THEORY OF COMPUTING, Volume 20 (2), 2024, pp. 1–19
www.theoryofcomputing.org

New Distinguishers for Negation-Limited Weak Pseudorandom Functions

Zhihuai Chen Siyao Guo* Qian Li† Chengyu Lin‡ Xiaoming Sun§

Received March 13, 2021; Revised August 1, 2022; Published July 29, 2024

Abstract. We show how to distinguish circuits with $\log k$ negations (a.k.a. k-monotone functions) from uniformly random functions in $\exp\left(\tilde{O}\left(n^{1/3}k^{2/3}\right)\right)$ time using random samples. The previous best distinguisher, due to the learning algorithm by Blais, Canonne, Oliveira, Servedio, and Tan (RANDOM'15), requires $\exp\left(\tilde{O}(n^{1/2}k)\right)$ time.

Our distinguishers are based on Fourier analysis on *slices of the Boolean cube*. We show that some "middle" slices of negation-limited circuits have strong low-degree Fourier concentration and then we apply a variation of the classic Linial, Mansour, and Nisan "Low-Degree algorithm" (JACM'93) on slices. Our techniques also lead to a slightly improved weak learner for negation-limited circuits under the uniform distribution.

*Supported by National Natural Science Foundation of China Grant No.62102260, NYTP Grant No. 20121201 and NYU Shanghai Boost Fund.

†Supported by Hetao Shenzhen-Hong Kong Science and Technology Innovation Cooperation Zone Project (No.HZQSWS-KCCYB-2024016), and the National Natural Science Foundation of China Grants No. 62002229.

‡Supported in part by the U.S. Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research under award number DE-SC-0001234, by a grant from the Columbia-IBM center for Blockchain and Data Transparency, by JPMorgan Chase & Co., and by LexisNexis Risk Solutions. Any views or opinions expressed herein are solely those of the authors listed.

§Supported by the National Natural Science Foundation of China Grants No. 62325210.

ACM Classification: F.2,G.2

AMS Classification: 68Q25,68Q32

Key words and phrases: pseudorandom functions, negation-limited circuits, Fourier analysis

DOI: 10.4086/toc.2024.v020a002

1 Introduction

One significant goal in the area of cryptography is to understand how simple cryptography can be. This motivates the study of low complexity cryptography which explores the possibility of implementing cryptographic primitives in low complexity classes. This line of research inherently lies at the intersection of computational complexity and cryptography. It links core problems in both areas and has become an essential source of new perspectives for both areas.

In this work, we continue this line of research and focus on *pseudorandom functions* (PRFs) in *negation-limited computation*. We start by introducing pseudorandom functions and negation-limited computation before connecting them to explain the main motivation of our work.

Pseudorandom functions. Pseudorandom functions (PRFs) [12] are fundamental primitives in symmetric cryptography. In particular, they yield direct solutions to most central goals of symmetric cryptography, such as encryption, authentication and identification. They are well studied in the theoretical community and widely used in practice.

As lightweight (computationally limited) devices become popular, the efficiency of cryptographic implementations also becomes increasingly significant. To obtain a better tradeoff between efficiency and security, a weaker notion of PRFs called *weak pseudorandom functions* (See Definition 1.1) has been considered. A distinguisher for a family of weak PRFs aims to distinguish a random member of the family from a truly random function after observing a number of random samples $(x_1, f_s(x_1)), \ldots, (x_m, f_s(x_m))$ where $x_1, \ldots, x_m$ are independent uniformly random strings from $\{0,1\}^n$ and $f_s : \{0,1\}^n \to \{0,1\}$ is the function in question. Weak PRFs suffice for many key applications such as encryption and authentication in symmetric cryptography. More importantly, weak PRFs may allow for significant gains in efficiency. Akavia et al. [1] pointed out weak PRFs have the potential to bypass the limitations of PRFs in low depth circuits. In particular, they provided candidate weak PRFs in a class of low depth circuits where PRFs provably cannot exist. This raises the following natural questions.

Can weak PRFs bypass the limitations of PRFs in other low complexity classes?

Besides cryptography, another important motivation for the study of low complexity PRFs comes from explaining the difficulties of obtaining circuit lower bounds and learning algorithms. We refer interested readers to the survey by Bogdanov and Rosen [7].

Negation-limited computation. The power of negations is a mystery in complexity theory. One of the main difficulties in proving lower bounds on circuit size using AND, OR, NOT gates is the presence of negation gates: the best such lower bound is linear, whereas if no negation gates are allowed, exponential lower bounds are known [23, 3, 2, 26, 4, 16]. In 1958, Markov [20] observed that every Boolean (even multiple-output) function of n variables can be computed by a circuit with only $\log n$ negation gates. In other words, the exponential gap between monotone computation and non-monotone computation exists due to as few as $\log n$ negations.

Besides circuit complexity, the divide between monotone and non-monotone computation exists in general: while we usually have a fairly good understanding of the monotone case, many things may fail to hold when negation gates are allowed. Aiming at bridging the gap between monotone and non-monotone computation, a body of recent work studies negation-limited computation from multiple angles including learning [5], cryptography [15], Boolean formulas [14, 24], property testing [8, 13], Boolean function conjectures [18]. Although the above works extend many results in monotone cases to as many as $O(\log n)$ negations, they also leave open several surprisingly basic questions about a single negation ranging from weak learning algorithms to the structure of their Fourier spectrum. More surprisingly, in the context of property testing, a single negation can be exponentially harder than the monotone case [8, 13]. Our understanding of a single negation remains largely a mystery.

When the circuit size is not of interest, the classes of circuits with $\log k$ negations are captured by the class of so-called k-monotone functions where each function in the family can be written as the parity of k monotone functions (see Section 2.2). To simplify the presentation, we will use k-monotone functions instead of circuits with $\log k$ negations in some of our discussions.

PRFs in negation-limited computation. Can pseudorandom functions be computed by a few negations? For pseudorandom functions, we have a fairly good understanding. Guo et al. [15] showed that PRFs are inherently highly non-monotone and require $\log n - O(1)$ negations, which is optimal up to an additive constant. However, the answer to weak PRFs is unsatisfying. Guo et al. [15] observed that weak PRFs cannot be monotone due to the weak learner for monotone functions by Blum et al. [6]. For general k, the best distinguisher, due to Blais et al. [5], runs in time $n^{O(k\sqrt{n})}$. Therefore even for a single negation (i. e., $k = 2$), the best distinguisher runs in time $n^{O(\sqrt{n})}$.

The above results demonstrate two strong separations. In negation-limited computation, weak PRFs have the potential to be much simpler than PRFs: for all we know, even a single negation may have $n^{\Omega(\sqrt{n})}$ hardness whereas PRFs cannot exist. From the angle of weak PRFs, the hardness gap between even a single negation and monotone could be as large as $n^{\Omega(\sqrt{n})}$. These separations are our main motivation to connect them together to study negation-limited weak PRFs.

1.1 Our results

Before presenting our main results, we define *weak pseudorandom functions* and *weak learning under uniform distribution*.

