{k-1}$ where $\alpha_i \in \{0,1\}^d$ for every $i \in [k-1]$. For each $i \in [k]$, we define the embedding function T_i as

$$T_i(x,\alpha) = (x_{\le n-d}, \alpha_1, \dots, \alpha_{i-1}, x_{>n-d}, \alpha_i, \dots, \alpha_{k-1}).$$

That is, T_i uses α to fill in the length-d suffix for all instances except for the i-th one, and use x to fill in the i-th instance. It is straightforward to verify that $\mathcal{G}$ satisfies the first two requirements of Definition 2.6.

To show that $\mathcal{G}$ satisfies the third requirement of Definition 2.6, we have to argue that for every $C \in \mathcal{F}_m$, (10) is still in $\mathcal{F}_n$. Observe that for every $j \in [k] \setminus \{i\}$, $u_j(\mathcal{G}(T_i(x,\alpha))_j)$ only depends on $x_{\le n-d}$, hence it belongs to $\mathcal{F}_n$. It is also easy to verify that $C(T_i(x,\alpha)) \in \mathcal{F}_n$. Since $\mathcal{F}_n$ is closed under multiplication, we can conclude that (10) belongs to $\mathcal{F}_n$ as well, which completes the proof. (See the proof of Lemma 6.3 for more details.)

2.4.2 The Bi-coloring Generator for Constructing Rigid Matrices

Now we turn to the generator for low-rank matrices, which will be used to construct the extremely rigid matrices in Theorem 1.5. First we define the class of "low-rank matrices". For every even $n \ge 1$, we can view a function $f\colon \{0,1\}^n$ as a $2^{n/2} \times 2^{n/2}$ matrix, denoted by M_f, where $M_f(x,y) =$

$f(x,y)$ for every $x,y \in \{0,1\}^{n/2}$. We call the first and last $n/2$-bits of inputs as the *row* index and the *column* index, respectively. Letting $r(n) \geq n^{\omega(1)}$ be the rank parameter, we let $\mathcal{M}_n$ denote the class of functions f whose matrix representation M_f satisfies $\mathrm{rank}(M_f) \leq r(n)$.

We will construct an $\mathcal{M}$-restrictable generator with seed length $n\sqrt{k}$, which improves upon the trivial seed length of nk. More precisely, assuming $\sqrt{k}$ is an integer for simplicity, and letting $t = \sqrt{k}$ and $m = nt$, we choose an arbitrary but fixed injective mapping $\rho\colon [k] \to [t] \times [t]$, denoted by $\rho(i) = (\rho(i)_u, \rho(i)_v)$. For every $z \in \{0,1\}^m$, we write $z = x_1 \circ \cdots \circ x_t \circ y_1 \circ \cdots \circ y_t$ where $|x_i| = |y_j| = n/2$ for every $i,j \in [t]$. Our generator $\mathcal{G}$ is then defined as

$$\mathcal{G}(z) := (x_{\rho(1)_u} \circ y_{\rho(1)_v}, \ldots, x_{\rho(k)_u} \circ y_{\rho(k)_v}).$$

It is then straightforward to construct the required mappings T_i: given $x \in \{0,1\}^n$ and $\alpha \in \{0,1\}^{n(t-1)}$, we simply set $x_{\rho(i)_u}$ to $x_{\leq n/2}$ and $y_{\rho(i)_v}$ to $x_{>n/2}$, and use α to fill the rest of z. It is easy to verify that $\mathcal{G}$ satisfies the first two requirements of Definition 2.6.

Intuitively, one can interpret the above construction by thinking about a bipartite graph, where the left and right side have t vertices each, and the seed z as a labeling of all vertices by strings in $\{0,1\}^{n/2}$(*i.e.*, the strings $x_1, \ldots, x_t$ and $y_1, \ldots, y_t$). For every $(i,j) \in [t] \times [t]$, we add an edge between the i-th vertex on the left side and the j-th vertex on the right side, and label this edge with the concatenation of the two $n/2$-bit strings on its endpoints (*i.e.* $x_i \circ y_j$).

Each edge can then be viewed as an instance generated by the seed z (so there are $t^2 \geq k$ instances in total. Now we argue why $\mathcal{G}$ satisfies the third requirement of Definition 2.6. The crucial property is that every two distinct edges can only share at most one common points. Hence, fixing $i \in [k]$ and $\alpha \in \{0,1\}^{n(t-1)}$, then for $j \neq i$, either $\rho(j)_u \neq \rho(i)_u$ or $\rho(j)_v \neq \rho(i)_v$.

We can now observe that, for fixed $\alpha \in \{0,1\}^{n(t-1)}$ and every function $u_j\colon \{0,1\}^n \to \{-1,1\}$, $u_j(\mathcal{G}(T_i(x,\alpha))_j)$ either only depends on $x_{\leq n/2}$, or only depends on $x_{>n/2}$, which means it is a matrix of rank at most 1. Taking an XOR (multiplication over the $\{-1,1\}$ basis is equivalent to XOR over the Boolean basis) of $k-1$ such matrices resulting in a matrix of rank at most $k-1$. Moreover, one can also observe that for a low-rank matrix C, $C(T_i(x,\alpha))$ has the same rank as of C. Therefore, (10) has low rank too, which completes the argument. (See the proof of Lemma 7.5.)

3 Preliminaries

3.1 Notation

We use $\mathbb{N}$ to denote the set of all non-negative integers and $\mathbb{N}_{\geq 1}$ to denote $\mathbb{N} \setminus \{0\}$. For every $n \in \mathbb{N}_{\geq 1}$, we let U_n denote the uniform distribution over $\{0,1\}^n$. By Boolean functions we mean functions that take binary strings $\{0,1\}^*$ and output $\{-1,1\}$, where -1 and 1 are interpreted as True and False respectively.

For a predicate P, we use $\mathbb{1}_P$ to denote its corresponding Boolean value. That is, $\mathbb{1}_P = 1$ if P is true, and 0 otherwise. For a real v, we define $\mathrm{sign}(v) := (-1) \cdot \mathbb{1}_{v<0} + 1 \cdot \mathbb{1}_{v \geq 0}$.

For two strings $\alpha, \beta \in \{0,1\}^*$, we write $\alpha \circ \beta$ to denote the concatenation of α and β. For a string $\alpha \in \{0,1\}^m$ of length m and $k \in [1,m]$, let $\alpha_{\leq k}$ and $\alpha_{>k}$ denote the length-k prefix and length-$(n-k)$ suffix of α respectively.

For two functions $f : A \to B$, $g\colon B \to C$, we write $g \circ f$ to denote the composition of g with f. (Therefore, $g \circ f$ is a function mapping from the set A to the set C.) Note that we are using the notion $\circ$ in two completely different contexts. It would not introduce any ambiguity.

Collections of functions. For $n \in \mathbb{N}_{\geq 1}$, an *$n$-input function collection* $\mathcal{F}_n$ is a subset of all n-input Boolean functions. A *function collection* $\mathcal{F} = \bigcup_{n \in \mathbb{N}_{\geq 1}} \mathcal{F}_n$ is a subset of all Boolean functions, where $\mathcal{F}_n$ is an n-input function collection. We say f is an $\mathcal{F}_n$-function (resp. $\mathcal{F}$-function) if $f \in \mathcal{F}_n$ (resp. $f \in \mathcal{F}$).

By *probabilistic $\mathcal{F}$-function* we mean a *distribution* $\mathcal{D}$ over $\mathcal{F}$-functions of same input length. For every n-bit input probabilistic $\mathcal{F}$-function $\mathcal{D}$, we define the *expectation function* of $\mathcal{D}$ as $P_{\mathcal{D}} \colon \{0,1\}^n \to \mathbb{R}$, where $P_{\mathcal{D}}(x) = \mathbb{E}_{D \leftarrow \mathcal{D}}[D(x)]$.

We say a function collection $\mathcal{F}$ is a typical function collection, if it is closed under negation, and flipping a subset of input bits. (That is, for $f \in \mathcal{F}_n$ and every $w \in \{0,1\}^n$, the function $g(x) := f(x \oplus w)$ and $-f$ both belong to $\mathcal{F}$ as well.)

Correlation and approximation. For a Boolean function $f \colon \{0,1\}^m \to \{-1,1\}$ and a function collection $\mathcal{F}$, we define the (maximum) correlation between f and $\mathcal{F}$-functions as

$$\mathrm{corr}(f, \mathcal{F}) := \max_{C \in \mathcal{F}_m} \langle f, C \rangle.$$

Slightly abusing the notation, we also use $\mathrm{corr}(f, d)$ to denote $\mathrm{corr}(f, \mathcal{P}_d)$, where $\mathcal{P}_d$ is the collection of all degree-2 $\mathcal{F}_2$-polynomials. Note that this is consistent with the definition of $\mathrm{corr}(f, d)$ in Section 1.1.

For two functions $f \colon \{0,1\}^n \to \{-1,1\}$ and a function collection $\mathcal{F}_n$, we say that f cannot be γ-approximated by $\mathcal{F}_n$ if $\Pr_{x \leftarrow U_n}[f(x) = g(x)] < \gamma$ for every $g \in \mathcal{F}_n$. By a standard connection, $\mathrm{corr}(f, \mathcal{F}_n) < \varepsilon$ if and only if f cannot be $(1/2 + \varepsilon/2)$-approximated by $\mathcal{F}_n$.

Arithmetization. We will crucially exploit multi-linear extension of Boolean functions of the following form. For every function $f \colon \{-1,1\}^n \to \{0,1\}$[19], we use $\tilde{f} \colon \mathbb{R} \to \mathbb{R}$ to denote the multi-linear extension of f. That is, we define

$$\tilde{f}(x) := \sum_{y \in \{-1,1\}^n} f(y) \cdot \prod_{j=1}^{n} \left(y_i \cdot \frac{x_i + y_i}{2} \right).$$

One can verify that $\tilde{f}(x)$ is multi-linear and is indeed an extension of f. Moreover, one can observe that the absolute value of the coefficient of each monomial in $\tilde{f}$ is at most 2^n.

3.2 Norms and Inner Products

For two functions $f, g \colon \{0,1\}^n \to \mathbb{R}$, we define their inner product as

$$\langle f, g \rangle := \mathop{\mathbb{E}}_{x \leftarrow U_n} [f(x) \cdot g(x)].$$

For every real $p \geq 1$, recall that the ℓ_p-norm of f is defined as

$$\|f\|_p := \left(\mathop{\mathbb{E}}_{x \leftarrow U_n} |f(x)|^p \right)^{1/p}.$$

We also define the ℓ_∞-norm of f as

$$\|f\|_\infty := \max_{x \in \{0,1\}^n} |f(x)|.$$

[19] Note that the input of f is from $\{-1,1\}^n$, while the output is in $\{0,1\}$. We will use this form of arithmetization in both Section 5 and Section 7. Check Section 5.1 and Section 7.1 for corresponding discussions.

We need the concept of duality between ℓ_p-norms for different choices of p. We first recall the definition of Hölder conjugates.

Definition 3.1. *Let $p, q \in \mathbb{R}_{\geq 1} \cup \{\infty\}$. We say that p and q are Hölder conjugates of each other, if it holds that $1/p + 1/q = 1$. (In particular, it can be the case that $p = 1$ and $q = \infty$ and vice versa.)*

The following inequality is useful for us.

Lemma 3.2 (Hölder's inequality). *Let $p, q \in \mathbb{R}_{\geq 1} \cup \{\infty\}$ be such that p and q are Hölder conjugates of each other. Let $f, g \colon \{0,1\}^n \to \mathbb{R}$. Then it holds that*

$$\langle f, g\rangle \leq \|f\|_p \|g\|_q.$$

In particular, when $p = q = 2$, we get the Cauchy-Schwarz inequality $\langle f, g\rangle \leq \|f\|_2 \|g\|_2$.

Based on Lemma 3.2, the following lemmas can be established.

Lemma 3.3. *For every function f and $p, q \in \mathbb{R} \cup \{\infty\}$ such that $1 \leq p \leq q$, it holds that $\|f\|_p \leq \|f\|_q$.*

Lemma 3.4 (Duality between ℓ_p spaces). *Let $n \in \mathbb{N}_{\geq 1}$. Let $p, q \in \mathbb{R}_{\geq 1} \cup \{\infty\}$ be such that p and q are Hölder conjugates of each other. For every function $f \colon \{0,1\}^n \to \mathbb{R}$, it holds that*

$$\max_{h : \|h\|_q = 1} \{\langle f, h\rangle\} = \|f\|_p.$$

3.3 Hardness Amplification

We also need the standard worst-case to strong average-case hardness amplification. We refer to [STV01] for an excellent exposition.

Theorem 3.5 ([STV01]). *There is a constant $c \geq 1$ such that, for any time-constructible function $S(n)$ and every $f \colon \{0,1\}^n \to \{-1,1\}$ that does not have (general) circuits of size $S(n)$. There is a function $g \colon \{0,1\}^{O(n)} \to \{-1,1\}$ that cannot be $(1/2 + S(n)^{-1/c})$-approximated by circuits of size $S(n)^{1/c}$. Furthermore, given the 2^n-length truth table of f, the truth table of g can be constructed in $2^{O(n)}$ time.*

4 Derandomized XOR Lemma

In this section we prove our derandomized XOR lemma, which is stated formally as below.

Throughout this section, for every $p \in \mathbb{R}_{>1}$, we set $c_p^{\mathsf{k}} \in (0,1)$ to be a small universal constant such that

$$H_{\mathrm{b}}(c_p^{\mathsf{k}}/(1 + c_p^{\mathsf{k}}))(1 + c_p^{\mathsf{k}}) < \frac{1}{p}(1 - c_p^{\mathsf{k}}),$$

where $H_{\mathrm{b}}(q) := -q \log_2 q - (1-q)\log_2(1-q)$ is the binary entropy function. Recall that we say a function collection $\mathcal{F}$ is typical, if it is closed under negation and flipping a subset of input bits.

Lemma 4.1 (Derandomized XOR lemma). *Let $\delta \in (0, 0.1)$ and $p \in \mathbb{R}_{>1}$ be three constants. For every sufficiently large $n \in \mathbb{N}$, every $\varepsilon \in [2^{-n}, 1)$ and every function $f \colon \{0,1\}^n \to \{-1,1\}$.*[20] *Let $\mathcal{F}$ be a typical function collection, and $k = \left\lceil c_p^{\mathsf{k}} \cdot \log \varepsilon^{-1}/5 \right\rceil$. Suppose the following two conditions hold:*

[20] By Lemma 1.7, the original XOR Lemma takes $O(n \log 1/\varepsilon)$ bits of inputs. Therefore, we mainly focus on the case that ε is sufficiently small.

1. *(**Weak inapproximability by** $\mathsf{Sum} \circ \mathcal{F}_n$.) $\langle f, C\rangle < (1-\delta)$ for every $\mathsf{Sum} \circ \mathcal{F}_n$-function C such that $\mathsf{complexity}(C) \le 10 \cdot n/\varepsilon^2$ and $\|C\|_p \le 1$.*

2. *(**Existence of an $\mathcal{F}$-restrictable generator**.) There is an $\mathcal{F}$-restrictable generator $\mathcal{G}_{\mathsf{res}} \colon \{0,1\}^m \to \{0,1\}^{nk}$ with seed length $m \ge n$, which is computable in $\mathrm{poly}(m)$ time.*

Then there is a polynomial-time computable generator $\mathcal{G} \colon \{0,1\}^{m+\ell} \to \{0,1\}^{nk}$ such that

$$\mathrm{corr}(f^{\oplus k} \circ \mathcal{G}, \mathcal{F}) \le \varepsilon^{\Omega(\delta)},$$

where $\ell \le O(m \log m)$.

Moreover, if $\varepsilon \ge 2^{-n^{1-c}}$ for some constant $c \in (0, 0.1)$, then this bound on ℓ can be further improved to $\ell \le O_c(m)$.

Our proof of Lemma 4.1 will follow the proof outline in Section 2.3: In Section 4.1, we show how to construct $(\delta,\varepsilon)_{\ell_p}$-witnesses from weak inapproximability against $\mathsf{Sum} \circ \mathcal{F}_n$-functions, and formally prove Lemma 2.3. In Section 4.2, we first establish a partially derandomized XOR lemma, which is captured by Lemma 4.2. In Section 4.3, we apply PRGs for space-bounded computation to finish the proof of Lemma 4.1.

The "moreover" part says that if ε is slightly sub-exponential (*i.e.* $\varepsilon \ge 2^{-n^{1-\Omega(1)}}$), then we can obtain an optimal linear-seed generator.

4.1 The Existence of (δ, ε)-Witnesses from Hardness Against Linear Sum of Functions

In this section, we prove Lemma 2.3 (restated below).

Reminder of Lemma 2.3. *Let $n \in \mathbb{N}_{\ge 1}$, and let $\mathcal{F}_n$ be a collection of n-input functions that is closed under negation. Let $p, q \in \mathbb{R}_{\ge 1} \cup \{\infty\}$ be such that p and q are Hölder conjugates of each other. For every function $f \colon \{0,1\}^n \to \{-1,1\}$ and $\delta, \varepsilon > 0$, if we have*

$$\langle f, C\rangle < (1-\delta)$$

for every $\mathsf{Sum} \circ \mathcal{F}_n$-function C such that $\mathsf{complexity}(C) \le 10 \cdot n/\varepsilon^2$ and $\|C\|_q \le 1$, then there is a $(\delta,\varepsilon)_{\ell_p}$-witness h for f against $\mathcal{F}_n$-functions.

Moreover, for the case $p = \infty$ and $q = 1$, the condition can be replaced by that for every $\mathsf{MAJ} \circ \mathcal{F}$-function C with top-sparsity bounded by $10n/\varepsilon^2$, it holds that $\langle f, C\rangle < 1 - 2\delta$.

Proof. For the general case, we argue as follows.

The Challenger-Distinguisher game. We consider the following two-player zero-sum game:

1. The Max-Player (Distinguisher) chooses an $\mathcal{F}_n$-probabilistic function $\mathcal{D}$ on n-bit inputs. (That is, $\mathcal{D}$ is a distribution over $\mathcal{F}_n$-functions.) Recall that we use $P_{\mathcal{D}} \colon \{0,1\}^n \to \mathbb{R}$ to denote the expectation function of $\mathcal{D}$. It is defined as $P_{\mathcal{D}}(x) = \mathbb{E}_{D \leftarrow \mathcal{D}}[D(x)]$ for every $x \in \{0,1\}^n$.

2. The Min-Player (Challenger) chooses a function $h \colon \{0,1\}^n \to \mathbb{R}$ such that $\|h\|_p \le 1$ and $\|h\|_1 \le 1 - \delta$. (*i.e.*, h satisfies the norm conditions for being a $(\delta,\varepsilon)_{\ell_p}$-witness.)

3. The payoff of game is $\langle P_{\mathcal{D}}, f - h\rangle$. (Note that $f \colon \{0,1\}^n \to \{-1,1\}$ is a fixed function.) The Min-Player (resp. Max-Player) wants to minimize (resp. maximize) the payoff.

Note that the strategy spaces of both players are compact convex sets and the payoff has a bilinear form. Therefore, by the minimax theorem, the game has a unique equilibrium payoff V_{game} when both players play optimally. We claim that $V_{\mathsf{game}} \leq \varepsilon$.

Indeed, suppose on the contrary that $V_{\mathsf{game}} > \varepsilon$. This implies that the Max-Player has a strategy $\mathcal{D}$, which is a probabilistic $\mathcal{F}_n$-function, such that for every Min-Player strategy h, it holds that

$$\langle P_{\mathcal{D}}, f - h \rangle > \varepsilon. \tag{11}$$

Next, we sample $t = 10 \cdot n/\varepsilon^2$ independent functions from $\mathcal{D}$, denoted by $D_1, \dots, D_t$. By a Chernoff bound, for every $x \in \{0,1\}^n$, it holds that

$$\Pr_{D_1,\dots,D_t}\left[\left|P_{\mathcal{D}}(x) - \mathop{\mathbb{E}}_{i \leftarrow [t]} D_i(x)\right| \geq \frac{\varepsilon}{2}\right] \leq 2^{-n-1}.$$

Then we can fix a set of functions $\{D_i\}_{i\in[t]}$ such that

$$\left|P_{\mathcal{D}}(x) - \mathop{\mathbb{E}}_{i \leftarrow [t]} D_i(x)\right| \leq \varepsilon/2 \quad \text{holds for every } x \in \{0,1\}^n. \tag{12}$$

We now construct a $\mathsf{Sum} \circ \mathcal{F}_n$-function D' as

$$D'(x) := \sum_{i=1}^{t} \frac{1}{t} \cdot D_i(x).$$

Note that D' has sparsity bounded by $t = 10 \cdot n/\varepsilon^2$. For every strategy h of Min-Player, it follows that

$$\begin{aligned}
\langle D', f - h \rangle &= \langle P_{\mathcal{D}}, f - h \rangle - \langle P_{\mathcal{D}} - D', f - h \rangle \\
&> \varepsilon - \|D' - P_{\mathcal{D}}\|_\infty \|f - h\|_1 && \text{((11) and Lemma 3.2)} \\
&\geq \varepsilon - \frac{\varepsilon}{2}(1 + \|h\|_1) && \text{((12) and } \|f\|_1 \leq \|f\|_\infty = 1) \\
&\geq 0, && (13)
\end{aligned}$$

where the last inequality follows from $\|h\|_1 \leq 1 - \delta \leq 1$. By Lemma 3.4 and the assumption that p and q are Hölder conjugates of each other, the Min-Player can choose a strategy h such that $\|h\|_p \leq 1 - \delta$ and $\langle D', h \rangle = (1-\delta) \cdot \|D'\|_q$. Note that $\|h\|_1 \leq \|h\|_p \leq 1 - \delta$ as well, so h is a valid strategy. Therefore, (13) in particular implies that

$$\begin{aligned}
\langle D', f \rangle &= \langle D', h \rangle + \langle D', f - h \rangle \\
&> (1-\delta)\|D'\|_q.
\end{aligned}$$

Next, we give a lower bound on $\|D'\|_q$. Choosing $h \equiv 0$, (11) implies that $\|P_{\mathcal{D}}\|_1 \geq \varepsilon$. Therefore, $\|D'\|_q \geq \|D'\|_1 \geq \|P_{\mathcal{D}}\|_1 - \frac{\varepsilon}{2} \geq \varepsilon/2$.

Letting $C = D'/\|D'\|_q$, we have $\langle C, f \rangle > 1 - \delta$, $\|C\|_q = 1$ and $\mathsf{complexity}(C) \leq \max(t, 1/\|D\|_q) \leq 10 \cdot n/\varepsilon^2$. This contradicts the assumption of the lemma.

Finally, given that $V_{\mathsf{game}} \leq \varepsilon$, it is straightforward to verify that an optimal strategy h of the Min-Player satisfies the requirement of being a $(\delta, \varepsilon)_{\ell_p}$-witness: First, we have that $\|h\|_1 \leq 1 - \delta$ and $\|h\|_p \leq 1$. Second, for every $C \in \mathcal{F}$ it holds that $\langle C, f - h \rangle \leq \varepsilon$. Since $\mathcal{F}$ is closed under negation, it in turn implies that $|\langle C, f - h \rangle| \leq \varepsilon$.

Impagliazzo's hardcore lemma. In the following we prove the "moreover" part in the statement of Lemma 2.3. We consider the same Challenger-Distinguisher game as before with $p = \infty$. (*i.e.*, the Min-Player chooses a function h with $\|h\|_1 \le 1 - \delta$ and $\|h\|_\infty \le 1$.)

Again by the minimax theorem, this game has a unique payoff V_{game} when both players play optimally. We claim $V_{\mathsf{game}} \le \varepsilon$.

Suppose that $V_{\mathsf{game}} > \varepsilon$, then there is a probabilistic $\mathcal{F}_n$-function $\mathcal{D}$ such that for every strategy h by the Min-Player, it holds that

$$\langle P_{\mathcal{D}}, f - h \rangle > \varepsilon. \tag{14}$$

Call an input $x \in \{0,1\}^n$ bad if $|f(x) - P_{\mathcal{D}}(x)| > (1 - \varepsilon)$. Let B be the set of bad inputs.

We claim that (14) implies $|B| \le \delta 2^n$. Otherwise, suppose that $|B| > \delta 2^n$. We define a function h_B as $h_B(x) = \mathbb{1}_{x \notin B} \cdot f(x)$, and observe that $\|h_B\|_1 \le 1 - \delta$. Then we have

$$\langle P_{\mathcal{D}}, f - h_B \rangle = \mathop{\mathbb{E}}_{x \leftarrow U_n} [\mathbb{1}_{x \in B} \cdot P_{\mathcal{D}}(x) \cdot f(x)] \le \frac{|B|}{2^n} \varepsilon \le \varepsilon,$$

which contradicts to (14). The penultimate inequality above follows from the fact that when $x \in B$ and $|f(x) - P_{\mathcal{D}}(x)| > (1 - \varepsilon)$, we have $f(x) \cdot P_{\mathcal{D}}(x) \le \varepsilon$ since $P_{\mathcal{D}}(x) \in [-1, 1]$.

Now, given $|B| \le \delta 2^n$, we can use a $\mathsf{Sum} \circ \mathcal{F}_n$ function C with $\mathsf{complexity}(C) \le 10n/\varepsilon^2$ to pointwise approximate $P_{\mathcal{D}}$ within an additive error of $\varepsilon/2$. It follows that for every $x \notin B$, we have $f(x) = \mathsf{sign}(C(x))$. Therefore, we can convert C to a $\mathsf{MAJ} \circ \mathcal{F}_n$-function C' such that $C'(x) = f(x)$ for every $x \notin B$. Since $|B| \le \delta 2^n$, we have

$$\langle C', f \rangle \ge 1 - \frac{2|B|}{2^n} \ge 1 - 2\delta.$$

this is a contradiction. So it must be the case that $V_{\mathsf{game}} \le \varepsilon$.

Finally, similar to the general case above, given that $V_{\mathsf{game}} \le \varepsilon$, it is straightforward to verify that an optimal strategy h of the Min-Player satisfies the requirement of being a $(\delta, \varepsilon)_{\ell_p}$-witness, which completes the proof of the moreover part.

The above is essentially identical to Nisan's proof of the hardcore lemma. □

4.2 Partial Derandomization Using $\mathcal{F}$-Restrictable Generators

Following our proof overview, in this subsection, we first recall the concept of $\mathcal{F}$-restrictable generators and then prove a partially derandomized XOR lemma (Lemma 4.2).

Reminder of Definition 2.6. *Given a function collection $\mathcal{F}$ and $n \in \mathbb{N}_{\ge 1}$, a generator $\mathcal{G} \colon \{0,1\}^m \to \{0,1\}^{nk}$ is called $\mathcal{F}$-restrictable, if there are k embedding functions $T_1, \ldots, T_k \colon \{0,1\}^n \times \{0,1\}^{m-n} \to \{0,1\}^m$ such that the following hold:*

1. *All the T_i are bijections.*

2. *For every $i \in [k]$ and $(x, \alpha) \in \{0,1\}^n \times \{0,1\}^{m-n}$, $\mathcal{G}(T_i(x, \alpha))_i = x$. That is, $T_i(x, \alpha) \in \{0,1\}^m$ is a seed to $\mathcal{G}$ which fixes the i-th instance of $\mathcal{G}(T_i(x, \alpha))$ to be x.*

3. *For every $\mathcal{F}_m$-function $C \colon \{0,1\}^m \to \{-1,1\}$, $i \in [k]$, $\alpha \in \{0,1\}^{m-n}$ and functions $u_1, \ldots, u_k \colon \{0,1\}^n \to \{-1,1\}$, the function*

$$D(x) := C(T_i(x, \alpha)) \cdot \prod_{j \in [k] \setminus \{i\}} u_j(\mathcal{G}(T_i(x, \alpha))_j)$$

belongs to $\mathcal{F}_n$.

The following lemma gives a partially derandomized XOR lemma, modulo the need of nk fresh random bits w.

Lemma 4.2. *Let $\delta \in (0, 0.1)$ and $p \in \mathbb{R}_{>1}$ be two constants. For every sufficiently large $n \in \mathbb{N}$, every $\varepsilon \in [2^{-n}, 1)$ and every function $f\colon \{0,1\}^n \to \{-1,1\}$. Let $\mathcal{F}$ be a typical function collection and $k = \left\lceil c_p^{\mathsf{k}} \cdot \log \varepsilon^{-1}/5 \right\rceil$. Suppose the following two conditions hold:*

1. *(**Weak inapproximability by** $\mathsf{Sum} \circ \mathcal{F}_n$.) $\langle f, C\rangle < (1-\delta)$ for every $\mathsf{Sum} \circ \mathcal{F}_n$-function C such that $\mathsf{complexity}(C) \le 10 \cdot n/\varepsilon^2$ and $\|C\|_p \le 1$.*
2. *(**Existence of an $\mathcal{F}$-restrictable generator**.) There is an $\mathcal{F}$-restrictable generator $\mathcal{G}_{\mathsf{res}}\colon \{0,1\}^m \to \{0,1\}^{nk}$ with seed length $m \ge n$, which is computable in $\mathrm{poly}(m)$ time.*

For every sequence $w = (w_1, \ldots, w_k) \in (\{0,1\}^n)^k$, we define a function $g^w : \{0,1\}^m \to \{-1,1\}$ as

$$g^w(r) := \prod_{i=1}^{k} f(\mathcal{G}_{\mathsf{res}}(r)_i \oplus w_i).$$

Then we have

$$\mathop{\mathbb{E}}_{w \leftarrow U_{nk}}[\mathrm{corr}(g^w, \mathcal{F})] \le \varepsilon^{\Omega(\delta)}.$$

The rest of this subsection is devoted to the proof of Lemma 4.2.

