A Polynomial Lower Bound for Testing Monotonicity

Aleksandrs Belovs
University of Latvia
stiboh@gmail.com

Eric Blais
University of Waterloo
eric.blais@uwaterloo.ca

November 27, 2018

Abstract

We show that every algorithm for testing n-variate Boolean functions for monotonicity must have query complexity $\tilde{\Omega}(n^{1/4})$. All previous lower bounds for this problem were designed for non-adaptive algorithms and, as a result, the best previous lower bound for general (possibly adaptive) monotonicity testers was only $\Omega(\log n)$. Combined with the query complexity of the non-adaptive monotonicity tester of Khot, Minzer, and Safra (FOCS 2015), our lower bound shows that adaptivity can result in at most a quadratic reduction in the query complexity for testing monotonicity.

By contrast, we show that there is an exponential gap between the query complexity of adaptive and non-adaptive algorithms for testing regular linear threshold functions (LTFs) for monotonicity. Chen, De, Servedio, and Tan (STOC 2015) recently showed that non-adaptive algorithms require almost $\Omega(n^{1/2})$ queries for this task. We introduce a new adaptive monotonicity testing algorithm that has query complexity $O(\log n)$ when the input is a regular LTF.

1 Introduction

The Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ is *monotone* iff $f(x) \leq f(y)$ for all $x \preceq y$, where $\preceq$ is the bitwise partial order on the Boolean hypercube $\{0,1\}^n$ (i.e., $x \preceq y$ iff $x_i \leq y_i$ for all $i \in [n]$). Conversely, the function f is *ϵ-far from monotone* for some $\epsilon > 0$ if for every monotone function $g\colon \{0,1\}^n \to \{0,1\}$, there are at least $\epsilon 2^n$ points $x \in \{0,1\}^n$ such that $f(x) \neq g(x)$. An *ϵ-tester for monotonicity* is a bounded-error randomized algorithm that distinguishes monotone functions from those that are ϵ-far from monotone. The tester has oracle access to the function f. It is *non-adaptive* if its queries do not depend on the oracle's responses to the previous queries; otherwise, it is *adaptive*.

The study of the monotonicity testing problem was initiated in 1998 by Goldreich, Goldwasser, Lehman, and Ron [19], who introduced the natural *edge tester* for monotonicity. This tester selects edges $x \preceq y$ of the hypercube $\{0,1\}^n$ uniformly at random and verifies that $f(x) \leq f(y)$ on each of these edges. The original analysis of the edge tester in [19] only gave a query complexity bound of $\mathrm{poly}(n)/\epsilon$. The correct bound $\Theta(n/\epsilon)$ on its complexity was obtained soon afterwards by the same authors and Samorodnitsky in the journal version [20] of the paper. (See also [16] where this bound was first presented.) This raised a question: are there any other ϵ-testers for monotonicity with significantly smaller query complexity?

1.1 Previous work on monotonicity testing

In 2002, Fischer *et al.* [18] showed that every non-adaptive tester for monotonicity has query complexity $\Omega(\log n)$.[1] This immediately implies an $\Omega(\log \log n)$ lower bound for the more general class of adaptive testers for monotonicity. Stronger lower bounds were established for more restricted classes of algorithms, like 1-sided non-adaptive algorithms [18] and even more limited *pair testers* [7]—algorithms that select pairs $x \preceq y$ of inputs from some distribution over the comparable pairs of inputs in the hypercube and check that $f(x) \leq f(y)$ on each of the selected pairs. Algorithms and strong lower bounds were also introduced for the related problem of testing monotonicity of functions with non-Boolean ranges and other domains [4, 5, 10, 16]. However, there was no further progress on Goldreich *et al.*'s original question for more than a decade, until a recent outburst of activity.

In 2013, Chakrabarty and Seshadhri [9] showed that there are indeed testers for monotonicity with query complexity asymptotically smaller than that of the edge tester. They introduced a pair tester with query complexity $\widetilde{O}(n^{7/8}\epsilon^{-3/2})$. Chen, Servedio and Tan [13] further developed these ideas to obtain a pair tester with query complexity $\widetilde{O}(n^{5/6}\epsilon^{-4})$. Khot, Minzer, and Safra [22] showed that a directed version of Talagrand's isoperimetric inequality yields a pair tester with query complexity $\widetilde{O}(\sqrt{n}/\epsilon^2)$. The authors [1] used this inequality to develop a *quantum* tester for monotonicity with query complexity $\widetilde{O}(n^{1/4}\epsilon^{-1/2})$.

On the lower bound side, in 2014, Chen, Servedio and Tan [13] established a lower bound of $\widetilde{\Omega}(n^{1/5})$ queries for all non-adaptive testers for monotonicity. This lower bound was later improved to almost $\Omega(\sqrt{n})$ by Chen, De, Servedio and Tan [11]. These recent developments essentially give a complete answer to the question of Goldreich *et al.* for non-adaptive algorithms: there exists a non-adaptive tester for monotonicity with query complexity that is quadratically smaller than that of the edge tester, and this is the best possible.

1.2 Our results

Despite all the recent progress on monotonicity, our understanding of the query complexity of *adaptive* testers for monotonicity remains far from complete. The best lower bound for

[1] Throughout the paper, we assume that $\epsilon = \Theta(1)$ in the lower bound settings.

the problem is $\Omega(\log n)$, which follows directly from the non-adaptive lower bound of Chen *et al.* [13]. This lower bound leaves open the possibility that there exist testers for monotonicity with query complexity that is *exponentially* smaller than that of the edge tester or of any other non-adaptive tester for monotonicity. Our main result eliminates this possibility.

Theorem 1. *There exists an absolute constant $\epsilon > 0$ such that any (adaptive) randomized algorithm that ϵ-tests whether an n-variate Boolean function f is monotone makes $\Omega\left(\frac{n^{1/4}}{\log^2 n}\right)$ queries to f.*

Theorem 1 shows that the query complexity of any tester for monotonicity (adaptive or not) is at most a quartic (4th power) factor better than that of the edge tester, and that adaptivity can result in at most a quadratic reduction in the query complexity for the monotonicity testing problem.

The proof of Theorem 1 is established by considering a family of functions known as *Talagrand's random DNFs*. This family of monotone functions have previously appeared in many different contexts—including DNF approximation [28], hardness amplification [6], and learning theory [24]—and are of particular interest because of their extremal noise sensitivity properties [26]. We use the same noise sensitivity properties to show that Talagrand's random DNF with $\sqrt{n}$ random input variables negated is $\Omega(1)$-far from monotone with high probability, and that a randomized algorithm with small query complexity cannot reliably distinguish original Talagrand's random DNFs from this modified version.[2]

Our approach represents a notable departure from previous lower bounds for the monotonicity testing problem, in that all the previous lower bounds [18, 13, 11] were obtained by considering linear threshold functions (LTFs)—Boolean functions of the form $f(x) = \mathrm{sgn}(\sum_{i\in[n]} w_i x_i - \theta)$ with appropriate *weight* $w_1, \ldots, w_n \in \mathbb{R}$ and *threshold* $\theta \in \mathbb{R}$ parameters. In fact, the previous lower bounds for monotonicity testing were obtained by considering a special class of LTFs known as *regular LTFs*. An LTF is *τ-regular* when the magnitude of each weight w_i is bounded by $|w_i| \leq \tau \cdot \sqrt{\sum_{j\in[n]} w_j^2}$. Regular LTFs have been studied in the context of approximating [15], learning [27], and testing [25] LTFs; the lower bounds in [13, 11] are obtained by showing that non-adaptive algorithm with small query complexity cannot reliably distinguish $O(\frac{1}{\sqrt{n}})$-regular LTFs that are monotone from those that are far from monotone.

Chen, De, Servedio, and Tan [11] asked if their approach could be generalized to obtain polynomial lower bounds on the query complexity of adaptive testers for monotonicity. We answer this question in the negative, by showing that there exists an adaptive algorithm with logarithmic query complexity that can ϵ-test monotonicity when its input is promised to be a regular LTF.

Theorem 2. *Fix $\epsilon > 0$ and $\tau > 0$. There is an adaptive algorithm $\mathcal{A}$ with query complexity*[3] *$O_{\epsilon,\tau}(1) + \log n$ that, given oracle access to the n-variate Boolean function f,*

1. *Always accepts when f is a monotone function, and*
2. *Rejects with probability at least $\frac{1}{2}$ when f is a $\frac{\tau}{\sqrt{n}}$-regular LTF that is ϵ-far from monotone.*

Combined with the lower bound of Chen *et al.* [11], Theorem 2 shows that there are natural classes of functions for which adaptivity can reduce query complexity of monotonicity testers

[2] For a more detailed discussion of why Talagrand's random DNFs are useful for establishing lower bounds on the monotonicity testing problem, see Section 3.

[3] In fact, we can restrict $\mathcal{A}$ to only query the value of the function on inputs from the middle layers of the hypercube, so it also ϵ-tests *truncated* regular LTFs for monotonicity. See Definition 7 for the definition of truncation, and Section 5.4 for more detail.

by an exponential amount. By the standard reduction between adaptive and non-adaptive algorithms, this is best possible.

The proof of Theorem 2 is obtained by analyzing a variant of the following natural adaptive tester for monotonicity. The tester selects pairs of inputs $x, y \in \{0,1\}^n$ independently at random until it finds a pair for which $f(x) \neq f(y)$, then it performs a random binary search between x and y to identify an edge (z, z') of the hypercube on which $f(z) \neq f(z')$. It accepts if and only if f is monotone on this edge. This algorithm was first suggested to the second author by Sofya Raskhodnikova. And while this algorithm and other adaptive monotonicity testers have been considered for quite some time now, to the best of our knowledge Theorem 2 yields the first separation for the query complexity of adaptive and non-adaptive monotonicity testers over any class of Boolean functions over the hypercube. (See for example the discussions in [22, §1.5], [8] and [3].)

1.3 Subsequent work

Following the publication of the conference version of this paper [2], Chen, Waingarten, and Xie [14] improved on Theorem 1 to show that $\widetilde{\Omega}(n^{1/3})$ queries are required to test monotonicity of n-variate Boolean functions. In the same paper, Chen *et al.* showed that the lower bound in Theorem 1 is tight (up to polylogarithmic factors) for the class of Talagrand random DNFs that we consider in our proof; they obtain the stronger bound by considering a different class of functions that they call *two-level Talagrand functions*. Determining whether $\Theta(n^{1/2})$ queries are required to test monotonicity or not remains an intriguing open problem.

The result in Theorem 2 has also since been generalized: Chen, Servedio, Tan, and Waingarten [12] showed that there is an algorithm that can test monotonicity of *all* linear threshold functions (and not just regular LTFs) with $\mathrm{polylog}(n)$ queries.