Definition 1.1 (Weak pseudorandom functions)**.** Let S be a distribution over $\{0,1\}^m$ and $\{F_s : \{0,1\}^n \to \{0,1\}\}$ be a family of functions indexed by string s in the support of S. We say that $\{F_s\}$ is a family of (c, ϵ)-secure weak pseudorandom functions (wPRFs) if for every Boolean circuit D of size at most c,

$$\left| \Pr_s[D^{F_s} \text{ accepts}] - \Pr_R[D^R \text{ accepts}] \right| \leq \varepsilon, \tag{1.1}$$

where s is distributed according to S, R is a function sampled uniformly at random from the set of all functions from $\{0,1\}^n$ to $\{0,1\}$, and D^h denotes the execution of D with random oracle access to a Boolean function $h : \{0,1\}^n \to \{0,1\}$. In other words, the distinguisher D^h only has access to random examples of the form $(x, h(x))$ where x is uniformly distributed over $\{0,1\}^n$. The two probabilities in Equation (1.1) are both also over the random samples x.

Definition 1.2 (Weak learning under the uniform distribution)**.** We say that an algorithm A weakly learns a family $\mathcal{F}$ of Boolean functions under the uniform distribution, if given any $f \in \mathcal{F}$ it accesses pairs $(x, f(x))$ where x is picked from $\{0,1\}^n$ uniformly at random, and then outputs a hypothesis h such that with high probability (over the random samples and the randomness of A)

$$\Pr_{x \sim U_{\{0,1\}^n}} \left[f(x) \neq h(x) \right] \leq \frac{1}{2} - \frac{1}{\mathrm{poly}(n)} \tag{1.2}$$

where $U_{\{0,1\}^n}$ is the uniform distribution over $\{0,1\}^n$.

A weak learner works slightly better than random guessing. But from this small advantage, if it is non-negligible, one can naturally derive an efficient distinguisher against a random function. Any weak learner explicitly gives an attack on candidate families of weak pseudorandom functions. Conversely, families of weak pseudorandom functions are hard to learn. Our main result is new distinguishers for negation-limited families of weak pseudorandom functions. Our results hold for inefficient circuits and are stated in terms of k-monotone functions.

Theorem 1.3. *Any family of k-monotone functions can be distinguished from uniformly random functions in* $\exp\left(O(n^{1/3}(k \log n)^{2/3})\right)$ *time. In other words, any family of k-monotone functions is not a* $\left(\exp\left(O(n^{1/3}(k \log n)^{2/3})\right), 1/3\right)$*-secure weak pseudorandom family.*

The previous best distinguisher for k-monotone weak PRFs is the learning algorithm by Blais et al. [5] which runs in $\exp\left(O(n^{1/2} k \log n)\right)$ time. Our result improves this bound by an $\Omega\left(n^{1/6}(k \log n)^{1/3}\right)$ factor in the exponent.

Theorem 1.3 implies that exponentially secure weak PRFs require $\log n - O(\log \log n)$ negations, which is optimal up to an additive $O(\log \log n)$ term. Therefore, weak PRFs cannot bypass the limitations of PRFs in terms of achieving exponential security.

Theorem 1.3 also implies that 1-negation functions can be distinguished in $\exp\left(O(n^{1/3} \log^{2/3} n)\right)$ time. Therefore, unlike testing 1 negation (using 1-sided non-adaptive tester) [13] and learning 1 negation to high accuracy [5], distinguishing 1 negation does not suffer from the $\exp(\sqrt{n})$ barrier.

It is natural to ask if we can leverage the distinguisher to a learning algorithm. Our second result gives weak learning algorithms for k-monotone functions under the uniform distribution.

Theorem 1.4. *k-monotone functions are weakly learnable in time* $\exp\left(O\left(k\sqrt{n \log n}\right)\right)$.

Our result slightly improves the previous best weak learner due to Blais et al. [5], by a $\Omega\left(\sqrt{\log n}\right)$ factor in the exponent.

We conjecture that both Theorem 1.3 and Theorem 1.4 are not tight. However, we believe that any further improvement of our results, even for a single negation, requires completely new techniques or proving rather hard conjectures which seem out of reach. See Section 6 for more details.

Our techniques. Blais et al. [5] showed a Fourier concentration of k-monotone functions on low degree monomials, by bounding the total influence of k-monotone functions. Then they apply the "Low-Degree Algorithm" established by Linial, Mansour, and Nisan [19] to learn k-monotone functions. One natural idea to improve their learning algorithm is to show Fourier concentration on lower levels. However, their influence bound is tight and even for monotone functions, we cannot show concentration bound on fewer than $\Omega(\sqrt{n})$ levels [9], which will require at least $n^{\Omega(\sqrt{n})}$ time by applying the "Low-Degree Algorithm".

Our main technique is using Fourier analysis on slices [10, 25]. Although the Fourier concentration on the Boolean cube cannot be improved, we show some "middle" slices of k-monotone functions can have much stronger Fourier concentration. Then by adapting the "Low-Degree Algorithm" to the slices, we obtain a distinguisher with significantly improved running time. Our weak learner is a simple variant of the "Low-Degree Algorithm" on slices.

Fourier analysis on slices has a notion of total influence which allows us to show Fourier concentration on a slice in a similar way. We give an upper bound on the sum of total influences for all "middle" slices of any k-monotone function. It implies the existence of a "middle" slice function with small total influence, and therefore good concentration. Then we optimize the number of "middle" slices to be analyzed to get an efficient algorithm.

Organization of the paper. We begin with basic notation in Section 2, then present the structural results for k-monotone functions in Section 3. In Sections 4 and 5, we present the distinguisher and the weak learner, respectively.

2 Preliminaries

2.1 Notation, terminology: sets, strings, slices

In this paper, all the logarithms are base 2.

As usual, for $n \in \mathbb{N}$ we write $[n] = \{1, 2, \ldots, n\}$. The set of k-subsets (subsets of size k) of $[n]$ is denoted $\binom{[n]}{k}$.

We refer to the set $\{0,1\}^n$ as the n-dimensional *Boolean hypercube*; its elements are the $(0,1)$-*strings* of length n. For $0 \le r \le n$, the r-*slice* of the n-cube is the set

$$\{(x_1, \ldots, x_n) \in \{0,1\}^n \;:\; \sum_i x_i = r\}\,. \tag{2.1}$$

We identify subsets $A \subseteq [n]$ with their indicator strings $x_A = (x_1, \ldots, x_n)$ defined by

$$x_i = \begin{cases} 1 & \text{if } i \in A \\ 0 & \text{if } i \notin A\,. \end{cases} \tag{2.2}$$

With this identification, the r-slice of the n-dimensional Boolean hypercube becomes identical with the set $\binom{[n]}{r}$.

2.2 Alternating number, negation complexity, k-monotone functions

For any two strings $x, y \in \{0,1\}^n$, we say $x \prec y$ (or $y \succ x$) if $x \neq y$ and $x_i \leq y_i$ for all $i \in [n]$. A chain $X = (x^1, x^2, \ldots, x^\ell)$ of length ℓ is an increasing sequence of strings in $\{0,1\}^n$ where $x^i \prec x^{i+1}$ for $i \in [\ell - 1]$. For a Boolean function $f : \{0,1\}^n \to \{0,1\}$, we define the *alternating number* of f on chain X to be the number of value flips on this chain:

$$a(f, X) = \left|\{i \in [\ell - 1] : f(x^i) \neq f(x^{i+1})\}\right|. \tag{2.3}$$

Let C be the set of all chains on $\{0,1\}^n$, the alternating number of f is

$$a(f) = \max_{X \in C} a(f, X). \tag{2.4}$$

Note that the alternating number of a monotone function is no more than 1.

A celebrated result of Markov connects the alternating number of a Boolean function f to the negation complexity $\mathbf{N}(f)$ – the minimum number of negation gates required in any Boolean $\mathsf{AND} - \mathsf{OR}$ circuits to compute f.