4.2.1 Notation and Construction of the Hybrids

We begin by introducing some notation, which will be used throughout Section 4.2 and Section 4.3. We set a parameter τ as $\lceil \frac{1}{5} \log \varepsilon^{-1} \rceil$ so that

$$k = \left\lceil c_p^{\mathsf{k}} \cdot \log \varepsilon^{-1}/5 \right\rceil \le c_p^{\mathsf{k}} \cdot \left\lceil \frac{1}{5} \log \varepsilon^{-1} \right\rceil \le c_p^{\mathsf{k}} \cdot \tau.$$

For a seed $r \in \{0,1\}^m$ to the generator $\mathcal{G}_{\mathsf{res}}$, we use the same notation as in Section 2.3 and write

$$\tilde{r}_j = \mathcal{G}_{\mathsf{res}}(r)_j \quad \text{for every } j \in [k].$$

The witness h. First, we observe that by Lemma 2.3 and the first condition of Lemma 4.2, there is a $(\delta, \varepsilon)_{\ell_{p/(p-1)}}$-witness h' for f against $\mathcal{F}_n$ functions. That is:

1. **(Indistinguishability.)** $\mathrm{corr}(f - h', \mathcal{F}_n) \le \varepsilon$.
2. **(Bounded ℓ_1-norm)** h' has ℓ_1-norm at most $1-\delta$ (*i.e.*, $\mathbb{E}_{x \leftarrow U_n}[|h'(x)|] \le 1 - \delta$).
3. **(Bounded $\ell_{p/(p-1)}$-norm.)** h' has $\ell_{p/(p-1)}$-norm at most 1.

It would be convenient to have an ℓ_∞-norm bound on the witness. To achieve this, we define from h' another function h as

$$h(x) := \mathrm{sign}(h'(x)) \cdot \min(|h'(x)|, \varepsilon^{1-p}).$$

We observe that (see also (15))

$$\left\| h'(x) \cdot \mathbb{1}_{|h'(x)| > \varepsilon^{1-p}} \right\|_1 \le \varepsilon.$$

Therefore, $\|h' - h\|_1 \le \varepsilon$. Also, since $\|h\|_{p/(p-1)} \le \|h'\|_{p/(p-1)}$ and $\|h\|_1 \le \|h'\|_1 \le 1$, it turns out that h is a $(\delta, 2\varepsilon)_{\ell_{p/(p-1)}}$-witness for f against $\mathcal{F}_n$-functions, which has ℓ_∞-norm bounded above by ε^{1-p}. This witness h will play the pivotal role in the derandomization of the XOR lemma.

The decomposition of the witness h. Letting $\sigma = \min(n, \lceil \log \varepsilon^{1-p} \rceil)$, as discussed in Section 2.3, we decompose the witness h into $\sigma + 1$ components as follows. For $\ell \in [\sigma]$, we define the function

$$h_{=\ell}(x) := h(x) \cdot \mathbb{1}_{|h(x)| \in (2^{\ell-1}, 2^{\ell}]}.$$

And we also define

$$h_{=0}(x) := h(x) \cdot \mathbb{1}_{|h(x)| \leq 1}.$$

Since $\|h\|_\infty \leq 2^\sigma$, it follows that $h = \sum_{\ell=0}^{\sigma} h_{=\ell}$. We collect the following two important properties of the functions $h_{=\ell}$.

Claim 4.3. *For all $\ell \in \{0, 1, \ldots, \sigma\}$, the following hold:*

1. $\|h_{=\ell}\|_1 \leq 2^{-(\ell-1)/(p-1)}$,
2. $\|h_{=\ell}\|_\infty \leq 2^{\ell}$.

In other words, in the decomposition, the higher the absolute values in $h_{=\ell}$ are, the smaller the ℓ_1-norm of h_ℓ is. This observation is the key for our new derandomized XOR lemma.

Proof. The second claim just follows from the definition of the $h_{=\ell}$.

For every $t \geq 1$, we have

$$\mathop{\mathbb{E}}_{x \leftarrow U_n} \left[|h(x)| \cdot \mathbb{1}_{|h(x)| \geq t} \right] \leq t^{-1/(p-1)} \mathop{\mathbb{E}}_{x \leftarrow U_n} \left[|h(x)|^{p/(p-1)} \right] \leq t^{-1/(p-1)}. \tag{15}$$

If $\ell = 0$, then clearly $\|h_{=0}\|_1 \leq \|h\|_1 \leq 1 \leq 2^{-(\ell-1)/(p-1)}$. For $\ell \in \mathbb{N}_{\geq 1}$, applying (15), it follows that

$$\|h_{=\ell}(x)\|_1 \leq \mathop{\mathbb{E}}_{x \leftarrow U_n} \left[|h(x)| \cdot \mathbb{1}_{|h(x)| \geq 2^{\ell-1}} \right] \leq 2^{-(\ell-1)/(p-1)}. \qquad \square$$

Hybrid functions and their decompositions. Following the outline in Section 2.3, we will apply a hybrid argument. Recall that we have defined the hybrid functions $H_i^{h,f} = h^{\oplus i} \otimes f^{\oplus k-i}$ in Section 2.2, which was later upgraded to $\mathcal{G}_i^{h,f} = H_i^{h,f} \circ \mathcal{G}_{\mathsf{res}}$ in Section 2.3.

Here we will also need to define the hybrid functions with respect to the sequence $w = (w_1, \ldots, w_k)$. We use $\mathcal{G}_{\mathsf{res}}^{w}$ to denote the generator

$$\mathcal{G}_{\mathsf{res}}^{w}(r) := (\tilde{r}_1 \oplus w_1, \ldots, \tilde{r}_k \oplus w_k) \quad \text{for every } r \in \{0,1\}^m.$$

For each $i \in \{0, \ldots, k\}$ and $w \in (\{0,1\}^n)^k$, we define a hybrid function $\mathcal{G}_i^{h,f;w} := H_i^{h,f} \circ \mathcal{G}_{\mathsf{res}}^{w}$. That is, for every $r \in \{0,1\}^m$, we have

$$\mathcal{G}_i^{h,f;w}(r) = \prod_{j=1}^{i} h(\tilde{r}_j \oplus w_j) \cdot \prod_{j=i+1}^{k} f(\tilde{r}_j \oplus w_j).$$

Now we decompose the h functions in $\mathcal{G}_i^{h,f;w}$ using the decomposition of the function h. For an i-tuple $(\ell_1, \ldots, \ell_i) \in \{0, 1, \ldots, \sigma\}^i$, we define

$$\mathcal{G}_{i;(\ell_1,\ldots,\ell_i)}^{h,f;w}(r) = \prod_{j=1}^{i} h_{=\ell_j}(\tilde{r}_j \oplus w_j) \cdot \prod_{j=i+1}^{k} f(\tilde{r}_j \oplus w_j).$$

That is, for each $j \in [i]$, for the j-th h-function in the product defining $\mathcal{G}_i^{h,f;w}$, we take the ℓ_j-th component $h_{=\ell_j}$ of h.

As discussed in Section 2.3, such a decomposition gives us a very *fine-grained* trade-off between ℓ_∞-norm and the ℓ_∞-norm of the h-functions in $\mathcal{G}_{i;(\ell_1,\ldots,\ell_i)}^{h,f;w}$. This will be very helpful for implementing our hybrid argument later.

Final hybrid functions. Now we are ready to define our hybrid functions. For $i \in [k]$, we let $\mathcal{L}_i$ be a set of i-tuples defined as

$$\mathcal{L}_i := \left\{ (\ell_1, \dots, \ell_i) : (\ell_1, \dots, \ell_i) \in \{0, 1, \dots, \sigma\}^i \text{ and } \sum_{j=1}^{i} \ell_j \le \tau \right\}.$$

We will use ε to denote the empty tuple, and then we define $\mathcal{L}_0 := \{\varepsilon\}$. We now define

$$\mathcal{W}_i^{h,f;w} := \sum_{(\ell_1,\dots,\ell_i) \in \mathcal{L}_i} \mathcal{G}_{i;(\ell_1,\dots,\ell_i)}^{h,f;w}.$$

Note that when $i = 0$, since $\mathcal{G}_i^{h,f;w}$ has no h-component to be decomposed, we will let $\mathcal{G}_{0;\varepsilon}^{h,f;w} := \mathcal{G}_0^{h,f;w}$ and hence we have $\mathcal{W}_0^{h,f;w} = \mathcal{G}_0^{h,f;w} = f^{\oplus k} \circ \mathcal{G}_{\mathsf{res}}^{w}$, which is simply g^w.

For every $i \in [k]$, we also let $\mathcal{R}_i$ be a set of i-tuples defined as

$$\mathcal{R}_i := \left\{ (\ell_1, \dots, \ell_i) : (\ell_1, \dots, \ell_i) \in \{0, 1, \dots, \sigma\}^i \text{ and } \sum_{j=1}^{i} \ell_j > \tau \text{ and } \sum_{j=1}^{i-1} \ell_j \le \tau \right\}.$$

The following upper bounds on the size of sets $\mathcal{L}_i$ and $\mathcal{R}_i$ will be useful for us.

Claim 4.4 (Upper bounds on $|\mathcal{L}_i|$ and $|\mathcal{R}_i|$)**.** *For every $i \in \{0, 1, \dots, k\}$, it holds that $|\mathcal{L}_i| = \binom{\tau+i}{i}$. For every $i \in \{1, \dots, k\}$, it holds that $|\mathcal{R}_i| \le |\mathcal{L}_{i-1}|(\sigma + 1)$.*

4.2.2 Implementing the Hybrid Argument

Note that Lemma 4.2 is then asking us to bound

$$\mathop{\mathbb{E}}_{w} \left[\max_{C \in \mathcal{F}} \langle C, \mathcal{W}_0^{h,f;w} \rangle \right].$$

We will now apply a standard hybrid argument with respect to the following list of hybrids:

$$g^w = \mathcal{W}_0^{h,f;w}, \quad \mathcal{W}_1^{h,f;w}, \quad \dots, \quad \mathcal{W}_k^{h,f;w}.$$

The following lemma implement the hybrid argument.

Lemma 4.5 ($\mathcal{W}_i^{h,f;w}$ and $\mathcal{W}_{i+1}^{h,f;w}$ are indistinguishable by $\mathcal{F}_m$-functions)**.** *For every $i \in \{0, 1, \dots, k-1\}$, it holds that*

$$\mathop{\mathbb{E}}_{w \leftarrow U_{nk}} \left[\max_{C \in \mathcal{F}_m} \left| \langle C, \mathcal{W}_i^{h,f;w} \rangle - \langle C, \mathcal{W}_{i+1}^{h,f;w} \rangle \right| \right] \le \varepsilon^{\Omega(1)}.$$

Lemma 4.6 ($\mathcal{W}_k^{h,f;w}$ has small ℓ_1-norm)**.** *It holds that*

$$\mathop{\mathbb{E}}_{w \leftarrow U_{nk}} \left\| \mathcal{W}_k^{h,f;w} \right\|_1 \le \varepsilon^{\Omega(\delta)}.$$

Assuming the two lemmas above, we can derive Lemma 4.2 as shown below.

Proof of Lemma 4.2. We have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{w\leftarrow U_{nk}}[\mathrm{corr}(g^w,\mathcal{F})] \\
&= \mathop{\mathbb{E}}_{w\leftarrow U_{nk}}\left[\max_{C\in\mathcal{F}}\{\langle C,\mathcal{W}_0^{h,f;w}\rangle\}\right] && \text{(by definition)} \\
&= \mathop{\mathbb{E}}_{w\leftarrow U_{nk}}\left[\max_{C\in\mathcal{F}}\left\{\langle C,\mathcal{W}_k^{h,f;w}\rangle+\sum_{i=0}^{k-1}\left(\langle C,\mathcal{W}_i^{h,f;w}\rangle-\langle C,\mathcal{W}_{i+1}^{h,f;w}\rangle\right)\right\}\right] \\
&\le \mathop{\mathbb{E}}_{w\leftarrow U_{nk}}\left[\max_{C\in\mathcal{F}}\{\langle C,\mathcal{W}_k^{h,f;w}\rangle\}\right]+k\cdot\varepsilon^{\Omega(\delta)} && \text{(by Lemma 4.5)} \\
&\le \mathop{\mathbb{E}}_{w\leftarrow U_{nk}}\left\|\mathcal{W}_k^{h,f;w}\right\|_1+\varepsilon^{\Omega(\delta)} && (\|C\|_\infty=1\text{ and }k\le O(\log\varepsilon^{-1})) \\
&\le \varepsilon^{\Omega(\delta)}. && \text{(by Lemma 4.6)}
\end{aligned} \tag{16}$$

□

4.2.3 Proofs of Lemma 4.5 and Lemma 4.6

We begin with the proof of Lemma 4.6, which is the simple one.

Proof of Lemma 4.6. Recall that $k\ge\Omega(\log\varepsilon^{-1})$, we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{w\leftarrow U_{nk}}\|\mathcal{W}_k^{h,f;w}\|_1\le\mathop{\mathbb{E}}_{w\leftarrow U_{nk}}\|\mathcal{G}_k^{h,f;w}\|_1 &\le \mathop{\mathbb{E}}_{w\leftarrow U_{nk}}\mathop{\mathbb{E}}_{r\leftarrow U_m}\left[\left|\prod_{j=1}^k h(\tilde{r}_j\oplus w_j)\right|\right] \\
&\le \mathop{\mathbb{E}}_{r\leftarrow U_m}\|h\|_1^k \\
&\le (1-\delta)^k \\
&\le \varepsilon^{\Omega(\delta)}.
\end{aligned}$$

□

The following lemma will be useful for the proof of Lemma 4.5. Its proof is based on some simple but lengthy manipulations, we defer its proof to the end of this subsection.

Lemma 4.7. *For every $j\in[k]$, $C\in\mathcal{F}_m$ and functions $q_1,\dots,q_{j-1}\colon\{0,1\}^n\to\mathbb{R}$, it holds that*

$$\begin{aligned}
&\mathop{\mathbb{E}}_{r\leftarrow U_m}\left[C(r)\cdot\prod_{i=1}^{j-1}q_i(\tilde{r}_i)\cdot\prod_{i=j}^{k}f(\tilde{r}_i\oplus w_i)\right]-\mathop{\mathbb{E}}_{r\leftarrow U_m}\left[C(r)\cdot h(\tilde{r}_j\oplus w_j)\cdot\prod_{i=1}^{j-1}q_i(\tilde{r}_i)\cdot\prod_{i=j+1}^{k}f(\tilde{r}_i\oplus w_i)\right] \\
&\le 2\varepsilon\cdot\prod_{i=1}^{j-1}\|q_i\|_\infty.
\end{aligned}$$

Next we prove Lemma 4.5.

Proof of Lemma 4.5. For every $i\in\{0,\dots,k\}$, recall that $\mathcal{L}_i$ is the set of i-tuples $(\ell_1,\dots,\ell_i)\in\{0,1,\dots,\sigma\}^i$ such that $\sum_{j=1}^i\ell_j\le\tau$, and $\mathcal{R}_i$ is the set of i-tuples $(\ell_1,\dots,\ell_i)\in\{0,1,\dots,\sigma\}^i$ such that $\sum_{j=1}^i\ell_j>\tau$ and $\sum_{j=1}^{i-1}\ell_j\le\tau$.

By Claim 4.4, $|\mathcal{L}_i| = \binom{\tau+i}{i}$. For each tuple $(\ell_1, \ldots, \ell_i)$ in $\mathcal{L}_i$, we define

$$\begin{aligned} Q_i(\ell_1, \ldots, \ell_i) &:= \langle C, \mathcal{G}^{h,f;w}_{i;(\ell_1,\ldots,\ell_i)} \rangle \\ &= \mathop{\mathbb{E}}_{r \leftarrow U_m} \left[C(r) \cdot \prod_{j=1}^{i} h_{=\ell_j}(\tilde{r}_j \oplus w_j) \cdot \prod_{j=i+1}^{k} f(\tilde{r}_j \oplus w_j) \right], \\ R_i(\ell_1, \ldots, \ell_i) &:= \mathop{\mathbb{E}}_{r \leftarrow U_m} \left[C(r) \cdot h(\tilde{r}_{i+1} \oplus w_{i+1}) \cdot \prod_{j=1}^{i} h_{=\ell_j}(\tilde{r}_j \oplus w_j) \cdot \prod_{j=i+2}^{k} f(\tilde{r}_j \oplus w_j) \right]. \end{aligned}$$

Applying Lemma 4.7, for each $(\ell_1, \ldots, \ell_i) \in \mathcal{L}_i$, we have

$$\begin{aligned} |Q_i(\ell_1, \ldots, \ell_i) - R_i(\ell_1, \ldots, \ell_i)| &\le \varepsilon \cdot \prod_{j=1}^{i} \|h_{=\ell_j}\|_\infty. \\ &\le \varepsilon \cdot 2^{\sum_{j=1}^{i} \ell_j} && \text{(Item (2) of Claim 4.3)} \\ &\le \varepsilon \cdot 2^{\tau}. && (\sum_{j=1}^{i} \ell_j \le \tau) \qquad (17) \end{aligned}$$

From the definition of $\mathcal{W}^{h,f;w}_i$, it follows that

$$\begin{aligned} \langle C, \mathcal{W}^{h,f;w}_i \rangle &= \left\langle C, \sum_{(\ell_1,\ldots,\ell_i) \in \mathcal{L}_i} \mathcal{G}^{h,f;w}_{i;(\ell_1,\ldots,\ell_i)} \right\rangle \\ &= \sum_{(\ell_1,\ldots,\ell_i) \in \mathcal{L}_i} \langle C, \mathcal{G}^{h,f;w}_{i;(\ell_1,\ldots,\ell_i)} \rangle && (18) \\ &= \sum_{(\ell_1,\ldots,\ell_i) \in \mathcal{L}_i} Q_i(\ell_1, \ldots, \ell_i). && (19) \end{aligned}$$

Similarly, from the definition of $\mathcal{W}^{h,f;w}_{i+1}$, we have

$$\begin{aligned} \langle C, \mathcal{W}^{h,f;w}_{i+1} \rangle &= \left\langle C, \sum_{(\ell_1,\ldots,\ell_{i+1}) \in \mathcal{L}_{i+1}} \mathcal{G}^{h,f;w}_{i+1;(\ell_1,\ldots,\ell_{i+1})} \right\rangle \\ &= \sum_{(\ell_1,\ldots,\ell_i) \in \mathcal{L}_i} R_i(\ell_1, \ldots, \ell_i) - \left\langle C, \sum_{(\ell_1,\ldots,\ell_{i+1}) \in \mathcal{R}_{i+1}} \mathcal{G}^{h,f;w}_{i+1;(\ell_1,\ldots,\ell_{i+1})} \right\rangle && (20) \end{aligned}$$

To proceed, we need the following bound on $\binom{\tau+k}{k}$.

Claim 4.8. *For every $i \in \{0, 1, \ldots, k\}$*

$$\binom{\tau+i}{i} \le \binom{\tau+k}{k} \le 2^{(\tau-k)/p}$$

Now we first prove Claim 4.8. It suffices to bound $\binom{\tau+k}{k}$ by monotonicity of the $\binom{\tau+i}{i}$.

We have

$$\begin{aligned}\binom{\tau+k}{k} &\le 2^{H_{\mathrm{b}}(k/(k+\tau))\cdot(k+\tau)} && (\tbinom{n}{m} \le 2^{H_{\mathrm{b}}(m/n)\cdot n})\\ &\le 2^{H_{\mathrm{b}}(c_p^{\mathsf{k}}/(1+c_p^{\mathsf{k}}))\cdot(1+c_p^{\mathsf{k}})\tau} && (k\le c_p^{\mathsf{k}}\cdot\tau)\\ &\le 2^{1/p\cdot(1-c_p^{\mathsf{k}})\cdot\tau} && (H_{\mathrm{b}}(c_p^{\mathsf{k}}/(1+c_p^{\mathsf{k}}))(1+c_p^{\mathsf{k}}) < \tfrac{1}{p}(1-c_p^{\mathsf{k}}) \text{ from our choice of } c_p^{\mathsf{k}})\\ &\le 2^{(\tau-k)/p}, && (k\le c_p^{\mathsf{k}}\cdot\tau)\end{aligned}$$

which completes the proof of Claim 4.8.

Finally, combining (17), (19) and (20), it follows that

$$\begin{aligned}&\left|\langle C, \mathcal{W}_i^{h,f;w}\rangle - \langle C, \mathcal{W}_{i+1}^{h,f;w}\rangle\right|\\ &\le \sum_{(\ell_1,\ldots,\ell_i)\in\mathcal{L}_i} |R_i(\ell_1,\ldots,\ell_i) - Q_i(\ell_1,\ldots,\ell_i)| + \left|\left\langle C, \sum_{(\ell_1,\ldots,\ell_{i+1})\in\mathcal{R}_{i+1}} \mathcal{G}_{i+1;(\ell_1,\ldots,\ell_{i+1})}^{h,f;w}\right\rangle\right|\\ &\le \varepsilon\cdot 2^{\tau}\cdot\binom{\tau+i}{i} + \left\|\sum_{(\ell_1,\ldots,\ell_{i+1})\in\mathcal{R}_{i+1}} \mathcal{G}_{i+1;(\ell_1,\ldots,\ell_{i+1})}^{h,f;w}\right\|_1 && (\|C\|_\infty = 1)\\ &\le \varepsilon^{\Omega(1)} + \left\|\sum_{(\ell_1,\ldots,\ell_{i+1})\in\mathcal{R}_{i+1}} \mathcal{G}_{i+1;(\ell_1,\ldots,\ell_{i+1})}^{h,f;w}\right\|_1, && (21)\end{aligned}$$

where the last inequality follows from Claim 4.8 and the fact that $1/\varepsilon > 8^{\tau}$ (recall that $\tau = \lceil\frac{1}{5}\log\varepsilon^{-1}\rceil$).

Now it remains to bound

$$\mathop{\mathbb{E}}_{w\leftarrow U_{nk}} \left\|\sum_{(\ell_1,\ldots,\ell_{i+1})\in\mathcal{R}_{i+1}} \mathcal{G}_{i+1;(\ell_1,\ldots,\ell_{i+1})}^{h,f;w}\right\|_1,$$

which is itself bounded by

$$\sum_{(\ell_1,\ldots,\ell_{i+1})\in\mathcal{R}_{i+1}} \mathop{\mathbb{E}}_{w\leftarrow U_{nk}} \left\|\mathcal{G}_{i+1;(\ell_1,\ldots,\ell_{i+1})}^{h,f;w}\right\|_1.$$

We now fix an $(i+1)$-tuple $(\ell_1,\ldots,\ell_{i+1})\in\mathcal{R}_{i+1}$. From the definition of $\mathcal{R}_{i+1}$, it holds that $\sum_{j=1}^{i+1}\ell_j > \tau$.

Therefore,

$$\begin{aligned}\mathop{\mathbb{E}}_{w\leftarrow U_{nk}} \left\|\mathcal{G}_{i+1;(\ell_1,\ldots,\ell_{i+1})}^{h,f;w}\right\|_1 &\le \mathop{\mathbb{E}}_{r\leftarrow U_m}\mathop{\mathbb{E}}_{w\leftarrow U_{nk}} \left|\prod_{j=1}^{i+1} h_{=\ell_j}(\tilde{r}_j\oplus w_j)\cdot\prod_{j=i+2}^{k} f(\tilde{r}_j\oplus w_j)\right| && (\|f\|_\infty = 1)\\ &= \mathop{\mathbb{E}}_{w\leftarrow U_{(i+1)k}}\mathop{\mathbb{E}}_{r\leftarrow U_m} \left|\prod_{j=1}^{i+1} h_{=\ell_j}(\tilde{r}_j\oplus w_j)\right|\\ &= \prod_{j=1}^{i+1}\|h_{=\ell_j}\|_1\\ &\le \prod_{j=1}^{i+1} 2^{-(\ell_j-1)/(p-1)} && (\text{Item (1) of Claim 4.3})\\ &\le 2^{-(\tau-k)/(p-1)}. && (\textstyle\sum_{j=1}^{i+1}\ell_j > \tau \text{ and } i\in\{0,1,\ldots,k-1\})\end{aligned}$$

Finally, putting everything together, we have

$$\begin{aligned}
\sum_{(\ell_1,\ldots,\ell_{i+1})\in\mathcal{R}_{i+1}} \mathop{\mathbb{E}}_{w\leftarrow U_{nk}} \left\|\mathcal{G}^{h,f;w}_{i+1;(\ell_1,\ldots,\ell_{i+1})}\right\|_1 &\le 2^{-(\tau-k)/(p-1)}|\mathcal{R}_{i+1}| \\
&\le \binom{\tau+k}{k}(\sigma+1)2^{-(\tau-k)/(p-1)} && \text{(Claim 4.4)} \\
&\le (\sigma+1)\cdot 2^{(\tau-k)/p}\cdot 2^{-(\tau-k)/(p-1)}. && \text{(Claim 4.8)} \\
&\le 2^{-\Omega(k)}\cdot(\sigma+1) && (p>1) \\
&\le \varepsilon^{\Omega(1)}. && (k=\Omega(\tau)=\Omega(\log\varepsilon^{-1}) \text{ and } \sigma = O(\log \varepsilon^{1-p}))
\end{aligned}$$

This completes the proof of Lemma 4.5.

□

In the following remark, we record two very useful facts from the proof of Lemma 4.5, which will be very helpful for the next section.

Remark 4.9. *For every $i \in \{0,1,\ldots,k-1\}$, the following hold:*

1. *For every $w \in \{0,1\}^{nk}$,*

$$\max_{C\in\mathcal{F}_m}\left|\langle C, \mathcal{W}^{h,f;w}_i\rangle - \langle C, \mathcal{W}^{h,f;w}_{i+1}\rangle\right| \le \varepsilon^{\Omega(1)} + \sum_{(\ell_1,\ldots,\ell_{i+1})\in\mathcal{R}_{i+1}} \left\|\mathcal{G}^{h,f;w}_{i+1;(\ell_1,\ldots,\ell_{i+1})}\right\|_1.$$

2.

$$\sum_{(\ell_1,\ldots,\ell_{i+1})\in\mathcal{R}_{i+1}} \mathop{\mathbb{E}}_{w\leftarrow U_{nk}} \left\|\mathcal{G}^{h,f;w}_{i+1;(\ell_1,\ldots,\ell_{i+1})}\right\|_1 \le \varepsilon^{\Omega(1)}.$$

4.2.4 Proof of Lemma 4.7

Finally, we finish this subsection by proving Lemma 4.7 (restated below).

Reminder of Lemma 4.7. *For every $j \in [k]$, $C \in \mathcal{F}_m$ and functions $q_1,\ldots,q_{j-1}\colon \{0,1\}^n \to \mathbb{R}$, it holds that*

$$\mathop{\mathbb{E}}_{r\leftarrow U_m}\left[C(r)\cdot\prod_{i=1}^{j-1} q_i(\tilde r_i)\cdot\prod_{i=j}^{k} f(\tilde r_i\oplus w_i)\right] - \mathop{\mathbb{E}}_{r\leftarrow U_m}\left[C(r)\cdot h(\tilde r_j\oplus w_j)\cdot\prod_{i=1}^{j-1} q_i(\tilde r_i)\cdot\prod_{i=j+1}^{k} f(\tilde r_i\oplus w_i)\right]$$
$$\le 2\varepsilon\cdot\prod_{i=1}^{j-1}\|q_i\|_\infty.$$

Proof of Lemma 4.7. For every $i \in [j-1]$, we let $M_i = \|q_i\|_\infty = \max_{x\in\{0,1\}^n}\{|q_i(x)|\}$. Recall that for $r \in \{0,1\}^m$, we use $\tilde r_i$ to denote $\mathcal{G}_{\mathsf{res}}(r)_i$ for every $i \in [k]$.

We claim that, for every $j \in [k]$ and $\alpha \in \{0,1\}^{m-n}$, the following holds

$$\mathop{\mathbb{E}}_{\substack{x\leftarrow U_n\\ r=T_j(x,\alpha)}}\left[C(r)\cdot\prod_{i=1}^{j-1} q_i(\tilde r_i)\cdot\prod_{i=j}^{k} f(\tilde r_i\oplus w_i)\right] - \mathop{\mathbb{E}}_{\substack{x\leftarrow U_n\\ r=T_j(x,\alpha)}}\left[C(r)\cdot\prod_{i=1}^{j-1} q_i(\tilde r_i)\cdot h(\tilde r_j\oplus w_j)\cdot\prod_{i=j+1}^{k} f(\tilde r_i\oplus w_i)\right]$$
$$\le 2\varepsilon\cdot\prod_{i=1}^{j-1} M_i \tag{22}$$

We will prove (22) shortly, but assuming it holds now. The lemma follows directly by taking an expectation over all $\alpha \in \{0,1\}^{m-n}$ (Note that here we used the condition that $T_j(\star)$ is a bijection).