Organization. We discuss the proofs of Theorems 1 and 2 at a high-level in Section 3, after introducing preliminary facts and terminology. The complete proofs follow in Sections 4 and 5, respectively.

2 Preliminaries

2.1 Probability theory

We use standard concentration inequalities.

Lemma 3 (Hoeffding's inequality)**.** *Let $w \in \mathbb{R}^n$ be any real-valued vector. Then for any $t > 0$, when $X_1, \ldots, X_n$ are independent random variables taking the values $+1$ and -1 with probability $\frac{1}{2}$ each,*

$$\Pr\left[\left|\sum_{i\in[n]} w_i X_i\right| > t\right] \leq 2\mathsf{e}^{-\frac{t^2}{2\|w\|_2^2}},$$

where $\|w\|_2 = \sqrt{\sum_{i\in[n]} w_i^2}$ is the ℓ_2-norm.

Lemma 4 (Bernstein's inequality)**.** *Consider a set of n independent random variables $X_1, \ldots, X_n$, where $-1 \leq X_i \leq 1$ for all i. Let $X = \sum_{i\in[n]} X_i$. Then, for all $0 < t < \mathrm{Var}[X]$, we have*

$$\Pr\Big[\big|X - \mathbb{E}[X]\big| > t\Big] \leq 2\mathsf{e}^{-\frac{t^2}{4\,\mathrm{Var}[X]}}.$$

We also use an anti-concentration inequality that follows directly from the Berry–Esséen theorem. (See, e.g., [27].)

Lemma 5 (Berry–Esséen corollary)**.** *Fix $\tau > 0$. Let $w \in \mathbb{R}^n$ be any real-valued vector that satisfies $\max_j |w_j| \le \tau \|w\|_2$. Then for any real $a < b$, when $X_1, \ldots, X_n$ are independent random variables taking the values $+1$ and -1 with probability $\frac{1}{2}$ each,*

$$\Pr\left[a \le \sum_{i \in [n]} w_i X_i \le b\right] \le \frac{b-a}{\|w\|_2} + 2\tau.$$

2.2 Property testing lower bounds

Theorem 1 is established via a standard lemma concerning the general setting where $\mathcal{P}$ and $\mathcal{N}$ are two disjoint families of n-variate Boolean functions, an algorithm is given oracle access to a function $f \in \mathcal{P} \cup \mathcal{N}$, and its task is to determine whether $f \in \mathcal{P}$ or $f \in \mathcal{N}$. The following lemma is essentially folklore—see, e.g., [17] for usage in property testing and [29] for a related lemma. We include a short proof for completeness.

Lemma 6. *Let* Yes *and* No *be probability distributions on n-variate Boolean functions satisfying*

$$\Pr_{f \sim \mathsf{Yes}}\big[f \in \mathcal{P}\big] = 1 \qquad \text{and} \qquad \Pr_{g \sim \mathsf{No}}\big[g \in \mathcal{N}\big] = \Omega(1).$$

If q is a positive integer such that for any sequences $x_1, \ldots, x_q \in \{0,1\}^n$ and $b_1, \ldots, b_q \in \{0,1\}$,

$$\Pr_{f \sim \mathsf{Yes}}\Big[\forall i\colon f(x_i) = b_i\Big] \le (1 + o(1)) \Pr_{g \sim \mathsf{No}}\Big[\forall i\colon g(x_i) = b_i\Big] + o(2^{-q}), \tag{1}$$

then any bounded-error randomized algorithm that decides whether $f \in \mathcal{P}$ or $f \in \mathcal{N}$ makes $\Omega(q)$ queries to f.

Proof. Let $\mathcal{A}$ be a randomized decision tree that distinguishes $\mathcal{P}$ from $\mathcal{N}$. Denote $p = \Pr_{g \sim \mathsf{No}}\big[g \in \mathcal{N}\big] = \Omega(1)$. With a constant number of repetitions of $\mathcal{A}$, we may assume that $\mathcal{A}$ accepts any function $f \in \mathcal{P}$ with probability at least $1 - p/2$, and accepts each $g \in \mathcal{N}$ with probability at most $1/3$. Then,

$$\Pr_{f \sim \mathsf{Yes}}\big[\mathcal{A} \text{ accepts on } f\big] \ge 1 - \frac{p}{2} \qquad \text{and} \qquad \Pr_{g \sim \mathsf{No}}\big[\mathcal{A} \text{ accepts on } g\big] \le (1-p) + \frac{p}{3} = 1 - \frac{2p}{3},$$

Assume towards a contradiction that $\mathcal{A}$ makes at most q queries. As $\mathcal{A}$ corresponds to a probability distribution on deterministic decision trees, there exists a decision tree $\mathcal{D}$ of depth at most q such that

$$\Pr_{f \sim \mathsf{Yes}}\big[\mathcal{D} \text{ accepts on } f\big] - \Pr_{g \sim \mathsf{No}}\big[\mathcal{D} \text{ accepts on } g\big] \ge \frac{p}{6}. \tag{2}$$

Without loss of generality, we may assume that every leaf of $\mathcal{D}$ is at depth exactly q. Let L denote the set of leaves of $\mathcal{D}$. Each leaf $\ell \in L$ is characterized by two sequences $x_1, \ldots, x_q \in \{0,1\}^n$ and $b_1, \ldots, b_q \in \{0,1\}$ such that $\mathcal{D}$ ends its work in ℓ on f iff $f(x_i) = b_i$ for all i. Let $L_1 \subseteq L$ be the set of leaves on which $\mathcal{D}$ accepts. Then, by (1),

$$\begin{aligned}
\Pr_{f \sim \mathsf{Yes}}\big[\mathcal{D} \text{ accepts on } f\big] &= \sum_{\ell \in L_1} \Pr_{f \sim \mathsf{Yes}}\big[\mathcal{D} \text{ terminates in } \ell \text{ on } f\big] \\
&\le \big(1 + o(1)\big) \sum_{\ell \in L_1} \Pr_{g \sim \mathsf{No}}\big[\mathcal{D} \text{ terminates in } \ell \text{ on } g\big] + o\big(|L_1| 2^{-q}\big) \\
&= \Pr_{g \sim \mathsf{No}}\big[\mathcal{D} \text{ accepts on } g\big] + o(1),
\end{aligned}$$

contradicting (2). □

The following operation is often useful in lower bounds on monotonicity on the hypercube. It essentially reduces monotonicity testing on the whole hypercube to monotonicity testing on its middle layers. This idea comes from [18].

Definition 7. For $\delta > 0$, the *δ-truncation* of the function $f\colon \{0,1\}^n \to \{0,1\}$ is the function $\mathrm{Truncate}_\delta(f)$ defined by

$$x \mapsto \begin{cases} 0, & \text{if } |x| < \frac{n}{2} - \delta\sqrt{n}; \\ f(x), & \text{if } \frac{n}{2} - \delta\sqrt{n} \le |x| \le \frac{n}{2} + \delta\sqrt{n}; \\ 1, & \text{if } |x| > \frac{n}{2} + \delta\sqrt{n}; \end{cases}$$

where $|x|$ denotes the Hamming weight.

When f is monotone, then $\mathrm{Truncate}_\delta(f)$ is also monotone. Furthermore, for every $\epsilon > 0$, there exists $\delta > 0$ such that $\mathrm{Truncate}_\delta(f)$ is $\frac{\epsilon}{2}$-far from monotone whenever f is ϵ-far from monotone. Note that it only makes sense to query $\mathrm{Truncate}_\delta(f)$ on the inputs $x \in \{0,1\}^n$ satisfying $|x| = \frac{n}{2} \pm O(\sqrt{n})$, since otherwise the response is known in advance. We call such inputs *nearly balanced*.

2.3 Harris–Kleitman inequality and distance to monotonicity

A set $S \subseteq \{0,1\}^n$ is *monotone increasing* if for every $x \in S$ and every $y \succeq x$, we have $y \in S$. Similarly, S is *monotone decreasing* if for every $x \in S$ and every $y \preceq x$, we have $y \in S$. The Harris–Kleitman inequality bounds the size of the intersection of two monotone increasing or two monotone decreasing sets.

Lemma 8 (Harris [21], Kleitman [23])**.** *Let $A, B \subseteq \{0,1\}^n$ be a pair of sets that are both monotone increasing or both monotone decreasing. Then when x is drawn uniformly at random from $\{0,1\}^n$,*

$$\Pr[x \in A \cap B] \ge \Pr[x \in A] \cdot \Pr[x \in B].$$

Recall that the Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ is *monotone* iff $f(x) \le f(y)$ for all $x \preceq y$. The function f is *anti-monotone* if $f(x) \ge f(y)$ for all $x \preceq y$. A corollary of the Harris–Kleitman inequality shows that the distance of an anti-monotone function to monotonicity is equal to the distance to the closest constant function.

Proposition 9. *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a monotone function and $g\colon \{0,1\}^n \to \{0,1\}$ be an anti-monotone function. Then*

$$\Pr[f(x) \ne g(x)] \ge \min\{\Pr[g(x) = 0], \Pr[g(x) = 1]\}.$$

Proof. We begin by noting that

$$\Pr[f(x) \ne g(x)] = \Pr[f(x) = 1 \wedge g(x) = 0] + \Pr[f(x) = 0 \wedge g(x) = 1].$$

The sets $f^{-1}(1)$ and $g^{-1}(0)$ are both monotone increasing, and the sets $f^{-1}(0)$ and $g^{-1}(1)$ are both monotone decreasing, so applying the Harris–Kleitman inequality to both terms of the sum yields

$$\begin{aligned} \Pr[f(x) \ne g(x)] &\ge \Pr[f(x) = 1]\ \Pr[g(x) = 0] + \Pr[f(x) = 0]\ \Pr[g(x) = 1] \\ &\ge (\Pr[f(x) = 1] + \Pr[f(x) = 0]) \cdot \min\{\Pr[g(x) = 0], \Pr[g(x) = 1]\} \\ &= \min\{\Pr[g(x) = 0], \Pr[g(x) = 1]\}. \end{aligned}$$

□

2.4 Talagrand's random DNFs and noise sensitivity

Talagrand's random DNF on n variables [30] is a disjunction of $2^{\sqrt{n}}$ independent random clauses of size $\sqrt{n}$.[4] Let $\mathcal{C}$ denote the set of functions $C\colon [\sqrt{n}] \to [n]$. We call such functions C *clauses*, with $C(i)$ denoting the index of the ith variable in the clause. A clause C gives rise to the Boolean function $f_C\colon \{0,1\}^n \to \{0,1\}$ defined by $f_C(x) = \bigwedge_{i\in[\sqrt{n}]} x_{C(i)}$.