Theorem 2.1 (Markov's Theorem [20])**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a function which is not identically* 0 *with $f(0^n) = 0$, then* $\mathbf{N}(f) = \lceil \log(a(f) + 1) \rceil - 1$.

Blais et al. [5] showed decomposition for functions with low alternating number [5].

Theorem 2.2 (Theorem 1.1 in [5])**.** *Let f be a k-alternating function, then $f(x) = h(m_1(x), \ldots, m_k(x))$ where $m_i(x)$ is monotone and h is the parity function or its negation. Conversely, any function of this form is k-alternating.*

The above characterization shows a simple structure for functions with a low alternating number, which are computable by few negation gates. To simplify notation, we will focus on the parity of few monotone functions.

Definition 2.3 (k-monotone function)**.** A function $f : \{0,1\}^n \to \{0,1\}$ is said to be k-monotone, if there exist k monotone functions $g_1, g_2, \ldots, g_k$ such that $f = g_1 \oplus g_2 \oplus \cdots \oplus g_k$.

2.3 Orthogonal basis for functions over a slice

Given a subset of the r-slice, $A \subseteq \binom{[n]}{r}$, denote its density in this slice by $\mu(A)$, i. e., $\mu(A) = |A| / \binom{n}{r}$. Define its *upper shadow* as

$$\partial^+ A := \left\{ x \in \binom{[n]}{r+1} : x \succ y \text{ for some } y \in A \right\} \tag{2.5}$$

and its *lower shadow* as

$$\partial^- A := \left\{ x \in \binom{[n]}{r-1} : x \prec y \text{ for some } y \in A \right\}. \tag{2.6}$$

Filmus [10] and Srinivasan [25] independently introduced an orthogonal basis for functions over a slice $\binom{[n]}{r}$ of the Boolean hypercube, which plays a central role in our proofs. Now we describe the explicit formula provided by Filmus [10] (see also Section 9.2 in [11]).

Definition 2.4. For $d \leq n/2$, define a $\mathcal{B}_{n,d}$*-sequence* to be a sequence $B = b_1, b_2, \cdots, b_d$ where (i) all $b_1, \cdots, b_d$ belong to $[n]$, (ii) $b_1 < b_2 < \cdots < b_d$, and (iii) $b_i \geq 2i$ for any $1 \leq i \leq d$. We will also use $\mathcal{B}_{n,d}$ to denote the set of all $\mathcal{B}_{n,d}$-sequences.

Given $B \in \mathcal{B}_{n,d}$, define $\mathcal{T}(B)$ to be the set consisting of all sequences $A = a_1, a_2, \cdots, a_d$ of d distinct elements of $[n]$ satisfying that (i) $\{a_1, \cdots, a_d\} \cap \{b_1, \cdots, b_d\} = \emptyset$ and (ii) $a_i < b_i$ for all i. Furthermore, we define

$$\chi_B = \sum_{A \in \mathcal{T}(B)} \prod_{i=1}^{d} (x_{a_i} - x_{b_i}). \tag{2.7}$$

Theorem 2.5 (Theorem 15 in [10])**.** *Let $r \leq n/2$ be an integer, the set $\{\chi_B : B \in \mathcal{B}_{n,d}$ for some $d \leq r\}$ is an orthogonal basis for the vector space of functions over the slice $\binom{[n]}{r}$. The Young-Fourier expansion of $f : \binom{[n]}{r} \to \mathbb{R}$ is the unique representation*

$$f = \sum_{B \in \mathcal{B}_{n,d},\ d \leq r} \hat{f}(B) \chi_B, \tag{2.8}$$

where $\hat{f}(B) = \frac{\langle f, \chi_B \rangle}{\|\chi_B\|_2^2}$. Here $\langle f, g \rangle := \mathbb{E}_{x \sim U}[f(x)g(x)]$. In addition for $B \in \mathcal{B}_{n,d}$,

1. *$\|\chi_B\|_2^2 = \prod_{i=1}^{d} \frac{(b_i - 2(i-1))(b_i - 2(i-1)-1)}{2} \cdot 2^d \frac{r^{\underline{d}}(n-r)^{\underline{d}}}{n^{\underline{2d}}} = n^{O(d)}$. In particular, if $r \geq \frac{n}{4}$ and $d = o(n)$, then $\|\chi_B\|_2^2 = 2^{-O(d)}$. Here, $r^{\underline{d}} = \prod_{i=0}^{d-1}(r-i)$.*

2. *$\|\chi_B\|_\infty \leq \sum_{A \in \mathcal{T}(B)} \|\chi_{A,B}\|_\infty = n^{O(d)}$.*

By Boolean duality, we can extend the above Young-Fourier expansion to where $r > n/2$ naturally. This can be done by replacing the basis $\{\chi_B(x)\}$ by $\{\chi_{\bar{B}}(x) := \chi_B(\bar{x})\}$ where $\bar{x}$ is obtained by flipping all bits of x.

Corollary 2.6. *Let $r > n/2$ be an integer, the set $\{\chi_{\bar{B}}(x) : B \in \mathcal{B}_{n,d}$ for some $d \leq n-r\}$ is an orthogonal basis for the vector space of functions over the slice $\binom{[n]}{r}$. The Young-Fourier expansion of $f : \binom{[n]}{r} \to \mathbb{R}$ is the unique representation*

$$f = \sum_{B \in \mathcal{B}_{n,d},\ d \leq n-r} \hat{f}(B) \chi_{\bar{B}}, \tag{2.9}$$

where $\hat{f}(B) = \frac{\langle f, \chi_{\bar{B}} \rangle}{\|\chi_{\bar{B}}\|_2^2}$. In addition, we have $\|\chi_{\bar{B}}\|_2^2 = \|\chi_B\|_2^2$ and $\|\chi_{\bar{B}}\|_\infty = \|\chi_B\|_\infty$.

Like for functions over the Boolean hypercube, we can define the *total weight* on level d:

Definition 2.7. Let $f : \binom{[n]}{r} \to \mathbb{R}$, for any $r \le n$, define

$$\mathbf{W}^d(f) = \sum_{B \in \mathcal{B}_{n,d}} \hat{f}(B)^2 \|\chi_B\|_2^2, \tag{2.10}$$

and denote $\mathbf{W}^{>d}(f) = \sum_{d'>d} \mathbf{W}^d$ and $\mathbf{W}^{\le d}(f) = \sum_{d' \le d} \mathbf{W}^d$.