In the rest of the proof we prove (22). Now we fix $j \in [k]$ and $\alpha \in \{0,1\}^{m-n}$. We consider a probabilistic algorithm $\mathcal{A}$ specified as follows: $\mathcal{A}$ first samples functions $u_1 \dots u_{j-1}$ by the following rule. For every $i \in [j-1]$ and $y \in \{0,1\}^n$, we independently set $u_i(y)$ as

$$u_i(y) = \begin{cases} 1 & \text{with probability } \dfrac{1+q_i(y)}{2 \cdot M_i}, \\ -1 & \text{with probability } \dfrac{1-q_i(y)}{2 \cdot M_i}, \\ \text{a uniform random bit in } \{-1,1\} & \text{otherwise.} \end{cases}$$

We can verify that

$$\mathop{\mathbb{E}}_{u_i}[u_i(y)] = \frac{q_i(y)}{M_i} \quad \text{for each } i \in [j-1],$$

where the expectation is taken over the sample distribution of u_i. Then, for an input $x \in \{0,1\}^n$, letting $r = T_j(x, \alpha)$, $\mathcal{A}$ outputs

$$C(T_j(x,\alpha)) \cdot \prod_{i=1}^{j-1} u_i(\tilde{r}_i) \cdot \prod_{i=j+1}^{k} f(\tilde{r}_i \oplus w_i). \tag{23}$$

By Definition 2.6, the formula (23) is an $\mathcal{F}_n$-function with input x for every sampled $(u_1, \dots, u_{i-1})$ (recall that α is fixed). Therefore, $\mathcal{A}$ can be implemented by a probabilistic $\mathcal{F}_n$-function $\mathcal{D}$. Let $P_{\mathcal{D}}$ be the expectation function of $\mathcal{D}$. For $x \in \{0,1\}^n$, again letting $r = T_j(x, \alpha)$, we have

$$P_{\mathcal{D}}(x) = \frac{1}{\prod_{i=1}^{j-1} M_i} \cdot C(T_j(x,\alpha)) \cdot \prod_{i=1}^{j-1} q_i(\tilde{r}_i) \cdot \prod_{i=j+1}^{k} f(\tilde{r}_i \oplus w_i).$$

We construct another probabilistic $\mathcal{F}$-function $\mathcal{D}'$ such that $P_{\mathcal{D}'}(x) = P_{\mathcal{D}}(x \oplus w_j)$ (this step uses the assumption that $\mathcal{F}$ is typical, and hence it is closed under flipping a subset of inputs). Since h is a $(\delta, 2\varepsilon)_{\ell_{p/(p-1)}}$-witness, we have

$$\langle f - h, P_{\mathcal{D}'} \rangle \le 2\varepsilon. \tag{24}$$

Now, it follows from (24) that

$$\mathop{\mathbb{E}}_{x \leftarrow U_n}[f(x \oplus w_j) \cdot P_{\mathcal{D}}(x)] - \mathop{\mathbb{E}}_{x \leftarrow U_n}[h(x \oplus w_j) \cdot P_{\mathcal{D}}(x)] \le 2\varepsilon.$$

This is equivalent to (22) after scaling both sides by $\prod_{i=1}^{j-1} M_i$: since $\mathcal{G}_{\mathsf{res}}(T_j(x,\alpha))_j = x$, it follows that $f(\tilde{r}_j \oplus w_j) = f(x \oplus w_j)$ and $h(\tilde{r}_j \oplus w_j) = h(x \oplus w_j)$. This finishes the proof of (22).

□

4.3 Full Derandomization by PRGs for Space-Bounded Computation

Lemma 4.2 tells us that for a randomly chosen $(w_1, \dots, w_k)$, the function $g^w(r)$ is hard with high probability. However, it requires nk bits to describe a list of good w_i. In this section, we will further derandomize Lemma 4.2 using PRGs for space-bounded computation.

Branching programs and PRGs for them. We first define read-once branching programs, which captures space-bounded computation.

Definition 4.10. *A (probabilistic, read-once, and oblivious) branching program of size s with block size n is a finite state machine with s states, over the alphabet $\{0,1\}^n$ (with a fixed start state, and an arbitrary number of accepting states). Each edge is labeled with a symbol in $\{0,1\}^n$. For every state s and a symbol $\alpha \in \{0,1\}^n$, the edges leaving a and labelled with α is assigned a probability distribution. The computation proceeds as follows. The input is read sequentially, one block of n bits at a time. If the machine is in state x and it reads α, then it chooses an edge leaving x and labeled with α according to its probability, and moves along it.*

From now on, for breivity, we will always use branching programs to refer to read-once and oblivious branching programs. Next we define pseudorandom generators for branching programs.

Definition 4.11. *A generator $G\colon \{0,1\}^\ell \to \{0,1\}^{nk}$ is ε-pseudorandom for branching programs of size s and block size n if for every branching program B of size s and block size n, it holds that*

$$|\Pr[B(G(U_\ell)) = 1] - \Pr[B(U_{nk}) = 1]| \le \varepsilon.$$

Nisan's PRG. We need the well-known construction of Nisan's PRGs fooling branching programs [Nis92].

Theorem 4.12 ([Nis92])**.** *For every n and $k \le 2^n$, there exists a generator*

$$\mathcal{G}^{\mathsf{Nisan}}_{n,k}\colon \{0,1\}^\ell \to \{0,1\}^{nk}$$

such that the following hold:

- *$\mathcal{G}^{\mathsf{Nisan}}_{n,k}$ is 2^{-3n}-pseudorandom for branching programs of size 2^{3n} and block size n.*
- *$\mathcal{G}^{\mathsf{Nisan}}_{n,k}$ has seed length $\ell = O(n \log k)$.*
- *$\mathcal{G}^{\mathsf{Nisan}}_{n,k}$ can be computed in $\mathrm{poly}(n,k)$ time.*

Fully derandomized XOR lemma. Now we are prove our fully derandomized XOR lemma except the "moreover" part.

Reminder of "Moreover" of Lemma 4.1. *Let $\delta \in (0, 0.1)$ and $p \in \mathbb{R}_{>1}$ be two constants. For every sufficiently large $n \in \mathbb{N}$, every $\varepsilon \in [2^{-n}, 1)$ and every function $f\colon \{0,1\}^n \to \{-1,1\}$.*[21] *Let $\mathcal{F}$ be a typical function collection, and $k = \left\lceil c_p^{\mathsf{k}} \cdot \log \varepsilon^{-1}/5 \right\rceil$. Suppose the following two conditions hold:*

1. *(**Weak inapproximability by** $\mathsf{Sum} \circ \mathcal{F}_n$.) $\langle f, C\rangle < (1-\delta)$ for every $\mathsf{Sum} \circ \mathcal{F}_n$-function C such that $\mathsf{complexity}(C) \le 10 \cdot n/\varepsilon^2$ and $\|C\|_p \le 1$.*
2. *(**Existence of an $\mathcal{F}$-restrictable generator**.) There is an $\mathcal{F}$-restrictable generator $\mathcal{G}_{\mathsf{res}}\colon \{0,1\}^m \to \{0,1\}^{nk}$ with seed length $m \ge n$, which is computable in $\mathrm{poly}(m)$ time.*

[21] By Lemma 1.7, the original XOR Lemma takes $O(n \log 1/\varepsilon)$ bits of inputs. Therefore, we mainly focus on the case that ε is sufficiently small.

Then there is a polynomial-time computable generator $\mathcal{G}\colon \{0,1\}^{m+\ell} \to \{0,1\}^{nk}$ such that

$$\mathrm{corr}(f^{\oplus k} \circ \mathcal{G}, \mathcal{F}) \leq \varepsilon^{\Omega(\delta)},$$

where $\ell \leq O(m \log m)$.

Proof of Lemma 4.1. We let $\mathcal{G}^{\mathsf{Nisan}}_{m,k}\colon \{0,1\}^{\ell} \to \{0,1\}^{mk}$ be the PRG from Theorem 4.12, which fools every branching program of size at most 2^{3m} within error 2^{-3m}. By Theorem 4.12 we know that $\ell = O(m \log k)$. We construct from $\mathcal{G}^{\mathsf{Nisan}}_{m,k}$ a generator $\mathcal{G}_2\colon \{0,1\}^{\ell} \to \{0,1\}^{nk}$ by only keeping the first n-bits of each block of $\mathcal{G}^{\mathsf{Nisan}}_{m,k}(r)$. We construct the final generator as $G(r_1, r_2) := \mathcal{G}_{\mathsf{res}}(r_1) \oplus \mathcal{G}_2(r_2)$.

Now we show that the above generator $\mathcal{G}$ satisfies the requirement of Lemma 4.1. For this purpose we need to prove result analogous to Lemma 4.2. Recall that g^w is defined as $g^w(r) = \prod_{i=1}^{k} f(\mathcal{G}_{\mathsf{res}}(r)_i \oplus w_i)$, and Lemma 4.2 says that for a randomly chosen $w \in \{0,1\}^{nk}$, it holds that

$$\mathop{\mathbb{E}}_{w \leftarrow U_{nk}}[\mathrm{corr}(g^w, \mathcal{F})] \leq \varepsilon^{\Omega(\delta)}.$$

We will prove a derandomized version of Lemma 4.2.

Lemma 4.13. *It holds that*

$$\mathop{\mathbb{E}}_{w \leftarrow \mathcal{G}_2(U_\ell)}[\mathrm{corr}(g^w, \mathcal{F})] \leq \varepsilon^{\Omega(\delta)}. \tag{25}$$

We will prove Lemma 4.13 shortly. Assuming Lemma 4.13 and noting that $\mathcal{F}$ is closed under restriction, for every $C \in \mathcal{F}$ we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{r \leftarrow U_{m+\ell}}[(f^{\oplus k} \circ \mathcal{G})(r) \cdot C(r)] &= \mathop{\mathbb{E}}_{r_2 \leftarrow U_\ell}\ \mathop{\mathbb{E}}_{r_1 \leftarrow U_m}[g^{\mathcal{G}_2(r_2)}(r_1) \cdot C(r_1, r_2)] \\
&= \mathop{\mathbb{E}}_{w \leftarrow \mathcal{G}_2(U_\ell)} \langle g^w(r_1), C(\cdot, r_2) \rangle \\
&\leq \mathop{\mathbb{E}}_{w \leftarrow \mathcal{G}_2(U_\ell)}[\mathrm{corr}(g^w, \mathcal{F})] \leq \varepsilon^{\Omega(\delta)}.
\end{aligned}$$

This establishes the hardness of $f^{\oplus k} \circ \mathcal{G}$ as desired.

□

Now we prove Lemma 4.13.

Proof of Lemma 4.13. Throughout the proof we will use the same notation as in the proof of Lemma 4.2. In particular, the witness function h will play a key role in the proof. We will also follow the structure of its proof structure.

We first recall the following crucial bounds.

Reminder of Lemma 4.5. *For every $i \in \{0, 1, \ldots, k-1\}$, it holds that*

$$\mathop{\mathbb{E}}_{w \leftarrow U_{nk}}\left[\max_{C \in \mathcal{F}_m} \left| \langle C, \mathcal{W}_i^{h,f;w} \rangle - \langle C, \mathcal{W}_{i+1}^{h,f;w} \rangle \right|\right] \leq \varepsilon^{\Omega(1)}.$$

Reminder of Lemma 4.6. *It holds that*

$$\mathop{\mathbb{E}}_{w \leftarrow U_{nk}} \left\| \mathcal{W}_k^{h,f;w} \right\|_1 \leq \varepsilon^{\Omega(\delta)}.$$

We will derandomize Lemma 4.5 and Lemma 4.6 by proving the following two lemmas.

Lemma 4.14 ($\mathcal{W}_i^{h,f;w}$ and $\mathcal{W}_{i+1}^{h,f;w}$ are indistinguishable by $\mathcal{F}_m$-functions, derandomized)**.** *For every $i \in \{0, 1, \ldots, k-1\}$, it holds that*

$$\mathop{\mathbb{E}}_{w \leftarrow \mathcal{G}_2(U_\ell)} \left[\max_{C \in \mathcal{F}_m} \left| \langle C, \mathcal{W}_i^{h,f;w} \rangle - \langle C, \mathcal{W}_{i+1}^{h,f;w} \rangle \right| \right] \leq \varepsilon^{\Omega(1)}.$$

Lemma 4.15 ($\mathcal{W}_k^{h,f;w}$ has small ℓ_1-norm, derandomized)**.** *It holds that*

$$\mathop{\mathbb{E}}_{w \leftarrow \mathcal{G}_2(U_\ell)} \left\| \mathcal{W}_k^{h,f;w} \right\|_1 \leq \varepsilon^{\Omega(\delta)}.$$

We can then proceed almost identically as in the proof of Lemma 4.2:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{w \leftarrow \mathcal{G}_2(U_\ell)} [\mathrm{corr}(g^w, \mathcal{F})] \\
&= \mathop{\mathbb{E}}_{w \leftarrow \mathcal{G}_2(U_\ell)} \left[\max_{C \in \mathcal{F}} \{ \langle C, \mathcal{W}_0^{h,f;w} \rangle \} \right] && \text{(by definition)} \\
&= \mathop{\mathbb{E}}_{w \leftarrow \mathcal{G}_2(U_\ell)} \left[\max_{C \in \mathcal{F}} \left\{ \langle C, \mathcal{W}_k^{h,f;w} \rangle + \sum_{i=0}^{k-1} \left(\langle C, \mathcal{W}_i^{h,f;w} \rangle - \langle C, \mathcal{W}_{i+1}^{h,f;w} \rangle \right) \right\} \right] \\
&\leq \mathop{\mathbb{E}}_{w \leftarrow \mathcal{G}_2(U_\ell)} \left[\max_{C \in \mathcal{F}} \{ \langle C, \mathcal{W}_k^{h,f;w} \rangle \} \right] + k \cdot \varepsilon^{\Omega(1)} && \text{(by Lemma 4.14)} \\
&\leq \mathop{\mathbb{E}}_{w \leftarrow \mathcal{G}_2(U_\ell)} \left\| \mathcal{W}_k^{h,f;w} \right\|_1 + \varepsilon^{\Omega(1)} && (\|C\|_\infty = 1 \text{ and } k = O(\log 1/\varepsilon)) \\
&\leq \varepsilon^{\Omega(\delta)}. && \text{(by Lemma 4.15)}
\end{aligned}$$

□

To prove Lemma 4.14 and Lemma 4.15, we need the following lemma, showing that $\mathcal{G}_2$ can be used to derandomize certain computation.

Lemma 4.16. *Let $q_1, \ldots, q_k \colon \{0,1\}^n \to [0, 2^n]$ be k functions such that:*

- $\|q_i\|_1 \leq 1$ *for every $i \in [k]$, and*
- $\prod_{i=1}^k \|q_i\|_\infty \leq 2^{2n}$.

For every $w \in (\{0,1\}^n)^k$, let $\mu^w \colon \{0,1\}^m \to \mathbb{R}$ be such that

$$\mu^w(r) = \prod_{i=1}^k q_i(\tilde{r}_i \oplus w_i).$$

Then, it holds that

$$\mathop{\mathbb{E}}_{w \leftarrow \mathcal{G}_2(U_\ell)} \|\mu^w(r)\|_1 \leq 2^{-m} + \mathop{\mathbb{E}}_{w \leftarrow U_{nk}} \|\mu^w(r)\|_1 = 2^{-m} + \prod_{i=1}^k \|q_i\|_1.$$

We will prove Lemma 4.16 later, but assuming it for now, we finish the proofs of Lemma 4.14 and Lemma 4.15.

We begin by the proof of Lemma 4.15.

Proof of Lemma 4.15. We recall the decomposition of $\mathcal{W}_k^{h,f;w}$:

$$\mathcal{W}_k^{h,f;w} := \sum_{(\ell_1,\dots,\ell_k)\in\mathcal{L}_k} \mathcal{G}^{h,f;w}_{k;(\ell_1,\dots,\ell_k)}.$$

For each $(\ell_1,\dots,\ell_k)\in\mathcal{L}_k$, we have

$$\mathcal{G}^{h,f;w}_{k;(\ell_1,\dots,\ell_k)}(r) = \prod_{i=1}^{k} h_{=\ell_i}(\tilde{r}_i \oplus w_i).$$

From the definition of $\mathcal{L}_k$ and Item (2) of Claim 4.3, we have $\prod_{i=1}^{k}\|h_{\ell=i}\|_\infty \le 2^{2n}$. Also, note that $\|h_{=\ell_i}\|_1 \le 1$ for every $i\in[k]$. Setting q_i as $h_{=\ell_i}$ for each $i\in[k]$ and applying Lemma 4.16, we have

$$\mathop{\mathbb{E}}_{w\leftarrow G_2(U_\ell)} \|\mathcal{G}^{h,f;w}_{k;(\ell_1,\dots,\ell_k)}\| \le \mathop{\mathbb{E}}_{w\leftarrow U_{nk}} \|\mathcal{G}^{h,f;w}_{k;(\ell_1,\dots,\ell_k)}\| + 2^{-m}.$$

Recall that $\varepsilon \ge 2^{-n}$ and $|\mathcal{L}_k| \le \binom{\tau+k}{k} \le \varepsilon^{-1/2}$ since $\tau = \lceil \frac{1}{5}\cdot\log\varepsilon^{-1}\rceil$. Taking a summation over all tuples in $|\mathcal{L}_k|$ completes the proof, since

$$2^{-m}\cdot|\mathcal{L}_k| \le \varepsilon^{\Omega(1)}$$

and

$$\sum_{(\ell_1,\dots,\ell_k)\in\mathcal{L}_k i} \mathop{\mathbb{E}}_{w\leftarrow U_{nk}} \|\mathcal{G}^{h,f;w}_{k;(\ell_1,\dots,\ell_k)}\| \le \mathop{\mathbb{E}}_{w\leftarrow U_{nk}} \|\mathcal{G}^{h,f;w}_{k}\| \le \varepsilon^{\Omega(\delta)},$$

where the last inequality follows from Lemma 4.6. □

Next we prove Lemma 4.14.

Proof of Lemma 4.14. From Item (1) of Remark 4.9, it suffices to bound

$$\sum_{(\ell_1,\dots,\ell_{i+1})\in\mathcal{R}_{i+1}} \mathop{\mathbb{E}}_{w\leftarrow\mathcal{G}_2(U_\ell)} \left\|\mathcal{G}^{h,f;w}_{i+1;(\ell_1,\dots,\ell_{i+1})}\right\|_1.$$

Fix an $(i+1)$-tuple $(\ell_1,\dots,\ell_{i+1})\in\mathcal{R}_{i+1}$, for every $r\in\{0,1\}^m$, we have

$$\mathcal{G}^{h,f;w}_{i+1;(\ell_1,\dots,\ell_{i+1})}(r) = \prod_{j=1}^{i+1} h_{=\ell_j}(\tilde{r}_j\oplus w_j)\cdot\prod_{j=i+2}^{k} f(\tilde{r}_j\oplus w_j).$$

Now, for each $j\in[i+1]$, we set $q_j = h_{=\ell_j}$ and for each $j\in\{i+2,\dots,k\}$, we set $q_j = f$. It is straightforward to verify that the functions $q_1,\dots,q_j$ satisfy the requirement of Lemma 4.16. Applying Lemma 4.16, we have

$$\mathop{\mathbb{E}}_{w\leftarrow\mathcal{G}_2(U_\ell)} \|\mathcal{G}^{h,f;w}_{i+1;(\ell_1,\dots,\ell_{i+1})}(r)\|_1 \le 2^{-m} + \mathop{\mathbb{E}}_{w\leftarrow U_{nk}} \|\mathcal{G}^{h,f;w}_{i+1;(\ell_1,\dots,\ell_{i+1})}(r)\|_1.$$

Recall that $|\mathcal{R}_{i+1}| \le \binom{\tau+i}{i}\cdot(\sigma+1)$ by Claim 4.4, summing up for all $(\ell_1,\dots,\ell_{i+1})\in\mathcal{R}_{i+1}$ and applying Item (2) of Remark 4.9 completes the proof. □

4.3.1 Proof of Lemma 4.16

Proof of Lemma 4.16. For every $j \in [k]$, let $M_j = \|q_j\|_\infty$, it follows that $\prod_{j=1}^k M_j \le 2^{2n}$. Let us consider the following (probabilisitc) branching program, denoted by B.

1. First, sample $r \leftarrow U_m$.
2. Read k blocks $w_1, \ldots, w_k$ in sequence. For every $i \in [k]$, after reading w_i, reject immediately with probability $1 - \frac{q_j(w_j \oplus \tilde{r}_j)}{M_j}$.
3. After reading k blocks without rejection, accept.

Note that B can be implemented by a probabilisitc branching program of size at most 2^{3m}. Now, associate with B an expectation function Q_B, where $Q_B(w_1, \ldots, w_k)$ denotes the probability of B outputting "accept" on input $(w_1, \ldots, w_k)$. By definition, we have

$$\mathop{\mathbb{E}}_{w \leftarrow U_{nk}}[Q_B(w)] = \frac{1}{\prod_{j \in [k]} M_j} \mathop{\mathbb{E}}_{w \leftarrow U_{nk}} \|\mu^w\|_1. \tag{26}$$

By Theorem 4.12, it follows that

$$\left| \mathop{\mathbb{E}}_{w \leftarrow \mathcal{G}_2(U_\ell)}[Q_B(w)] - \mathop{\mathbb{E}}_{w \leftarrow U_{nk}}[Q_B(w)] \right| \le 2^{-3m}. \tag{27}$$

Also, observe that

$$\mathop{\mathbb{E}}_{w \leftarrow \mathcal{G}_2(U_\ell)}[Q_B(w)] = \frac{1}{\prod_{j \in [k]} M_j} \mathop{\mathbb{E}}_{w \leftarrow \mathcal{G}_2(U_\ell)} \|\mu^w\|_1. \tag{28}$$

We combine (26), (27) and (28) together and note that $\prod_{j \in [k]} M_j \le 2^{2n} \le 2^{2m}$. This completes the proof.

□

4.3.2 Achieving Linear Seed-Length in Slightly Sub-Exponential Error Regime

In the case that $\varepsilon \ge 2^{-n^{1-c}}$ for a Aconstant $c > 0$, it is possible to use a linear-length PRG for space-bounded computation, so that we can reduce the seed length of the pseudorandom instance generator to be linear in n.

The Nisan-Zuckerman PRG. For this purpose, we will need the following PRG for space-bounded computation by Nisan and Zuckerman.

Theorem 4.17 ([NZ96]). *For every $c > 0$, the following is true. For every $n \in \mathbb{N}$ and $k \le O(n)$, there is a generator*

$$\mathcal{G}^{\mathsf{NZ}}_{n,k} \colon \{0,1\}^\ell \to \{0,1\}^{nk}$$

such that the following hold:

- *$\mathcal{G}^{\mathsf{NZ}}_{n,k}$ is $2^{-3n^{1-c}}$-pseudorandom for branching programs of size 2^{3n} and block size n.*
- *$\mathcal{G}^{\mathsf{NZ}}_{n,k}$ has seed length $\ell = O_c(n)$.*
- *$\mathcal{G}^{\mathsf{NZ}}_{n,k}$ can be computed in $\mathrm{poly}(n,k)$ time.*

Fully derandomized XOR lemma with linear seed length. We now prove the linear-length seed generators. (*i.e.* the "moreover" part in Lemma 4.1.)

Reminder of The "moreover" part of Lemma 4.1. *Let $\delta, c \in (0, 0.1)$ and $p \in \mathbb{R}_{>1}$ be three constants. For every sufficiently large $n \in \mathbb{N}$, every $\varepsilon \in [2^{-n^{1-c}}, 1)$ and every function $f\colon \{0,1\}^n \to \{-1,1\}$. Let $\mathcal{F}$ be a typical function collection and let $k = \left\lceil c_p^{\mathsf{k}} \cdot \log \varepsilon^{-1}/5 \right\rceil$. Suppose the following two conditions hold:*

1. *(**Weak inapproximability by** $\mathsf{Sum} \circ \mathcal{F}_n$.) $\langle f, C\rangle < (1-\delta)$ for every $\mathsf{Sum} \circ \mathcal{F}_n$-function C such that $\mathsf{complexity}(C) \le 10 \cdot n/\varepsilon^2$ and $\|C\|_p \le 1$.*
2. *(**Existence of an $\mathcal{F}$-restrictable generator**.) There is an $\mathcal{F}$-restrictable generator $\mathcal{G}_{\mathsf{res}}\colon \{0,1\}^m \to \{0,1\}^{nk}$ with seed length $m \ge n$, which is computable in $\mathrm{poly}(m)$ time.*

Then there is a polynomial-time computable generator $\mathcal{G}\colon \{0,1\}^{m+\ell} \to \{0,1\}^{nk}$ such that

$$\mathrm{corr}(f^{\oplus k} \circ \mathcal{G}, \mathcal{F}) \le \varepsilon^{\Omega(\delta)},$$

where $\ell \le O_c(m)$.

Proof Sketch. We let $\mathcal{G}_2\colon \{0,1\}^\ell \to \{0,1\}^{nk}$ be the PRG from Theorem 4.17, which can fool every branching program of size at most 2^{3m} within error $2^{-3m^{1-c}}$. It follows that $\ell \le O_c(m)$. We construct the final generator as $\mathcal{G}(r_1, r_2) := \mathcal{G}_{\mathsf{res}}(r_1) \oplus \mathcal{G}_2(r_2)$. To show that $\mathcal{G}$ works, we use a similar argument as in the proof of Lemma 4.1 except one difference: when applying the full derandomization based on the partial derandomization, we use the following lemma to replace Lemma 4.16.

Lemma 4.18 (A variant of Lemma 4.16)**.** *Let $q_1, \dots, q_k\colon \{0,1\}^n \to [0, 2^n]$ be k functions such that*

- *$\|q_i\|_1 \le 1$ for every $i \in [k]$, and*
- *$\prod_{i=1}^k \|q_i\|_\infty \le \varepsilon^{-p}$.*

For every $w \in (\{0,1\}^n)^k$, let $\mu^w\colon \{0,1\}^m \to \mathbb{R}$ be such that

$$\mu^w(r) = \prod_{i=1}^k q_i(\tilde{r}_i \oplus w_i).$$

Then, it holds that

$$\mathop{\mathbb{E}}_{w \leftarrow \mathcal{G}_2(U_\ell)} \|\mu^w(r)\|_1 \le 2^{-m^{1-c}} + \mathop{\mathbb{E}}_{w \leftarrow U_{nk}} \|\mu^w(r)\|_1 = 2^{-m^{1-c}} + \prod_{i=1}^k \|q_i\|_1.$$

The proof of Lemma 4.18 is also analogous to Lemma 4.16.

□

5 Weak-inapproximability by Linear Sums from Non-Trivial Circuit-Analysis Algorithms

In this section we show that for a function collection $\mathcal{F}$ which admits a sufficient circuit-analysis algorithm, one can use Williams' algorithmic method [Wil13, Wil18, CW19, CLW20] to construct a hard function f which cannot be weak-approximated by $\mathsf{Sum} \circ \mathcal{F}$-functions. In later sections, this will be combined with our new derandomized XOR lemma to construct strong average-case hard functions against $\mathcal{F}$.

Our connection works for every typical function collections. We first summarize the necessary requirements for the target function collection below.

Definition 5.1 (Applicable function collections)**.** *Let $S(n) \geq n^{\omega(1)}$ be a non-decreasing time-constructible function, and $\mathcal{F} = \bigcup_{n \in \mathbb{N}_{\geq 1}} \mathcal{F}_n$ be a function collection. We say that $\mathcal{F}$ is $S(n)$-applicable, if the following hold:*

1. *There is a* #SAT *algorithm for $\mathsf{AND}_4 \circ \mathcal{F}_n$-functions that runs in $O(2^n/S(n))$ time, where $\mathsf{AND}_4 \circ \mathcal{F}_n$ denotes the subset of functions which can be computed by taking an* AND *of four $\mathcal{F}_n$-functions.*
2. *For every $n \in \mathbb{N}_{\geq 1}$ and $S \subseteq [n]$, the function $\chi_S(x) := \prod_{i \in S} (-1)^{x_i}$ is in $\mathcal{F}_n$. In other words, $\mathcal{F}$ contains all the* parities.
3. *$\mathcal{F}$ is closed under negation.*

Now we state the main theorem of this section, which claims that for every $S(n)$-applicable function collection $\mathcal{F}$, one can construct an E^{NP} function f, that is weakly-inapproximable by $\mathcal{F}$-functions. We also remark that the following theorem is the only result in this section which is used outside, so reader not interested in its proof details may skip this section entirely and jump to the later sections.

Theorem 5.2. *There are absolute constants $\alpha, \delta \in (0,1)$ and $K \geq 1$ such that the following hold. Let $\mathcal{F}$ be an $S(n)$-applicable collection. There is an E^{NP} machine which, for every sufficiently large n, on input 1^n, outputs in $2^{O(n)}$ time a Boolean function $f\colon \{0,1\}^\ell \to \{-1,1\}$ where $\ell \in [n, Kn]$ such that one of the following holds.*

1. *f cannot be computed by $S(\ell)^\alpha$-size general circuits.*
2. *f is hard in the following sense: for every* Sum $\circ\, \mathcal{F}_\ell$*-functions H such that* complexity$(H) \leq S(\ell)^{3\alpha}$ *and $\|H\|_4 \leq 1$, it holds that*
$$\langle f, H \rangle < (1-\delta).$$

The rest of this section is devoted to the proof of Theorem 5.2 and is organized as follows: In Section 5.1 we introduced some previous results which will be crucial to our proof of Theorem 5.2. In Section 5.2 we design a "cheating algorithm" $\mathcal{A}_{\mathsf{cheat}}$ which attempts to break a certain NTIME hierarchy theorem, this part is very similar to the proof of [CLW20, Theorem 1.2], and is also crucial to our proof of Theorem 5.2. In Section 5.3, we analyze $\mathcal{A}_{\mathsf{cheat}}$ and conclude the proof of Theorem 5.2.