Definition 10. A *Talagrand random DNF* is a function $f\colon \{0,1\}^n \to \{0,1\}$ defined by

$$f(x) = \bigvee_{j=1}^{2^{\sqrt{n}}} f_{C_j}(x),$$

where $C_1, \ldots, C_{2^{\sqrt{n}}}$ are drawn independently and uniformly at random from $\mathcal{C}$. We denote the distribution of n-variate Talagrand random DNF functions by Tal.

Talagrand random DNF functions are monotone. They are also highly sensitive to noise in the following sense. Let $\mathcal{B}(n,\delta)$ be the probability distribution on the subsets of $[n]$, in which each element is included in the subset independently with probability δ.

Definition 11. The *noise sensitivity* of a function $f\colon \{0,1\}^n \to \{0,1\}$ at noise rate δ is

$$\mathrm{NS}_\delta(f) = \Pr_{x\sim\{0,1\}^n,\, S\sim\mathcal{B}(n,\delta)}\big[f(x) \neq f(x^S)\big],$$

where x^S denotes the input string x with the variables in S flipped.

As the following result shows, with high probability a Talagrand random DNF has high noise sensitivity.[5]

Theorem 12 (Mossel–O'Donnell [26])**.** *Talagrand's random DNF f satisfies* $\mathrm{NS}_{1/\sqrt{n}}(f) = \Omega(1)$ *with probability* $\Omega(1)$.

2.5 Linear threshold functions

In studying linear threshold functions, it is more convenient to assume that the function is of the form $f\colon \{-1,1\}^n \to \{-1,1\}$.

Definition 13 (LTF)**.** The function $f\colon \{-1,1\}^n \to \{-1,1\}$ is a *linear threshold function* (alternatively: *LTF*, or *halfspace*) with associated *weights* $w_1, \ldots, w_n \in \mathbb{R}$ and *threshold* θ if it satisfies

$$f(x) = \mathrm{sgn}\left(\sum_{i=1}^n w_i x_i - \theta\right)$$

for every $x \in \{-1,1\}^n$ where sgn is the sign function defined by $\mathrm{sgn}(x) = 1$ for non-negative x and $\mathrm{sgn}(x) = -1$ for negative x.

Definition 14 (Regular LTFs)**.** The LTF $f\colon \{-1,1\}^n \to \{-1,1\}^n$ is *τ-regular* if it can be represented with a set of weights $w_1, \ldots, w_n$ that satisfy $\max_{i\in[n]} |w_i| \leq \tau \cdot \sqrt{\sum_{i=1}^n w_i^2}$.

[4] Talagrand's original definition was for random *CNF*s. However, DNFs are more convenient than CNFs for our intended applications, and all the results about CNFs easily carry over to the DNF case by duality.

[5] Mossel and O'Donnell only postulate the existence of one such function f. However, Theorem 12 easily follows from the equation before the Proof of Theorem 3 in Section 4 of [26].

3 High-level overview and intuition

3.1 The bisection algorithm and regular LTFs

The intuition behind the proofs of Theorems 1 and 2 is best described by first examining the previous non-adaptive query complexity lower bounds of Chen *et al.* [13, 11]. In these lower bounds, two distributions $\mathcal{D}_{\mathsf{Yes}}$ and $\mathcal{D}_{\mathsf{No}}$ over a finite set of weights are defined under the two constraints that

1. Every weight in the support of $\mathcal{D}_{\mathsf{Yes}}$ is non-negative, and
2. A weight $w \sim \mathcal{D}_{\mathsf{No}}$ is negative with constant probability.

Two distributions Yes and No over n-variate LTFs are defined by drawing weights $w_1, \dots, w_n$ independently at random from the distributions $\mathcal{D}_{\mathsf{Yes}}$ and $\mathcal{D}_{\mathsf{No}}$, respectively, and then by letting

$$f(x_1, \dots, x_n) = \operatorname{sgn}(w_1 x_1 + \cdots + w_n x_n).$$

Since $\mathcal{D}_{\mathsf{Yes}}$ and $\mathcal{D}_{\mathsf{No}}$ are over finite domains (of size independent of n), the resulting function f is always an $O(\frac{1}{\sqrt{n}})$-regular LTF [11, Claim B.2]. Furthermore, the functions drawn from $\mathcal{D}_{\mathsf{Yes}}$ are always monotone, and the functions drawn from $\mathcal{D}_{\mathsf{No}}$ are $\Omega(1)$-far from monotone with large probability [11, Theorem B.9]. Chen *et al.* then obtain a lower bound on the number of queries required to test monotonicity non-adaptively by bounding the number of queries required to distinguish functions drawn from $\mathcal{D}_{\mathsf{Yes}}$ from those drawn by $\mathcal{D}_{\mathsf{No}}$. Thus, their argument also yields the following lower bound.

Theorem 15 (Chen–De–Servedio–Tan [11])**.** *For each $\delta > 0$, there exist $\epsilon, \tau = \Theta(1)$ such that $\Omega(n^{1/2-\delta})$ non-adaptive nearly balanced queries are required to ϵ-test $\frac{\tau}{\sqrt{n}}$-regular LTFs for monotonicity.*

Regular LTFs are used in the proofs of [13, 11] because with suitable weight distributions $\mathcal{D}_{\mathsf{Yes}}$ and $\mathcal{D}_{\mathsf{No}}$, appropriate central limit theorems can be used to bound the query complexity of non-adaptive algorithms. Regular LTFs, however, also have one other notable characteristic: when a $O(\frac{1}{\sqrt{n}})$-regular LTF is $\Omega(1)$-far from monotone, then a *constant* fraction of the f-sensitive edges $x \preceq y$ of the hypercube—edges of the hypercube that satisfy $f(x) \neq f(y)$—are edges where $f(x) > f(y)$ and are thus witnesses to the non-monotonicity of f.

This observation suggests a natural approach for testing monotonicity of regular LTFs: draw an edge $x \preceq y$ uniformly at random from the set of f-sensitive edges of the hypercube, and test whether f is monotone on this edge. While we do not know of any query-efficient algorithm for drawing edges from this distribution, there is a simple way—described in the bisection algorithm below—to at least guarantee that we return *some* f-sensitive edge, and this algorithm uses only a logarithmic number of queries when f is not too close to a constant function. In this algorithm, for $x, y \in \{0,1\}^n$, $\operatorname{Hybrid}(x, y)$ denotes the set of inputs $z \in \{0,1\}^n$ that satisfy $z_i = x_i$ for every index $i \in [n]$ where $x_i = y_i$.

The proof of Theorem 2 is completed by showing that a slight variant of this algorithm does indeed identify a non-monotone edge with constant probability when the input function is a regular LTF that is far from monotone. Specifically, we consider the random process on subsets of $[n]$ defined by the bisection algorithm and show that with constant probability, after $\log n - \Theta(1)$ iterations of the while loop, the set $\{i \in [n] : x_i \neq y_i\}$ has cardinality $O(1)$ and contains some coordinates with negative weights. The details are in Section 5.

Algorithm 1 Bisection algorithm

1: Draw $x, y \in \{0,1\}^n$ uniformly and independently at random until $f(x) = 0$ and $f(y) = 1$.
2: If $O(1/\epsilon)$ pairs are drawn without satisfying the condition, **accept**.
3: **while** $|\mathrm{Hybrid}(x, y)| > 2$ **do**
4: Draw $z \in \mathrm{Hybrid}(x, y)$ uniformly at random.
5: If $f(z) = 0$, update $x \leftarrow z$.
6: Otherwise if $f(z) = 1$, update $y \leftarrow z$.
7: **end while**
8: If $x \preceq y$, **accept**; otherwise **reject**.

3.2 Noise sensitivity and polynomial lower bound

Theorem 2 shows that we need functions other than regular LTFs to prove a polynomial lower bound for adaptive monotonicity testing. To find such functions, we can start by identifying functions that are far from monotone but for which the bisection algorithm rejects only with small probability.

On a function $f\colon \{0,1\}^n \to \{0,1\}$, the bisection algorithm ends its work in an f-sensitive edge xy of the hypercube. Let us say in this case that the algorithm *ends its work in variable* i, where i is the only variable where x and y differ. Thus, on each f, the bisection algorithm defines the corresponding *output probability distribution* on the variables in $[n]$. Our first observation is that negating some input variables of a function does not affect the output probability distribution of the bisection algorithm.

Proposition 16. *For each $f\colon \{0,1\}^n \to \{0,1\}$ and $S \subseteq [n]$, the output probability distributions on $[n]$ defined by the bisection algorithm on the functions f and $g(x) = f(x^S)$ are identical.*

Proof. Let

$$(x_1, y_1), (x_2, y_2), \ldots, (x_t, y_t)$$

be a transcript of the execution of the bisection algorithm on the input function f. That is, (x_i, y_i) is the value of x and y before the ith iteration of the loop in Algorithm 1. Then

$$(x_1^S, y_1^S), (x_2^S, y_2^S), \ldots, (x_t^S, y_t^S)$$

is an equiprobable transcript of the bisection algorithm on the function g, which ends its work in the same variable. □

Our next observation is that negating a small random subset of the variables to a noise-sensitive monotone function results, with high probability, in a function that is far from monotone.

Lemma 17. *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a monotone function and $0 < \delta < 1$ be a real number. Assume $\mathrm{NS}_\delta(f) = \Omega(1)$. Then, with probability $\Omega(1)$ over the choice of $S \sim \mathcal{B}(n, \delta)$, the function $g(x) = f(x^S)$ is $\Omega(1)$-far from being monotone.*

Proof. By the definition of noise sensitivity,

$$\Pr_{x \sim \{0,1\}^n,\, S \sim \mathcal{B}(n,\delta)} \big[f(x) \neq f(x^S)\big] = \Omega(1).$$

By Markov's inequality, with probability $\Omega(1)$ over the choice of $S \sim \mathcal{B}(n, \delta)$, we have

$$\Pr_{x \sim \{0,1\}^n} \big[f(x) \neq f(x^S)\big] = \Omega(1). \tag{3}$$

Let $g(x) = f(x^S)$ be defined for such a set S, and let $D(g)$ denote the least number of inputs on which we have to modify the value of g in order to make it monotone. We aim to estimate $D(g)$.

Write $x = (y, z)$ with $y \in \{0,1\}^{[n]\setminus S}$ and $z \in \{0,1\}^S$. For each y, consider the function $g_y(z) = g(y, z)$. We have $D(g) \geq \sum_y D(g_y)$. Each g_y is anti-monotone, so by Proposition 9 we have that $D(g_y) \geq \min\{g_y^{-1}(0), g_y^{-1}(1)\}$. We can lower bound the latter quantity by the number of pairs $\{z, z^S\}$ satisfying $g_y(z) \neq g_y(z^S)$. Summing over all y, we get that $D(g)$ is at least the number of pairs $\{x, x^S\}$ satisfying $f(x) \neq f(x^S)$. By (3), g is $\Omega(1)$-far from being monotone. □

These observations, along with Theorem 12, show that there are indeed functions that are far from monotone but are rejected by the bisection algorithm with only a small probability.