Definition 2.8. Let $f : \binom{[n]}{r} \to \{\pm 1\}$. For $i, j \in [n]$, define the *influence* of f on the pair (i, j) as

$$\mathbf{I}_{ij}[f] = 2\Pr[f(x^{(i,j)}) \ne f(x)]. \tag{2.11}$$

Here $x^{(i,j)}$ is obtained by switching x_i and x_j. The *total influence* of f is

$$\mathbf{I}[f] = \frac{1}{n} \sum_{1 \le i < j \le n} \mathbf{I}_{ij}[f]. \tag{2.12}$$

Lemma 2.9 (Proposition 3.1 in [22]). *Let $f : \binom{[n]}{r} \to \{\pm 1\}$ and $A = \{x \in \binom{[n]}{r} : f(x) = -1\}$, then*

$$\min\{\mu(\partial^+ A), \mu(\partial^- A)\} \ge \mu(A) + \frac{n}{4r(n-r)} \cdot \mathbf{I}[f] \ge \mu(A) + \frac{1}{n}\mathbf{I}[f]. \tag{2.13}$$

Theorem 2.10 (Lemma 24 in [10]). *Let $f : \binom{[n]}{r} \to \{\pm 1\}$. Then*

$$\mathbf{I}[f] = \sum_{d \le \min(r, n-r)} \frac{d(n+1-d)}{n} \cdot \mathbf{W}^d = \sum_{B \in \mathcal{B}_{n,d}, d \le \min(r,n-r)} \frac{d(n+1-d)}{n} \cdot \hat{f}(B)^2 \|\chi_B\|_2^2. \tag{2.14}$$

In addition, according to Parseval's identity,

$$\sum_d \mathbf{W}^d = \sum_{B \in \mathcal{B}_{n,d}, d \le \min(r,n-r)} \hat{f}(B)^2 \|\chi_B\|_2^2 = \|f\|_2^2 = 1. \tag{2.15}$$

2.4 Basic inequalities

Finally, we will make use of the Hoeffding bound.

Theorem 2.11 (Hoeffding Bound, Theorem 2 in [17]). *Let $X = \sum_{i=1}^n X_i$, where $X_i \in [a_i, b_i]$ are independent random variables. Then for any $\theta > 0$,*

$$\Pr(|X - \mathbb{E}(X)| \ge \theta) \le 2\exp\left(-\frac{2\theta^2}{\sum_i (b_i - a_i)^2}\right). \tag{2.16}$$

Corollary 2.12. *Let X be a random variable with distribution $\mathcal{D}$ whose range is $[l, u]$. Let $X_1, \dots, X_m$ be its independent samples. Then w.p. $\ge 1 - \delta$, for any $\epsilon > 0$,*

$$\left| \frac{1}{m} \sum_{i=1}^m X_i - \mathbb{E}(X) \right| \le \epsilon \tag{2.17}$$

as long as $m \ge (u-l)^2 \log(2/\delta)/(2\epsilon^2)$.

The following fact will also be used.

Proposition 2.13. *For $t = o(n)$ we have $\binom{n}{n/2-t/2}/2^n = \frac{1}{\sqrt{n}} \cdot 2^{-\Theta(t^2/n)}$.*

Proof. By Stirling's approximation,

$$\log\binom{n}{n/2-t/2} = \log(1+o(1)) + \log\sqrt{\frac{n}{2\pi(n/2-t/2)(n/2+t/2)}} + n\cdot H\left(\frac{n/2-t/2}{n}\right), \quad (2.18)$$

where $H(p) = -p\log p - (1-p)\log(1-p)$ is the binary entropy function. As $t/2n = o(1)$, by the Taylor expansion of the entropy function around $1/2$, we have

$$H\left(\frac{1}{2} - \frac{t}{2n}\right) = 1 - \frac{1+o(1)}{2\ln 2}\left(\frac{t}{n}\right)^2. \quad (2.19)$$

The conclusion follows immediately. □

3 Concentration property of k-monotone functions

In the rest of this paper, for a function $f : \{0,1\}^n \to \{0,1\}$, we convert the range to $\{\pm 1\}$. The mapping from $\{0,1\}$ to $\{-1,1\}$ is given by $1-2b$, sending 0 to 1 and 1 to -1. So a function $f : \{0,1\}^n \to \{\pm 1\}$ is said to be k-monotone if $(1-f)/2$ is k-monotone.

In this section, we show that some "middle" slice of a k-monotone function has Fourier concentration. For functions $f : \{0,1\}^n \to \{\pm 1\}$, let $f|_r$ be the subfunction of f restricted to $\binom{[n]}{r}$ and $\mu(f|_r) := \mu(f|_r^{-1}(-1))$.

Definition 3.1 ((t,d,ϵ)-concentration)**.** We say $f : \{0,1\}^n \to \{\pm 1\}$ is (t,d,ϵ)-concentrated if the following holds: for some r such that $n/2 - t/2 \le r \le n/2 + t/2$,

$$\mathbf{W}^{>d}(f|_r) = \sum_{B\in\mathcal{B}_{n,d'}:d'>d} \widehat{f|_r}(B)^2\|\chi_B\|_2^2 < \epsilon. \quad (3.1)$$

Intuitively, f has low-degree Fourier concentration on at least one of the middle slices.

Lemma 3.2. *Let $f : \{0,1\}^n \to \{\pm 1\}$ be a k-monotone function. For any $1 < t \le n$ and any d,ϵ such that $d\epsilon \ge 2kn/t$, we have that f is (t,d,ϵ)-concentrated.*

Lemma 3.2 follows from an upper bound on the sum of total influences on slices.

Proposition 3.3. *Let $f : \{0,1\}^n \to \{\pm 1\}$ be a k-monotone function. Then $\sum_{r=0}^{n-1} \mathbf{I}[f|_r] \le kn$.*

We first prove Lemma 3.2 using Proposition 3.3.

Proof of Lemma 3.2. By contradiction, we assume that $\mathbf{W}^{>d}(f|_r) > \epsilon \geq \frac{2kn}{dt}$ for any $n/2 - t/2 \leq r \leq n/2 + t/2$. According to Proposition 3.3, we have $\sum_{r=\lceil n/2-t/2\rceil}^{\lfloor n/2+t/2\rfloor} \mathbf{I}[f|_r] \leq kn$. By averaging, let $n/2 - t/2 \leq r \leq n/2 + t/2$ be such that $\mathbf{I}[f|_r] \leq kn/t$. By Theorem 2.10, we can deduce that,

$$\frac{kn}{t} \geq \mathbf{I}[f|_r] = \sum_{d' \leq \min(r, n-r)} \frac{d'(n+1-d')}{n} \cdot \mathbf{W}^{d'}(f|_r) \tag{3.2}$$

$$\geq \sum_{d < d' \leq \min(r, n-r)} \frac{d'(n+1-d')}{n} \cdot \mathbf{W}^{d'}(f|_r) \tag{3.3}$$

$$\geq \frac{d(n+1-d)}{n} \cdot \sum_{d < d' \leq \min(r, n-r)} \mathbf{W}^{d'}(f|_r) \tag{3.4}$$

$$> \frac{d}{2} \cdot \epsilon \geq \frac{d}{2} \cdot \frac{2kn}{dt} \geq \frac{kn}{t}\ , \tag{3.5}$$

a contradiction. □

Now we prove Proposition 3.3.

Proof of Proposition 3.3. Suppose f is the parity of $h_1, \cdots, h_k$ where each h_i is monotone. For any r, when we switch x_i and x_j, $f|_r(x)$ changes only if at least one $h_i|_r(x)$ changes for $i = 1, 2, \ldots, k$. Thus, combining with the union bound, we have

$$\mathbf{I}[f|_r] \leq \sum_{i=1}^{k} \mathbf{I}[h_i|_r]. \tag{3.6}$$

Since h_i is monotone, the upper shadow of $h_i|_r^{-1}(-1)$ is a subset of $h_i|_{r+1}^{-1}(-1)$. Then according to Lemma 2.9, we have

$$\mu(h_i|_{r+1}) \geq \mu(h_i|_r) + \frac{1}{n} I(h_i|_r), \tag{3.7}$$

which implies

$$\frac{1}{n} \sum_{r=0}^{n-1} \mathbf{I}[h_i|_r] \leq \mu(h_i|_n) - \mu(h_i|_0) \leq 1. \tag{3.8}$$

Equation (3.6) and Equation (3.8) imply the desired conclusion. □

4 Distinguishers for k-monotone functions

In this section, we prove the following theorem.