5.1 Preliminaries

We will need some technical ingredients from the literature.

Robustly-often NTIME hiearchy theorem and P^{NP} refuter for it. We start with the following robustly-often hard $\mathsf{NTIME}[T(n)]$ language with a corresponding refuter algorithm for it.

Theorem 5.3 (Robustly-often NTIME hierarchy [FS17])**.** *For every polynomial $T(n) = n^K$ and for some constant $k \geq 1$, there exists a language $L \in \mathsf{NTIME}[T(n)]$ such that, for any $L' \in \mathsf{NTIME}[o(T(n))]$, for all but finitely many n, there exists $m \in [n, n+T(n)]$ such that L and L' cannot agree on all inputs of length m.*

Theorem 5.4 (Refuter for the robustly-often NTIME hierarchy [CLW20, Theorem 4.8])**.** *For every polynomial $T(n) = n^K$ for some constant $k \geq 1$, there is an* $\mathsf{NTIME}[T(n)]$ *machine* $\mathcal{A}^T_{\mathsf{FS}}$ *and a* P^{NP} *algorithm* $\mathcal{R}^T$ *such that:*

1. ***Input.*** *The input for $\mathcal{R}^T$ is a pair $(M, 1^n)$ with the promise that M is nondeterministic Turing machine runs in $o(T(n))$ time.*

2. ***Output.*** *For every fixed M and all large enough n, $\mathcal{R}^T(M, 1^n)$ outputs a string x such that $|x| \in [n, n + T(n)]$ and $\mathcal{R}^T(x) \neq M(x)$.*

We remark that the original theorems in [FS17] and [CLW20] apply to a large class of functions $T(n)$, but here we will just state them for the special case that $T(n) = n^K$, since this is all we need in the proofs.

Efficient Construction of PCPs. We recall the following Probabilistically Checkable Proofs (PCP) systems and PCP of Proximity systems, which are also used in [CR20] and [CLW20].

Theorem 5.5 ([BV14])**.** *Let M be an algorithm running in time $T = T(n) \geq n$ on inputs of the form (x, y) where $|x| = n$. Given $x \in \{0,1\}^n$, one can output in $\mathrm{poly}(n, \log T)$ time circuits $Q\colon \{0,1\}^r \to \{0,1\}^{rt}$ for $t = \mathrm{poly}(r)$ and $R\colon \{-1,1\}^t \to \{0,1\}$ such that:*

- ***Proof length.*** $2^r \leq T \cdot \mathrm{polylog} T$.

- ***Completeness.*** *If there is a $y \in \{0,1\}^{T(n)}$ such that $M(x, y)$ accepts then there is a map $\pi\colon \{0,1\}^r \to \{-1,1\}$ such that for all $z \in \{0,1\}^r$, $R(\pi(q_1), \ldots, \pi(q_t)) = 1$ where $(q_1, \ldots, q_t) = Q(z)$.*

- ***Soundness.*** *If no $y \in \{0,1\}^{T(n)}$ causes $M(x, y)$ to accept, then for every map $\pi\colon \{0,1\}^r \to \{-1,1\}$, at most $\frac{2^r}{n^{10}}$ distinct $z \in \{0,1\}^r$ have $R(\pi(q_1), \ldots, \pi(q_t)) = 1$ where $(q_1, \ldots, q_t) = Q(z)$.*

- ***Complexity.*** *Q is a projection, i.e., each output bit of Q is a bit of input, the negation of a bit, or a constant. R is a* 3CNF*.*

Note that this is an extremely efficient PCP, where the 3CNF R and the projection Q collectively form the verifier for the PCP. The following lemma from [CW19, VW20] is a slight modification of the probabilistically checkable proof of proximity (PCPP) system in [BGHSV06].

Theorem 5.6 ([CW19, VW20])**.** *There are constants $0 < s_{\mathsf{pcpp}} < c_{\mathsf{pcpp}} < 1$ and a polynomial-time transformation that, given a circuit D on n inputs of size $m \geq n$, outputs a 2-SAT instance F on the variable set $\mathcal{Y} \cup \mathcal{Z}$ where $|\mathcal{Y}| \leq \mathrm{poly}(n)$, $|\mathcal{Z}| \leq \mathrm{poly}(m)$, and the following hold for all $x \in \{0,1\}^n$:*

- *If $D(x) = 1$, then $F|_{\mathcal{Y} = \mathsf{Enc}(x)}$ on variable set $\mathcal{Z}$ has a satisfying assignment $\mathcal{Z}_x$ such that at least c_{pcpp}-fraction of the clauses are satisfied. Furthermore, there is a $\mathrm{poly}(m)$ time algorithm that given x outputs $\mathcal{Z}_x$.*

- *If $D(x) = 0$, then there is no assignment to the $\mathcal{Z}$ variables in $F|_{\mathcal{Y} = \mathsf{Enc}(x)}$ satisfies more than s_{pcpp}-fraction of the clauses.*

Moreover, the number of clauses in the 2-SAT instance F is a power of 2, and for each $i \in [|\mathcal{Y}|]$, $\mathsf{Enc}_i(x)$ is a parity function depending on at most $n/2$ bits of x.

Arithmetization. In the following, we will frequently apply arithmetization to these PCP verifiers. For input Boolean values to PCP verifier, we always interpret Boolean True and False as real -1 and 1 respectively. This is consistent with the proof fed into it (recall that PCP verifiers get oracle Boolean functions as proof.). For output of PCP verifiers, we always interpret Boolean True and False (Accept and Reject) as real 1 and 0 respectively. By doing so, the expectation of output of PCP verifier is naturally its probability of acceptance. See also Section 3.1 for more details of the arithmetization.

5.2 Description of the Cheating Algorithm $\mathcal{A}_{\mathsf{cheat}}$

Now, we first describe a nondeterministic algorithm $\mathcal{A}_{\mathsf{cheat}}$ to speed up the computation $\mathcal{A}_{\mathsf{FS}}^{T}(x)$, where $\mathcal{A}_{\mathsf{FS}}^{T}$ is defined in Theorem 5.4. We will borrow most notation from [CLW20]. Our algorithm $\mathcal{A}_{\mathsf{cheat}}$ is basically the same as the algorithm $\mathcal{A}_{\mathsf{PCPP}}$ used in the proof of [CLW20, Theorem 1.2].

Set up. The algorithm $\mathcal{A}_{\mathsf{cheat}}$ is parameterized by two sufficiently small constants $\alpha, \delta \in (0,1)$ and a sufficiently large constant $K \geq 1$ specifying the time bound $T(n) = n^K$. We assume that K is large enough so that the PCP construction in Theorem 5.5 can be done in $\mathrm{poly}(n) \leq n^{K/2}$ time.

Using PCP first. On an input z of length n, $\mathcal{A}_{\mathsf{cheat}}$ applies the PCP from Theorem 5.5 to the computation $\mathcal{A}_{\mathsf{FS}}^{T}(x)$, and obtains an oracle circuit VPCP_z. Recall that both of VPCP_z and its oracle take inputs of length $\ell = \ell(n) = \log(T(n)) + O(\log\log T(n))$. Theorem 5.5 implies the following.

Claim 5.7. *The following statements hold.*

1. *If $\mathcal{A}_{\mathsf{FS}}^{T}(z) = 1$, then there an oracle $\mathcal{O}$ such that $\mathsf{VPCP}_z^{\mathcal{O}}(x) = 1$ for every $x \in \{0,1\}^{\ell}$.*
2. *It $\mathcal{A}_{\mathsf{FS}}^{T}(z) = 0$, then for every oracle $\mathcal{O}$, it holds that $\Pr_{x\in\{0,1\}^{\ell}}[\mathsf{VPCP}_z^{\mathcal{O}}(x) = 1] \leq \frac{1}{n^{10}} \leq \frac{1}{\ell^{10}}$.*

Then $\mathcal{A}_{\mathsf{cheat}}$ guesses a general circuit $C\colon \{0,1\}^{\ell} \to \{-1,1\}$ of size at most $S(\ell)^{\alpha}$ as the oracle for VPCP_z. Feeding C into VPCP_z, it obtains the circuit VPCP_z^{C}, with circuit size bounded above by $\mathrm{poly}(|C|) = S(\ell)^{O(\alpha)}$. By Claim 5.7, the algorithm $\mathcal{A}_{\mathsf{cheat}}$ needs to distinguish between the following two cases:

1. $\mathsf{VPCP}_z^{C}\colon \{0,1\}^{\ell} \to \{0,1\}$ is a tautology.
2. VPCP_z^{C} accepts at most $2^{\ell}/\ell^{10}$ many inputs.

The PCPP construction and notation. $\mathcal{A}_{\mathsf{cheat}}$ then applies the PCPP from Theorem 5.6 to the circuit VPCP_z^{C}. It produces a 2SAT instance Φ with $m = \mathrm{poly}(|C|) = S(\ell)^{O(\alpha)}$ many clauses over the variable set $\mathcal{Y} \cup \mathcal{Z}$, as well as an encoding function $\mathsf{Enc}\colon \{0,1\}^{\ell} \to \{0,1\}^{|\mathcal{Y}|}$. For $(s,t) \in [|\mathcal{Y}|] \times [|\mathcal{Z}|]$, we use $\mathcal{Y}_s$ and $\mathcal{Z}_t$ to denote the s-th variable in $\mathcal{Y}$ and the t-th variable in $\mathcal{Z}$, respectively.

Recall by Theorem 5.6 that s is a power of two. For brevity, we set $r = \ell + 1 + \log m$.

To elegantly discuss the algorithm and its analysis, we introduce some useful notation. Let the clauses of Φ be $(\mathsf{Cons}_i)_{i=1}^{m}$, where each of Cons_i involves 2 variables from $\mathcal{Y} \cup \mathcal{Z}$. For each clause Cons_i, it extends to a degree-2 polynomial, denoted by $\widetilde{\mathsf{Cons}}_i$.[22] For every $i \in [m]$ and $j \in [2]$, we set an indicator $\mathcal{T}_{i,j} \in \mathcal{Y} \cup \mathcal{Z}$ to indicate the j-th variable in Cons_i.

[22] For inputs to $\widetilde{\mathsf{Cons}}_i$, we identify Boolean False and True as real 1 and -1 respectively. For outputs of $\widetilde{\mathsf{Cons}}_i$, we identify Boolean False and True as real 0 and 1 respectively.

1. By a "real-valued proof" we mean a pair of two lists of proof functions (Y, Z) for PCPP, where $Y = (Y_s)_{s\in[|\mathcal{Y}|]}$, $Z = (Z_t)_{t\in[|\mathcal{Z}|]}$ and each of Y_s and Z_t is a function from $\{0,1\}^\ell$ to $\mathbb{R}$. Based on (Y, Z), we define the following terminologies:

   - Recall that each clause Cons_i involves two variables. We define indicators $T_{i1}^{(Y,Z)}$ and $T_{i2}^{(Y,Z)}$ to indicate the corresponding functions in (Y, Z).
   - Recall that each clause Cons_i extends to a polynomial $\widetilde{\mathsf{Cons}_i}$. We define $F_i^{(Y,Z)} := \widetilde{\mathsf{Cons}_i}(T_{i1}^{(Y,Z)}, T_{i2}^{(Y,Z)})$.

   Note that these objects all depend on the given proof (Y, Z), when the context is clear, we also omit the superscript, and simply write them as T_{ij} and F_i.

2. By a "Boolean-valued proof" we mean a pair of two lists of proof functions $(\widehat{Y} = \mathsf{Enc}(x), \widehat{Z})$ where $\widehat{Y}_s(x) = \mathsf{Enc}_s(x)$ for every $x \in \{0,1\}^\ell$ and $s \in [|\mathcal{Y}|]$, $\widehat{Z} = (\widehat{Z}_t)_{t\in[|\mathcal{Z}|]}$, and each $\widehat{Z}_t$ is a function from $\{0,1\}^\ell \to \{-1,1\}$. Recall that $\mathsf{Enc}\colon \{0,1\}^\ell \to \{-1,1\}^{|\mathcal{Y}|}$ is the corresponding function in Theorem 5.6. Similar to the case of real-valued proofs, the proof $(\widehat{Y} = \mathsf{Enc}(x), \widehat{Z})$ induces $\widehat{T}_{ij}^{(\widehat{Y},\widehat{Z})}$ and $\widehat{F}_i^{(\widehat{Y},\widehat{Z})}$. When the context is clear, we omit the superscript and write them as $\widehat{T}_{ij}$ and $\widehat{F}_i$.

To clarify, we always use (Y, Z) to denote a real-valued proof, and $(\widehat{Y}, \widehat{Z})$ to denote a Boolean-valued proof. We summarize the properties of the PCPP construction in the following claim. Recall that s_{pcpp} and c_{pcpp} are the soundness and completeness parameters in Theorem 5.6.

Claim 5.8. *The following statements hold.*

1. *If* $\mathsf{VPCP}_z{}^C$ *is a tautology, then there is a Boolean proof* $(\widehat{Y} = \mathsf{Enc}(x), \widehat{Z})$ *such that*
$$\mathop{\mathbb{E}}_{x\leftarrow U_\ell}\ \mathop{\mathbb{E}}_{i\in[m]} \widehat{F}_i(x) \geq c_{\mathsf{pcpp}}.$$

2. *If* $\mathsf{VPCP}_z{}^C(x) = 1$ *for at most a* $\frac{1}{\ell^{10}} \leq o(1)$ *fraction of* x*, then for every sufficiently large* n*, for every Boolean proof* $(\widehat{Y} = \mathsf{Enc}(x), \widehat{Z})$*, we have*
$$\mathop{\mathbb{E}}_{x\leftarrow U_\ell}\ \mathop{\mathbb{E}}_{i\in[m]} \widehat{F}_i(x) < c_{\mathsf{pcpp}} - \frac{9}{10}(c_{\mathsf{pcpp}} - s_{\mathsf{pcpp}}).$$

Guess proof function for PCPP. Next, $\mathcal{A}_{\mathsf{cheat}}$ guesses a $\mathsf{Sum} \circ \mathcal{F}_r$-functions of complexity at most $S(r)^{3\alpha}$, denoted by $H\colon \{0,1\}^{\log m} \times \{0,1\}^1 \times \{0,1\}^\ell \to \mathbb{R}$. We identify the first $\log m$ bits of inputs as an index in $[m]$, so that the first $\log m + 1$ bits can identify a variable $\mathcal{T}_{i,j}$. Based on H, we construct a real-valued proof (Y, Z) as
$$Y_s(x) := \mathop{\mathbb{E}}_{i,j:\mathcal{T}_{ij}=\mathcal{Y}_s} H(i,j,x) \ \text{ for } s \in [|\mathcal{Y}|],$$
$$Z_t(x) := \mathop{\mathbb{E}}_{i,j:\mathcal{T}_{ij}=\mathcal{Z}_t} H(i,j,x) \ \text{ for } t \in [|\mathcal{Z}|].$$

Recall that we defined $T_{i,j} \in Y \cup Z$ as the circuit corresponds to variable $\mathcal{T}_{i,j} \in \mathcal{Y} \cup \mathcal{Z}$. We define
$$P_{ij}(x) = \begin{cases} (1 + T_{ij}(x))^2(1 - T_{ij}(x))^2, & \text{if } T_{ij} \in Z, \\ (\mathsf{Enc}_s(x) - T_{ij}(x))^2, & \text{if } T_{ij} \in Y. \end{cases} \tag{29}$$

Verification. Finally, let $\mathcal{A}_{\mathsf{cheat}}$ verify the following:

$$\mathop{\mathbb{E}}_{i,j\in[m]\times[2]}\ \mathop{\mathbb{E}}_{x\leftarrow U_\ell} P_{i,j}(x) \le \delta, \tag{30}$$

$$\mathop{\mathbb{E}}_{i,j\in[m]\times[2]}\ \mathop{\mathbb{E}}_{x\leftarrow U_\ell} H(i,j,x)^2 \le 1, \tag{31}$$

$$\mathop{\mathbb{E}}_{i\in[m]}\ \mathop{\mathbb{E}}_{x\leftarrow U_\ell} F_i(x) \ge c_{\mathsf{pcpp}} - \frac{1}{2}(c_{\mathsf{pcpp}} - s_{\mathsf{pcpp}}). \tag{32}$$

$\mathcal{A}_{\mathsf{cheat}}$ accepts if and only if all of the conditions above hold. Note that each of those summations above can reduce to solving $S(\ell)^{O(\alpha)}$ many #SAT tasks for $\mathsf{AND}_4 \circ \mathcal{F}$-functions[23].

Running time of $\mathcal{A}_{\mathsf{cheat}}$. We verify that the algorithm $\mathcal{A}_{\mathsf{cheat}}$ runs in $\mathsf{NTIME}[o(T(n))]$ for small enough α: the construction of VPCP_z requires $\mathrm{poly}(n) < T(n)^{1/2}$ time for sufficiently large constant K; the PCPP construction runs in $S(\ell)^{O(1)}$ time; the guess and verification run in $2^r/S(r) \cdot S(r)^{O(\alpha)} \le 2^\ell/S(\ell)^{\Omega(1)} \le o(T(n))$ time for small enough α. (Recall that $r = \ell + O(\alpha \log S(\ell))$, and $S(\ell) \ge \ell^{\omega(1)}$.) This completes the description of algorithm.

5.3 Proof of Theorem 5.2

We state two crucial properties of $\mathcal{A}_{\mathsf{cheat}}$ below. First, we observe that the algorithm $\mathcal{A}_{\mathsf{cheat}}$ only makes one-sided error.

Lemma 5.9. *For every small enough constants $\alpha, \delta \in (0,1)$ and for every sufficiently large constant $K \ge 1$, the following holds: $\mathcal{A}_{\mathsf{cheat}}(z) \le \mathcal{A}_{\mathsf{FS}}^{T}(z)$ for all but finitely many inputs z.*

The proof of Lemma 5.9 can be found in Appendix B.

Combining Lemma 5.9 with Theorem 5.4. We conclude that for every sufficiently large n, one can apply the refuter $\mathcal{R}^T$ to find an input z of length $|z| \in [n, n^K + n]$ such that $\mathcal{A}_{\mathsf{cheat}}(z) = 0$ and $\mathcal{A}_{\mathsf{FS}}^{T}(z) = 1$. Considering one such z, by $\mathcal{A}_{\mathsf{FS}}^{T}(z) = 1$ and Item (1) of Claim 5.7 we know that there is an oracle $\mathcal{O}$ such that $\mathsf{VPCP}_z^{\mathcal{O}}$ is a tautology. If there is no circuit C of size at most $S(\ell)^\alpha$ such that VPCP_z^{C} is a tautology, then in particular it implies that this $\mathcal{O}$ cannot be computed by circuits of size at most $S(\ell)^\alpha$. This proves the Case (1) in Theorem 5.2 assuming that no circuit C of size at most $S(\ell)^\alpha$ can make VPCP_z^{C} a tautology.

In the following, we will show Case (2) in Theorem 5.2 holds if there is a circuit C of size at most $S(\ell)^\alpha$ such that VPCP_z^{C} is a tautology. This will finish the proof of Theorem 5.2 as at least one of Case (1) or Case (2) holds regardless of such a circuit C exists or not.

Lemma 5.10. *For every small enough constants $\alpha, \delta \in (0,1)$ and for every sufficiently large constant $K \ge 1$, the following holds: Suppose for some input z of length n such that $\mathcal{A}_{\mathsf{cheat}}(z) = 0$, there is a circuit of size at most $S(\ell)^\alpha$ such that VPCP_z^{C} is a tautology with $(\widehat{Y} = \mathsf{Enc}(x), \widehat{Z})$ being its correct Boolean-valued proof. Then the function $H^{\widehat{Y},\widehat{Z}}$ defined by*

$$\begin{aligned} H^{\widehat{Y},\widehat{Z}} \colon \{0,1\}^{\log(m)+1+\ell} &\to \{-1,1\} \\ (i,j,x) &\mapsto \widehat{T}_{ij}(x) \end{aligned} \tag{33}$$

is hard in the following sense: letting $r = \log(m) + 1 + \ell$, for every $\mathsf{Sum} \circ \mathcal{F}_r$-functions H such that $\mathsf{complexity}(H) \le S(r)^{3\alpha}$ and $\|H\|_4 \le 1$, it holds that

$$\langle H^{\widehat{Y},\widehat{Z}}, H \rangle < (1 - \delta/5). \tag{34}$$

[23] This is the same as the algorithm used in [CLW20]. We refer interested readers to [CLW20, Theorem 4.8] for the details about how this works.

The proof of Lemma 5.10 can be found in Appendix B.

We are finally ready to prove Theorem 5.2.

Proof of Theorem 5.2. We design the E^{NP} algorithm $\mathcal{A}_{\mathsf{hard}}$ as follows. Let n be a sufficiently large input length. On an input 1^n, $\mathcal{A}_{\mathsf{hard}}$ sets $m = 2^{n/K}$. By Theorem 5.4, $\mathcal{A}_{\mathsf{hard}}$ can find in $\mathrm{poly}(m) \le 2^{O(n)}$ time (with access to an NP oracle) an input z of length $|z| \in [m, m + m^K]$ such that $\mathcal{A}^{T}_{\mathsf{FS}}(z) = 1$ and $\mathcal{A}_{\mathsf{cheat}}(z) = 0$.

Consider the PCP system VPCP_z. Since $\mathcal{A}^{T}_{\mathsf{FS}}(z) = 1$, by Item (1) of Claim 5.7, there is an oracle $\mathcal{O}\colon \{0,1\}^{\ell} \to \{0,1\}$ for $\ell = K \log|z| + O(\log\log|z|)$ such that $\mathsf{VPCP}^{\mathcal{O}}_z$ is a tautology. Recall that $m = 2^{n/K}$ and $|z| \in [m, m + m^K]$, it follows that $n \le \ell \le n(K+1)$. $\mathcal{A}_{\mathsf{hard}}$ then construct the lexicographically first such oracle, still denoted by $\mathcal{O}$ for convenience, which can be found with the help of an NP oracle in $\mathrm{poly}(m) \le 2^{O(n)}$ time. Depending on whether $\mathcal{O}$ can be computed by small circuits or not, we have the following two cases:

1. ($\mathcal{O}$ **is hard.**) That is, $\mathcal{O}$ cannot be computed by a (general) circuit of size at most $S(\ell)^{\alpha}$. In this case, $\mathcal{O}$ induces a hard function on ℓ-bit inputs, and Case (1) of Theorem 5.2 holds. We let $\mathcal{A}_{\mathsf{hard}}$ output $\mathcal{O}$.

2. ($\mathcal{O}$ **is easy.**) That is, $\mathcal{O}$ can be computed by a (general) circuit of size at most $S(\ell)^{\alpha}$. In this case, $\mathcal{A}_{\mathsf{hard}}$ first constructs the lexicographically first such circuit C (with access to an NP oracle in $2^{O(n)}$ time). Feeding C into the oracle circuit VPCP_z, $\mathcal{A}_{\mathsf{hard}}$ then obtains a circuit VPCP^{C}_z. Consider the PCP of Proximity proof (the 2SAT instance) Φ for VPCP^{C}_z over variables $(\mathcal{Y}, \mathcal{Z})$. Since VPCP^{C}_z is a tautology, we have a list of proof functions $(\widehat{Y} = \mathsf{Enc}(x), \widehat{Z})$ such that for every $x \in \{0,1\}^{\ell}$, at least c_{pcpp}-fraction of clauses are satisfied by the assignment $(\mathcal{Y}, \mathcal{Z}) = (\widehat{Y}(x) = \mathsf{Enc}(x), \widehat{Z}(x))$. Now, given that $\mathcal{A}_{\mathsf{cheat}}(z) = 0$, it follows from Lemma 5.10 that the function $H^{\widehat{Y},\widehat{Z}}$ defined by
$$H^{\widehat{Y},\widehat{Z}}(i,j,x) := \widehat{T}_{ij}(x)$$
satisfies Case (2) of Theorem 5.2 statement[24]. We let $\mathcal{A}_{\mathsf{hard}}$ output $H^{\widehat{Y},\widehat{Z}}$.

□

6 Strong Correlation Bounds Against $\mathbb{F}_2$-Polynomials

In this section, we apply Theorem 5.4 to prove Theorem 1.1 (the strong correlation bound against $\mathbb{F}_2$-polynomials). For an $\mathbb{F}_2$-polynomial $P\colon \mathbb{F}_2^n \to \mathbb{F}_2$, we will consider its corresponding Boolean function (recall that we take Boolean functions to be functions from $\{0,1\}^*$ to $\{-1,1\}$) defined by $B_P(x) := (-1)^{P(x)}$ for every $x \in \{0,1\}^n$.

6.1 Special Collections of Functions Extending F_2-Polynomials

We will work with two special collections of functions, which contains low-degree $\mathbb{F}_2$-polynomials as a sub-collection. We give their definitions below.

[24] Note that the inapproximability parameter here is $(1-\delta/5)$ instead of $(1-\delta)$. This is OK since Theorem 5.2 only claims the existence of one such $\delta > 0$.

Definition 6.1. *For every $n, d, p \in \mathbb{N}_{\geq 1}$ such that $d, p \leq n$, we define the n-bit function collection $\mathcal{H}_n^{d,p}$ as the set of all functions $C\colon \{0,1\}^n \to \{-1,1\}$ that can be written as*

$$C(x) = (-1)^{P(x)} \cdot Q(x_{\leq n-p}),$$

where $P\colon \mathbb{F}_2^n \to F_2$ is an $\mathbb{F}_2$-polynomial of degree at most d and Q is an arbitrary function from $\{0,1\}^{n-p}$ to $\{-1,1\}$ (recall that $x_{\leq n-p}$ is the length-$(n-p)$ prefix of x). For convenience, we say that $P(x)$ and $Q(x)$ are the polynomial part and the free part of C, respectively.

We also define a function collection

$$\mathcal{F}^{n,d,p} := \bigcup_{q=n}^{\infty} \mathcal{H}_q^{d,(q-n)+p}.$$

Clearly, $\mathcal{H}_n^{d,p}$ and $\mathcal{F}^{n,d,p}$ are closed under XOR (which is multiplication over $\{-1,1\}$). In the following, we will state and prove the following two crucial properties of the collections:

1. $\mathcal{H}_n^{d,p}$ admits a non-trivial #SAT algorithm, as shown in Lemma 6.2.

2. There are $\mathcal{F}^{n,d,p}$-restrictable generators with relatively short seeds, as shown in Lemma 6.3.

Lemma 6.2. *For every $n, d, p \in \mathbb{N}_{\geq 1}$ such that $d, p \leq n$, there is an algorithm which given any $\mathcal{H}_n^{d,p}$-function C can evaluate*[25] *$\sum_{x \in \{0,1\}^n} C(x)$ in $2^{n-\Omega(\min(n/d,p))+O(\log(n))}$ time.*

Lemma 6.3. *For every $n, d, p \in \mathbb{N}_{\geq 1}$ such that $d, p \leq n$ and for every $k \in \mathbb{N}_{\geq 1}$, there is an $\mathcal{F}^{n,d,p}$-restrictable generator $\mathcal{G}\colon \{0,1\}^m \to \{0,1\}^{nk}$ with seed length $m = n + (k-1)p$.*

In the rest of this subsection we prove Lemma 6.2 and Lemma 6.3 separately.

Proof of Lemma 6.2. The new algorithm is a slight adaptation of the algorithm from [Wil18, Theorem 6.1]. It is instructive to describe the original algorithm here. Given a degree-d $\mathbb{F}_2$-polynomial P, the algorithm from [Wil18] computes the sum $\sum_{x \in \{0,1\}^n} P(x)$ over *integers* as follows:

1. Choose K to be a sufficiently large constant and let $\delta = \frac{1}{K \cdot d}$. Let $\ell = \lfloor \delta n \rfloor$. We will work with the ring $\mathbb{Z}_{2^\ell}$. Recall that there is a modulus-amplifying polynomial ([Yao90, BT94]) $V(x)$ of degree $2\ell - 1$ such that the following hold: (1) if $x \equiv 0 \pmod 2$ then $V(x) \equiv 0 \pmod{2^\ell}$; (2) if $x \equiv 1 \pmod 2$ then $V(x) \equiv 1 \pmod{2^\ell}$.

2. By the modulus-amplifying property of V, for every $(x_1, \ldots, x_{n-\ell+1}) \in \{0,1\}^{n-\ell+1}$, it holds that

$$\sum_{(x_{n-\ell+2},\ldots,x_n) \in \{0,1\}^{\ell-1}} P(x_1, \ldots, x_n) \equiv \sum_{(x_{n-\ell+2},\ldots,x_n) \in \{0,1\}^{\ell-1}} V(P(x_1, \ldots, x_n)) \pmod{2^\ell}.$$

   Note that in above, the two sums on both sides of the equality are taken over *integers* and not $\mathbb{F}_2$.

[25] That is, the algorithm is given a description of the polynomial part P of C by listing all coefficients in P, and a description of the free part Q of C by giving the truth-table of Q. Such description takes roughly $\sum_{i=0}^{d} \binom{n}{i} + 2^{n-p}$ bits.

3. We construct an $(n-\ell+1)$-variable polynomial P' over $\mathbb{Z}_{2^\ell}$ as

$$P'(x_1,\ldots,x_{n-\ell+1}) := \sum_{(x_{n-\ell+2},\ldots,x_n)\in\{0,1\}^{\ell-1}} V(P(x_1,\ldots,x_n)).$$

   Since P' is a $\mathbb{Z}_{2^\ell}$-polynomial, the sum on the right side above is taken over $\mathbb{Z}_{2^\ell}$ (not $\mathbb{F}_2$).