Proposition 18. *There exists a Boolean function $g\colon \{0,1\}^n \to \{0,1\}$ that is $\Omega(1)$-far from being monotone, but such that the bisection algorithm rejects g with probability only $O(1/\sqrt{n})$.*

Proof. Let f be a monotone functions satisfying $\mathrm{NS}_{1/\sqrt{n}}(f) = \Omega(1)$. By Theorem 12, a Talagrand random DNF satisfies this condition with probability $\Omega(1)$. Let p_i denote the probability that the Bisection algorithm ends its work in variable i when it is run on the function f. Let $S \sim \mathcal{B}(n, 1/\sqrt{n})$. Then by linearity of expectation

$$\mathbb{E}_S\left[\sum_{i\in S} p_i\right] = \sum_{i\in[n]} p_i \Pr_S[i \in S] = \frac{1}{\sqrt{n}} \sum_{i\in[n]} p_i = \frac{1}{\sqrt{n}}.$$

By Markov's inequality, and using Lemma 17, there exists S such that the function $g(x) = f(x^S)$ is $\Omega(1)$-far from being monotone and $\sum_{i\in S} p_i = O(1/\sqrt{n})$. By Proposition 16, the latter sum is exactly equal to the rejection probability of the bisection algorithm on the function g. □

This result shows that there are functions obtained by negating some variables in a Talagrand random DNF that are $\Omega(1)$-far from monotone, but such that the bisection algorithm requires $\Omega(\sqrt{n})$ queries to detect that it is non-monotone. The proof of Theorem 1 uses a very different approach—after all, there is no direct analogue of Proposition 16 that can hold for all adaptive algorithms—but the underlying ideas are the same. We show that for *any* set of $q \ll n^{1/4}$ queries, the distribution of the values returned by the monotone and the non-monotone Talagrand random DNFs are very similar. After that, we can apply Lemma 6 to complete the proof.

4 Polynomial lower bound

In this section, we prove Theorem 1. We will heavily use set-theoretic operations on the input strings of our functions. Because of that, let us substitute bit-strings $x \in \{0,1\}^n$ with the corresponding subsets $X \subseteq [n]$ such that $x_i = 1$ iff $i \in X$. Thus, our functions f will map subsets of $[n]$ into $\{0,1\}$.

We will use notation $X^{(S)}$ to denote the symmetric difference of X and S. Throughout this section we use $\mathcal{B} = \mathcal{B}(n, 1/\sqrt{n})$ to denote the probability distribution on subsets of $[n]$ where each element is included in the subset independently with probability $1/\sqrt{n}$.

4.1 Outline of the proof of Theorem 1

In this section we describe our proof of Theorem 1 assuming a crucial Lemma 24. In the following sections we prove the latter lemma.

Let us recall the distribution Tal of Talagrand's random DNFs. First, a clause is a function $C\colon [\sqrt{n}] \to [n]$, and $\mathcal{C}$ denotes the set of all clauses. A clause can be interpreted as a multiset of $[n]$. In particular, for $X \subseteq [n]$, we have $C \subseteq X$ if and only if $C(i) \in X$ for all $i \in [\sqrt{n}]$. We have

$$\Pr_{C\sim\mathcal{C}}\big[C \subseteq X\big] = \left(\frac{|X|}{n}\right)^{\sqrt{n}}, \tag{4}$$

where the probability distribution on $\mathcal{C}$ is uniform.

Denote $J = [2^{\sqrt{n}}]$, and recall that $\mathcal{C}^J$ denotes the set of functions from J to $\mathcal{C}$. We let $\mathbf{C}$ denote a sequence of clauses $(C_j)_{j\in J} = \big(C_1, C_2, \ldots, C_{2^{\sqrt{n}}}\big)$ in $\mathcal{C}^J$. The distribution Tal of Talagrand's random DNFs is defined by

$$f_{\mathbf{C}}(X) = \bigwedge\nolimits_{j\in J}\big[C_j \subseteq X\big], \tag{5}$$

where $\mathbf{C}$ is sampled from $\mathcal{C}^J$ uniformly, and $\big[C_j \subseteq X\big]$ is 1 if $C_j \subseteq X$ and 0 otherwise.

Following the discussion in Section 3.2, let us define the following distribution $\mathsf{Tal}^{\pm}$ on n-variate Boolean functions

$$\mathsf{Tal}^{\pm} = \Big\{X \mapsto f\big(X^{(S)}\big) \mid f \sim \mathsf{Tal},\ S \sim \mathcal{B}\Big\}.$$

We define two distributions for a sufficiently large constant $\delta > 0$:

$$\mathsf{Yes} = \{\mathrm{Truncate}_{\delta}(f) \mid f \sim \mathsf{Tal}\} \qquad \text{and} \qquad \mathsf{No} = \big\{\mathrm{Truncate}_{\delta}(f) \mid f \sim \mathsf{Tal}^{\pm}\big\}.$$

All the functions in Yes are monotone, and, by Theorem 12 and Lemma 17, a function in No is $\Omega(1)$-far from monotone with probability $\Omega(1)$.

In view of Lemma 6, it suffices to show that for all $q = O(n^{1/4}\log^{-2} n)$, nearly balanced input strings $X_1, \ldots, X_q \in \{0,1\}^n$, and Boolean outcomes $b_1, \ldots, b_q \in \{0,1\}$, we have

$$\Pr_{f\sim\mathsf{Tal}}\Big[\forall i\colon f(X_i) = b_i\Big] \le (1 + o(1)) \Pr_{g\sim\mathsf{Tal}^{\pm}}\Big[\forall i\colon g(X_i) = b_i\Big] + o(2^{-q}). \tag{6}$$

From this point on, we fix q, $X_1, \ldots, X_q$ and $b_1, \ldots, b_q$ as above.

Definition 19. For any set $T \subseteq [n]$, $\mathcal{M}^{(T)}$ denotes the set of sequences $\mathbf{C} \in \mathcal{C}^J$ such that the corresponding function $f_{\mathbf{C}}$ from (5) satisfies $f_{\mathbf{C}}\big(X_i^{(T)}\big) = b_i$ for all $i \in [q]$. We use $\mathcal{M}$ as a shorthand for $\mathcal{M}^{(\emptyset)}$.

The connection of this definition with (6) is via the identities

$$\Pr_{f\sim\mathsf{Tal}}\Big[\forall i\colon f(X_i) = b_i\Big] = \frac{|\mathcal{M}|}{|\mathcal{C}^J|} \qquad \text{and} \qquad \Pr_{g\sim\mathsf{Tal}^{\pm}}\Big[\forall i\colon g(X_i) = b_i\Big] = \frac{\mathbb{E}_{S\sim\mathcal{B}}\left[|\mathcal{M}^{(S)}|\right]}{|\mathcal{C}^J|}. \tag{7}$$

Here is the high-level plan for the proof of Theorem 1. Since the clauses in the sequence $\mathbf{C} = (C_j)_{j\in J}$ are drawn independently from each other, a natural idea is to decompose $\mathcal{M}^{(T)}$, where $T = S$ or $T = \emptyset$, into a Cartesian product and argue about each coordinate $j \in J$ independently. However, $\mathcal{M}^{(T)}$ does not readily decompose into a Cartesian product. But that is easy to fix in the following way. We will write $\mathcal{M}^{(T)}$ as a disjoint union $\mathcal{M}^{(T)} = \bigcup_{\tau} \mathcal{M}_{\tau}^{(T)}$ for some indices τ to be described shortly. These indices are chosen so that each $\mathcal{M}_{\tau}^{(T)}$ is decomposable into a Cartesian product $\mathcal{M}_{\tau}^{(T)} = \prod_{j\in J} \mathcal{M}_{\tau,j}^{(T)}$. We will divide all τs into good τs and bad τs. The contribution from the bad ones is estimated using the $o(2^{-q})$ term in (6). For

the good ones, we show that $|\mathcal{M}_\tau|$ and $\mathbb{E}_{S\sim\mathcal{B}}\big[|\mathcal{M}_\tau^{(S)}|\big]$ are close. The latter is proven considering each $j \in J$ independently.

Let us start with the implementation of this plan. Let $\mathcal{X} = (X_1, \dots, X_q)$ be the set of input strings and partition it into the two sets

$$\mathcal{X}_0 = \big\{X_i \mid i \in [q], b_i = 0\big\} \qquad \text{and} \qquad \mathcal{X}_1 = \big\{X_i \mid i \in [q], b_i = 1\big\}.$$

The indices τ are functions $\tau\colon \mathcal{X}_1 \to J$ that specify the index j of the first clause C_j contained in an input string $X \in \mathcal{X}_1$.

Definition 20. For each function $\tau\colon \mathcal{X}_1 \to J$, let $\mathcal{M}_\tau^{(T)}$ be the set of sequences $\mathbf{C} \in \mathcal{M}^{(T)}$ satisfying the following three conditions:

$$\begin{aligned}
&\bullet \text{ for each } X \in \mathcal{X}_0 \text{ and } j \in J, \text{ we have } C_j \not\subseteq X^{(T)};\\
&\bullet \text{ for each } X \in \mathcal{X}_1 \text{ and } j < \tau(X), \text{ we have } C_j \not\subseteq X^{(T)}; \text{ and}\\
&\bullet \text{ for each } X \in \mathcal{X}_1, \text{ we have } C_{\tau(X)} \subseteq X^{(T)}.
\end{aligned} \tag{8}$$

The sets $\{\mathcal{M}_\tau^{(T)}\}$ partition $\mathcal{M}^{(T)}$ since any sequence of clauses $\mathbf{C} \in \mathcal{M}^{(T)}$ satisfies the conditions in (8) for exactly one function τ. As a result, for every set $T \subseteq [n]$ we have that

$$|\mathcal{M}^{(T)}| = \sum_\tau |\mathcal{M}_\tau^{(T)}|,$$

where the sum is taken over all functions $\tau\colon \mathcal{X}_1 \to J$.

Definition 21. For each $j \in J$ and $\tau\colon \mathcal{X}_1 \to J$, define

$$\begin{aligned}
\mathcal{X}_{1,j}^\tau &= \{X \in \mathcal{X}_1 \mid \tau(X) = j\},\\
\mathcal{X}_{0,j}^\tau &= \mathcal{X}_0 \cup \{X \in \mathcal{X}_1 \mid j < \tau(X)\}
\end{aligned}$$

and

$$\mathcal{M}_{\tau,j}^{(T)} = \Big\{C \in \mathcal{C} \;\Big|\; \forall X \in \mathcal{X}_{1,j}^\tau\colon C \subseteq X^{(T)} \quad \text{and} \quad \forall X \in \mathcal{X}_{0,j}^\tau\colon C \not\subseteq X^{(T)}\Big\},$$

where C is an individual clause. Again, we use $\mathcal{M}_\tau$ and $\mathcal{M}_{\tau,j}$ as shorthand for $\mathcal{M}_\tau^{(\emptyset)}$ and $\mathcal{M}_{\tau,j}^{(\emptyset)}$, respectively.