Theorem 1.3. *Any family of k-monotone functions can be distinguished from uniformly random functions in* $\exp\left(O(n^{1/3}(k \log n)^{2/3})\right)$ *time. In other words, any family of k-monotone functions is not a* $\left(\exp\left(O(n^{1/3}(k \log n)^{2/3})\right), 1/3\right)$*-secure weak pseudorandom family.*

Algorithm 1: A Distinguisher for $(t, d, 1/2)$-Concentrated Functions

1 Let C be a large enough constant;
2 **for** $r \leftarrow \lceil \frac{n}{2} - \frac{t}{2} \rceil$ **to** $\lfloor \frac{n}{2} + \frac{t}{2} \rfloor$ **do**
3     $S \leftarrow 0$;
4   **for** $B \in \mathcal{B}_{n,d'}$ *with* $d' \leq d$ **do**
5   **if** $r \leq n/2$ **then**
6   Estimate $\langle f|_r, \chi_B \rangle$ with accuracy $n^{-C \cdot d}$;
7             $S \leftarrow S + \widehat{f|_r}(B)^2 \|\chi_B\|_2^2$;
              // $\widehat{f|_r}(B)^2 \|\chi_B\|_2^2 = \langle f|_r, \chi_B \rangle^2 / \|\chi_B\|_2^2$
8   **else**
9   Estimate $\langle f|_r, \chi_{\bar{B}} \rangle$ with accuracy $n^{-C \cdot d}$;
10            $S \leftarrow S + \widehat{f|_r}(\bar{B})^2 \|\chi_{\bar{B}}\|_2^2$;
              // $\widehat{f|_r}(B)^2 \|\chi_{\bar{B}}\|_2^2 = \langle f|_r, \chi_{\bar{B}} \rangle^2 / \|\chi_{\bar{B}}\|_2^2$
11   **if** $S \geq 3/8$ **then**
12   Return True;
13 Return False;

We prove this theorem by giving a distinguisher for $(t, d, 1/2)$-concentrated functions.

Proposition 4.1. *For $t \leq n/4$ and $d = o(n/\log n)$, any family of $(t, d, 1/2)$-concentrated functions can be distinguished from uniform random functions in $2^{O(d \log n + t^2/n)}$ time.*

By Lemma 3.2, every k-monotone function is $\left((kn^2 \log n)^{1/3}, 4\left(\frac{k^2 n}{\log n}\right)^{1/3}, 1/2\right)$-concentrated, then Theorem 1.3 follows. Now we prove the proposition.

Proof. The distinguisher is given in Algorithm 1. We will show that

- (**Soundness**) It accepts a uniform random function w.p. $o(1)$;
- (**Completeness**) It accepts any $(t, d, 1/2)$-concentrated function w.p. $1 - o(1)$;
- (**Complexity**) Its sample/time complexity is $2^{O(d \log n + t^2/n)}$.

Soundness. Let f be a uniform random function. We claim that for each $\frac{n}{2} - \frac{t}{2} \leq r \leq \frac{n}{2} + \frac{t}{2}$, the variable S in Line 11 is at most $1/4$ w.p. $1 - o(\frac{1}{n})$, which concludes the soundness by the union bound.

Fix such an r. W.l.o.g., we assume that $r \leq n/2$. For any $B \in \mathcal{B}_{n,d'}$ where $d' \leq d$, it is easily seen that $\mathbb{E}_f \left[\langle f|_r, \chi_B \rangle\right] = 0$, then by the Hoeffding bound,

$$\Pr_f \left[\left|\langle f|_r, \chi_B \rangle\right| \geq \theta\right] \leq 2\exp\left(-2\theta^2 \binom{n}{r} / \|\chi_B\|_2^2\right) \leq 2\exp\left(-\theta^2 \left(\frac{4}{3}\right)^{n/4 - o(n)}\right), \tag{4.1}$$

where the last inequality is due to that $\binom{n}{r} \geq (\frac{n}{r})^r \geq (\frac{4}{3})^{n/4}$ and $\|\chi_B\|_2^2 = 2^{O(d\log n)} = (\frac{4}{3})^{o(n)}$. In particular, by letting $\theta = \left(\frac{3}{4}\right)^{n/10}$ and using the union bound, we have that with probability at least $1 - n^{O(d)} \cdot \exp\left(-\theta^2 \left(\frac{4}{3}\right)^{n/4-o(n)}\right) = 1 - o(\frac{1}{n})$, $\left|\langle f|_r, \chi_B\rangle\right| < \theta$ for every $B \in \mathcal{B}_{n,d'}$ where $d' \leq d$. Thus, w.p. $1 - o(\frac{1}{n})$,

$$\mathbf{W}^{\leq d}[f|_r] = \sum_{B\in\mathcal{B}_{n,d'},d'\leq d} \widehat{f|_r}(B)^2\|\chi_B\|_2^2 = \sum_{B\in\mathcal{B}_{n,d'},d'\leq d} \frac{\langle f|_r, \chi_B\rangle^2}{\|\chi_B\|_2^2} \leq \sum_{B\in\mathcal{B}_{n,d'},d'\leq d} \frac{\theta^2}{\|\chi_B\|_2^2} \leq \frac{1}{8}, \tag{4.2}$$

where the last inequality holds for sufficiently large n. Finally, S is an estimate of $\mathbf{W}^{\leq d}[f|_r]$ with additive error $n^{-\Omega(d)}$.

Completeness. Let f be a $(t, d, 1/2)$-concentrated function. By definition, there is some r such that $n/2 - t/2 \leq r \leq n/2 + t/2$ and $W^{\leq d}[f|_r] > 1/2$. As S is an estimate of $\mathbf{W}^{\leq d}[f|_r]$ with additive error $n^{-\Omega(d)}$, we conclude that Algorithm 1 accepts f with high probability.

Complexity. The loop in Line 2 is repeated at most t times. In Line 4, the number of strings $B \in \mathcal{B}_{n,d'}$ with $d' \leq d$ we enumerated is at most $n^{O(d)}$. Furthermore, for each $n/2 - t/2 \leq r \leq n/2 + t/2$ and each $B \in \mathcal{B}_{n,d'}$ with $d' \leq d$, according to the Hoeffding bound, $n^{O(d)}$ uniform random samples on the slice $\binom{[n]}{r}$ are sufficient to estimate $\langle f|_r, \chi_B\rangle$ with accuracy $n^{-C\cdot d}$. In addition, a random uniform sample is from the slice $\binom{[n]}{r}$ with probability $\binom{n}{r}/2^n$, which is $\frac{1}{\sqrt{n}} \cdot 2^{-O(t^2/n)}$ according to Proposition 2.13. Thus, the total number of random samples used is at most $t \cdot n^{O(d)} \cdot n^{O(d)} \cdot 2^{O(t^2/n)} = 2^{O(d\log n + t^2/n)}$.

Besides, the function $\chi_B = \sum_{A\in\mathcal{T}(B)} \chi_{A,B}$ can be computed by enumerating all $n^{O(d)}$ strings A in $\mathcal{T}(B)$. Thus, the time complexity is also $2^{O(d\log n + t^2/n)}$. □

5 Weak learners for k-monotone functions

In this section, we prove the following theorem.