   For the sufficiently large K, the number of non-zero coefficients in the polynomial P' is bounded by $2^{0.01n}$. Fix one such K. We can first compute the description of the polynomial $P(x)^i$ (that is, a list of all its coefficients) for $i \in [2\ell-1]$, and then compute the description of the polynomial $V(P(x))$ as well. Last, we enumerate all possible assignments to last $\ell-1$ input bits and then take a sum to compute the description of P'. This takes $2^{0.02n}\cdot\mathrm{poly}(n)+2^{0.01n+\ell}\le 2^{0.1n}$ time.

4. By the modulus-amplifying property of V, we know that $P'(x_1,\ldots,x_{n-\ell+1})$ is exactly the number of $(x_{n-\ell+2},\ldots,x_n)$ such that $P(x_1,\ldots,x_n)=1$. Using dynamic programming, we can then produce the table $(P'(x_1,\ldots,x_{n-\ell+1}))_{x_1,\ldots,x_{n-\ell+1}}$ in $O(2^{n-\ell}\cdot\mathrm{poly}(n))\le O(2^{n-n/Kd})$ time. We take a summation over the table and report the answer, which completes the description of algorithm.

Our adaptation modifies the Step (4). Recall that we want to evaluate $\sum_x(-1)^{P(x)}\cdot Q(x_{\le n-p})$, where Q is an arbitrary Boolean function depending only on the length-$(n-p)$ prefix. Observe that Q just applies a "global XOR" to all the inputs sharing the same $(n-p)$-length prefix, which can fit perfectly in Step (3) of the algorithm above.

More precisely, note that in Step (4), entries of the table $(P'(x_1,\ldots,x_{n-\ell+1}))_{x_1,\ldots,x_{n-\ell+1}}$ correspond to disjoint sets of inputs. For ease of presentation, we say that an entry "contains" its corresponding inputs. A crucial observation is, as long as $\ell-1\le p$ [26], the effect of Q is the same on every inputs in one entry, since all inputs in a single entry share the same $(n-\ell+1)$-prefix. Therefore, we can easily handle the effect of Q to the summation after producing the table in Step (4) for the polynomial P: If $Q(x_{\le n-p})=1$, then the contribution from the entry does not change. Otherwise, all outputs in this entry should change the sign. In conclusion, the evaluation of the summation can be done in $2^{n-\Omega(\min(n/d,p))+O(\log n)}$ time. □

Proof of Lemma 6.3. For a given input $r\in\{0,1\}^m$ to the generator $\mathcal{G}$, we write $r=y\circ s_1\circ\cdots\circ s_k$, where $|y|=n-p$ and $|s_i|=p$ for each $i\in[k]$ (recall that $m=n+(k-1)p$). We simply define

$$\mathcal{G}(r)=(y\circ s_1,\ldots,y\circ s_k).$$

That is, all the generated instances share the same $(n-p)$-length prefix, and each of them holds an independent p-length suffix.

We now verify that this is an $\mathcal{F}^{n,d,p}$-restrictable generator. According to Definition 2.6, we need to define the functions T_i and verify the three items of Definition 2.6. For every $(m-n)$-bit string α, we write $\alpha=\alpha_1\circ\cdots\circ\alpha_{k-1}$ where each of α_i has length p. For every $i\in[k]$, set

$$T_i(x,\alpha)=x_{\le n-p}\circ\alpha_1\circ\cdots\circ\alpha_{i-1}\circ x_{>n-p}\circ\alpha_i\circ\cdots\circ\alpha_k.$$

Here recall that $x_{\le n-p}$ and $x_{>n-p}$ denote the length-$(n-p)$ prefix and length-p suffix of x, respectively.

[26] we can assume that this condition holds. Otherwise, we apply the algorithm for a larger degree parameter $d'=\Theta(n/p)$

It is straightforward to verify that Item (1) and (2) of Definition 2.6 hold. We establish Item (3) below. For every list of functions $(u_j \colon \{0,1\}^n \to \{-1,1\})_{j\in[k]\setminus\{i\}}$, $\alpha \in \{0,1\}^{m-n}$ and $C \in \mathcal{H}_m^{d,(m-n)+p}$, we consider the function

$$D(x) := C(T_i(x,\alpha)) \cdot \prod_{j\in[k]\setminus\{i\}} u_j(\mathcal{G}(T_i(x,\alpha))_j).$$

Clearly $C(T_i(x,\alpha))$ is an $\mathcal{H}_n^{d,p}$-function in x: the polynomial part of C does not increase its degree in a restriction, and the free part of C only depends on the length-$(n-p)$ prefix of its input, which is also the length-$(n-p)$ prefix of x in computing $C(T_i(x,\alpha))$. Moreover, for every $j \in [k] \setminus \{i\}$, the function $u_j(\mathcal{G}(T_i(x,\alpha))_j)$ only depends on the $(n-p)$-length prefix of x, which is also an $\mathcal{H}_n^{d,p}$-function. Since $\mathcal{H}_n^{d,p}$ is closed under XOR (that is, multiplication over $\{-1,1\}$), the function $D(x)$ is in $\mathcal{H}_n^{d,p}$. This completes the proof. □

6.2 Applying the New XOR Lemma

Let $\beta > 0$ be a sufficiently enough constant. We will consider the function collection

$$\mathcal{F}^{\mathsf{poly}} = \bigcup_{n\geq 1} \mathcal{H}_n^{\beta\sqrt{n/\log n}, \sqrt{n\log n}/\beta}.$$

Also let $S(n) := 2^{\beta\sqrt{n\log n}}$. By Lemma 6.2, there exists $\beta > 0$ such that $\mathcal{F}^{\mathsf{poly}}$ is $S(n)$-applicable: $\mathcal{F}^{\mathsf{poly}}$ is closed under negation; there is a $2^{n-\Omega(\sqrt{n\log n}/\beta)} \leq 2^n/S(n)$-time algorithm solving #SAT for $\mathsf{AND}_4 \circ \mathcal{F}^{\mathsf{poly}}$ (which can reduce to solving 16 #SAT tasks for $\mathsf{XOR}_4 \circ \mathcal{F}^{\mathsf{poly}}$ by standard Fourier analysis); $\mathcal{F}^{\mathsf{poly}}$ contains all parity functions, which are just polynomials of degree 1. Applying Theorem 5.2 to $\mathcal{F}^{\mathsf{poly}}$, we obtain the following.

Corollary 6.4. *There are constants $\alpha, \delta > 0$ and $K \geq 1$ such that the following hold. There is an E^{NP} machine which, for all sufficiently large n, on input 1^n, outputs in $2^{O(n)}$ time a Boolean function $f \colon \{0,1\}^\ell \to \{-1,1\}$ where $\ell \in [n, Kn]$ such that one of the following holds.*

1. *f cannot be computed by $2^{\alpha\sqrt{\ell\log\ell}}$-size general circuits.*
2. *For every $\mathsf{Sum} \circ \mathcal{F}_\ell^{\mathsf{poly}}$-function $H \colon \{0,1\}^\ell \to \mathbb{R}$ such that $\mathsf{complexity}(H) \leq 2^{3\alpha\sqrt{\ell\log\ell}}$ and $\|H\|_4 \leq 1$ it holds that*
$$\langle f, H\rangle < (1-\delta).$$

Using the algorithm of Corollary 6.4 we can construct in $2^{O(n)}$ time a function f that meets one of two conditions above. Now we apply hardness amplification on the function f, depending on which case in Corollary 6.4 holds.

Case 1: f is worst-case hard against general circuits. In this case, we apply Theorem 3.5 to the function f_ℓ. In $2^{O(n)}$ time we can get an $s = \Theta(\ell)$-bit function g'_s such that g'_s cannot be $\left(\frac{1}{2} + 2^{-o(\sqrt{n/\log n})}\right)$-approximated by (general) circuits of size $2^{o(\sqrt{n\log n})}$. Since every n-variable $\mathbb{F}_2$-polynomial of degree at most b can be simulated by a circuit of size $2^{O(b\log n)}$. It follows that g'_s cannot be $\left(\frac{1}{2} + 2^{-o(\sqrt{n/\log n})}\right)$-approximated by $\mathbb{F}_2$-polynomials of degree at most $o(\sqrt{n/\log n})$.

Case 2: f is weakly average-case hard against $\mathsf{Sum} \circ \mathcal{F}_\ell^{\mathsf{poly}}$. In this case, note that

$$\mathcal{H}_\ell^{\beta\sqrt{\ell/\log\ell},\sqrt{\ell\log\ell}/\beta} \subseteq \mathcal{F}^{\mathsf{poly}} \cap \mathcal{F}^{\ell,\beta\sqrt{\ell/\log\ell},\sqrt{\ell\log\ell}/\beta}.$$

We apply Lemma 4.1 to the function f_ℓ with $\mathcal{F}^{\ell,\beta\sqrt{\ell/\log\ell},\sqrt{\ell\log\ell}/\beta}$-restrictable generator of Lemma 6.3, where we set the inapproximability parameter as $\varepsilon = 2^{-\alpha\sqrt{\ell/\log\ell}}$. Let the number of instances generated in applying Lemma 4.1 be $k = \Theta(\log\varepsilon^{-1}) = \Theta(\alpha\sqrt{\ell/\log\ell})$. Also let the instance generator be $\mathcal{G}\colon \{0,1\}^s \to \{0,1\}^{nk}$. Since the seed length to the restrictable generator is $m = \ell + k\sqrt{\ell\log\ell}/\alpha \le O(\ell)$, it follows from Lemma 4.1 that $s \le O(m) = O(\ell)$. Now, we define a function $g'_s\colon \{0,1\}^s \to \{-1,1\}$ as $g'_s := f^{\oplus k} \circ \mathcal{G}$. By Lemma 4.1, it follows that g'_s cannot be $\left(\frac{1}{2} + 2^{-o(k)}\right)$-approximated by $\mathcal{F}^{\mathsf{poly}}$-functions. We have in particular that g'_s cannot be $\left(\frac{1}{2} + 2^{-o(\sqrt{n/\log n})}\right)$-approximated by $\mathbb{F}_2$-polynomials of degree at most $\beta\sqrt{n/\log n}$.

Padding. In both cases, given n, we can find in $2^{O(n)}$ time a Boolean function $g'_s\colon \{0,1\}^s \to \{-1,1\}$ with $s = \Theta(n)$, such that g'_s cannot be $\left(\frac{1}{2} + 2^{-o(\sqrt{n/\log n})}\right)$-approximated by $\mathbb{F}_2$-polynomials of degree at most $o(\sqrt{n/\log n})$. Now we choose a large enough constant C so that $s \le Cn$ for all sufficiently large n.

Design of the E^{NP} function. We design the final E^{NP} algorithm $\mathcal{A}$ as follows. Given an input z, $\mathcal{A}$ sets $n = |z|/C$ and finds the function g'_s as described before. Since $s \le nC \le |z|$, we just let $\mathcal{A}$ output $g'_s(z_{\le s})$. If follows that for sufficiently large input length m, $\mathcal{A}$ on m-bit inputs computes a function that cannot be $\left(\frac{1}{2} + 2^{-o(\sqrt{m/\log m})}\right)$-approximated by $\mathbb{F}_2$-polynomials of degree at most $o(\sqrt{m/\log m})$. This completes the proof of Theorem 1.1.

7 Better Degree-Error Trade-off against $\mathbb{F}_2$-Polynomials and P^{NP} Construction of Extremely Rigid Matrices

In this section, we prove degree-error trade-off for E^{NP} against $\mathbb{F}_2$-polynomials (Theorem 1.3) and present P^{NP} construction of extremely rigid matrices (Theorem 1.5).

Before we proceed, we remark that one can already combine known techniques from [CLW20] and [BHPT20] to prove the following (which is also independently utilized by [Lu20] and [HV20]).

Theorem 7.1. *For every $\beta \in (0,1)$, there is an E^{NP} function f such that, for every sufficiently large n, it holds that* $\operatorname{corr}(f, n^\beta/\log n) \le \exp(-\Omega(n^{\frac{1}{2}(1-\beta)}))$.

Using our new derandomized XOR lemma, we can substantially improve the correlation parameters from Theorem 7.1 to those stated in Theorem 1.3 and Theorem 1.5 (restated below).

Reminder of Theorem 1.3. *For every $\beta \in (0,1)$, there is an E^{NP} function f such that, for every sufficiently large n, it holds that* $\operatorname{corr}(f, n^\beta/\log n) \le \exp(-\Omega(n^{\frac{2}{3}(1-\beta)}))$.

Reminder of Theorem 1.5. *For every constant $\varepsilon \in (0,1)$, there is a P^{NP} algorithm which on input 1^n outputs an $n \times n$ $\mathbb{F}_2$-matrix H_n satisfying $\mathcal{R}_{H_n}(2^{\log^{1-\varepsilon} n}) \ge (1/2 - \exp(-\log^{2/3\cdot\varepsilon} n)) \cdot n^2$, for every sufficiently large n.*

In fact, [HV20] and [Lu20] stated a more fine-grained trade-off: for every $n, \rho, k \in \mathbb{N}_{\geq 1}$ such that $\log \rho \leq \delta \log n / k(\log\log n + k)$ for a sufficiently small $\delta > 0$, they constructed an $n \times n$ matrix $H_n \in \mathbb{F}_2^{n\times n}$ such that $\mathcal{R}_{H_n}(\rho) \geq (1/2 - 2^{-k}) \cdot n^2$. Through a more careful calculation, our approach (which is based on the derandomized XOR Lemma) can recover their results and indeed applies to a wider regime $\log \rho \leq \delta \log n / \sqrt{k}(\log\log n + k)$. For the sake of a clearer exposition, in this section we will only focus on the case where ρ, the rank parameter, is set to 2^{n^β} for a constant $\beta \in (0, 1)$.

Now, let us formally define the function collection we will work with.

Definition 7.2 (Collection of low-rank functions)**.** *For every $n, r \in \mathbb{N}$ such that $r \leq 2^n$, let $\mathcal{M}_{2n}^r$ denote the collection of functions such that, for every $f \in \mathcal{M}$, there is a $2^n \times 2^n$ matrix M over $\mathbb{F}_2$ of rank at most r such that $f(x, y) = (-1)^{M(x,y)}$ for every $(x, y) \in \{0, 1\}^{2n}$. Typically, we describe an $\mathcal{M}_{2n}^r$-function by giving the low-rank decomposition of the matrix $M := A \cdot B^T$ where $A, B \in \mathbb{F}^{2^n\times r}$, which only requires $O(2^n \cdot r)$ bits. We also let $\mathcal{M}_{2n+1}^r$ be an empty collection for every $n \in \mathbb{N}$.*

In this section, we will frequently view a $2n$-bit input function as a $2^n \times 2^n$ matrix and vice versa. For convenience, for every $n \in \mathbb{N}_{\geq 1}$ and every $2n$-bit function $f\colon \{0, 1\}^{2n} \to \{-1, 1\}$, for every input x to f, we call the first n bits of x (*i.e.*, $x_{\leq n}$) as the *row* index, and the last n bits of x (*i.e.*, $x_{>n}$) as the *column* index.

We will prove the following theorem, which implies Theorem 1.5 and Theorem 1.3.

Theorem 7.3. *For every $\beta \in (0, 1)$, there is an E^{NP} function f such that, for every sufficiently large n, f_{2n} is a function that cannot be $\left(\frac{1}{2} + 2^{-o(n^{\frac{2}{3}(1-\beta)})}\right)$-approximated by $\mathcal{M}_{2n}^{2^{2n^\beta}}$-functions.*

We sketch how we obtain Theorem 1.3 and Theorem 1.5 from Theorem 7.3,

Proof of Theorem 1.3. We show that $\mathcal{M}_{2n}^r$ contains all degree-d $\mathbb{F}_2$-polynomials on $2n$ variables, given that $\sum_{i=0}^{d}\binom{2n}{i} \leq r$. In fact, for every $\mathbb{F}_2$-polynomial $P(x_1, \dots, x_n, y_1, \dots, y_n)$ of degree at most d, if we write the truth-table of P as a $2^n \times 2^n$ matrix, then each monomial corresponds to a rank-1 matrix, and the matrix is a sum of at most $\sum_{i=0}^{2n}\binom{2n}{i} \leq r$ rank-1 matrix (*i.e.*, the monimials).

We consider the function f constructed in Theorem 7.3, given that f_{2n} cannot be $\left(\frac{1}{2} + 2^{-o(n^{\frac{2}{3}(1-\beta)})}\right)$-approximated by $\mathcal{M}_{2n}^{2^{2n^\beta}}$-functions, it follows naturally that $\mathrm{corr}(f_{2n}, n^\beta / \log n) \leq \exp(-\Omega(n^{\frac{2}{3}(1-\beta)}))$. To handle the odd input lengths, we simply define $f_{2n+1}(x) := f_{2n}(x_{\leq 2n})$ for every $x \in \{0, 1\}^{2n+1}$, and this completes the proof. □

Proof of Theorem 1.5. Let $\varepsilon \in (0, 1)$ be a constant. First, we set $\beta = 1 - \varepsilon$ and let f be the function constructed in Theorem 7.3 with parameter β.

We will use a padding argument. Given a sufficiently large integer $n \geq 1$, we set $\ell = \lceil \log \sqrt{n} \rceil$. Consider the function $f_{2\ell}$ constructed in Theorem 7.3. We view $f_{2\ell}$ as a $2^\ell \times 2^\ell$ matrix A, and "pad" it into a larger matrix. We set $k = \left\lfloor \frac{n}{2^\ell} \right\rfloor$ and defining a $(k2^\ell) \times (k2^\ell)$ matrix $M' := 1_k \otimes A$ where 1_k denotes all-ones $k \times k$ matrix and $\otimes$ denotes the Kronecker product of matrices. We then further pad the $(k2^\ell) \times (k2^\ell)$ matrix M' to an $n \times n$ matrix M by filling zeros in the empty entries.

The rigidity of M' follows from the fact that $\mathcal{R}_{\mathbb{1}_k \otimes A}(r) = \mathcal{R}_A(r) \cdot k^2$ (see, *e.g.*, [AC19, Lemma 2.7]). The rigidity of M follows from the rigidity of M', and the observation that we only add $O(n\sqrt{n}) \leq n^2 \cdot \exp\left(-\log^{\frac{2}{3}(1-\beta)}(n)\right)$ zeros in M. □

7.1 Technical Ingredients

We collect some crucial technical ingredients required by the proof of this section. First, we need a fast #SAT-algorithm for $\mathcal{M}_n^r$-functions.

Lemma 7.4 ([AC19, CW16]). *For every even integer $n \geq 1$ and $r \leq 2^{o(n)}$, there is a $2^{n-\Omega(n/\log r)}$-time deterministic algorithm for solving #SAT problem for $\mathcal{M}_n^r$-functions.*

Second, we introduce a restrictable generator for $\mathcal{M}$-functions with seed length shorter than nk.

Lemma 7.5. *For every $n, r, k \in \mathbb{N}_{\geq 1}$ such that $r \leq 2^n$ and $2 \leq k \leq r$, we define*

$$\mathcal{N}^{2n,r,k} := \mathcal{M}_{2n}^r \cup \mathcal{M}_{2n\lceil\sqrt{k}\rceil}^{r-k}.$$

There is an $\mathcal{N}^{2n,r,k}$-restrictable generator $\mathcal{G}\colon \{0,1\}^m \to \{0,1\}^{nk}$ with seed length $m = 2n\left\lceil\sqrt{k}\right\rceil$.

Proof. For brevity, let $t = \left\lceil\sqrt{k}\right\rceil$. We choose an arbitrary but fixed injective mapping $\rho\colon [k] \to [t] \times [t]$, denoted by $\rho(i) = (\rho(i)_u, \rho(i)_v)$. For every $z \in \{0,1\}^m$, we write $z = x_1 \circ \cdots \circ x_t \circ y_1 \circ \cdots \circ y_t$ where $|x_i| = |y_j| = n$ for every $i, j \in [t]$. We design our generator as

$$\mathcal{G}(z) := (x_{\rho(1)_u} \circ y_{\rho(1)_v}, \ldots, x_{\rho(k)_u} \circ y_{\rho(k)_v}).$$

For every $\alpha \in \{0,1\}^{m-2n}$, write $\alpha = \alpha_1 \circ \cdots \circ \alpha_{2t-2}$ where $|\alpha_j| = n$ for each $j \in [2t-2]$. For every $i \in [k]$, $x \in \{0,1\}^{2n}$ and $\alpha \in \{0,1\}^{m-2n}$, we construct the embedding function as

$$T_i(x, \alpha) := (\alpha_1, \ldots, \alpha_{\rho(i)_u - 1}, x_{\leq n}, \ldots, \alpha_{t-1}) \circ (\alpha_t, \ldots, \alpha_{t+\rho(i)_v - 2}, x_{>n}, \ldots, \alpha_{2t-2}).$$

It is easy to verify that Item (1) and (2) of Definition 2.6 both hold. We establish Item (3) below. For every list of functions $(u_j\colon \{0,1\}^{2n} \to \{-1,1\})_{j\in[k]\setminus\{i\}}$, $\alpha \in \{0,1\}^{m-2n}$ and $C \in \mathcal{M}_m^{r-k}$, we consider the function

$$D(x) := C(T_i(x,\alpha)) \cdot \prod_{j\in[k]\setminus\{i\}} u_j(\mathcal{G}(T_i(x,\alpha))_j).$$

By the design of T_i, the first and last n bits of x occur in the first and last $m/2$ bits of $T_i(x,\alpha)$ respectively, and consequently we have $C(T_i(x,\alpha))$ is an $\mathcal{M}_{2n}^{r-k}$-function in x. Moreover, for every $j \in [k] \setminus \{i\}$, the function $u_j(\mathcal{G}(T_i(x,\alpha))_j)$ only depends on either $x_{\leq n}$ or $x_{>n}$, so it is an $\mathcal{M}_{2n}^1$-function. Therefore, the function $D(x)$ is in $\mathcal{M}_{2n}^{r-k+(k-1)} \subset \mathcal{M}_{2n}^r$, which completes the proof. □

We also need the rectangular PCP of [BHPT20], stated below.

Theorem 7.6 ([BHPT20, Theorem 8.2 and Remark 8.3]). *Let M be an algorithm running in time $T = T(n) \geq n$ on inputs of the form (z, y) where $|z| = n$. For any odd constant integer $m \in \mathbb{N}$ such that $T(n)^{1/m} \geq n$, given $z \in \{0,1\}^n$ one can output in time $\mathrm{poly}(n, T^{1/m})$ a PCP verifier $\mathsf{VrecPCP}_z$ with proof length 2^ℓ, randomness γ, completeness 1 and soundness $s \in (0,1)$ such that the following hold.*

- ***Shortness.*** *$T(n) \leq 2^\ell \leq 2^\gamma \leq T \cdot \mathrm{polylog}(T)$, and ℓ is even.*
- ***Query complexity.*** *There is a constant q such that $\mathsf{VrecPCP}_z$ makes only q queries to the proof.*

- ***Rectangular.*** *The randomness $r \in \{0,1\}^\gamma$ can be split into three parts $r = (r_{\mathsf{row}}, r_{\mathsf{col}}, r_{\mathsf{shared}})$ such that $|r_{\mathsf{row}}| = |r_{\mathsf{col}}| \geq \frac{m-6}{2m} \log T(n)$. For a given proof $\pi\colon \{0,1\}^\ell \to \{-1,1\}$ and fixed r_{shared}, we write $\mathsf{VrecPCP}_z^\pi(r_{\mathsf{row}}, r_{\mathsf{col}}, r_{\mathsf{shared}})$ to denote the output of $\mathsf{VrecPCP}_z$ given $(r_{\mathsf{row}}, r_{\mathsf{col}}, r_{\mathsf{shared}})$ as randomness.*

  For every $i \in [q]$, the row index of the i-th query of $\mathsf{VrecPCP}_z$ only depends on the pair $(r_{\mathsf{shared}}, r_{\mathsf{row}})$, and the column index of the i-th query of $\mathsf{VrecPCP}_z$ only depends on $(r_{\mathsf{shared}}, r_{\mathsf{col}})$. After reading the proof, the decision of $\mathsf{VrecPCP}_z$ only depends on r_{shared}, the q queried bits, and p parity check bits over $(r_{\mathsf{row}}, r_{\mathsf{col}})$, where the specification of each parity check is determined only by r_{shared} and z. Here p is a constant.

- ***Completeness.*** *If there is a y such that $M(z,y)$ accepts, then there is a proof H such that*

$$\Pr_{r \leftarrow U_\gamma}\left[\mathsf{VrecPCP}_z^H(r) = 1\right] = 1.$$

- ***Soundness.*** *If no y causes $M(z,y)$ to accept, then for every proof H, we have*

$$\Pr_{r \leftarrow U_\gamma}\left[\mathsf{VrecPCP}_z^H(r) = 1\right] \leq s.$$

- ***Smoothness.*** *For every $p \in \{0,1\}^\ell$, the quantity*

$$\left|\{(r,i) : \mathsf{VrecPCP}_z^H(r) \textit{ makes the i-th query to } H(p)\}\right|$$

  is the same.

Arithmetization. By the rectangular property, fixing r_{shared}, the output of $\mathsf{VrecPCP}_z$ depends only on the q queried bits and p parity check bits over $(r_{\mathsf{row}}, r_{\mathsf{col}})$. For each r_{shared}, and $j \in [p]$, we define a function $C_j^{r_{\mathsf{shared}}}\colon \{0,1\}^{|r_{\mathsf{row}}|+|r_{\mathsf{col}}|} \to \{-1,1\}$ which maps the random bits $(r_{\mathsf{row}}, r_{\mathsf{col}})$ to its parity check result. Note that $C_j^{r_{\mathsf{shared}}}$ can be written as $C_j^{r_{\mathsf{shared}}}(r_{\mathsf{row}}, r_{\mathsf{col}}) = (-1)^{\mathrm{Par}_j^{r_{\mathsf{shared}}}(r_{\mathsf{row}}, r_{\mathsf{col}})}$ where $\mathrm{Par}_j^{r_{\mathsf{shared}}}(r_{\mathsf{row}}, r_{\mathsf{col}})$ computes an affine function over $(r_{\mathsf{row}}, r_{\mathsf{col}})$. Hence $\mathrm{Par}_j^{r_{\mathsf{shared}}}(r_{\mathsf{row}}, r_{\mathsf{col}})$ can be written as a rank-2 matrix[27], and $C_j^{r_{\mathsf{shared}}} \in \mathcal{M}^2_{|r_{\mathsf{row}}, r_{\mathsf{col}}|}$.

Now, given a randomness $r = (r_{\mathsf{shared}}, r_{\mathsf{row}}, r_{\mathsf{col}})$, we can write the output of $\mathsf{VrecPCP}_z$ as a multi-linear polynomial (over $\mathbb{R}$) of the query answers $(v_1, \ldots, v_q)$ and the parity check $(c_1, \ldots, c_p)$, denoted by $Q_{r_{\mathsf{shared}}}\colon \mathbb{R}^{q+p} \to \mathbb{R}$, which maps $(v_1, \ldots, v_q, c_1, \ldots, c_p)$ to a bit from $\{0,1\}$. Moreover, for every $r = (r_{\mathsf{shared}}, r_{\mathsf{row}}, r_{\mathsf{col}})$, we can write the decision as a multi-linear polynomial of queried bits, denoted by $P_r\colon \mathbb{R}^q \to \mathbb{R}$ (see Section 3.1 for details). Since both of P_r and $Q_{r_{\mathsf{shared}}}$ are multi-linear polynomials, we have

$$P_r(v_1, \ldots, v_q) = Q_{r_{\mathsf{shared}}}(v_1, \ldots, v_q, C_1^{r_{\mathsf{shared}}}(r_{\mathsf{row}}, r_{\mathsf{col}}), \ldots, C_p^{r_{\mathsf{shared}}}(r_{\mathsf{row}}, r_{\mathsf{col}})). \tag{35}$$

Then we can define an "arithmetized verifier", denoted by $\widetilde{\mathsf{VrecPCP}}_z$, which takes a real-valued function $H\colon \{0,1\}^\ell \to \mathbb{R}$ as proof. $\widetilde{\mathsf{VrecPCP}}_z$ samples a random string $(r_{\mathsf{row}}, r_{\mathsf{col}}, r_{\mathsf{shared}}) \leftarrow U_\gamma$, reads some values $(v_1, \ldots, v_q)$ from H according to the randomness, and outputs $P_r(v_1, \ldots, v_q)$. Intuitively, for a given real-valued function $H\colon \{0,1\}^\ell \to \mathbb{R}$ as a proof to $\widetilde{\mathsf{VrecPCP}}_z$, as long as H

[27] Note that every affine function over only r_{row} or r_{col} can be written as a rank-1 matrix. Therefore, being an XOR over two such affine functions, $\mathrm{Par}_j^{r_{\mathsf{shared}}}(r_{\mathsf{row}}, r_{\mathsf{col}})$ can be written as a rank-2 matrix.

is close to some Boolean function $H' \colon \{0,1\}^\ell \to \{-1,1\}$, we will have that $\mathbb{E}_r \left[\widetilde{\mathsf{VrecPCP}}_z^H(r)\right] \approx \mathbb{E}_r \left[\mathsf{VrecPCP}_z^{H'}(r)\right]$.

In fact, by the smoothness of $\mathsf{VrecPCP}_z$, we have the following lemma.