By Definitions 20 and 21, for every τ and T, we have the following decomposition into a Cartesian product:

$$\mathcal{M}_\tau^{(T)} = \prod_{j\in J} \mathcal{M}_{\tau,j}^{(T)}.$$

In particular,

$$\big|\mathcal{M}_\tau^{(T)}\big| = \prod_{j\in J} |\mathcal{M}_{\tau,j}^{(T)}|. \tag{9}$$

Definition 22. We call a function $\tau\colon \mathcal{X}_1 \to J$ *good* if

$$\forall j \in J,\ \forall X, Y \in \mathcal{X}_{1,j}^\tau\colon |X \cap Y| \geq \frac{n}{2} - n^{3/4}. \tag{10}$$

Otherwise, we call τ *bad*.

We have the following estimates for bad and good τs, respectively.

Lemma 23. *We have*

$$\sum_{\tau \text{ is bad}} \frac{|\mathcal{M}_\tau|}{|\mathcal{C}^J|} = o(2^q).$$

Lemma 24. *For any good τ,*

$$\big|\mathcal{M}_\tau\big| \le \big(1 + o(1)\big) \mathop{\mathbb{E}}_{S\sim\mathcal{B}}\big[\big|\mathcal{M}_\tau^{(S)}\big|\big].$$

We prove these two lemmata later, but now show how Theorem 1 follows from them.

Proof of Theorem 1 from Lemmata 23 and 24. Recall that to complete the proof of the theorem, we wish to establish the inequality (6). By (7), the left-hand side of this inequality satisfies

$$\Pr_{f\sim\mathsf{Tal}}\Big[\forall i\colon f(X_i) = b_i\Big] = \frac{|\mathcal{M}|}{|\mathcal{C}^J|} = \frac{\sum_\tau |\mathcal{M}_\tau|}{|\mathcal{C}^J|} = \sum_{\tau \text{ is good}} \frac{|\mathcal{M}_\tau|}{|\mathcal{C}^J|} + \sum_{\tau \text{ is bad}} \frac{|\mathcal{M}_\tau|}{|\mathcal{C}^J|}.$$

Lemma 23 shows that the second sum is bounded by $o(2^q)$. For the first sum, Lemma 24 yields

$$\sum_{\tau \text{ is good}} \frac{|\mathcal{M}_\tau|}{|\mathcal{C}^J|} \le (1 + o(1)) \sum_{\tau \text{ is good}} \frac{\mathbb{E}_{S\sim\mathcal{B}}\big[\big|\mathcal{M}_\tau^{(S)}\big|\big]}{|\mathcal{C}^J|} \le (1 + o(1)) \frac{\mathbb{E}_{S\sim\mathcal{B}}\big[\big|\mathcal{M}^{(S)}\big|\big]}{|\mathcal{C}^J|},$$

and (6) holds by (7). □

Now we continue with the proof of Lemmata 23 and 24.

Proof of Lemma 23. Since each sequence of clauses $\mathbf{C}$ is in exactly one set $\mathcal{M}_\tau$,

$$\sum_{\tau \text{ is bad}} \frac{|\mathcal{M}_\tau|}{|\mathcal{C}^J|} = \Pr_{\mathbf{C}\sim\mathcal{C}^J}\Big[\text{exists bad } \tau \text{ such that } \mathbf{C} \in \mathcal{M}_\tau\Big]. \tag{11}$$

A τ is bad if $\exists j \in J\ \exists X, Y \in \mathcal{X}_{1,j}^\tau\colon |X \cap Y| < n/2 - n^{3/4}$. Also, $\mathbf{C} \in \mathcal{M}_\tau$ implies that $C_j \subseteq X$ for any $X \in \mathcal{X}_{1,j}$. Hence, any $\mathbf{C}$ satisfying the condition in (11) also satisfies

$$\exists j \in J,\ \exists X, Y \in \mathcal{X}\colon \Big(|X \cap Y| < \frac{n}{2} - n^{3/4}\Big) \wedge \Big(C_j \subseteq X \cap Y\Big). \tag{12}$$

By (4) and the union bound, the probability that $\mathbf{C}$ satisfies (12) is at most

$$2^{\sqrt{n}} q^2 \left(\frac{\frac{n}{2} - n^{3/4}}{n}\right)^{\sqrt{n}} = q^2\Big(1 - 2n^{-1/4}\Big)^{\sqrt{n}} \le q^2 \mathsf{e}^{-2n^{1/4}},$$

which is $o(2^{-q})$ when $q = O(n^{1/4}/\log^2 n)$. □

Lemma 24, which is the crux of our proof, follows from two Lemmata 26 and 27, which we prove in Sections 4.3 and 4.4, respectively. From this point on assume a good function τ is fixed.

Definition 25. For a fixed τ, we decompose the set of indices J into two parts $J = J_1 \cup J_0$, where

$$J_1 = \{\tau(X) \mid X \in \mathcal{X}_1\} \qquad \text{and} \qquad J_0 = J \setminus J_1.$$

Note that $|J_1| \le q < n^{1/4}$, while $|J_0| \approx 2^{\sqrt{n}}$. Also, J_1 is precisely the set of those j for which $\mathcal{X}_{1,j}$ is non-empty.

We prove Lemma 24 by considering the sets J_1 and J_0 separately. For J_1, the main technical challenge is that in general, for indices $j \in J_1$, both $X^\tau_{1,j}$ and $X^\tau_{0,j}$ are non-empty. But the set J_1 is small so we can use relatively crude estimates to obtain the following result.

Lemma 26. *Assume τ is good. Then*

$$\Pr_{S\sim\mathcal{B}}\Big[\prod_{j\in J_1}|\mathcal{M}_{\tau,j}| \le \big(1+o(1)\big)\prod_{j\in J_1}|\mathcal{M}^{(S)}_{\tau,j}|\Big] = 1-o(1).$$

For J_0, the analysis is a bit easier, since for every $j \in J_0$ we have $X^\tau_{1,j} = \emptyset$. However, we must use only accurate estimates since $|J_0| \approx 2^{\sqrt{n}}$. Doing so, we obtain the following bound.

Lemma 27. *Assume τ is good. Then*

$$\Pr_{S\sim\mathcal{B}}\Big[\prod_{j\in J_0}|\mathcal{M}_{\tau,j}| \le \big(1+o(1)\big)\prod_{j\in J_0}|\mathcal{M}^{(S)}_{\tau,j}|\Big] = 1-o(1).$$

Proof of Lemma 24 from Lemmata 26 and 27. Using (9) and the union bound, we get from Lemmata 26 and 27 that

$$\Pr_{S\sim\mathcal{B}}\Big[\big|\mathcal{M}_\tau\big| \le \big(1+o(1)\big)\big|\mathcal{M}^{(S)}_\tau\big|\Big] = 1-o(1).$$

Denote the above event by W. Then,

$$\mathop{\mathbb{E}}_{S\sim\mathcal{B}}\big[|\mathcal{M}^{(S)}_\tau|\big] \ge \Pr_{S\sim\mathcal{B}}[W]\cdot\mathop{\mathbb{E}}_{S\sim\mathcal{B}}\Big[|\mathcal{M}^{(S)}_\tau| \;\Big|\; W\Big] \ge (1-o(1))\cdot(1-o(1))\big|\mathcal{M}_\tau\big|,$$

from which Lemma 24 follows. □

4.2 A simple lemma

In this section, we prove a simple lemma that will be used in the proofs of both Lemmata 26 and 27. Let $\gamma = \omega(\sqrt{n})$. Define a graph G^γ on the vertex set $\mathcal{X}$ defined in Section 4.1, where two vertices X and Y are connected iff $|X \cap Y| \ge n/2 - \gamma$.

Lemma 28. *For every non-empty connected subset $\mathcal{A}$ of vertices of G^γ, we have*

$$\Big|\bigcap_{X\in\mathcal{A}} X\Big| \ge \frac{n}{2} - O\big(|\mathcal{A}|\gamma\big) \qquad \text{and} \qquad \Big|\bigcup_{X\in\mathcal{A}} X\Big| \le \frac{n}{2} + O\big(|\mathcal{A}|\gamma\big). \tag{13}$$

Proof. We prove the first inequality in (13), the second one being similar. The proof is by induction on the size of $\mathcal{A}$. The base case $|\mathcal{A}| = 1$ follows from the fact that $X \in \mathcal{X}$ is nearly balanced.

For the inductive step, take a vertex $Y \in \mathcal{A}$ such that $\mathcal{A}\setminus\{Y\}$ is connected. Let Z be a neighbour of Y in $\mathcal{A}\setminus\{Y\}$. By the inductive hypothesis,

$$\Big|\bigcap_{X\in\mathcal{A}\setminus\{Y\}} X\Big| \ge \frac{n}{2} - O\big((|\mathcal{A}|-1)\gamma\big).$$

Also,

$$\Big|\bigcap_{X\in\mathcal{A}\setminus\{Y\}} X\Big| - \Big|\bigcap_{X\in\mathcal{A}} X\Big| \le |Z| - |Z\cap Y| = O(\gamma),$$

since Z is nearly balanced and $|Y \cap Z| \ge n/2 - \gamma$. Combining the last two inequalities, we obtain (13). □

4.3 Proof of Lemma 26

Recall that a good τ is fixed in this section. Also, we only work with J_1, so a sequence of clauses $\mathbf{C}$ will be over the set J_1, that is $\mathbf{C} = (C_j)_{j\in J_1}$. Let, for $T \subseteq [n]$,

$$\mathcal{P}^{(T)} = \prod_{j\in J_1} \mathcal{M}^{(T)}_{\tau,j}.$$

Again, $\mathcal{P} = \mathcal{P}^{(\emptyset)}$. Thus, we have to show that

$$\Pr_{S\sim\mathcal{B}}\Big[\big|\mathcal{P}\big| \le (1+o(1))\big|\mathcal{P}^{(S)}\big|\Big] = 1 - o(1). \tag{14}$$

For $j \in J_1$, let us denote

$$Y_j = \bigcap_{X\in\mathcal{X}^\tau_{1,j}} X, \qquad \text{and} \qquad Z_j = \bigcup_{X\in\mathcal{X}^\tau_{1,j}} X.$$

Since every X in $\mathcal{X}^\tau_{1,j}$ is nearly balanced, $|Y_j| \le \frac{n}{2} + O(\sqrt{n})$. Furthermore, using Lemma 28 with $\gamma = n^{3/4}$, we get from (10) that

$$\frac{n}{2} - O(q\cdot n^{3/4}) \le |Y_j| \qquad \text{and} \qquad |Z_j| \le \frac{n}{2} + O\big(|\mathcal{X}^\tau_{1,j}|\cdot n^{3/4}\big).$$

Claim 29. *With probability $1-o(1)$, a set $S\sim\mathcal{B}$ satisfies*

$$|S| = O(\sqrt{n}), \tag{15}$$

$$|S\cap Y_j| \le \frac{\sqrt{n}}{2} + O\Big(n^{1/4}\sqrt{\log n}\Big), \quad \textit{and} \tag{16}$$

$$|S\setminus Z_j| \ge \frac{\sqrt{n}}{2} - O\Big(|\mathcal{X}^\tau_{1,j}|n^{1/4}\sqrt{\log n}\Big) \tag{17}$$

for all $j\in J_1$.