Theorem 1.4. *k-monotone functions are weakly learnable in time* $\exp\left(O\left(k\sqrt{n\log n}\right)\right)$.

We prove Theorem 1.4 by giving a weak learner for $(t, d, 1/2)$-concentrated functions. By Lemma 3.2, k-monotone functions are $\left(\sqrt{n\log n}, 4k\sqrt{\frac{n}{\log n}}, 1/2\right)$-concentrated, then Theorem 1.4 follows.

Proposition 5.1. *For $t = O(\sqrt{n\log n})$ and $d = o(n/\log n)$, Algorithm 2 weakly learns $(t, d, 1/2)$-concentrated functions in $2^{O(d\log n + t^2/n)}$ time.*

Algorithm 2: A weak learner for $(t, d, 1/2)$-concentrated functions

1 Let C be a large enough constant;
2 **for** $r \leftarrow \frac{n}{2} - \frac{t}{2}$ **to** $\frac{n}{2} + \frac{t}{2}$ **do**
3 $\quad S \leftarrow 0$;
4 $\quad p \leftarrow 1$;
5 $\quad$**for** $B \in \mathcal{B}_{n,d'}$ *with* $d' \leq d$ **do**
6 $\quad\quad$**if** $r \leq n/2$ **then**
7 $\quad\quad\quad$Estimate $\langle f|_r, \chi_B \rangle$ with accuracy $n^{-C \cdot d}$;
8 $\quad\quad\quad S \leftarrow S + \widehat{f|_r}(B)^2 \|\chi_B\|_2^2$;
9 $\quad\quad$**else**
10 $\quad\quad\quad$Estimate $\langle f|_r, \chi_{\bar{B}} \rangle$ with accuracy $n^{-C \cdot d}$;
11 $\quad\quad\quad S \leftarrow S + \widehat{f|_r}(B)^2 \|\chi_{\bar{B}}\|_2^2$;
12 $\quad$**if** $S \geq 3/8$ **then**
13 $\quad\quad$**if** $r \leq n/2$ **then**
14 $\quad\quad\quad g(x) \leftarrow \sum_{B \in \mathcal{B}_{n,d'} : d' \leq d} \widehat{f|_r}(B) \chi_B(x)$;
15 $\quad\quad$**else**
16 $\quad\quad\quad g(x) \leftarrow \sum_{B \in \mathcal{B}_{n,d'} : d' \leq d} \widehat{f|_r}(B) \chi_{\bar{B}}(x)$;
17 $\quad\quad$**while** $p > \sqrt{3}/4$ **do**
18 $\quad\quad\quad$Pick $\theta \in [-1, 1]$ uniformly at random;
19 $\quad\quad\quad$Estimate $p \leftarrow \Pr[f|_r \neq \operatorname{sign}(g - \theta)]$;
20 $\quad\quad$Estimate $\mu_{\neq r} \leftarrow \mathbb{E}_x[f(x) \mid |x| \neq r]$;
21 $\quad\quad$Return $h(x) = \begin{cases} \operatorname{sign}(g(x) - \theta) & \text{if } |x| = r; \\ \operatorname{sign}(\mu_{\neq r}) & \text{if } |x| \neq r. \end{cases}$
22 Return $h(x) \equiv 0$.

To learn $(t, d, 1/2)$-concentrated functions f, Algorithm 2 tries to find out the slice $\binom{[n]}{r}$ on which $f|_r$ is concentrated, and then figures out a function $g : \binom{[n]}{r} \to \mathbb{R}$ which is very close to $f|_r$. To convert the approximated function g to a Boolean-valued function, we can utilize Claim 5.2 similar to Exercise 3.34 in [21]. For the rest of the slices, the learner just outputs the most frequent value. Since $t = O(\sqrt{n \log n})$, each slice in $[n/2 - t/2, n/2 + t/2]$ is at least a $\frac{1}{\sqrt{n}} \cdot 2^{-O(t^2/n)} = 1/\text{poly}(n)$ fraction according to Proposition 2.13. Hence we get a $(1/2 - 1/\text{poly}(n))$-close function h.

Proof of Proposition 5.1. We first show that Algorithm 2 weakly learns $(t, d, 1/2)$-concentrated functions. Let f be a $(t, d, 1/2)$-concentrated function. For each $n/2 - t/2 \leq r \leq n/2 + t/2$, the variable S in Line 12 is an estimate of $\mathbf{W}^{\leq d}[f|_r]$ with additive error $n^{-\Omega(d)}$. Then for some r, the condition $S \geq 3/8$ in Line 12 holds, and Algorithm 2 executes Lines 13-21. For the function g obtained in Line 14 or Line 16, and sufficiently large n,

$$\|f|_r - g\|_1 \leq \|f|_r - g\|_2 = \sqrt{\sum_{B \in \mathcal{B}_{n,d'}, d' \leq d} (\widehat{f|_r}(B) - \hat{g}(B))^2 \|\chi_B\|_2^2 + W^{>d}[f|_r]} \leq \sqrt{o(1) + 5/8} \leq \sqrt{3}/2.$$

To convert g to a Boolean-valued function, we utilize the following claim.

Claim 5.2. *Suppose $f : \binom{[n]}{r} \to \{-1, 1\}$ and $g : \binom{[n]}{r} \to \mathbb{R}$. Pick $\theta \in [-1, 1]$ uniformly at random and define $g' = \operatorname{sign}(g(x) - \theta)$, we have $\mathbb{E}_\theta \left[\Pr_x \left(f(x) \neq g'(x)\right)\right] \leq \|f - g\|_1/2$.*

Proof. By rewriting the last formula and swapping the expectation operators, we have

$$\begin{aligned}\mathop{\mathbb{E}}_\theta \left[\Pr_x \left(f(x) \neq g'(x)\right)\right] &= \mathop{\mathbb{E}}_\theta \mathop{\mathbb{E}}_x \left[1_{f(x)\neq g'(x)}\right] = \mathop{\mathbb{E}}_x \mathop{\mathbb{E}}_\theta \left[1_{f(x)\neq g'(x)}\right] = \mathop{\mathbb{E}}_x \left[\Pr_\theta \left(f(x) \neq \operatorname{sign}(g(x) - \theta)\right)\right] \\ &\leq \mathop{\mathbb{E}}_x \left[\frac{|f(x) - g(x)|}{2}\right] = \frac{\|f - g\|_1}{2}.\end{aligned}$$

□

Thus, for a random $\theta \in [-1, 1]$, $\Pr\left[f|_r(x) \neq \operatorname{sign}(g(x) - \theta)\right] \leq \sqrt{3}/4$ holds with a constant probability. That is, with high probability, the loop of Lines 17-19 is repeated a constant number of times, and we will get a $\theta^\star$ such that $\Pr\left[f|_r(x) \neq \operatorname{sign}(g(x) - \theta^\star)\right] \leq \sqrt{3}/4$. Finally, we have

$$\Pr[h(x) \neq f(x)] \tag{5.1}$$

$$= \Pr[|x| \neq r] \Pr[f(x) \neq \operatorname{sign}(\mu_{\neq r}) \mid |x| \neq r] + \Pr[|x| = r] \Pr\left[f|_r(x) \neq \operatorname{sign}(g(x) - \theta^\star)\right] \tag{5.2}$$

$$\leq \left(1 - \frac{\binom{n}{r}}{2^n}\right) \cdot \frac{1}{2} + \frac{\binom{n}{r}}{2^n} \cdot \frac{\sqrt{3}}{4} = \frac{1}{2} + \frac{\binom{n}{r}}{2^n} \cdot \left(\frac{\sqrt{3}}{4} - \frac{1}{2}\right) \tag{5.3}$$

$$= \frac{1}{2} - \left(\frac{1}{2} - \frac{\sqrt{3}}{4}\right) \cdot \frac{1}{\sqrt{n}} \cdot 2^{-O(t^2/n)} = \frac{1}{2} - \frac{1}{\operatorname{poly}(n)}, \tag{5.4}$$

where the penultimate equality is according to Proposition 2.13 and the last equality is due to the assumption that $t = O(\sqrt{n \log n})$.