Lemma 7.7. *For two functions $H \colon \{0,1\}^\ell \to \mathbb{R}$ and $H' \colon \{0,1\}^\ell \to \{-1,1\}$, it holds that*

$$\mathop{\mathbb{E}}_{r \leftarrow U_\gamma} \left| \widetilde{\mathsf{VrecPCP}}_z^H(r) - \mathsf{VrecPCP}_z^{H'}(r) \right| \le 2^q \cdot q \cdot \left(2 + q^{1/(2q-2)} \|H\|_{2q-2}\right)^{2q-2} \|H - H'\|_2.$$

We prove Lemma 7.7 in Appendix C.

7.2 Proof of Theorem 7.3 via the Algorithmic Method

Now we try to prove Theorem 7.3 by the algorithmic method. First, we choose a constant K and set $T(n) = n^K$. We also choose a proper parameter m for Theorem 7.6. We choose K and m properly so that the construction of VrecPCP of Theorem 7.6 can be done in $T(n)^{1/2}$ time.

We prove the trade-off for every $\beta \in (0,1)$. We first set $c = \frac{1-\beta}{1+\beta/2}$. We let $\alpha, \delta > 0$ be two small enough constants and set $\tau \ge 1$ to be a *large enough* constant. For the algorithm $\mathcal{A}_{\mathsf{FS}}^T$ defined in Theorem 5.4, we design a cheating algorithm $\mathcal{A}_{\mathsf{matrix}}$ to speed up the computation of $\mathcal{A}_{\mathsf{FS}}^T$ as follows:

- Given an input $z \in \{0,1\}^n$, $\mathcal{A}_{\mathsf{matrix}}$ applies the rectangular PCP of Theorem 7.6 (with the constant m set to 13) to $\mathcal{A}_{\mathsf{FS}}^T(z)$ and obtains $\mathsf{VrecPCP}_z$. Let the proof length to $\mathsf{VrecPCP}_z$ be 2^ℓ. Note that $2^\ell = T(n) \cdot \mathrm{polylog}(T(n))$.

- Then $\mathcal{A}_{\mathsf{matrix}}$ guesses a $\mathsf{Sum} \circ \mathcal{M}_\ell^{2^{\tau \ell^{1-c}}}$-function of complexity at most $2^{\alpha \ell^c}$, denoted by H, as the proof. It then applies the following tests:

$$\mathop{\mathbb{E}}_{(x,y) \leftarrow U_\ell} (1 - H(x,y))^2 (1 + H(x,y))^2 \le \delta, \tag{36}$$

$$\mathop{\mathbb{E}}_{(x,y) \leftarrow U_\ell} H(x,y)^{2q-2} \le 1, \tag{37}$$

$$\mathop{\mathbb{E}}_{r \leftarrow U_\gamma} \left[\widetilde{\mathsf{VrecPCP}}_z^H(r)\right] \ge \frac{1+s}{2}. \tag{38}$$

  It accepts the proof H if all the above three tests pass, and reject otherwise. We will show in Lemma 7.8 (its proof is deferred to Appendix C) that for fixed $\tau \ge 1$, each of these tests can be done in $o(n^K)$-time for sufficiently small α. This completes the design of $\mathcal{A}_{\mathsf{matrix}}$.

Lemma 7.8. *For every constant $\tau \ge 1$, there is a sufficiently small $\alpha > 0$ such that the following is true. For every $z \in \{0,1\}^*$ and H being a $\mathsf{Sum} \circ \mathcal{M}_\ell^{2^{\tau \ell^{1-c}}}$-function with $\mathsf{complexity}(H) \le 2^{\alpha \ell^c}$, evaluations of the left-hand sides of (36)-(38) can be done in $2^{\ell - \Omega(\ell^c/\alpha)} \le o(n^K)$ time.*

Useful Lemmas. The tests (36)-(38) play a role that is similar to that of the tests (30)-(32) in the proof of Theorem 5.2. Analogously, the following lemmas can be established, and we defer their proofs to Appendix C.

First, we can verify that the algorithm $\mathcal{A}_{\mathsf{matrix}}$ only makes one-sided error.

Lemma 7.9. *For every sufficiently small $\delta > 0$, it holds that $\mathcal{A}_{\mathsf{matrix}}(z) \le \mathcal{A}_{\mathsf{FS}}^T(z)$ for every $z \in \{0,1\}^*$.*

Second, for every $z \in \{0,1\}^*$ such that $\mathcal{A}^T_{\mathsf{FS}}(z) = 1$ and $\mathcal{A}_{\mathsf{matrix}}(z) = 0$, the correct proof for $\mathsf{VrecPCP}_z$ is rigid in the following sense. (We say a proof H is correct for $\mathsf{VrecPCP}_z$ if it makes $\mathsf{VrecPCP}_z$ always accepts.)

Lemma 7.10. *For every sufficiently small $\alpha > 0$, the following is true. For every z such that $\mathcal{A}_{\mathsf{matrix}}(z) = 0$ and $\mathcal{A}^T_{\mathsf{FS}}(z) = 1$, every correct proof for $\mathsf{VrecPCP}_z$ is a function $H' \colon \{0,1\}^\ell \to \{-1,1\}$ such that, for every $\mathsf{Sum} \circ \mathcal{M}_\ell^{2^{\tau\ell^{1-c}}}$-function H with $\mathsf{complexity}(H) \le 2^{\alpha\ell^c}$, it holds that $\langle H, H' \rangle \le (1 - \delta/5)\|H\|_{2(q-1)}$.*

Applying the derandomized XOR lemma. In the following we will combine the derandomized XOR lemma Lemma 4.1 and Lemma 7.10 to finish the proof of Theorem 7.3. First, we show that there is a constant $C \ge 1$ and an E^{NP} algorithm $\mathcal{A}_{\mathsf{xor}}$ such that, for every sufficiently large n, $\mathcal{A}_{\mathsf{xor}}$ constructs a function $g \colon \{0,1\}^s \to \{-1,1\}$ that cannot be $\left(\frac{1}{2} + 2^{-o(s^{\frac{2}{3}(1-\beta)})}\right)$-approximated by $\mathcal{M}_s^{2^{o(s^\beta)}}$-functions, where $s \in [n, Cn]$ is an even integer.

Given $n \in \mathbb{N}_{\ge 1}$, $\mathcal{A}_{\mathsf{xor}}$ sets $n' = n^{2/(2+c)}$ and $m = 2^{n'/K}$ (for simplicity, we ignore the rounding issue here and pretend both n' and m are integers). By Theorem 5.4, $\mathcal{A}_{\mathsf{xor}}$ can find in $\mathrm{poly}(m) \le 2^{O(n)}$ time an input z of length $|z| \in [m, m + m^K]$ such that $\mathcal{A}^T_{\mathsf{FS}}(z) = 1$ and $\mathcal{A}_{\mathsf{matrix}}(z) = 0$. We consider the rectangular PCP system $\mathsf{VrecPCP}_z$. Let ${N_1}^2$ be the proof length of $\mathsf{VrecPCP}_z$. It holds that $N_1 \le O(|z|^{K/2}) \le 2^{O(n')}$. Also let $\ell = 2 \log N_1 \le O(n')$.

Since $\mathcal{A}^T_{\mathsf{FS}}(z) = 1$, there exists a function $H' \colon \{0,1\}^\ell \to \{-1,1\}$ being a correct proof for $\mathsf{VrecPCP}_z$. *i.e.*, $\mathsf{VrecPCP}_z^{H'}$ is a tautology. Using an NP oracle, $\mathcal{A}_{\mathsf{xor}}$ can find the lexicographically first such H' in $\mathrm{poly}(m) \le 2^{O(n)}$ time. By Lemma 7.10, it follows that for every $\mathsf{Sum} \circ \mathcal{M}_{2\ell}^{2^{\tau\ell^{1-c}}}$-function C with complexity bounded by $2^{\alpha\ell^c}$, we have $\langle H', C \rangle \le (1 - \delta/5)\|C\|_{2q-2}$.

We will apply the derandomized XOR Lemma (Lemma 4.1) with the restrictable generator given by Lemma 7.5. We set the inapproximability parameter as $\varepsilon = 2^{-\frac{\alpha}{3}\ell^c}$. Then we let $k = \Theta(\log \varepsilon^{-1}) = \Theta(\ell^c)$ be the number of instances given by Lemma 4.1. Consider the function collection $\mathcal{N}^{2\ell, 2^{\tau\ell^{1-c}}, k}$. We let $\mathcal{A}_{\mathsf{xor}}$ apply Lemma 4.1 to the function H' with the $\mathcal{N}^{2\ell, 2^{\tau\ell^{1-c}}, k}$-restrictable generator of Lemma 7.5. $\mathcal{A}_{\mathsf{xor}}$ obtains from H' a function $g \colon \{0,1\}^s \to \{-1,1\}$ where $s = \Theta(n\sqrt{k}) = \Theta(\ell^{1+c/2})$, such that g cannot be $\left(\frac{1}{2} + 2^{-o(\ell^c)}\right)$-approximated by $\mathcal{M}_s^{2^{\tau'\ell^{1-c}}}$-functions for some $\tau' \ge \Omega(\tau)$.

Note that $\ell = \Theta(n')$, $s = \Theta(\ell^{1+c/2}) = \Theta(n)$ and $\beta = \frac{2-2c}{2+c}$. So we have that $\ell^c = \Theta(s^{\frac{2}{3}(1-\beta)})$ and $\ell^{1-c} = \Theta(s^\beta)$. Hence, it follows that g cannot be $\left(\frac{1}{2} + 2^{-o(s^{\frac{2}{3}(1-\beta)})}\right)$-approximated by $\mathcal{M}_s^{2^{\tau''s^\beta}}$-functions for some $\tau'' \ge \Omega(\tau')$. Since $s \le O(n)$, there is a constant $C \ge 1$ such that $s \le Cn$ holds for every sufficiently large n.

Padding. Now we design the final E^{NP} algorithm $\mathcal{A}$. On an input x, $\mathcal{A}$ sets $n = |x|/C$ and invokes $\mathcal{A}_{\mathsf{xor}}$ to find a function $g \colon \{0,1\}^s \to \{-1,1\}$ as shown before. Since $s \le Cn \le |x|$, $\mathcal{A}$ just outputs $g(x_{\le s/2}, x_{>n-s/2})$. For every sufficiently large n, $\mathcal{A}$ on $2n$-bit inputs computes a function that cannot be $\left(\frac{1}{2} + 2^{-o(n^{\frac{2}{3}(1-\beta)})}\right)$-approximated by $\mathcal{M}_{2n}^{2^{\tau^{(3)}n^\beta}}$-functions for $\tau^{(3)} \ge \Omega(\tau'') \ge \Omega(\tau)$.

Choice of parameters. Finally, we specify our choice of the parameters, and which completes the proof. We first choose τ such that $\tau^{(3)} \ge 1$. Then we choose α accordingly such that $\mathcal{A}_{\mathsf{matrix}}$ runs in time $o(T(n))$. We also choose δ such that Lemma 7.9 and Lemma 7.10 hold.

8 A Conditional Construction of Nondeterministic PRGs

In this section, we give a construction of nondeterministic PRGs fooling a circuit $\mathscr{C}$, given that a sufficiently efficient CAPP algorithm for $\mathscr{C}$ exists.

We first recall the definition of nondeterministic pseudorandom generators (NPRGs), borrowed from [CLW20, Definition 3.4].

Definition 8.1. *Let $S, \ell \colon \mathbb{N} \to \mathbb{N}$ and $\varepsilon \colon \mathbb{N} \to (0,1)$ be functions, and $\mathscr{C}$ be a circuit class. Let $M(x,y)$ be a deterministic Turing machine. We say M is a* nondeterministic pseudo-random generator (NPRG) *with seed length $\ell(n)$ that ε-fools $\mathscr{C}$-circuits of size S if following holds:*

1. *On every input $x \in \{0,1\}^{\ell(n)}$ and $y \in \{0,1\}^{2^{O(\ell(n))}}$, $M(x,y)$ either rejects or outputs a string, and whether $M(x,y)$ rejects depends only on $|x|$ and y (but not on x).*
2. *If $M(x,y)$ does not reject, then there is a function $G_y \colon \{0,1\}^{\ell(n)} \to \{0,1\}^n$ such that for every $\mathscr{C}$ circuit C of size S,*

$$\left| \Pr_{\mathbf{x} \in \{0,1\}^{\ell(n)}} [C(G_y(\mathbf{x})) = 1] - \Pr_{\mathbf{z} \in \{0,1\}^n} [C(\mathbf{z}) = 1] \right| < \varepsilon(n),$$

   and $M(x,y)$ outputs $G_y(x)$ in nondeterministic $2^{O(\ell(n))}$ time.

3. *There is at least one input y such that $M(x,y)$ does not reject.*

If the above conditions only hold for infinitely many integers n, then we say M is an infinitely often NPRG *(i.o.-NPRG).*

We define CAPP (Circuit Acceptance Probability Problem) formally.

Definition 8.2. *In the CAPP problem, one is given a circuit $C \colon \{0,1\}^n \to \{-1,1\}$ of size $S(n)$, and is asked to estimate $\Pr_{x \leftarrow \{0,1\}^n}[C(x) = 1]$ within an additive error of $1/S(n)$.*

We use Junta_k to denote the family of k-juntas, *i.e.*, functions that only depend on k input bits. The main claim of this section is the following.

Theorem 8.3. *Let $\mathcal{C}$ be a nice circuit class[28]. If there is an $\varepsilon > 0$ such that, the $\#\mathsf{SAT}$ (or CAPP) problem of $\mathcal{C} \circ \mathsf{Junta}_{2^{\varepsilon n}}$-circuit of size $2^{\varepsilon n}$ can be solved in deterministic $2^{(1-\varepsilon)n}$ time, then there exists an infinitely often NPRG which takes $O(\log n \log\log^2 n)$ bits seeds, runs in time $\mathrm{poly}(n)$ and fools $\mathscr{C}$-circuits of size n.*

8.1 Technical Ingredients

Before proving Theorem 8.3, we collect some necessary technical ingredients. The first one is the NTIME hierarchy theorem for unary language.

Theorem 8.4 ([Žák83])**.** *There is a unary language $L \in \mathsf{NTIME}[2^n] \setminus \mathsf{NTIME}[2^n/n]$.*

The second one is the well-known Nisan-Wigderson construction of PRGs from hard functions.

[28] By nice, we mean the circuit class $\mathscr{C}$ is closed under projections and negations. Also, given k circuits $C_1, \dots, C_k \colon \{0,1\}^n \to \{-1,1\}$, the circuit $C(x) := \prod_{i=1}^k C_i(x)$ is also a $\mathscr{C}$-circuit of size $\mathrm{poly}(n,k) \cdot \sum_{i=1}^k \mathsf{size}(C_i)$. (*i.e.*, $\mathscr{C}$ is closed under XOR.)

Lemma 8.5 ([NW94]). *Let m, ℓ, a be integers such that $a \le \ell$, and $t = O(\ell^2 \cdot m^{1/a}/a)$. Let $\mathscr{C}$ be a nice circuit class. There is a function $G\colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ such that the following hold. For any function $Y\colon \{0,1\}^\ell \to \{0,1\}$ represented as a length-2^ℓ truth table, if Y cannot be $(1/2 + \varepsilon/m)$-approximated by $\mathscr{C} \circ \mathsf{Junta}_a$-circuits (where the top $\mathscr{C}$ circuit has size S), then $G(Y, \mathcal{U}_t)$ ε-fools every $\mathscr{C}$ circuit (of size S). That is, for any $\mathscr{C}$ circuit C (of size S),*

$$\left| \Pr_{\mathbf{s}\in\{0,1\}^t}[C(G(Y,\mathbf{s})) = 1] - \Pr_{\mathbf{x}\in\{0,1\}^m}[C(\mathbf{x}) = 1] \right| \le \varepsilon.$$

Moreover, the function G is computable in $\mathrm{poly}(m, 2^t)$ time.

The third one is the Nisan-Wigderson combinatorial design, which, in our language, is a restrictable generator for $\mathcal{C} \circ \mathsf{Junta}_{2^{\varepsilon n}}$-circuits. For a $\mathscr{C} \circ \mathsf{Junta}_a$ circuit C, we define its size as the number of bottom Junta_a gates.

Lemma 8.6 ([NW94]). *For every sufficiently small constant $\varepsilon > 0$, the following is true. For every $n \in \mathbb{N}_{\ge 1}$ and $k \le O(n)$, there is an integer $m = O(n)$ such that the following holds. Let $\mathcal{C}_1$ denote the class of $\mathscr{C} \circ \mathsf{Junta}_{2^{\varepsilon n}}$-circuits on n-bit inputs of size at most $2^{\varepsilon n}$, also let $\mathcal{C}_2$ denote the class of $\mathscr{C} \circ \mathsf{Junta}_{2^{\varepsilon n}}$-circuit on m-bit inputs of size at most $2^{\varepsilon n/2}$. There is a generator $\mathcal{G}\colon \{0,1\}^m \to \{0,1\}^{nk}$ that can be computed in $\mathrm{poly}(m)$ time and is $(\mathcal{C}_1 \cup \mathcal{C}_2)$-restrictable.*

The last one is the standard PRG construction for the general circuit class.

Lemma 8.7 ([Uma03]). *There is a universal constant $g \in (0,1)$ and a function $G\colon \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ such that, for every s and $Y\colon \{0,1\}^\ell \to \{0,1\}$, if Y cannot be computed by (general) circuits of size s^g, then $G(Y, \mathcal{U}_{g\ell})$ $1/s$-fools (general) circuits of size s. That is, for all circuits C of size at most s, it holds:*

$$\left| \Pr_{x\in\{0,1\}^{\ell g}}[C(Y,x) = 1] - \Pr_{x\sim\{0,1\}^s}[C(x) = 1] \right| \le \frac{1}{s}.$$

Furthermore, G is computable in $\mathrm{poly}(|Y|)$ time.

8.2 The Construction of NPRG

Now we prove Theorem 8.3. The proof is similar to the proof of [CLW20, Theorem 7.1]. We let $\mathcal{A}_{\mathsf{Zak}}$ be the $\mathsf{NTIME}[2^n]$ algorithm computing the unary language of Theorem 8.4. Let $\alpha > 0$ be a small enough constant. We design another algorithm $\mathcal{A}_{\mathsf{fast}}$ trying to speed up $\mathcal{A}_{\mathsf{Zak}}$ as follows:

1. $\mathcal{A}_{\mathsf{fast}}$ rejects all non-unary inputs. On input $z = 1^n$, $\mathcal{A}_{\mathsf{fast}}$ applies PCP from Lemma 5.5 first, obtaining an oracle circuit VPCP_{1^n}. For brevity, we just write it as $\mathsf{VPCP}_{(n)}$ from now on. Recall that $\mathsf{VPCP}_{(n)}$ and its oracle take input of length $\ell = \ell(n) = n + O(\log n)$.

2. Then $\mathcal{A}_{\mathsf{fast}}$ guesses a $2^{\alpha\ell}$-size general circuit $C\colon \{0,1\}^\ell \to \{-1,1\}$. Feeding C into $\mathsf{VPCP}_{(n)}$, $\mathcal{A}_{\mathsf{fast}}$ obtains a circuit $\mathsf{VPCP}^C_{(n)}$.

3. $\mathcal{A}_{\mathsf{fast}}$ applies the PCPP from Lemma 5.6, and gets a 2SAT instance over $m = 2^{O(\alpha\ell)}$ clauses. It then guesses a $\mathsf{Sum} \circ \mathscr{C} \circ \mathsf{Junta}_{2^{\alpha\ell}}$-circuit $H\colon \{0,1\}^{\log m + \ell + 1} \to \mathbb{R}$ with complexity$(H) \le 2^{\alpha\ell}$ as the proof for PCPP. $\mathcal{A}_{\mathsf{fast}}$ runs the tests that are similar to the tests used by $\mathcal{A}_{\mathsf{cheat}}$ in proof of Theorem 5.2 ((30)-(32)) and accepts if and only if all tests are passed.

Applying the assumed CAPP algorithm, for a sufficiently small $\alpha > 0$, it holds that $\mathcal{A}_{\mathsf{fast}} \in \mathsf{NTIME}[2^n/n]$.[29] Also, we can show that $\mathcal{A}_{\mathsf{fast}}$ only makes one-sided error as we had done in Lemma 5.9. Therefore, for infinitely many n, we have $\mathcal{A}_{\mathsf{fast}}(1^n) = 0$ and $\mathcal{A}_{\mathsf{Zak}}(1^n) = 1$. Let $\mathcal{S}$ be the set of all such positive integers n: $\mathcal{S} = \{n \in \mathbb{N}_{\geq 1} : \mathcal{A}_{\mathsf{fast}}(1^n) = 0 \text{ and } \mathcal{A}_{\mathsf{Zak}}(1^n) = 1\}$. Depending on whether there is a succinct circuit for $\mathsf{VPCP}_{(n)}$ as a correct oracle (meaning that $\mathsf{VPCP}^C_{(n)}(r) = 1$ for all r), we use two different constructions.

Case 1. There are infinitely many $n \in \mathcal{S}$ such that, there is no circuit C of size $2^{\alpha\ell}$ such that $\mathsf{VPCP}^C_{(n)}$ is a tautology.

In the following we only consider these n, as there are infinitely many of them. Then we use nondeterminism to find an oracle $\mathcal{O}\colon \{0,1\}^\ell \to \{-1,1\}$ such that $\mathsf{VPCP}^{\mathcal{O}}_{(n)}$ is a tautology in $2^{O(\ell)} = 2^{O(n)}$ time. It follows that for infinitely many n, $\mathcal{O}$ cannot be computed by $2^{o(\ell)}$-size (general) circuits. Then we can apply the standard PRG construction of Lemma 8.7.

Case 2. For all but finitely many $n \in \mathcal{S}$, there exists a $2^{\alpha\ell}$-size circuit C such that $\mathsf{VPCP}^C_{(n)}$ is a tautology. Again we only consider these n as $|\mathcal{S}|$ is infinite.

On these n, it must be the case that $\mathcal{A}_{\mathsf{fast}}$ does not accept any $\mathsf{Sum} \circ \mathscr{C} \circ \mathsf{Junta}_{2^{\alpha\ell}}$-circuit with complexity bounded by $2^{\alpha\ell}$. Hence, the correct proof $H\colon \{0,1\}^{O(\ell)} \to \{-1,1\}$ for the PCPP of $\mathsf{VPCP}^C_{(n)}$ cannot be approximated by any $\mathsf{Sum} \circ \mathscr{C} \circ \mathsf{Junta}_{2^{\alpha\ell}}$-circuit with small complexity (*i.e.*, less than $2^{\alpha\ell}$).

We can then apply the derandomized XOR Lemma to H with the restrictable generator of Lemma 8.6, and obtain a function g with input length $O(\ell \log \ell)$ which cannot be $\left(\frac{1}{2} + 2^{-\gamma\ell}\right)$-approximated by $\mathscr{C} \circ \mathsf{Junta}_{2^{\alpha\ell}}$-functions of size at most $2^{\gamma\ell}$, where $\gamma > 0$ is a sufficiently small constant. Finally, we use the function g to apply Lemma 4.14, and obtain a PRG which produces m pseudorandom bits that can fool $\mathscr{C}$-circuits of size at most $m = 2^{\Omega(\gamma\ell)}$ with seed length being $O((\ell^2 \log^2 \ell)/(\alpha\ell)) = O(\ell \log^2 \ell) = O(\log m \log\log^2 m)$.

9 Open Problems and Discussions

There are several interesting open questions stemmed from our work, we highlight some of them below.

1. One interesting open question is whether we can improve the seed length of our new derandomized XOR lemma to be $O(n)$, even for $\varepsilon = 2^{-\Omega(n)}$. Using the Nisan-Zuckerman PRG, we managed to get the optimal seed length $O(n)$ when $\varepsilon \geq 2^{-n^{1-\Omega(1)}}$ (the moreover part of Lemma 4.1).

   To derandomize the proof of Lemma 4.2, we observe that one can also apply a low-error PRG for *combinatorial rectangles* (see the proof of Lemma 4.16). However, directly applying the best PRGs for combinatorial rectangles does not seem to improve our seed length here.

2. As we already discussed in Section 1, the degree-to-error trade off in Theorem 1.3 does not match Theorem 1.1 when $d = n^{0.49}$. We believe the following conjecture can be proved.

[29] Our proofs in Section 5 assumed the strong #SAT algorithms, but one can observe that CAPP algorithms also suffices. Similarly arguments can be found in [CLW20, Section 6].

Conjecture 9.1. *For every $\beta \in (0,1)$, there is an E^{NP} function f such that, for every sufficiently large n, it holds that* $\mathrm{corr}(f, n^{\beta}) \le \exp(-\Omega(n^{(1-\beta)} / \log n))$.

We believe such a better trade for $\beta > 1/2$ can be achieved by a better structured PCP.

For $\beta < 1/2$, the bottleneck becomes our restriction generator Lemma 6.3. In this case, we hope to prove correlation bounds of $2^{-n^{1-\beta}}$, and thus have to set $k = n^{1-\beta} > \sqrt{n}$, but Lemma 6.3 only works when $k \le \sqrt{n}$.

The generator in Lemma 6.3 appears to be optimal for the particular function collection we considered (*i.e.*, $\mathcal{F}^{\mathsf{poly}}$), so one probably has to consider a different super class $\mathcal{H}$ of n^{β}-degree $\mathbb{F}_2$-polynomials which admits a better restrictable-generator. Moreover, in order to prove a $2^{-n^{1-\beta}}$ correlation bound, our method has to solve CAPP for a linear sum of $2^{O(n^{1-\beta})}$ $\mathcal{H}$ functions, meaning that CAPP for a single function in $\mathcal{H}$ should admit an algorithm with running time at most $2^{n-n^{1-\beta}}$. This is a stringent restriction on $\mathcal{H}$ and we do not have a candidate function collection yet.

3. Can we also improve the rank-error-tradeoff in Theorem 1.5? We believe it can also be strengthened as follows.

   Conjecture 9.2. *For every constant $\varepsilon \in (0,1)$, there is a P^{NP} algorithm which on input 1^n outputs an $n \times n$ $\mathbb{F}_2$-matrix H_n satisfying $\mathcal{R}_{H_n}(2^{\log^{1-\varepsilon} n}) \ge (1/2 - \exp(-\log^{\varepsilon} n)) \cdot n^2$, for every sufficiently large n.*

   One potential approach to the conjecture above is to design a better restriction generator for low-rank matrices (*i.e.*, improve the parameters in Lemma 7.5).

Acknowledgment

We would like to thank Roei Tell and R. Ryan Williams for many helpful discussions during the project. We are greatful to Emanuele Viola for helpful comments on a draft of this paper. We would like to thank Zhenjian Lu for sharing a manuscript of [Lu20] to us and for comments on an early draft of this paper. We are also grateful to an anonymous reviewer for pointing out an issue in the proof of Theorem 1.5 in an early draft of this paper.

The first author is also grateful to William M. Hoza for pointing out that the Nisan-Zuckerman PRG can be used to improve the seed length in our applications. The second author wants to thank R. Ryan Williams for hosting him during the spring and the summer of 2020 in MIT.

Lijie Chen is supported by an IBM Fellowship.

References

[AC19] Josh Alman and Lijie Chen. *Efficient construction of rigid matrices using an NP oracle*. In *Proc. 60th FOCS*, pages 1034–1055. IEEE Comp. Soc., 2019.

[BCG20] Mark Braverman, Gil Cohen, and Sumegha Garg. *Pseudorandom pseudo-distributions with near-optimal error for read-once branching programs*. SIAM J. Comput., 49(5), 2020.

[BGHSV06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. *Robust PCPs of proximity, shorter PCPs, and applications to coding*. SIAM J. Comput., 36(4):889–974, 2006.

[BHK09] Boaz Barak, Moritz Hardt, and Satyen Kale. *The uniform hardcore lemma via approximate bregman projections*. In *Proc. 20th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'09)*, pages 1193–1200. ACM Press, 2009.

[BHPT20] Amey Bhangale, Prahladh Harsha, Orr Paradise, and Avishay Tal. *Rigid matrices from rectangular PCPs*. In *Proc. 61st FOCS*. IEEE Comp. Soc., 2020.

[BNS92] László Babai, Noam Nisan, and Mario Szegedy. *Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs*. J. Comput. System Sci., 45(2):204–232, 1992.

[Bou05] Jean Bourgain. *Estimation of certain exponential sums arising in complexity theory*. Comptes Rendus Mathematique, 340(9):627–631, 2005.

[BT94] Richard Beigel and Jun Tarui. *On* ACC. Comput. Complex., 4:350–366, 1994.

[BV14] Eli Ben-Sasson and Emanuele Viola. *Short PCPs with projection queries*. In *Proc. 41st Internat. Colloq. on Automata, Languages and Programming (ICALP'14)*, pages 163–173. Springer, 2014.

[CGLLS20] Eshan Chattopadhyay, Jason Gaitonde, Chin Ho Lee, Shachar Lovett, and Abhishek Shetty. *Fractional pseudorandom generators from any fourier level*. Electron. Colloquium Comput. Complex., 27:121, 2020.

[CHHLZ20] Eshan Chattopadhyay, Pooya Hatami, Kaave Hosseini, Shachar Lovett, and David Zuckerman. *XOR lemmas for resilient functions against polynomials*. In *Proc. 52nd STOC*, pages 234–246. ACM Press, 2020.

[CHLT19] Eshan Chattopadhyay, Pooya Hatami, Shachar Lovett, and Avishay Tal. *Pseudorandom generators from the second fourier level and applications to AC0 with parity gates*. In *Proc. 10th Innovations in Theoretical Computer Science Conf. (ITCS'19)*, pages 22:1–22:15. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019.