Proof. For $S\setminus Z_j$, we have

$$\mathbb{E}\big[|S\setminus Z_j|\big] \ge \frac{\sqrt{n}}{2} - O\Big(|\mathcal{X}^\tau_{1,j}|n^{1/4}\Big) \qquad \text{and} \qquad \operatorname{Var}\big[|S\setminus Z_j|\big] \le \sqrt{n}.$$

And similar estimates can be obtained for $S\cap Y_j$. Bernstein's inequality shows that for any $j\in J_1$ the conditions (15)–(17) each hold with probability $1-O(1/n)$ when the constants in the $O(\cdot)$ terms in the conditions are large enough. The claim is obtained using the union bound. □

In the remaining part of this section, we assume that the set S satisfies the conditions (15)—(17).

For each sequence of clauses $\mathbf{C}\in\mathcal{P}$, we have $C_j(a)\in Y_j$ for each $j\in J_1$ and $a\in[\sqrt{n}]$. Also, for each $j\in J_1$ and $X\in\mathcal{X}^\tau_{0,j}$, there exists $a\in[\sqrt{n}]$ such that $C_j(a)\notin X$.

Definition 30. For any $\mathbf{C}\in\mathcal{P}$, $j\in J_1$, and $X\in\mathcal{X}^\tau_{0,j}$, an index $a\in[\sqrt{n}]$ is the *pivotal index* for the clause C_j with respect to X if $C_j(a)\notin X$ and every $a'<a$ satisfies $C_j(a')\in X$.

When a is the pivotal index for C_j with respect to X, then $C_j(a)\in[n]$ is the *pivotal element* for C_j with respect to X. A *pivotal element* of a sequence of clauses $\mathbf{C}\in\mathcal{P}$ is an element which is pivotal for C_j with respect to X for some choice of $j\in J_1$ and $X\in\mathcal{X}^\tau_{0,j}$.

The pivotal element $C_j(a)$ can be interpreted as our choice of witness that $C_j \not\subseteq X$. Each clause C_j has at most $|\mathcal{X}_{0,j}| \le q$ pivotal elements, and the total number of pivotal elements over all clauses C_j for a fixed sequence $\mathbf{C} = (C_j)_{j\in J_1}$ is at most $q \cdot |J_1| \le q^2$.

Let us outline how we prove (14). For most S, we define a large (an $1 - o(1)$ fraction of) subset $\mathcal{R}_S$ of $\mathcal{P}$ and construct an injection that maps a sequence of clauses $\mathbf{C} \in \mathcal{R}_S$ into a sequence of clauses $\mathbf{C}' \in \mathcal{P}^{(S)}$. That is, we transform a clause C_j into a clause C'_j such that $C'_j \subseteq X^{(S)}$ for all $X \in \mathcal{X}^\tau_{1,j}$ and $C'_j \not\subseteq X^{(S)}$ for all $X \in \mathcal{X}^\tau_{0,j}$. Moreover, this transformation is injective.

Here are some more details. First, we do not touch the elements of C_j that lie outside of S since their membership in X is not altered by the mapping $X \mapsto X^{(S)}$. The remaining elements of C_j are *shifted* inside S. We avoid shifting the pivotal elements by simply requiring that they all lie outside of S. Since there are few of them, we can do this without reducing the size of $\mathcal{R}_S$ too much. Thus, we ensure that $C'_j \not\subseteq X^{(S)}$ for all $X \in \mathcal{X}^\tau_{0,j}$. The non-pivotal elements are shifted from $Y_j \cap S$ to $S \setminus Z_j$, which is "the new $Y_j \cap S$." This ensures that $C'_j \subseteq X^{(S)}$ for all $X \in \mathcal{X}^\tau_{1,j}$.

For each $S \subseteq [n]$ satisfying (15)—(17) and $j \in J_1$, let us fix an arbitrary subset S_j of $S \cap Y_j$ of size

$$|S_j| = \max\bigl\{0, |S \cap Y_j| - |S \setminus Z_j|\bigr\} = O\Bigl(|\mathcal{X}^\tau_{1,j}| n^{1/4}\sqrt{\log n}\Bigr). \tag{18}$$

We define two auxiliary subsets of $\mathcal{P}$.

Definition 31. A sequence $\mathbf{C} \in \mathcal{P}$ is called *half-restricted* (with respect to S) if all its pivotal elements lie outside of S. Denote the set of half-restricted $\mathbf{C}$ by $\mathcal{H}_S$.

A sequence $\mathbf{C} \in \mathcal{H}_S$ is called *restricted* (with respect to the set S) if for all $j \in J_1$ and $a \in [\sqrt{n}]$ we have $C_j(a) \notin S_j$. Denote the set of restricted $\mathbf{C}$ by $\mathcal{R}_S$.

Lemma 26 follows from the following three claims together with Claim 29.

Claim 32. *We have*

$$\Pr_{S\sim\mathcal{B}}\bigl[|\mathcal{P}| \le \bigl(1 + o(1)\bigr)|\mathcal{H}_S|\bigr] = 1 - o(1).$$

Proof. For each $i \in [n]$, let d_i denote the number of sequences $\mathbf{C} \in \mathcal{P}$ for which i is a pivotal element. For any set $S \subseteq [n]$, the number of half-restricted sequences with respect to S is $|\mathcal{H}_S| \ge |\mathcal{P}| - \sum_{i\in S} d_i$. To prove the claim, we want to bound $\sum_{i\in S} d_i$ when $S \sim \mathcal{B}$.

Each sequence $\mathbf{C}$ in $\mathcal{P}$ has at most q^2 pivotal elements, so

$$\sum_{i\in[n]} d_i \le |\mathcal{P}| \cdot q^2 = o\bigl(|\mathcal{P}|\sqrt{n}\bigr).$$

In particular, $\mathbb{E}_{S\sim\mathcal{B}}\Bigl[\sum_{i\in S} d_i\Bigr] = o(|\mathcal{P}|)$. By Markov's inequality, with probability $1 - o(1)$, we have $\sum_{i\in S} d_i = o(|\mathcal{P}|)$ and therefore $|\mathcal{H}_S| \ge (1 - o(1))|\mathcal{P}|$, implying the claim. □

Claim 33. *For every set $S \subseteq [n]$ that satisfies conditions (15)–(17), we have*

$$|\mathcal{H}_S| \le \bigl(1 + o(1)\bigr)|\mathcal{R}_S|.$$

Proof. In this case, it is easier to consider each $j \in J_1$ independently. Again, the conditions for different j are independent, hence[6]:

$$|\mathcal{R}_S| = \prod_{j\in J_1} |\mathcal{R}_j| \qquad \text{and} \qquad |\mathcal{H}_S| = \prod_{j\in J_1} |\mathcal{H}_j|,$$

[6]as S is fixed here, we will drop S from all the new pieces of notation introduced in the proof of this claim

where $\mathcal{R}_j$ and $\mathcal{H}_j$ are the projections of $\mathcal{R}_S$ and $\mathcal{H}_S$ onto the jth component. We prove that

$$|\mathcal{H}_j| \le \mathsf{e}^{O\left(n^{-1/4}\sqrt{\log n}|\mathcal{X}^\tau_{1,j}|\right)}|\mathcal{R}_j|, \tag{19}$$

which implies the claim, since then

$$\frac{|\mathcal{H}_S|}{|\mathcal{R}_S|} = \prod_{j\in J_1}\frac{|\mathcal{H}_j|}{|\mathcal{R}_j|} \le \mathsf{e}^{O\left(n^{-1/4}\sqrt{\log n}\sum_{j\in J_1}|\mathcal{X}^\tau_{1,j}|\right)} \le \mathsf{e}^{O\left(n^{-1/4}\sqrt{\log n}\cdot q\right)} = 1+o(1),$$

as $\mathcal{X}^\tau_{1,j}$ do not overlap for different $j \in J_1$.

Now let us prove (19). Fix j, and now C and C' will denote individual clauses. Let $\mathcal{H}_{j,k}$ denote the subset of clauses C' in $\mathcal{H}_j$ such that for exactly k values of $a \in [\sqrt{n}]$, we have $C'(a) \in S_j$. In particular, $\mathcal{R}_j = \mathcal{H}_{j,0}$, and $\mathcal{H}_j = \bigcup_{k\ge 0}\mathcal{H}_{j,k}$. We say that a clause $C \in \mathcal{R}_j$ is *in relation* with a clause $C' \in \mathcal{H}_{j,k}$ iff $C(a) = C'(a)$ whenever $C'(a) \notin S_j$.

Clearly, each clause $C \in \mathcal{R}_j$ is in relation with at most $\binom{\sqrt{n}}{k}|S_j|^k$ clauses in $\mathcal{H}_{j,k}$. Next, we claim that if we take a clause $C' \in \mathcal{H}_{j,k}$ and substitute each $C'(a) \in S_j$ with an element of $Y_j \setminus S$, we get a clause $C \in \mathcal{R}_j$.

In order to see this, we have to check few things. First, the new value of $C(a)$ lies in Y_j, hence, $C \subseteq X$ for all $X \in \mathcal{X}^\tau_{1,j}$, since $C' \subseteq X$ by definition. Second, any $C'(a)$ that was changed was not a pivotal element of C' since $S_j \subseteq S$ and $C' \in \mathcal{H}_j$. This shows that $C \not\subseteq X$ for all $X \in \mathcal{X}^\tau_{0,j}$. This together already shows that $C \in \mathcal{M}_{\tau,j}$.

To show that $C \in \mathcal{R}_j$ it remains to show that all its pivotal elements lie outside of S. But this is the case since each pivotal element of C is either a pivotal element of C', or a value of $C(a)$ that was changed.