What remains is to show that Algorithm 2 terminates in $2^{O(d \log n + t^2/n)}$ time. First, as shown in the analysis of Algorithm 1, for each $n/2 - t/2 \leq r \leq n/2 + t/2$, it costs $2^{O(d \log n + t^2/n)}$ time to execute Lines 5-11. For some r, Algorithm 2 would execute Lines 13-21. As shown above, the loop of Lines 17-19 is repeated a constant of times. So, it costs $2^{O(d \log n + t^2/n)}$ time to execute Lines 13-21. Therefore the total time complexity is $t \cdot 2^{O(d \log n + t^2/n)} = 2^{O(d \log n + t^2/n)}$. □

6 Discussion and open problems

Fourier analysis on slices. It is surprising to us that a simple variant of the "Low-Degree Algorithm" on slices can outperform the classic "Low-Degree Algorithm" in terms of attacking negation-limited weak PRFs. To the best of our knowledge, unlike Fourier analysis on the Boolean cube, Fourier analysis on slices has not been explored in cryptography. It is a very interesting direction to use this technique to attack more cryptographic constructions, particularly ones which are secure against attacks based on standard Fourier analysis.

The hardness of 1-negation weak PRFs. One of the most intriguing open problems is how hard can 1-negation weak PRFs be? Our bound suggests that, unlike testing 1 negation (using a 1-sided non-adaptive tester) [13] and learning 1 negation to high accuracy [5], distinguishing 1 negation is significantly more efficient than $2^{O(\sqrt{n})}$. Can we have polynomial time distinguishers? We believe that new structural results of 2-monotone functions are required for polynomial time distinguishers.

Fourier spectrum of k-monotone functions on low levels. It is a basic fact [21] that every monotone function $f : \{0,1\}^n \to \{-1,1\}$ has a large Fourier coefficient on the first two levels. Does a similar statement hold for k-monotone functions? We are particularly interested in and focus on the case when $k = o(n)$. On one hand, there exist k-monotone functions f (e.g., $f = x_1 \oplus \cdots \oplus x_k$) such that $\max_{|S|\le k-1} |\hat{f}(S)| = 0$. Second, the minimum $\max_{|S|\le k} |\hat{f}(S)|$ among all k-monotone functions that we are aware of so far is about $\left(\frac{\ln(n/k)}{n/k}\right)^k = \left(\frac{k}{n}\right)^{O(k)}$, which is achieved by the parity of k $\mathrm{Tribes}_{n/k}$ functions on disjoint variables. So we are curious about the following conjectures.

Conjecture 6.1 (Rocco Servedio, 2014)**.** *For $k = o(n)$ and any k-monotone function $f : \{0,1\}^n \to \{-1,1\}$, there exists a set $S \subseteq [n]$ of size at most k such that $|\hat{f}(S)| = \left(\frac{k}{n}\right)^{O(k)}$.*

If Conjecture 6.1 is true, then applying "Low-Degree Algorithm" on the Boolean cube [19], we can distinguish k-monotone functions from uniformly random functions in $(n/k)^{O(k)}$ time. When k is constant, it also leads to a polynomial time weak learner. We conjecture $(n/k)^{\Theta(k)}$ is the correct bound of distinguishers, specifically, there exists a $\left((n/k)^{\Theta(k)}, 1/3\right)$-secure k-monotone weak pseudorandom family.

In fact, our first attempt to distinguish k-monotone functions is to prove Conjecture 6.1. So far, even the following much weaker conjecture remains open.

Conjecture 6.2 (Rocco Servedio, 2014)**.** *Let $f : \{0,1\}^n \to \{-1,1\}$ be a 2-monotone function. There exists a set $S \subseteq [n]$ of size $o(\sqrt{n})$ such that $|\hat{f}(S)| > 0$.*

We note that it is easy to derive that there exists a set $S \subseteq [n]$ of size $O(\sqrt{n})$ such that $|\hat{f}(S)| > 0$ by the $O(\sqrt{n})$ total influence upper bound of 2-monotone function shown by Blais et al [5]. We are not aware of other relevant results or implications.

Acknowledgments. We sincerely thank the ToC editors and the anonymous ToC reviewers for their detailed and constructive comments. We thank Shengyu Zhang for fruitful discussions at the early stage of this work. Siyao Guo would like to thank Igor Carboni Oliveira for telling her Conjecture 6.2 and an early version of Conjecture 6.1.

References

[1] ADI AKAVIA, ANDREJ BOGDANOV, SIYAO GUO, AKSHAY KAMATH, AND ALON ROSEN: Candidate weak pseudorandom functions in $AC^0 \circ MOD_2$. In *Proc. 5th Innovations in Theoret. Comp. Sci. Conf. (ITCS'14)*, pp. 251–260. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2014. [doi:10.1145/2554797.2554821] 2

[2] NOGA ALON AND RAVI B BOPPANA: The monotone circuit complexity of Boolean functions. *Combinatorica*, 7(1):1–22, 1987. [doi:10.1007/BF02579196] 2

[3] ALEXANDER E ANDREEV: A method for obtaining efficient lower bounds for monotone complexity. *Algebra and Logic*, 26(1):1–18, 1987. Preliminary version in Doklady 282, 1985, pp.1033-1037. Original article in *Algebra i Logika* 26(1) 3–26, 1987 (Russian). [doi:10.1007/BF01978380] 2

[4] CHRISTER BERG AND STAFFAN ULFBERG: Symmetric approximation arguments for monotone lower bounds without sunflowers. *Comput. Complexity*, 8(1):1–20, 1999. [doi:10.1007/s000370050017] 2

[5] ERIC BLAIS, CLÉMENT L. CANONNE, IGOR CARBONI OLIVEIRA, ROCCO A. SERVEDIO, AND LI-YANG TAN: Learning circuits with few negations. In *Proc. 18th Internat. Workshop on Approximation Algorithms for Combinat. Opt. Probl. (APPROX'15)*, pp. 512–527. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015. [doi:10.4230/LIPIcs.APPROX-RANDOM.2015.512] 3, 4, 5, 6, 15

[6] AVRIM BLUM, CARL BURCH, AND JOHN LANGFORD: On learning monotone Boolean functions. In *Proc. 39th FOCS*, pp. 408–415. IEEE Comp. Soc., 1998. [doi:10.1109/SFCS.1998.743491] 3

[7] ANDREJ BOGDANOV AND ALON ROSEN: Pseudorandom functions: Three decades later. In YEHUDA LINDELL, editor, *Tutorials on the Foundations of Cryptography*, pp. 79–158. Springer, 2017. [doi:10.1007/978-3-319-57048-8_3] 2

[8] CLÉMENT L. CANONNE, ELENA GRIGORESCU, SIYAO GUO, AKASH KUMAR, AND KARL WIMMER: Testing k-monotonicity: The rise and fall of Boolean functions. *Theory of Computing*, 15(1):1–55, 2019. Preliminary version in ITCS'17. [doi:10.4086/toc.2019.v015a001] 3