[CL20] Eshan Chattopadhyay and Jyun-Jie Liao. *Optimal error pseudodistributions for read-once branching programs*. In *Proc. 35th Conf. Computational Complexity (CCC'20)*, pages 25:1–25:27. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020.

[CLW20] Lijie Chen, Xin Lyu, and Ryan Williams. *Almost-everywhere circuit lower bounds from non-trivial derandomization*. In *Proc. 61st FOCS*. IEEE Comp. Soc., 2020.

[CR20] Lijie Chen and Hanlin Ren. *Strong average-case lower bounds from non-trivial derandomization*. In *Proc. 52nd STOC*, pages 1327–1334. ACM Press, 2020.

[CW16] Timothy M. Chan and Ryan Williams. *Deterministic APSP, orthogonal vectors, and more: Quickly derandomizing Razborov-Smolensky*. In *Proc. 27th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'16)*, pages 1246–1255. ACM Press, 2016.

[CW19] LIJIE CHEN and R. RYAN WILLIAMS. *Stronger connections between circuit analysis and circuit lower bounds, via PCPs of proximity*. In *Proc. 34th Conf. Computational Complexity (CCC'19)*, pages 19:1–19:43. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2019.

[FK18] MICHAEL A. FORBES and ZANDER KELLEY. *Pseudorandom generators for read-once branching programs, in any order*. In *Proc. 59th FOCS*, pages 946–955. IEEE Comp. Soc., 2018.

[FS17] LANCE FORTNOW and RAHUL SANTHANAM. *Robust simulations and significant separations*. Inf. Comput., 256:149–159, 2017.

[GKW18] ALEXANDER GOLOVNEV, ALEXANDER S. KULIKOV, and R. RYAN WILLIAMS. *Circuit depth reductions*. Electron. Colloquium Comput. Complex., 25:192, 2018.

[GNW11] ODED GOLDREICH, NOAM NISAN, and AVI WIGDERSON. *On Yao's XOR-lemma*. In ODED GOLDREICH, ed., *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, volume 6650 of *Lecture Notes in Computer Science*, pages 273–301. Springer, 2011.

[GRS05] FREDERIC GREEN, AMITABHA ROY, and HOWARD STRAUBING. *Bounds on an exponential sum arising in boolean circuit complexity*. Comptes Rendus Mathematique, 341(5):279–282, 2005.

[Hås89] JOHAN HÅSTAD. *Almost optimal lower bounds for small depth circuits*. Adv. Comput. Res., 5:143–170, 1989.

[HLV18] ELAD HARAMATY, CHIN HO LEE, and EMANUELE VIOLA. *Bounded independence plus noise fools products*. SIAM J. Comput., 47(2):493–523, 2018.

[Hol05] THOMAS HOLENSTEIN. *Key agreement from weak bit agreement*. In *Proc. 37th STOC*, pages 664–673. ACM, 2005.

[HV20] XUANGUI HUANG and EMANUELE VIOLA. *Average-case rigidity lower bounds*. Electron. Colloquium Comput. Complex., 26:175, 2020.

[HVV06] ALEXANDER HEALY, SALIL P. VADHAN, and EMANUELE VIOLA. *Using nondeterminism to amplify hardness*. SIAM J. Comput., 35(4):903–931, 2006.

[Imp95] RUSSELL IMPAGLIAZZO. *Hard-core distributions for somewhat hard problems*. In *Proc. 36th FOCS*, pages 538–545. IEEE Comp. Soc., 1995.

[IW97] RUSSELL IMPAGLIAZZO and AVI WIGDERSON. $\mathsf{P} = \mathsf{BPP}$ *if* E *requires exponential circuits: Derandomizing the XOR lemma*. In *Proc. 29th STOC*, pages 220–229. ACM Press, 1997.

[KS18] SWASTIK KOPPARTY and SRIKANTH SRINIVASAN. *Certifying polynomials for* $\mathsf{AC}^0[\oplus]$ *circuits, with applications to lower bounds and circuit compression*. Theory Comput., 14(1):1–24, 2018.

[Lev87] LEONID A. LEVIN. *One-way functions and pseudorandom generators*. Combinatorica, 7(4):357–363, 1987.

[Lok09] Satyanarayana V. Lokam. *Complexity lower bounds using linear algebra*. Foundations and Trends in Theoretical Computer Science, 4(1-2):1–155, 2009.

[LSSTV19] Nutan Limaye, Karteek Sreenivasaiah, Srikanth Srinivasan, Utkarsh Tripathi, and S. Venkitesh. *A fixed-depth size-hierarchy theorem for* $\mathrm{AC}^0[\oplus]$ *via the coin problem*. In *Proc. 51st STOC*, pages 442–453. ACM Press, 2019.

[Lu20] Zhenjian Lu. *Personal communication*, 2020.

[LV17] Chin Ho Lee and Emanuele Viola. *More on bounded independence plus noise: Pseudorandom generators for read-once polynomials*. Electron. Colloquium Comput. Complex., 24:167, 2017.

[Nis92] Noam Nisan. *Pseudorandom generators for space-bounded computation*. Combinatorica, 12(4):449–461, 1992.

[NW94] Noam Nisan and Avi Wigderson. *Hardness vs randomness*. J. Comput. System Sci., 49(2):149–167, 1994.

[NZ96] Noam Nisan and David Zuckerman. *Randomness is linear in space*. J. Comput. System Sci., 52(1):43–52, 1996.

[OSS19] Igor Carboni Oliveira, Rahul Santhanam, and Srikanth Srinivasan. *Parity helps to compute majority*. In *Proc. 34th Conf. Computational Complexity (CCC'19)*, volume 137, pages 23:1–23:17. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019.

[Raz87] Alexander A. Razborov. *Lower bounds on the size of bounded depth circuits over a complete basis with logical addition*. Mathematical Notes of the Academy of Sciences of the USSR, 41(4):333–338, 1987.

[Raz89] ———. *On rigid matrices (in Russian)*, 1989. Steklov Mathematical Institute.

[RSS18] Ninad Rajgopal, Rahul Santhanam, and Srikanth Srinivasan. *Deterministically counting satisfying assignments for constant-depth circuits with parity gates, with implications for lower bounds*. In *Proc. 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS'18)*, volume 117, pages 78:1–78:15. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018.

[RTTV08] Omer Reingold, Luca Trevisan, Madhur Tulsiani, and Salil P. Vadhan. *Dense subsets of pseudorandom sets*. In *Proc. 49th FOCS*, pages 76–85. IEEE Comp. Soc., 2008.

[Smo87] Roman Smolensky. *Algebraic methods in the theory of lower bounds for boolean circuit complexity*. In *Proc. 19th STOC*, pages 77–82. ACM Press, 1987.

[Smo93] ———. *On representations by low-degree polynomials*. In *34th Annual Symposium on Foundations of Computer Science, Palo Alto, California, USA, 3-5 November 1993*, pages 130–138. IEEE Computer Society, 1993.

[STV01] Madhu Sudan, Luca Trevisan, and Salil P. Vadhan. *Pseudorandom generators without the XOR lemma*. J. Comput. Syst. Sci., 62(2):236–266, 2001.

[TTV09] Luca Trevisan, Madhur Tulsiani, and Salil P. Vadhan. *Regularity, boosting, and efficiently simulating every high-entropy distribution*. In *Proc. 24th Conf. Computational Complexity (CCC'09)*, pages 126–136. IEEE Comp. Soc., 2009.

[Uma03] Christopher Umans. *Pseudo-random generators for all hardnesses*. J. Comput. System Sci., 67(2):419–440, 2003.

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

[Vio09] Emanuele Viola. *Guest column: correlation bounds for polynomials over $\{0,1\}$*. SIGACT News, 40(1):27–44, 2009.

[Vio20a] ———. *Matching smolensky's correlation bound with majority*. Electron. Colloquium Comput. Complex., 20:193, 2020.

[Vio20b] ———. *New lower bounds for probabilistic degree and AC0 with parity gates*. Electron. Colloquium Comput. Complex., 27:15, 2020.

[VW08] Emanuele Viola and Avi Wigderson. *Norms, XOR lemmas, and lower bounds for polynomials and protocols*. Theory of Computing, 4(1):137–168, 2008.

[VW20] Nikhil Vyas and R. Ryan Williams. *Lower bounds against sparse symmetric functions of ACC circuits: Expanding the reach of #SAT algorithms*. In *Proc. 37th International Symposium on Theoretical Aspects of Computer Science (STACS'20)*, volume 154, pages 59:1–59:17. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020.

[Wil13] Ryan Williams. *Improving exhaustive search implies superpolynomial lower bounds*. SIAM J. Comput., 42(3):1218–1244, 2013.

[Wil14] ———. *Nonuniform ACC circuit lower bounds*. J. ACM, 61(1):2, 2014.

[Wil18] Richard Ryan Williams. *Limits on representing boolean functions by linear combinations of simple functions: Thresholds, ReLUs, and low-degree polynomials*. In *Proc. 33th Conf. Computational Complexity (CCC'18)*, pages 6:1–6:24. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018.

[Yao82] Andrew Chi-Chih Yao. *Theory and applications of trapdoor functions (extended abstract)*. In *Proc. 23rd FOCS*, pages 80–91. IEEE Comp. Soc., 1982.

[Yao90] ———. *On ACC and threshold circuits*. In *Proc. 31st FOCS*, pages 619–627. IEEE Comp. Soc., 1990.

[Žák83] Stanislav Žák. *A Turing machine time hierarchy*. Theoretical Computer Science, 26(3):327–333, 1983.

A Connections to Depth-3 Circuits Lower Bound

In this section, we prove Theorem 1.2 (restated below). For ease of presentation, we will assume that Boolean functions have image $\{0,1\}$ instead of $\{-1,1\}$ throughout this section.

Reminder of Theorem 1.2. *For any function $d(n)\colon \mathbb{N} \to \mathbb{N}$, if there is a function f in E^{NP} such that $\mathrm{corr}(f_n, d(n)) \le 2^{-d(n)}$ for infinitely many $n \in \mathbb{N}$, then there is a function g in E^{NP} that does not admit depth-3 AC^0-circuits of size at most $2^{o(d(n))}$.*

In the following, we will use AC^0_3 to denote depth-3 AC^0-circuits.

We will need ε-biased sets in this section. We first recall its definition below.

Definition A.1. *A set $S \subseteq \{0,1\}^n$ is called ε-biased if, for all non-zero $x \in \{0,1\}^n$, it holds that*

$$\left| \mathop{\mathbb{E}}_{y \leftarrow S} (-1)^{\langle y,x \rangle} \right| \le \varepsilon.$$

It can be shown by the probabilistic method that ε-biased sets of size $\mathrm{poly}(n, 1/\varepsilon)$ exists.

Theorem A.2. *For every $n \in \mathbb{N}_{\ge 1}$ and $\varepsilon \in (0,1)$, there is an ε-biased set $S \subseteq \{0,1\}^n$ of size $O(n/\varepsilon^2)$.*

First, we show that every AC^0_3-circuit with bounded bottom fan-in can be slightly approximated by $\mathbb{F}_2$-polynomials.

Theorem A.3. *For every $n, \tau \in \mathbb{N}_{\ge 1}$ such that $\tau \ge \log n$, every AC^0_3-circuit $C\colon \{0,1\}^n \to \{0,1\}$ of size at most 2^τ and with bottom fan-in at most τ can be $\left(1/2 + 2^{-O(\tau)}\right)$-approximated by $\mathbb{F}_2$-polynomials of degree τ.*

Proof. We can assume without loss of generality that C is an $\mathsf{AND} \circ \mathsf{OR} \circ \mathsf{AND}$-circuit. If not, we can proceed with the negation of C. We will show that such a circuit C is equivalent to a $\mathsf{MAJ} \circ \mathsf{XOR} \circ \mathsf{AND}$-circuit D with top gate fan-in bounded by $2^{O(\tau)}$ and bottom fan-in bounded by τ.

First, we write C as $C = \bigwedge_{i=1}^{m_1} C_i$ where C_i is the enumeration of medium layer $\mathsf{OR} \circ \mathsf{AND}$ sub-circuits. We additionally write C_i as $C_i = \bigvee_{i=1}^{m_2} C_{i,j}$, here we assume each of C_i has the same fan-in $m_2 \le 2^\tau$. The general case can be handled similarly.

We set $\varepsilon = 2^{-3\tau}$ and choose an ε-biased set $S \subseteq \{0,1\}^{m_2}$ such that $|S| \le 2^{O(\tau)}$, given by Theorem A.2. Then for each C_i, we consider the formula

$$F_i(x) := \sum_{y \in S} \bigoplus_{j : y_j = 1} C_{i,j}(x).$$

We have the following observations:

1. If $C_i(x) = 0$ (that is, none of $C_{i,j}(x)$ evaluates to 1), then $F_i(x) = 0$.
2. If $C_i(x) = 1$ (that is, at least one of $C_{i,j}(x)$ evaluates to 1), then it follows from the definition of ε-biased sets that
$$F_i(x) \in \left[|S| \cdot \frac{1-\varepsilon}{2}, |S| \cdot \frac{1+\varepsilon}{2} \right].$$

We then consider the formula

$$F(x) := \sum_{i=1}^{m_1} F_i(x).$$

The following hold:

1. If $C(x) = 1$ (that is, none of $C_i(x)$ evaluates to 0), then $F(x) \geq |S| \cdot m_1 \cdot \frac{1-\varepsilon}{2}$.
2. If $C(x) = 0$ (that is, at least one of $C_i(x)$ evaluates to 0), then $F(x) \leq |S| \cdot (m_1 - 1) \cdot \frac{1+\varepsilon}{2}$.

By the choice of ε, we know $\varepsilon \leq m_1^{-3}$ and hence $|S| \cdot m_1 \cdot \frac{1-\varepsilon}{2} > |S| \cdot (m_1 - 1) \cdot \frac{1+\varepsilon}{2}$. Therefore, we can set a threshold T such that $C(x)$ can be computed by checking whether $F(x) \geq T$. Observe that $F(x)$ is a (unweighted) sum of at most $2^{O(\tau)}$ $\mathsf{XOR} \circ \mathsf{AND}$-circuits, where the bottom fan-in of AND gates are at most τ. We can then write $\mathbb{1}_{F(x) \geq T}$ as a $\mathsf{MAJ} \circ \mathsf{XOR} \circ \mathsf{AND}$-circuit, denoted as D.

D can be seen as a majority over at most $2^{O(\tau)}$ $\mathbb{F}_2$-polynomials of degree τ. Therefore, at least one of these polynomials agrees with $C(x)$ on more than a $1/2 + 2^{-O(\tau)}$-fraction of inputs. This completes the proof. □

Remark A.4. *One might wonder whether it is possible to use the original polynomial method of Razborov-Smolensky to prove Theorem A.3 directly. In fact, from that it can be showed that for every size-s depth-3 AC^0 circuit C, there is a probabilistic polynomial $\mathcal{P}$ of degree $O(\log^2 s)$ that computes C with a constant error. However, for our setting (i.e., $s \geq 2^{\Omega(\sqrt{n})}$), this bound becomes trivial as $\log^2 s \geq n$, so it is not enough for showing Theorem A.3.*

We also observe that the restriction on the bottom fan-in in Theorem A.3 is not essential.

Lemma A.5. *For any function $b(n)\colon \mathbb{N} \to \mathbb{N}$, suppose there is a function $f\colon \{0,1\}^n \to \{0,1\}$ which cannot be computed by AC^0_3-circuits of size at most $2^{b(n)}$ and with bottom fan-in bounded by $b(n)$. Then the function $g\colon \{0,1\}^{2n} \to \{0,1\}$ defined by $g(x,y) := f(x_1 \oplus y_1, \ldots, x_n \oplus y_n)$ for every $x, y \in \{0,1\}^n$ cannot be computed by AC^0_3-circuits of size at most $2^{o(b(n))}$.*

Proof Sketch. Suppose on the contrary that g can be computed by an AC^0_3-circuit C of size at most $s(n) = 2^{o(b(n))}$. We consider a random restriction $\rho \leftarrow \mathcal{D}$ specified as follows: for every $i \in [n]$, we randomly select $v_i \in \{x_i, y_i\}$ and fix v_i to a uniform bit in $\{0,1\}$. After the restriction, each bottom gate in C of fan-in larger than $b(n)$ is killed with probability $2^{-\Omega(b(n))}$.

Given that $s(n) = 2^{o(b(n))}$, by a union bound, we know that there exists a restriction $\rho \in \mathcal{D}$ such that $g\big|_\rho$ can be computed by an AC^0_3-circuit of size at most $2^{b(n)}$ with bottom fan-in bounded by $b(n)$. Also, note that for every $\rho \in \mathcal{D}$, the restricted function $g\big|_\rho$ is equivalent to $f(x \oplus w_\rho)$ for some $w_\rho \in \{0,1\}^n$. This is a contradiction and we are done. □

Finally, we sketch the proof of Theorem 1.2.

Proof of Theorem 1.2. By Theorem A.3, f being inapproximable by low-degree $\mathbb{F}_2$-polynomials implies that f cannot be computed by small AC^0_3-circuits with bounded bottom fan-in. We apply the gadget of Lemma A.5 to construct from f a function g that cannot be computed by small AC^0_3-circuits. □

B Missing Proofs in Section 5

B.1 A Useful Lemma

Before we proceed, we state and prove the following useful lemma. It will be used frequently in this section and Appendix C.

Lemma B.1. *Let $\mathcal{S}$ be a set. Let $P\colon \mathcal{S} \times \mathbb{R}^d \to \mathbb{R}$ be such that, for every $x \in \mathcal{S}$, $P(x, \star)$ is a multi-linear polynomial with absolute values of coefficients bounded by M. Let $f_1, \ldots, f_d\colon \mathcal{S} \to \mathbb{R}$ and $g_1, \ldots, g_d\colon \mathcal{S} \to \mathbb{R}$ be two list of functions. Then, it holds that*

$$\begin{aligned}&\mathop{\mathbb{E}}_{x \leftarrow \mathcal{S}} |P(x, f_1(x), \ldots, f_d(x)) - P(x, g_1(x), \ldots, g_d(x))| \\ &\le M \cdot \prod_{j=1}^{d} \left(\|g_j\|_{2(d-1)} + \|f_j\|_{2(d-1)} + 1 \right) \cdot \sum_{i=1}^{d} \|f_i - g_i\|_2 .\end{aligned}$$

To prove Lemma B.1, the following inequality will be used. It can be proved by iteratively applying Hölder's inequality.

Claim B.2. *Let $h_1, \ldots, h_k\colon \mathcal{S} \to \mathbb{R}$ be k functions, it holds that $\left\| \prod_{i=1}^{k} h_i \right\|_2 \le \prod_{i=1}^{k} \|h_i\|_{2k}$.*

Proof. Use induction on k. For $k = 1$, the statement is trivial. Assuming it is true for $k-1$, we have

$$\begin{aligned}\left\| \prod_{i=1}^{k} h_i \right\|_2 &\le \left\| \prod_{i=1}^{k-1} h_i \right\|_{2k/(k-1)} \cdot \|h_k\|_{2k} && \text{(Hölder's inequality with } (p,q) = (2k/(k-1), 2k)) \\ &\le \prod_{i=1}^{k} \|h_i\|_{2k}. && \text{(induction hypothesis)}\end{aligned}$$

□

Then we prove Lemma B.1.

Proof of Lemma B.1. We write

$$P(x, v_1, \ldots, v_d) = \sum_{T \subseteq [d]} \alpha_{x,T} \prod_{i \in T} v_i,$$

where $|\alpha_{x,T}| \le M$ for each $T \subseteq [d]$. Fixing an $T \subseteq [d]$, we consider

$$\mathop{\mathbb{E}}_{x \leftarrow \mathcal{S}} |\alpha_{x,T}| \cdot \left| \prod_{i \in T} f_i(x) - \prod_{i \in T} g_i(x) \right| .$$

It follows that

$$\begin{aligned}&\mathop{\mathbb{E}}_{x \leftarrow \mathcal{S}} |\alpha_{x,T}| \left| \prod_{i \in T} f_i(x) - \prod_{i \in T} g_i(x) \right| \\ &\le M \cdot \mathop{\mathbb{E}}_{x \leftarrow \mathcal{S}} \left| \sum_{i \in T} \left[(f_i(x) - g_i(x)) \left(\prod_{j: j \in T, j < i} f_j(x) \prod_{j: j \in T, j > i} g_j(x) \right) \right] \right| \\ &\le M \cdot \sum_{i \in T} \left\langle f_i - g_i, \prod_{j: j \in T, j < i} f_j \prod_{j: j \in T, j > i} g_j \right\rangle \\ &\le M \cdot \sum_{i \in T} \|f_i - g_i\|_2 \cdot \left\| \prod_{j: j \in T, j < i} f_j \prod_{j: j \in T, j > i} g_j \right\|_2 && \text{(By Cauchy-Schwartz)} \\ &\le M \cdot \sum_{i \in T} \|f_i - g_i\|_2 \left[\prod_{j \in T, j < i} \|f_j\|_{2(|T|-1)} \prod_{j \in T, j > i} \|g_j\|_{2(|T|-1)} \right] && \text{(By Claim B.2)} \\ &\le M \cdot \sum_{i \in T} \|f_i - g_i\|_2 \left[\prod_{j \in T, j < i} \|f_j\|_{2(d-1)} \prod_{j \in T, j > i} \|g_j\|_{2(d-1)} \right] . && \text{(By } \|f\|_{2(|T|-1)} \le \|f\|_{2(d-1)})\end{aligned}$$

Finally, taking a summation over $T \subseteq [d]$ proves the lemma.

□

B.2 $\mathcal{A}_{\mathsf{cheat}}$ Makes Only One-sided Error

In this subsection, we verify Lemma 5.9.

Reminder of Lemma 5.9. *For every small enough constants $\alpha, \delta \in (0,1)$ and for every sufficiently large constant $K \geq 1$, the following holds: $\mathcal{A}_{\mathsf{cheat}}(z) \leq \mathcal{A}^{T}_{\mathsf{FS}}(z)$ for all but finitely many inputs z.*

Proof of Lemma 5.9. Suppose that $\mathcal{A}^{T}_{\mathsf{FS}}(z) = 0$. We consider the execution of $\mathcal{A}_{\mathsf{cheat}}(z)$. Let $\ell = \log T(n) + O(\log\log T(n))$. By Item 2 of Claim 5.7, for every circuit $C\colon \{0,1\}^{\ell} \to \{-1,1\}$ fed to VPCP_z, the circuit $\mathsf{VPCP}^{C}_{z}\colon \{0,1\}^{\ell} \to \{0,1\}$ evaluates to 1 on at most $2^{\ell}/\mathrm{poly}(n)$ many inputs. Recall in the verification of $\mathcal{A}_{\mathsf{cheat}}$, it applies the following tests(rewriting (30) - (32)):

$$\mathop{\mathbb{E}}_{i,j\in[m]\times[2]} \mathop{\mathbb{E}}_{x\leftarrow U_\ell} P_{i,j}(x) \leq \delta, \tag{39}$$

$$\mathop{\mathbb{E}}_{i,j\in[m]\times[2]} \mathop{\mathbb{E}}_{x\leftarrow U_\ell} H(i,j,x)^2 \leq 1, \tag{40}$$

$$\mathop{\mathbb{E}}_{i\in[m]} \mathop{\mathbb{E}}_{x\leftarrow U_\ell} F_i(x) \geq c_{\mathsf{pcpp}} - \frac{1}{2}(c_{\mathsf{pcpp}} - s_{\mathsf{pcpp}}), \tag{41}$$

where P_{ij} is defined as

$$P_{ij}(x) = \begin{cases} (1+T_{ij}(x))^2(1-T_{ij}(x))^2, & \text{if } T_{ij} \in Z, \\ (\mathsf{Enc}_s(x) - T_{ij}(x))^2, & \text{if } T_{ij} \in Y. \end{cases}$$

Now suppose that there is a $\mathsf{Sum} \circ \mathcal{F}$ function $H(i,j,x)$ which can pass the tests above. We will derive a contradiction.

Let (Y, Z) be the real-valued proof constructed by $\mathcal{A}_{\mathsf{cheat}}$. We define from (Y, Z) a list of Boolean valued proof circuits $(\widehat{Y}, \widehat{Z})$ as follows. First, let $\widehat{Y}_i := \mathsf{Enc}_i(x)$. For each Z_i, let $\widehat{Z}_i(x) := \mathrm{sign}(Z_i(x))$. We can then analogously define $\widehat{T}_{ij} \in (\widehat{Y}, \widehat{Z})$ and $\widehat{F}_i(x) := \widetilde{\mathsf{Cons}}_i(\widehat{T}_{i,1}, \widehat{T}_{i,2})$. Note that since $\mathcal{A}^{T}_{\mathsf{FS}}(z) = 0$, we have

$$\mathop{\mathbb{E}}_{i\in[m]} \mathop{\mathbb{E}}_{x\leftarrow U_\ell} \widehat{F}_i(x) \leq (1 - 1/\mathrm{poly}(n)) \cdot s_{\mathsf{pcpp}} + c_{\mathsf{pcpp}}/\mathrm{poly}(n) \leq c_{\mathsf{pcpp}} - \frac{9}{10} \cdot (c_{\mathsf{pcpp}} - s_{\mathsf{pcpp}}) \tag{42}$$

by properties of PCP and PCPP. Also note from (41) that $\mathbb{E}_{i\in[m]} \mathbb{E}_{x\leftarrow U_\ell} F_i(x)$ is bounded below by $\frac{c_{\mathsf{pcpp}}+s_{\mathsf{pcpp}}}{2}$.

In the following, we further bound $\mathbb{E}_{i\in[m]} \mathbb{E}_{x\leftarrow U_\ell}[\widehat{F}_i(x) - F_i(x)]$, which will lead to a contradiction.

We will apply Lemma B.1. We let $\mathcal{S}$ be $[m] \times \{0,1\}^{\ell}$, and define a function $P_F\colon [m] \times \{0,1\}^{\ell} \times \mathbb{R}^2 \to \mathbb{R}$ as $P_F(i, x, v_1, v_2) := \widetilde{\mathsf{Cons}}_i(v_1, v_2)$. For $j \in \{1,2\}$, we set function $f_j\colon [m] \times \{0,1\}^{\ell} \to \mathbb{R}$ as $f_j(i,x) := T_{ij}(x)$, and function g_j as $g_j(i,x) := \widehat{T}_{ij}(x)$.

Applying Lemma B.1 with the function P_F, the set $\mathcal{S}$ and two list of functions (f_1, f_2), (g_1, g_2), it follows that[30]

$$\begin{aligned}&\mathop{\mathbb{E}}_{x \leftarrow \mathcal{S}} |P_F(x, f_1(x), f_2(x)) - P_F(x, g_1(x), g_2(x))| \\ &\leq 4 \cdot \prod_{j=1}^{2} \left(\|g_j\|_2 + \|f_j\|_2 + 1\right) \cdot \max_{j \in [2]} \left\{\|f_j - g_j\|_2\right\}. \end{aligned} \tag{43}$$

By definition, we have:

$$\begin{aligned}\mathop{\mathbb{E}}_{x \leftarrow \mathcal{S}} |P_F(x, f_1(x), f_2(x)) - P_F(x, g_1(x), g_2(x))| &= \mathop{\mathbb{E}}_{i \in [m]} \mathop{\mathbb{E}}_{x \leftarrow U_\ell} |\widehat{F}_i(x) - F_i(x)|, \\ \max_{j \in [2]} \left\{\|f_j - g_j\|_2\right\} &\leq \sum_{j \in [2]} \left(\left(\mathop{\mathbb{E}}_{i \in [m]} \|T_{ij} - \widehat{T}_{ij}\|_2^2 \right)^{1/2} \right), \\ \|f_j\|_2 &= \left(\mathop{\mathbb{E}}_{i \in [m]} \|T_{ij}\|_2^2 \right)^{1/2}, \\ \|g_j\|_2 &= 1. \end{aligned}$$

Therefore, (43) translates to

$$\begin{aligned}&\mathop{\mathbb{E}}_{i \in [m]} \mathop{\mathbb{E}}_{x \leftarrow U_\ell} |\widehat{F}_i(x) - F_i(x)| \\ &\leq 4 \cdot \sum_{j \in [2]} \left(\left(\mathop{\mathbb{E}}_{i \in [m]} \|T_{ij} - \widehat{T}_{ij}\|_2^2 \right)^{1/2} \right) \cdot \prod_{j \in [2]} \left(\left(\mathop{\mathbb{E}}_{i \in [m]} \|T_{ij}\|_2^2 \right)^{1/2} + 2 \right). \end{aligned} \tag{44}$$

Now, we make two observations. First, we have

$$\sum_{j \in [2]} \left(\left(\mathop{\mathbb{E}}_{i \in [m]} \|T_{ij} - \widehat{T}_{ij}\|_2^2 \right)^{1/2} \right) \leq 2 \left(\mathop{\mathbb{E}}_{i,j \in [m] \times [2]} \|T_{ij} - \widehat{T}_{ij}\|_2^2 \right)^{1/2} \leq 2 \left(\mathop{\mathbb{E}}_{i,j \in [m] \times [2]} \mathop{\mathbb{E}}_{x \leftarrow U_\ell} P_{ij}(x) \right)^{1/2}, \tag{45}$$

where the first step is due to $\sqrt{a} + \sqrt{b} \leq 2\sqrt{\frac{a+b}{2}}$, and the second step follows from $(1 - T_{ij}(x))^2(1 + T_{ij}(x))^2 \geq (T_{ij}(x) - \widehat{T}_{ij}(x))^2$. We also have

$$\prod_{j \in [2]} \left(\left(\mathop{\mathbb{E}}_{i \in [m]} \|T_{ij}\|_2^2 \right)^{1/2} + 2 \right) \leq \left(\left(2 \mathop{\mathbb{E}}_{ij \in [m] \times [2]} \|T_{ij}\|_2^2 \right)^{1/2} + 2 \right)^2, \tag{46}$$

since $(\sqrt{a} + 2)(\sqrt{b} + 2) \leq (\sqrt{a} + \sqrt{b} + 2)^2$. Finally, combining (44)-(46) with the conditions (39)-(40), it follows that

$$\mathop{\mathbb{E}}_{i \in [m]} \mathop{\mathbb{E}}_{x \leftarrow U_\ell} |\widehat{F}_i(x) - F_i(x)| \leq 200\sqrt{\delta}. \tag{47}$$

Now, it is clear that for $\delta < \sqrt{\frac{c_{\mathrm{pcpp}} - s_{\mathrm{pcpp}}}{1000}}$, combining (41), (42) and (47) leads to a contradiction. This completes the proof. □

[30] Recall that for each $x \in S$, P_F is the multi-linear extension of some Boolean function. So its coefficients are bounded by 4 (see Section 3.1).