Hence, each clause $C' \in \mathcal{H}_{j,k}$ is in relation with at least $|Y_j \setminus S|^k$ clauses in $\mathcal{R}_j$. Using double counting,

$$\frac{|\mathcal{H}_{j,k}|}{|\mathcal{R}_j|} \le \frac{\binom{\sqrt{n}}{k}|S_j|^k}{|Y_j\setminus S|^k} \le \frac{n^{k/2}/k!\cdot\left(O(|\mathcal{X}^\tau_{1,j}|n^{1/4}\sqrt{\log n})\right)^k}{\left(\Omega(n)\right)^k} = \frac{1}{k!}\left(O\Big(\frac{|\mathcal{X}^\tau_{1,j}|\sqrt{\log n}}{n^{1/4}}\Big)\right)^k.$$

Hence,

$$\frac{|\mathcal{H}_j|}{|\mathcal{R}_j|} \le 1+\sum_{k\ge 1}\frac{1}{k!}\left(O\Big(\frac{|\mathcal{X}^\tau_{1,j}|\sqrt{\log n}}{n^{1/4}}\Big)\right)^k = \mathsf{e}^{O\left(n^{-1/4}\sqrt{\log n}|\mathcal{X}^\tau_{1,j}|\right)}. \qquad \square$$

Claim 34. *For every S, we have $|\mathcal{R}_S| \le \left|\mathcal{P}^{(S)}\right|$.*

Proof. We use a shifting argument where we move elements from $S\cap Y_j$ to $S\setminus Z_j$. More precisely, let $\pi_j\colon S\cap Y_j\setminus S_j \to S\setminus Z_j$ be any injective mapping, which exists due to (18). Define the mapping $\pi\colon \mathcal{R}_S \to \mathcal{P}^{(S)}$ by setting $\pi\colon \mathbf{C}\mapsto\mathbf{C}'$, where for each $j\in J_1$ and $a\in[\sqrt{n}]$

$$C'_j(a) = \begin{cases}\pi_j(C_j(a)), & \text{if } C_j(a)\in S\cap Y_j\setminus S_j;\\ C_j(a), & \text{if } C_j(a)\in Y_j\setminus S.\end{cases}$$

The mapping π is well-defined since every sequence $\mathbf{C}$ in $\mathcal{R}_S$, each $j\in J_1$ and $a\in[\sqrt{n}]$ satisfy $C_j(a)\in Y_j\setminus S_j$. The mapping π is injective because the mapping π_j is injective.

To complete the proof, we show that the image of π is a subset of $\mathcal{P}^{(S)}$. Fix any $j\in J_1$. We complete the argument by showing that $C'_j \in \mathcal{M}^{(S)}_{\tau,j}$.

- For any $X\in\mathcal{X}^\tau_{1,j}$ and $a\in[\sqrt{n}]$, we have that $C_j(a)\in X$. If $C_j(a)\notin S$ then $C'_j(a) = C_j(a)\in X^{(S)}$ as well. If $C_j(a)\in S$ then $C'_j(a)\in S\setminus Z_j$ and we again have $C'_j(a)\in X^{(S)}$. Hence, $C'_j\subseteq X^{(S)}$.
- For any $X\in\mathcal{X}^\tau_{0,j}$, consider the pivotal index a corresponding to C_j and X. The corresponding element, $C_j(a)\notin S$ so $C'_j(a) = C_j(a)\notin X^{(S)}$ and therefore $C'_j\not\subseteq X^{(S)}$. $\square$

4.4 Proof of Lemma 27

Again, a good τ is fixed in this section. As in Section 4.1, we treat pairs of inputs that are far from each other separately. Let a parameter $\gamma = \Theta(\sqrt{n}\log n)$ be specified later. Define the graph G^γ as in Section 4.2. Let $\mathcal{A}_1, \dots, \mathcal{A}_\varkappa$ be the connected components of G^γ, and

$$Z_k = \bigcap_{X \in \mathcal{A}_k} X.$$

Using Lemma 28, we get that

$$|Z_k| \geq \frac{n}{2} - O(q\gamma) \geq \frac{n}{2} - O(n^{3/4}).$$

Claim 35. *With probability $1 - o(1)$, a set $S \sim \mathcal{B}$ satisfies*

$$|S| = O(\sqrt{n}), \tag{20}$$

$$|S \cap X| \geq \frac{\sqrt{n}}{2} - O\Big(n^{1/4}\sqrt{\log n}\Big), \quad \textit{and} \tag{21}$$

$$|S \setminus Z_k| \leq \frac{\sqrt{n}}{2} + O\Big(n^{1/4}\sqrt{\log n}\Big) \tag{22}$$

for all $X \in \mathcal{X}$ and $k \in [\varkappa]$.

Proof. By Bernstein's inequality as in Claim 29, if the $O(\cdot)$ factors are chosen appropriately. □

In this section, we work with individual $j \in J$ and C will denote a clause from $\mathcal{C}$.

Definition 36. For a subset $\mathcal{X}' \subseteq \mathcal{X}$ and a set $T \subseteq [n]$, define

$$\mathcal{P}^{(T)}[\mathcal{X}'] = \Big\{C \in \mathcal{C} \mid \exists x \in \mathcal{X}' \colon C \subseteq X^{(T)}\Big\}.$$

Again, $\mathcal{P}[\mathcal{X}'] = \mathcal{P}^{(\emptyset)}[\mathcal{X}']$. Note also that $\mathcal{P}^{(T)}[\mathcal{X}^\tau_{0,j}] = \mathcal{C} \setminus \mathcal{M}^{(T)}_{\tau,j}$ for all $j \in J_0$.

Lemma 27 follows from the following result:

Lemma 37. *For all S satisfying (20)–(22) and all $\mathcal{X}' \subseteq \mathcal{X}$,*

$$\Big|\mathcal{P}[\mathcal{X}']\Big| \geq \Big(1 - O\big(\big(n^{-1/4}\log^{3/2} n\big)\Big)\Big|\mathcal{P}^{(S)}[\mathcal{X}']\Big|.$$

Proof of Lemma 27 from Lemma 37. Fix S satisfying (20)–(22), and denote

$$p_j = \frac{\big|\mathcal{P}[\mathcal{X}^\tau_{0,j}]\big|}{|\mathcal{C}|} \qquad \text{and} \qquad p_j^{(S)} = \frac{\big|\mathcal{P}^{(S)}[\mathcal{X}^\tau_{0,j}]\big|}{|\mathcal{C}|}.$$

Since every $X \in \mathcal{X}$ is balanced and S satisfies (20), we have that $|X^{(S)}| \leq n/2 + O(\sqrt{n})$. Thus, using (4):

$$p_j^{(S)} = \Pr_{C\sim\mathcal{C}}\Big[C \in \mathcal{P}^{(S)}[\mathcal{X}^\tau_{0,j}]\Big] \leq \sum_{X \in \mathcal{X}^\tau_{0,j}} \Pr_{C\sim\mathcal{C}}\big[C \subseteq X^{(S)}\big] \leq |\mathcal{X}^\tau_{0,j}|\left(\frac{\frac{n}{2} + O(\sqrt{n})}{n}\right)^{\sqrt{n}} = O(q2^{-\sqrt{n}}).$$

By Lemma 37 we have

$$\begin{aligned}\frac{|\mathcal{M}_{\tau,j}|}{|\mathcal{M}^{(S)}_{\tau,j}|} = \frac{1-p_j}{1-p_j^{(S)}} &\leq \frac{1 - \big(1 - O(n^{-1/4}\log^{3/2} n)\big)p_j^{(S)}}{1 - p_j^{(S)}} \\ &= 1 + O\Big(n^{-1/4}\log^{3/2} n \cdot q2^{-\sqrt{n}}\Big) = 1 + o\big(2^{-\sqrt{n}}\big).\end{aligned} \tag{23}$$

Therefore, taking the product of (23) over all $j \in J_0$, we have

$$\prod_{j \in J_0} \frac{|\mathcal{M}_{\tau,j}|}{|\mathcal{M}^{(S)}_{\tau,j}|} \leq \Big(1 + o(2^{-\sqrt{n}})\Big)^{2^{\sqrt{n}}} = 1 + o(1).$$

This together with Claim 35 proves Lemma 27. □

We end this section with a proof of Lemma 37. Up to the end of the section, a subset $S \subseteq [n]$ satisfying (20)–(22) and a subset $\mathcal{X}' \subseteq \mathcal{X}$ are fixed. We will drop $\mathcal{X}'$ from notation, so that $\mathcal{P}^{(T)} = \mathcal{P}^{(T)}[\mathcal{X}']$.

The main idea, like in Section 4.3, is to perform shifting on the elements inside S. However, this time our situation is different, so we present the shifting argument in a different form. We use the following definition to group together different C that can be obtained by shifting their elements inside S.

Definition 38. A *partial clause* is a function $B\colon D \to [n] \setminus S$ defined on a subset $D \subseteq [\sqrt{n}]$. The domain D is denoted by $\mathrm{dom}(B)$. We say that a clause $C\colon [\sqrt{n}] \to [n]$ *extends* a partial clause B if $C(a) = B(a)$ for all $a \in \mathrm{dom}(B)$ and $C(a) \in S$ for all $a \in [\sqrt{n}] \setminus \mathrm{dom}(B)$. We use $\mathcal{P}^{(T)}_B$ to denote the subset of clauses in $\mathcal{P}^{(T)}$ extending a partial clause B.

Alternatively, one can say that a clause C extends a partial clause B if and only if B is C restricted to the domain $[\sqrt{n}] \setminus C^{-1}(S)$. Similarly as we did with clauses, we treat partial clauses B as multisets. Note that we have a decomposition into pairwise disjoint subsets:

$$\mathcal{P}^{(T)} = \bigcup_B \mathcal{P}^{(T)}_B.$$

Claim 39. *We have*

$$\Pr_{C \sim \mathcal{P}^{(S)}}\Big[|C^{-1}(S)| = \Omega(\log n)\Big] \leq \frac{1}{n}.$$

Proof. The claim holds because $|C^{-1}(S)|$ approximately follows a Poisson distribution, as we now show. Indeed, for a non-negative integer k, let

$$\mathcal{P}^{(S)}_k = \Big\{C \in \mathcal{P}^{(S)} \;\Big|\; |C^{-1}(S)| = k\Big\}.$$

We say that $C \in \mathcal{P}^{(S)}_0$ is *in relation* with $C' \in \mathcal{P}^{(S)}_k$ iff $C(a) = C'(a)$ for all $a \in [\sqrt{n}] \setminus C'^{-1}(S)$. Each $C \in \mathcal{P}^{(S)}_0$ is in relation with at most $\binom{\sqrt{n}}{k}|S|^k$ clauses in $\mathcal{P}^{(S)}_k$.