[9] DANA DACHMAN-SOLED, VITALY FELDMAN, LI-YANG TAN, ANDREW WAN, AND KARL WIMMER: Approximate resilience, monotonicity, and the complexity of agnostic learning. In *Proc. 26th Ann. ACM–SIAM Symp. on Discrete Algorithms (SODA'15)*, pp. 498–511. SIAM, 2015. [doi:10.1137/1.9781611973730.34] 5

[10] YUVAL FILMUS: An orthogonal basis for functions over a slice of the Boolean hypercube. *Electr. J. Combinat.*, 23(1):P1.23, 2016. [doi:10.37236/4567] 5, 7, 8

[11] YUVAL FILMUS AND ELCHANAN MOSSEL: Harmonicity and invariance on slices of the Boolean cube. In *Proc. 31st Comput. Complexity Conf. (CCC'16)*, pp. 16:1–13. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016. [doi:10.4230/LIPIcs.CCC.2016.16] 7

[12] ODED GOLDREICH, SHAFI GOLDWASSER, AND SILVIO MICALI: How to construct random functions. *J. ACM*, 33(4):792–807, 1986. [doi:10.1145/6490.6503] 2

[13] ELENA GRIGORESCU, AKASH KUMAR, AND KARL WIMMER: Flipping out with many flips: Hardness of testing k-monotonicity. *SIAM J. Comput.*, 33(4):2111–2125, 2019. [doi:10.1137/18M1217978] 3, 4, 15

[14] SIYAO GUO AND ILAN KOMARGODSKI: Negation-limited formulas. *Theoret. Comput. Sci.*, 660:75–85, 2017. [doi:10.1016/j.tcs.2016.11.027] 3

[15] SIYAO GUO, TAL MALKIN, IGOR C OLIVEIRA, AND ALON ROSEN: The power of negations in cryptography. In *Proc. Theory of Cryptography Conf. (TCC'15)*, pp. 36–65. Springer, 2015. [doi:10.1007/978-3-662-46494-6_3] 3

[16] DANNY HARNIK AND RAN RAZ: Higher lower bounds on monotone size. In *Proc. 32nd STOC*, pp. 378–387. ACM Press, 2000. [doi:10.1145/335305.335349] 2

[17] WASSILY HOEFFDING: Probability inequalities for sums of bounded random variables. *J. Amer. Statistical Assoc.*, 58(301):13–30, 1963. Also available in The Collected Works of Wassily Hoeffding, 1994, pp. 409–426, Springer and at JSTOR. See also Theorem 1 in Clayton Scott's lecture notes, U. Michigan, 2014. [doi:10.1080/01621459.1963.10500830] 8

[18] CHENGYU LIN AND SHENGYU ZHANG: Sensitivity conjecture and log-rank conjecture for functions with small alternating numbers. In *Proc. 44th Internat. Colloq. on Automata, Languages, and Programming (ICALP'17)*, pp. 51:1–13. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017. [doi:10.4230/LIPIcs.ICALP.2017.51] 3

[19] NATHAN LINIAL, YISHAY MANSOUR, AND NOAM NISAN: Constant depth circuits, Fourier transform, and learnability. *J. ACM*, 40(3):607–620, 1993. [doi:10.1145/174130.174138] 5, 15

[20] ANDREY A MARKOV: On the inversion complexity of a system of functions. *J. ACM*, 5(4):331–334, 1958. [doi:10.1145/320941.320945] 2, 6

[21] RYAN O'DONNELL: *Analysis of Boolean Functions*. Cambridge Univ. Press, 2014. [doi:10.1017/CBO9781139814782, arXiv:2105.10386] 13, 15

[22] RYAN O'DONNELL AND KARL WIMMER: KKL, Kruskal-Katona, and monotone nets. *SIAM J. Comput.*, 42(6):2375–2399, 2013. Preliminary version in FOCS'09. [doi:10.1137/100787325] 8

[23] ALEXANDER A RAZBOROV: Lower bounds for the monotone complexity of some Boolean functions. *Soviet Math. Doklady*, 31:354–357, 1985. Russian original at Mathnet.ru. 2

[24] BENJAMIN ROSSMAN: Correlation bounds against monotone NC^1. In *Proc. 30th Comput. Complexity Conf. (CCC'15)*, pp. 392–411. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015. [doi:10.4230/LIPIcs.CCC.2015.392] 3

[25] MURALI K SRINIVASAN: Symmetric chains, Gelfand–Tsetlin chains, and the Terwilliger algebra of the binary Hamming scheme. *J. Algebraic Combinatorics*, 34(2):301–322, 2011. [doi:10.1007/s10801-010-0272-2] 5, 7

[26] ÉVA TARDOS: The gap between monotone and non-monotone circuit complexity is exponential. *Combinatorica*, 8(1):141–142, 1988. [doi:10.1007/BF02122563] 2

AUTHORS

Zhihuai Chen
Researcher
Institute of Computing Technology
Chinese Academy of Sciences
Haidian District, Beijing, China
chenzhihuai@outlook.com
https://chenzhihuai.github.io/

Siyao Guo
Assistant Professor
Computer Science, NYU Shanghai
Pudong Xinqu, Shanghai, China
siyao.guo@nyu.edu
https://sites.google.com/site/siyaoguo/

Qian Li
Research Scientist
Shenzhen International Center For Industrial And Applied Mathematics
Shenzhen Research Institute of Big Data
Shenzhen, Guangdong Province, China
liqian.ict@gmail.com
https://www.sribd.cn/en/teacher/936

Chengyu Lin
Cryptographic Engineer
Espresso Systems
Menlo Park, CA, USA
linmrain@gmail.com
https://www.linkedin.com/in/chengyu-lin-3aa7687b/

Xiaoming Sun
Professor
Institute of Computing Technology
Chinese Academy of Sciences
Haidian District, Beijing, China
sunxiaoming@ict.ac.cn
http://theory.ict.ac.cn/sunxiaoming

ABOUT THE AUTHORS

ZHIHUAI CHEN received his Ph. D. from the Institute of Computing Technology, Chinese Academy of Sciences under the supervision of Xiaoming Sun. He has worked on game theory and mechanism design. Recently he has been working on a scheduling problem.

SIYAO GUO is an assistant professor of Computer Science at NYU Shanghai. She received her Ph. D. from the Chinese University of Hong Kong, advised by Andrej Bogdanov. Her research interests include computational complexity, cryptography and pseudorandomness.

QIAN LI is a research scientist at the Shenzhen International Center For Industrial And Applied Mathematics, Shenzhen Research Institute of Big Data. Previously he was an engineer at the Alibaba Group. He received his Ph. D. in 2018 from the Institute of Computing Technology, Chinese Academy of Sciences under the supervision of Xiaoming Sun. He has worked on algorithms, Boolean function analysis, quantum computing, and auction theory. He is currently interested in algorithms and machine learning.

CHENGYU LIN is a cryptographic engineer at Espresso Systems. He received his Ph. D. from Columbia University, advised by Tal Malkin. He works on multiparty computation, private information retrieval and lattice cryptography.

XIAOMING SUN is a professor at the Institute of Computing Technology, Chinese Academy of Sciences. He received his Ph. D. in 2005 from Tsinghua University, advised by Andrew Yao. He is currently interested in quantum computing.