B.3 Extract Hardness

Now we prove Lemma 5.10.

Recall that for an input z of length n, we let $\ell = \log T(n) + O(\log\log T(n))$ be the input length to VPCP_z^C and its oracle. For a circuit C, we apply a PCPP construction to VPCP_z^C and use $m = \mathrm{poly}(|\mathsf{VPCP}_z^C|)$ to denote the number of clauses.

Reminder of Lemma 5.10. *For every small enough constants $\alpha, \delta \in (0,1)$ and for every sufficiently large constant $K \geq 1$, the following holds: Suppose for some input z of length n such that $\mathcal{A}_{\mathsf{cheat}}(z) = 0$, there is a circuit C of size at most $S(\ell)^\alpha$ such that VPCP_z^C is a tautology with $(\widehat{Y} = \mathsf{Enc}(x), \widehat{Z})$ being its correct Boolean-valued proof. Then the function $H^{\widehat{Y},\widehat{Z}}$ defined by*

$$\begin{aligned} H^{\widehat{Y},\widehat{Z}} \colon \{0,1\}^{\log(m)+1+\ell} &\to \{-1,1\} \\ (i,j,x) &\mapsto \widehat{T}_{ij}(x) \end{aligned} \tag{48}$$

is hard in the following sense: letting $r = \log(m) + 1 + \ell$, for every $\mathsf{Sum} \circ \mathcal{F}_r$-functions H such that $\mathsf{complexity}(H) \leq S(r)^{3\alpha}$ and $\|H\|_4 \leq 1$, it holds that

$$\langle H^{\widehat{Y},\widehat{Z}}, H \rangle < (1 - \delta/5). \tag{49}$$

Proof of Lemma 5.10. Suppose on the contrary that there exists a $\mathsf{Sum} \circ \mathcal{F}_r$-function $H \colon \{0,1\}^r \to \mathbb{R}$ with $\mathsf{complexity}(H) \leq S(r)^{3\alpha}$ and $\|H\|_4 \leq 1$ such that (49) does not hold. We show that the algorithm $\mathcal{A}_{\mathsf{cheat}}$ can pass the final verification given H being the guessed function, and this contradicts to the assumption. We assume without loss of generality that $\|H\|_4 = 1$. (If not, just do some scaling.)

Recall that for every $s \in [|\mathcal{Y}|]$ and $t \in [|\mathcal{Z}|]$, $\mathcal{A}_{\mathsf{cheat}}$ constructed the following functions.

$$\begin{aligned} Y_s(x) &:= \mathop{\mathbb{E}}_{i,j \text{ s.t. } \mathcal{T}_{ij} = \mathcal{Y}_s} H(i,j,x), \\ Z_t(x) &:= \mathop{\mathbb{E}}_{i,j \text{ s.t. } \mathcal{T}_{ij} = \mathcal{Z}_t} H(i,j,x). \end{aligned} \tag{50}$$

Recall that $T_{ij} \in (Y \cup Z)$ denotes the proof function corresponding to the variable $\mathcal{T}_{ij} \in \mathcal{Y} \cup \mathcal{Z}$ and $F_i = \widetilde{\mathsf{Cons}}_i(T_{i1}, T_{i2})$. Also recall that we defined $\widehat{T}$ and $\widehat{F}$ for $(\widehat{Y}, \widehat{Z})$ similarly.

To show that H can pass the verification, we need to verify the following (rewriting (30) - (32)):

$$\mathop{\mathbb{E}}_{i,j \in [m] \times [2]} \mathop{\mathbb{E}}_{x \leftarrow U_\ell} P_{i,j}(x) \leq \delta, \tag{51}$$

$$\mathop{\mathbb{E}}_{i,j \in [m] \times [2]} \mathop{\mathbb{E}}_{x \leftarrow U_\ell} H(i,j,x)^2 \leq 1, \tag{52}$$

$$\mathop{\mathbb{E}}_{i \in [m]} \mathop{\mathbb{E}}_{x \leftarrow U_\ell} F_i(x) \geq c_{\mathsf{pcpp}} - \frac{1}{2}(c_{\mathsf{pcpp}} - s_{\mathsf{pcpp}}). \tag{53}$$

Here $P_{ij}(x)$ is defined as

$$P_{ij}(x) = \begin{cases} (1 - T_{ij}(x)^2)^2, & \text{if } T_{ij} \in Z, \\ (\mathsf{Enc}_s(x) - T_{ij}(x))^2, & \text{if } T_{ij} \in Y. \end{cases}$$

(52) clearly holds since $\|H\|_4 \le 1$. (51) is a little bit tricky. First, by definition (50) we observe that

$$\mathop{\mathbb{E}}_{i,j\in[m]\times[2]}\ \mathop{\mathbb{E}}_{x\leftarrow U_\ell}[T_{i,j}(x)\cdot \widehat{T}_{i,j}(x)] = \langle H^{\widehat{Y},\widehat{Z}}, H\rangle \ge (1-\delta/5). \tag{54}$$

$$\begin{aligned}
\mathop{\mathbb{E}}_{i,j\in[m]\times[2]}\ \mathop{\mathbb{E}}_{x\leftarrow U_\ell} |T_{ij}(x)|^2 &\ge \left(\mathop{\mathbb{E}}_{i,j}\ \mathop{\mathbb{E}}_{x\leftarrow U_\ell} |T_{ij}(x)|\right)^2 && \text{(by Jensen's inequality)}\\
&\ge \left(\mathop{\mathbb{E}}_{i,j}\ \mathop{\mathbb{E}}_{x\leftarrow U_\ell} [T_{i,j}(x)\cdot \widehat{T}_{i,j}(x)]\right)^2 && (\widehat{T}_{i,j}(x)\in\{-1,1\})\\
&\ge (1-2\delta/5).
\end{aligned} \tag{55}$$

By definition of P_{ij}, we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{ij\in[m]\times[2]}\ \mathop{\mathbb{E}}_{x\leftarrow U_\ell} P_{ij}(x) &\le \mathop{\mathbb{E}}_{ij}\ \mathop{\mathbb{E}}_{x\leftarrow U_\ell} \left((1-T_{ij}(x)^2)^2 + (T_{ij}(x)-\widehat{T}_{ij}(x))^2\right)\\
&\le \mathop{\mathbb{E}}_{ij}\ \mathop{\mathbb{E}}_{x\leftarrow U_\ell} \left(T_{ij}(x)^4 + 2 - 2\cdot T_{ij}(x)\cdot \widehat{T}_{ij}(x) - T_{ij}(x)^2\right)\\
&\le \|H\|_4^4 + 2 - \mathop{\mathbb{E}}_{ij}\ \mathop{\mathbb{E}}_{x\leftarrow U_\ell} \left(2\cdot T_{ij}(x)\cdot \widehat{T}_{ij}(x) + T_{ij}(x)^2\right)\\
&\le 1+2-(2(1-\delta/5)+(1-2\delta/5)) && \text{(by (54) and (55))}\\
&\le \delta.
\end{aligned}$$

Finally we verify (53). We note that

$$\mathop{\mathbb{E}}_{i,j\in[m]\times[2]} \|T_{ij}\|_2^2 \le \|H\|_2^2 \le 1. \tag{56}$$

Combining this fact with (54), we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{i,j\in[m]\times[2]} \|T_{ij}-\widehat{T}_{ij}\|_2^2 &= \mathop{\mathbb{E}}_{i,j\in[m]\times[2]}\ \mathop{\mathbb{E}}_{x\leftarrow U_\ell} (T_{ij}(x)-\widehat{T}_{ij}(x))^2\\
&= \mathop{\mathbb{E}}_{i,j\in[m]\times[2]} \left(\|T_{ij}\|_2^2 + \|\widehat{T}_{ij}\|_2^2 - 2\langle T_{ij}, \widehat{T}_{ij}\rangle\right)\\
&\le 1+1-2(1-\delta/5) && \text{(by (54) and (56))}\\
&\le \frac{2}{5}\delta.
\end{aligned} \tag{57}$$

Now, given (52) and (57), we can bound $\mathbb{E}_{i\in[m]}\mathbb{E}_{x\leftarrow U_\ell}|F_i - \widehat{F}_i|$ by utilizing Lemma B.1. This step is very similar to the application of Lemma B.1 in the proof of Lemma 5.9. In particular, we can show that

$$\mathop{\mathbb{E}}_{i\in[m]}\ \mathop{\mathbb{E}}_{x\leftarrow U_\ell}[|F_i(x)-\widehat{F}_i(x)|] \le 200\sqrt{\delta}.$$

Since $(\widehat{Y},\widehat{Z})$ is the correct proof to VPCP_z^C, it follows that

$$\mathop{\mathbb{E}}_{i\in[m]}\ \mathop{\mathbb{E}}_{x\leftarrow U_\ell}[\widehat{F}_i(x)] \ge c_{\mathsf{pcpp}}.$$

Hence, for sufficiently small $\delta > 0$, it holds that

$$\mathop{\mathbb{E}}_{i\in[m]}\ \mathop{\mathbb{E}}_{x\leftarrow U_\ell}[F_i(x)] \ge c_{\mathsf{pcpp}} - \frac{1}{2}(c_{\mathsf{pcpp}} - s_{\mathsf{pcpp}}).$$

In conclusion, we showed that $\mathcal{A}_{\mathsf{cheat}}$ on the PCPP of VPCP_z^C accepts H as a proof, and $\mathcal{A}_{\mathsf{cheat}}(z) = 1$. This contradicts to our assumption.

□

C Missing Proofs in Section 7

C.1 The Proof of Lemma 7.7

Reminder of Lemma 7.7. *For two matrices $H \in \mathbb{R}^{N_1 \times N_1}$ and $H' \in \{-1,1\}^{N_1 \times N_1}$, it holds that*

$$\mathop{\mathbb{E}}_{r \leftarrow U_\gamma} \left| \widetilde{\mathsf{VrecPCP}}_z^H(r) - \mathsf{VrecPCP}_z^{H'}(r) \right| \le 2^q \cdot q \cdot \left(2 + q^{1/(2q-2)} \|H\|_{2q-2}\right)^{2q-2} \|H - H'\|_2.$$

Proof. For every $r \in \{0,1\}^\gamma$ and $i \in [q]$, let the index of the i-th query to the proof be $(x_{r,i}, y_{r,i})$. Then we define q functions $H_1, \dots, H_q \colon \{0,1\}^\gamma \to \mathbb{R}$ by $H_i(r) := H(x_{r,i}, y_{r,i})$ for every $i \in [q]$. We also define q Boolean functions $H'_1, \dots, H'_q \colon \{0,1\}^\gamma \to \{-1,1\}$ by $H'_i(r) = H'(x_{r,i}, y_{r,i})$ for every i.

By the smoothness property of $\mathsf{VrecPCP}_z$ (Theorem 7.6), we have

$$\mathop{\mathbb{E}}_{i \leftarrow [q]} \|H_i\|_{2q-2}^{2q-2} = \|H\|_{2q-2}^{2q-2},$$
$$\mathop{\mathbb{E}}_{i \leftarrow [q]} \|H_i - H'_i\|_2^2 = \|H - H'\|_2^2.$$

Hence, for every $i \in [q]$, it holds that

$$\|H_i - H'_i\|_2 \le q \cdot \|H - H'\|_2, \tag{58}$$
$$\|H_i\|_{2(q-1)} \le q^{1/(2q-2)} \|H\|_{2(q-1)}. \tag{59}$$

Recall that $\widetilde{\mathsf{VrecPCP}}_z^H$ is the natural arithmetization of $\mathsf{VrecPCP}$: for each $r = (r_{\mathsf{row}}, r_{\mathsf{col}}, r_{\mathsf{shared}})$, $\widetilde{\mathsf{VrecPCP}}_z^H$ reads q values from the proof H, which are $H_1(r), \dots, H_q(r)$ by definition, and outputs $P_r(H_1(r), \dots, H_q(r))$ where $P_r \colon \mathbb{R}^q \to \mathbb{R}$ is the polynomial mapping the query answers to the decision. Observe that P_r is the multi-linear extension of a Boolean function[31]. So its coefficients are bounded by 2^q.

Therefore, we can apply Lemma B.1 to the function $\widetilde{\mathsf{VrecPCP}}_z$, with the set $\mathcal{S} := \{0,1\}^\gamma$ and two lists of functions $(H_1, \dots, H_q)$ and $(H'_1, \dots, H'_q)$. This completes the proof. □

C.2 The Proof of Lemma 7.8

Reminder of Lemma 7.8. *For every $\tau \ge 1$, there is a sufficiently small $\alpha > 0$ such that the following is true. For every $z \in \{0,1\}^*$ and H being a $\mathsf{Sum} \circ \mathcal{M}_\ell^{2^{\tau \ell^{1-c}}}$-function with $\mathsf{complexity}(H) \le 2^{\alpha \ell^c}$, evaluations of the left-hand sides of (36)-(38) can be done in $2^{\ell - \Omega(\ell^c/\alpha)} \le o(n^K)$ time.*

Before we proceed, we introduce the following lemma. It will be proved in the end of this subsection.

Lemma C.1. *For every $n, d, r, t \in \mathbb{N}_{\ge 1}$, let $P \colon \mathbb{R}^d \to \mathbb{R}$ be a multi-linear polynomial. Let $H_1, \dots, H_d \colon \{0,1\}^n \to \mathbb{R}$ be a list of $\mathsf{Sum} \circ \mathcal{M}_n^r$-functions with $\mathsf{complexity}(H_i) \le t$ for every $i \in [d]$. Then the evaluation*

$$\mathop{\mathbb{E}}_{x \leftarrow U_n} P(H_1(x), \dots, H_d(x))$$

reduces to at most $(2t)^d$-many $\#\mathsf{SAT}$ tasks for $\mathcal{M}_n^{dr}$-functions. The reduction can be computed in $O((2t)^d \cdot 2^{n/2} \cdot d \cdot r)$ time.

[31] Check Section 3.1 for the relevant discussion.

Proof of Lemma 7.8. First, we recall the left-hand sides of (36)-(38):

$$\mathop{\mathbb{E}}_{(x,y)\leftarrow U_\ell} (1-H(x,y))^2(1+H(x,y))^2 \tag{60}$$

$$\mathop{\mathbb{E}}_{(x,y)\leftarrow U_\ell} H(x,y)^{2q-2} \tag{61}$$

$$\mathop{\mathbb{E}}_{r\leftarrow U_\gamma} \left[\widetilde{\mathsf{VrecPCP}}_z^H(r)\right] \tag{62}$$

Note that for (60) and (61), we have to calculate expectations of constant-degree polynomials (over $\mathbb{R}$) of H.

For (62), we introduce some notation. For every $r = (r_{\mathsf{row}}, r_{\mathsf{col}}, r_{\mathsf{shared}})$ and for $i \in [q]$, let $x_{r,i}, y_{r,i}$ be the row index and column index of the i-th query given randomness r. Recall that we wrote the output of $\mathsf{VrecPCP}_z$ as a multi-linear polynomial (over $\mathbb{R}$) of the query answers $(v_1, \ldots, v_q)$ and the parity check $(c_1, \ldots, c_p)$, denoted by $Q_{r_{\mathsf{shared}}} \colon \mathbb{R}^{q+p} \to \mathbb{R}$, which maps $(v_1, \ldots, v_q, c_1, \ldots, c_p)$ to a bit in $\{0, 1\}$. Moreover, for every $r = (r_{\mathsf{shared}}, r_{\mathsf{row}}, r_{\mathsf{col}})$, we wrote the decision as a multi-linear polynomial of queried bits, denoted by $P_r \colon \mathbb{R}^q \to \mathbb{R}$.

Now, fixing r_{shared}, for every $i \in [q]$, we define a function $H_i^{r_{\mathsf{shared}}}$ as

$$H_i^{r_{\mathsf{shared}}}(r_{\mathsf{row}}, r_{\mathsf{col}}) := H(x_{r,i}, y_{r,i}).$$

Since $x_{r,i}$ (resp. $y_{r,i}$) only depends on r_{row} (resp. r_{col}), we conclude that $H_i^{r_{\mathsf{shared}}} \in \mathsf{Sum} \circ \mathcal{M}^{2^{\tau\ell^{1-c}}}_{|r_{\mathsf{row}}|+|r_{\mathsf{col}}|}$ with $\mathsf{complexity}(H_i^{r_{\mathsf{shared}}}) \le 2^{\alpha\ell^c}$, and the description of $H_i^{r_{\mathsf{shared}}}$ can be computed in $\widetilde{O}(2^{\ell/2})$ time. Also recall that for every $j \in [p]$ and $(r_{\mathsf{row}}, r_{\mathsf{col}})$, we defined $C_j^{r_{\mathsf{shared}}}(r_{\mathsf{row}}, r_{\mathsf{col}})$ to be the result of the j-th parity check given randomness r. Note that we have $C_j^{r_{\mathsf{shared}}} \in \mathcal{M}^2_{|r_{\mathsf{row}}|+|r_{\mathsf{col}}|}$. What we want to compute can be written as

$$\mathop{\mathbb{E}}_{(r_{\mathsf{row}}, r_{\mathsf{col}})}[P_r(H_1^{r_{\mathsf{shared}}}, \ldots, H_q^{r_{\mathsf{shared}}})] = \mathop{\mathbb{E}}_{(r_{\mathsf{row}}, r_{\mathsf{col}})}[Q_{r_{\mathsf{shared}}}(H_1^{r_{\mathsf{shared}}}, \ldots, H_q^{r_{\mathsf{shared}}}, C_1^{r_{\mathsf{shared}}}, \ldots, C_p^{r_{\mathsf{shared}}})]. \tag{63}$$

Fixing $\tau \ge 1$, we can choose $\alpha > 0$ being sufficiently small constant such that the following argument holds. First, assuming Lemma C.1 and given $\mathsf{complexity}(H_i) \le 2^{\alpha\ell^c}$, the evaluations of (60) and (61) reduce to solving $2^{O(\alpha\ell^c)}$ #SAT tasks for $\mathcal{M}_\ell^{2q2^{\tau\ell^{1-c}}}$-functions, which can be done in

$$2^{\ell-\Omega(\ell^c/\tau)+O(\alpha\ell^c)} = 2^{\ell-\Omega(\ell^c/\tau)} \le o(n^K)$$

time, by Lemma 7.4.

For (62), note that its evaluation reduces to calculating for each r_{shared} the right-hand side of (63). For every fixed r_{shared}, by Lemma C.1, the evaluation of (63) reduces to solving $2^{O(\alpha\ell^c)}$ #SAT tasks for $\mathcal{M}^{(q+1)2^{\tau\ell^{1-c}}}_{|r_{\mathsf{row}}|+|r_{\mathsf{col}}|}$-functions (since p and q are constants). Since $|r_{\mathsf{row}}| \ge \Omega(\ell)$, again by Lemma 7.4, evaluating (63) can be done in

$$2^{|r_{\mathsf{row}}|+|r_{\mathsf{col}}|-\Omega(\ell^c/\tau)+O(\alpha\ell^c)}$$

time. Enumerating r_{shared}, the evaluation of (62) can be done in

$$2^{|r_{\mathsf{shared}}|+|r_{\mathsf{row}}|+|r_{\mathsf{col}}|-\Omega(\ell^c/\tau)+O(\alpha\ell^c)} = 2^{\gamma-\Omega(\ell^c/\tau)} \le o(n^K)$$

time, which completes the proof.

□

Now we prove Lemma C.1.

Proof of Lemma C.1. We first show that for the special case when $P(v_1,\dots,v_d) = \prod_{i=1}^{d} v_i$, the desired evaluation in the lemma can be reduced to solving t^d #SAT tasks. The general case can be handled by considering monomials in P one by one (there are at most 2^d monomials in a multilinear polynomials on d variables). First, for each $i \in [q]$, we write

$$H_i(x) = \sum_{j=1}^{t} \alpha_{i,j} H_{i,j}(x).$$

Note that here we assume without loss of generality that all of $H_i(x)$ have sparsity exactly t. Then we have

$$\mathop{\mathbb{E}}_{x \leftarrow U_n} \prod_{i=1}^{d} H_i(x) = \sum_{(\ell_1,\dots,\ell_d)\in[t]^d} \left[\prod_{i\in[d]} \alpha_{i,\ell_i} \left(\mathop{\mathbb{E}}_{x\leftarrow U_n} \prod_{i=1}^{d} H_{i,\ell_i}(x) \right) \right].$$

For two functions $f, g \in \mathcal{M}_n^r$, we define two matrices $M_f, M_g \in \mathbb{F}_2^{2^{r/2}\times 2^{r/2}}$ such that $f(x,y) = (-1)^{M_f(x,y)}$ and $g(x,y) = (-1)^{M_g(x,y)}$ for $(x,y) \in \{0,1\}^n$. Then we have $f(x,y)\cdot g(x,y) = (-1)^{M_f(x,y)+M_g(x,y)}$, where the addition of M_f and M_g is over $\mathbb{F}_2$. It follows that $f \cdot g \in \mathcal{M}_n^{2r}$. Moreover, if we have the descriptions of f and g, denoted by $M_f = A_f \cdot B_f^T$ and $M_g = A_g \cdot B_g^T$, then the description of $f(x,y)\cdot g(x,y)$ is just $(A_f, A_g)\cdot(B_f, B_g)^T$, where we use $(A_f, A_g), (B_f, B_g)$ to denote the concatenations of matrices with same number of rows.

Hence, for every tuple $(\ell_1,\dots,\ell_d) \in [n^d]$, the function $\prod_{i=1}^{d} H_{i,\ell_i}(x)$ is in $\mathcal{M}_n^{dr}$, and its description can be computed in $O(2^{n/2}\cdot r\cdot d)$ time. This completes the proof. □

C.3 The Proof of Lemma 7.9

Reminder of Lemma 7.9. *For every sufficiently small $\delta > 0$, it holds that $\mathcal{A}_{\mathsf{matrix}}(z) \le \mathcal{A}_{\mathsf{FS}}^{T}(z)$.*

Proof of Lemma 7.9. Suppose on the contrary that there exists $z \in \{0,1\}^*$ such that $\mathcal{A}_{\mathsf{FS}}^{T}(z) = 0$ but $\mathcal{A}_{\mathsf{matrix}}(z) = 1$. We show a contradiction. By $\mathcal{A}_{\mathsf{matrix}}(z) = 1$ we know that there is a function $H\colon \{0,1\}^{\ell} \to \mathbb{R}$ such that the following hold (rewriting (36) - (38)):

$$\mathop{\mathbb{E}}_{(x,y)\leftarrow U_\ell} (1 - H(x,y))^2 (1 + H(x,y))^2 \le \delta, \tag{64}$$

$$\mathop{\mathbb{E}}_{(x,y)\leftarrow U_\ell} H(x,y)^{2q-2} \le 1, \tag{65}$$

$$\mathop{\mathbb{E}}_{r\leftarrow U_\gamma} \left[\widetilde{\mathsf{VrecPCP}_z^H(r)} \right] \ge \frac{1+s}{2}. \tag{66}$$

We define from H a Boolean function H' as $H'(x,y) = \mathrm{sign}(H(x,y))$. It follows from (64) that $\|H - H'\|_2 \le \|(1-H)(1+H)\|_2 \le \sqrt{\delta}$.

Using Lemma 7.7, it follows from (65) that

$$\mathop{\mathbb{E}}_{r\leftarrow U_\gamma} \left| \widetilde{\mathsf{VrecPCP}_z^H(r)} - \mathsf{VrecPCP}_z^{H'}(r) \right| \le 2^q \cdot q \cdot \left(2 + q^{1/(2q-2)} \|H\|_{2q-2} \right)^{2q-2} \|H - H'\|_2 \le 2^{O(q)} \sqrt{\delta}. \tag{67}$$

Since $\mathcal{A}_{\mathsf{FS}}^{T}(z) = 0$, by the soundness property of Theorem 7.6, it follows that

$$\mathop{\mathbb{E}}_{r \leftarrow U_\gamma}\left[\mathsf{VrecPCP}_z^{H'}(r)\right] < s. \tag{68}$$

Note that for sufficiently small δ ($\delta \ll (1-s)$), combining (66)-(68) leads to a contradiction. This completes the proof. □

C.4 The Proof of Lemma 7.10

Reminder of Lemma 7.10. *For every sufficiently small $\alpha > 0$, the following is true. For every z such that $\mathcal{A}_{\mathsf{matrix}}(z) = 0$ and $\mathcal{A}_{\mathsf{FS}}^{T}(z) = 1$, every correct proof for $\mathsf{VrecPCP}_z$ is a function $H' \colon \{0,1\}^{\ell} \to \{-1,1\}$ such that, for every $\mathsf{Sum} \circ \mathcal{M}_{\ell}^{2^{\tau \ell^{1-c}}}$-function H with $\mathsf{complexity}(H) \le 2^{\alpha \ell^{c}}$, it holds that $\langle H, H' \rangle \le (1-\delta/5)\|H\|_{2(q-1)}$.*

Proof. Suppose on the contrary that there exists an $H \colon \{0,1\}^{\ell} \to \mathbb{R}$ guessed by $\mathcal{A}_{\mathsf{matrix}}(z)$ that violates the lemma statement. In the following, we assume without loss of generality that $\|H\|_{2(q-1)} = 1$. (If not, just do a scaling.) We show that $\mathcal{A}_{\mathsf{matrix}}(z)$ accepts H as a proof, which contradicts the assumption that $\mathcal{A}_{\mathsf{matrix}}(z) = 0$.

We verify that after guessing H, $\mathcal{A}_{\mathsf{matrix}}(z)$ can pass the following tests (rewriting (36)-(38)):

$$\mathop{\mathbb{E}}_{(x,y) \leftarrow U_\ell}(1 - H(x,y))^2(1 + H(x,y))^2 \le \delta, \tag{69}$$

$$\mathop{\mathbb{E}}_{(x,y) \leftarrow U_\ell} H(x,y)^{2q-2} \le 1, \tag{70}$$

$$\mathop{\mathbb{E}}_{r \leftarrow U_\gamma}\left[\widetilde{\mathsf{VrecPCP}_z^{H}}(r)\right] \ge \frac{1+s}{2}. \tag{71}$$

In the following, we assume that $q \ge 3$. If it is not the case, we can construct an equivalent PCP which issues some additional queries to proof and ignore the answer.

First, (70) holds since we have assumed that $\|H\|_{2q-2} = 1$. For (71), we observe that

$$\|H\|_2^2 \le \|H\|_{2q-2}^2 \le 1$$

and

$$\langle H, H' \rangle > (1-\delta/5)\|H\|_{2q-2} \ge 1 - \delta/5.$$

Therefore, we have that

$$\|H - H'\|_2^2 = \|H\|_2^2 + \|H'\|_2^2 - 2\langle H, H' \rangle \le \frac{2}{5}\delta. \tag{72}$$

Note that δ measures the distance between H and H'. The less δ is, the closer H is to H'. Hence, for sufficiently small δ, it follows from Lemma 7.7, (70) and (72) that

$$\mathop{\mathbb{E}}_{r \leftarrow U_\gamma}\left[\widetilde{\mathsf{VrecPCP}_z^{H}}(r)\right] \ge \mathop{\mathbb{E}}_{r \leftarrow U_\gamma}\left[\mathsf{VrecPCP}_z^{H'}(r)\right] - 2^{O(q)} \cdot \delta \ge \frac{1+s}{2}.$$

Lastly, we verify (69). Note that $\|H\|_2 \geq \|H\|_1 \geq \langle H, H' \rangle \geq 1 - \delta/5$. It follows that

$$\begin{aligned}
\|(H+1)(H-1)\|_2^2 &= \|1 - H^2\|_2^2 \\
&= \mathop{\mathbb{E}}_{(x,y) \leftarrow U_\ell} \left[H(x,y)^4 + 1 - 2H(x,y)^2 \right] \\
&= \|H\|_4^4 + 1 - 2\|H\|_2^2 \\
&\leq 2 - 2(1 - 2\delta/5) && (\|H\|_4 \leq \|H\|_{2q-2} = 1) \\
&\leq \delta.
\end{aligned}$$

This shows that H can pass the tests and consequently $\mathcal{A}_{\mathsf{matrix}}(z) = 1$, a contradiction.

□