On the other hand, let $C' \in \mathcal{P}^{(S)}_k$. Then $C' \in \mathcal{P}^{(S)}$ and so there exists $X \in \mathcal{X}'$ such that $C' \subseteq X^{(S)}$. We can change each value $C'(a) \in S$ to a value in $X \setminus S$ and obtain a clause in $\mathcal{P}^{(S)}_0$. Hence, C' is in relation with at least $|X \setminus S|^k = \big(\Omega(n)\big)^k$ clauses in $\mathcal{P}^{(S)}_0$. Using double counting,

$$\frac{|\mathcal{P}^{(S)}_k|}{|\mathcal{P}^{(S)}_0|} \leq \frac{\binom{\sqrt{n}}{k}\big(O(\sqrt{n})\big)^k}{\big(\Omega(n)\big)^k} = \frac{\big(O(1)\big)^k}{k!}.$$

This implies the claim. □

Thus, we can focus exclusively on C with small $C^{-1}(S)$.

Definition 40. We say that a partial clause is *large* if $|\mathrm{dom}(B)| \geq \sqrt{n} - O(\sqrt{n})$. We call a large partial clause B *bad* if $B \subseteq X \cap Y$ for some vertices $X, Y \in \mathcal{X}'$ that belong to two different connected components of G^γ. Otherwise, we call B *good*.

Thus, using Claim 39 and that $\mathcal{P}_B^{(S)}$ are disjoint:

$$\sum_{B \text{ is bad}} \big|\mathcal{P}_B^{(S)}\big| + \sum_{B \text{ is good}} \big|\mathcal{P}_B^{(S)}\big| = \sum_{B \text{ is large}} \big|\mathcal{P}_B^{(S)}\big| \geq \Big(1 - \frac{1}{n}\Big)\big|\mathcal{P}^{(S)}\big|. \tag{24}$$

Claim 41. *If B is good and S satisfies (20)–(22), then*

$$|\mathcal{P}_B| \geq \Big(1 - O\big(n^{-1/4}\log^{3/2} n\big)\Big)|\mathcal{P}_B^{(S)}|.$$

Proof. Let $\mathcal{X}_B = \{X \in \mathcal{X}' \mid B \subseteq X\}$. If $\mathcal{X}_B$ is empty, then both $\mathcal{P}_B$ and $\mathcal{P}_B^{(S)}$ are empty, and we are done. For the rest of this proof, we consider $\mathcal{X}_B \neq \emptyset$. Fix any $X \in \mathcal{X}_B$ and define $Y_B = \bigcap_{Y \in \mathcal{X}_B} Y$. Since B is good, $\mathcal{X}_B$ is contained in some connected component $\mathcal{A}_k$ of G^γ. Hence, $Z_k \subseteq Y_B$.

Let $\overline{D} = [\sqrt{n}]\backslash\mathrm{dom}(B)$ be the complement of the domain of B. In particular, $|\overline{D}| = O(\log n)$. The size of $\mathcal{P}_B$ is at least the number of functions from $\overline{D}$ to $X \cap S$ since each such function can be used to extend B to a complete clause in $\mathcal{P}_B$. Similarly, the size of $\mathcal{P}_B^{(S)}$ is at most the number of functions from $\overline{D}$ to $S \setminus Y_B \subseteq S \setminus Z_k$. Using the bounds in (21) and (22) we then obtain that

$$\big|\mathcal{P}_B\big| \geq \big|S \cap X\big|^{|\overline{D}|} \geq \left(\frac{\sqrt{n}}{2} - O(n^{1/4}\sqrt{\log n})\right)^{O(\log n)}$$

and

$$\big|\mathcal{P}_B^{(S)}\big| \leq \big|S \setminus Z_k\big|^{|\overline{D}|} \leq \left(\frac{\sqrt{n}}{2} + O(n^{1/4}\sqrt{\log n})\right)^{O(\log n)}.$$

Thus,

$$\frac{\big|\mathcal{P}_B\big|}{\big|\mathcal{P}_B^{(S)}\big|} \geq \left(\frac{\frac{\sqrt{n}}{2} - O(n^{1/4}\sqrt{\log n})}{\frac{\sqrt{n}}{2} + O(n^{1/4}\sqrt{\log n})}\right)^{O(\log n)} \geq 1 - O(n^{-1/4}\log^{3/2} n). \qquad \square$$

Claim 42. *We have*

$$\sum_{B \text{ is bad}} \big|\mathcal{P}_B^{(S)}\big| \leq O\Big(\frac{1}{n}\Big)\,\big|\mathcal{P}^{(S)}\big|.$$

Proof. Fix a particular pair $X, Y \in \mathcal{X}'$ of vertices that lie in different connected components of G^γ. If $B \subseteq X \cap Y$, then for every clause $C \in \mathcal{P}_B^{(S)}$ we have $C(a) \in X \cap Y$ if $a \in \mathrm{dom}(B)$ and $C(a) \in S$ if $a \notin \mathrm{dom}(B)$. Hence, $C \subseteq (X \cap Y) \cup S$. Thus, using (20):

$$\sum_{B \subseteq X \cap Y} |\mathcal{P}_B^{(S)}| = \left|\bigcup_{B \subseteq x \cap y} \mathcal{P}_B^{(S)}\right| \leq \big(\big|(x \cap y) \cup S\big|\big)^{\sqrt{n}} \leq \Big(\frac{n}{2} - \Omega(\gamma)\Big)^{\sqrt{n}}.$$

On the other hand, we can lower bound the number of clauses contained in $\mathcal{P}^{(S)}$ by the number of clauses contained in $X^{(S)}$, that is,

$$|\mathcal{P}^{(S)}| \geq \big|X^{(S)}\big|^{\sqrt{n}} \geq \Big(\frac{n}{2} - O(\sqrt{n})\Big)^{\sqrt{n}}.$$

Thus,

$$\frac{\sum_{B \subseteq X \cap Y} |\mathcal{P}_B^{(S)}|}{|\mathcal{P}^{(S)}|} \leq \left(\frac{\frac{n}{2} - \Omega(\gamma)}{\frac{n}{2} - O(\sqrt{n})}\right)^{\sqrt{n}} \leq \left(1 - \Omega\Big(\frac{\gamma}{n}\Big)\right)^{\sqrt{n}} \leq \mathsf{e}^{-\Omega(\gamma/\sqrt{n})} = \mathsf{e}^{-\Theta(\log n)}. \tag{25}$$

Taking the Θ-factor sufficiently large and summing (25) over all X and Y in $\mathcal{X}'$, we obtain the claim. $\square$

From (24) and Claim 42, we have

$$\sum_{B \text{ is good}} |\mathcal{P}_B^{(S)}| \geq \Big(1 - O\Big(\frac{1}{n}\Big)\Big)|\mathcal{P}^{(S)}|,$$

and with Claim 41 this gives

$$|\mathcal{P}| \geq \sum_{B \text{ is good}} |\mathcal{P}_B| \geq \Big(1 - O\big(\big(n^{-1/4}\log^{3/2} n\big)\Big) \sum_{B \text{ is good}} |\mathcal{P}_B^{(S)}| \geq \Big(1 - O\big(n^{-1/4}\log^{3/2} n\big)\Big)|\mathcal{P}^{(S)}|,$$

proving Lemma 37.

5 Testing monotonicity of regular LTFs

5.1 Randomized bisection process

The key component of the analysis of the bisection algorithm and the proof of Theorem 2 is the analysis of *randomized bisection processes*, as defined below.

Definition 43 (Randomized bisection process)**.** Fix any finite set S. The *randomized bisection process* with initial set S is the sequence of random sets $S_0, S_1, S_2, \ldots$ defined as follows. Initially, $S_0 = S$. For each $k \geq 1$, S_{k-1} is partitioned uniformly at random into two sets A_k and B_k. Then the set S_k is chosen to be either A_k or B_k by some arbitrary (and possibly adversarial) external process.

Lemma 44. *For any $\delta > 0$, there exists $\kappa = 2\log_2(1/\delta) + O(1)$ such that the randomized bisection process $S_0, S_1, S_2, \ldots, S_k$ with the initial set S, for every $k \leq \log_2|S| - \kappa$, satisfies*

$$\frac{1}{2} \cdot \frac{|S|}{2^k} < |S_k| < \frac{3}{2} \cdot \frac{|S|}{2^k} \tag{26}$$

with probability at least $1 - \delta$.

Proof. Let us prove the lower bound first. The best strategy for the adversary is to take the smallest of A_k and B_k on each step, so we may assume that the sets S_k of size less than $|S_{k-1}|/2$ appear with probability twice as large as in the situation where we simply chose a random subset of S_{k-1}, whereas the sets S_k of size more than $|S_{k-1}|/2$ never appear at all. This implies, in particular, that $|S_k| \leq |S|/2^k$ for each $k \geq 1$.

Using Fubini's theorem and the Chernoff–Hoeffding bound, we obtain that for a fixed S_{k-1},

$$\begin{aligned}\mathbb{E}\big[|S_k| \mid S_{k-1}\big] &= \frac{|S_{k-1}|}{2} - 2\int_0^{+\infty} \Pr\Big[\mathcal{B} < \frac{|S_{k-1}|}{2} - t\Big]\, dt \\ &\geq \frac{|S_{k-1}|}{2} - 2\int_0^{+\infty} \mathsf{e}^{-2t^2/|S_{k-1}|}\, dt \geq \frac{|S_{k-1}|}{2} - O\Big(\sqrt{|S_{k-1}|}\Big),\end{aligned}$$

where $\mathcal{B}$ is the binomial probability distribution on $|S_{k-1}|$ elements with probability $\frac{1}{2}$.

Since $x \mapsto \frac{x}{2} - c\sqrt{x}$ is a convex function, if we unfix S_{k-1}, we get by Jensen's inequality

$$\mathbb{E}\big[|S_k|\big] \geq \frac{\mathbb{E}\big[|S_{k-1}|\big]}{2} - O\Big(\sqrt{\mathbb{E}\big[|S_{k-1}|\big]}\Big). \tag{27}$$

Using induction on k,

$$\mathbb{E}\big[|S_k|\big] \geq \frac{|S|}{2^k} - \sum_{j=1}^{k} \frac{1}{2^{k-j}} \cdot O\Big(\sqrt{\mathbb{E}\big[|S_{j-1}|\big]}\Big).$$

Using our earlier observation that we can assume that $|S_j| \le |S|/2^j$, we get

$$\mathbb{E}\big[|S_k|\big] \ge \frac{|S|}{2^k} - \sum_{j=1}^{k} \frac{1}{2^{k-j}} \cdot O\left(\sqrt{\frac{|S|}{2^{j-1}}}\right) = \frac{|S|}{2^k} - \sqrt{\frac{|S|}{2^k}} \sum_{j=1}^{k} \frac{O(1)}{2^{(k-j)/2}} \ge \frac{|S|}{2^k} - O\left(\sqrt{\frac{|S|}{2^k}}\right). \quad (28)$$

When $\frac{|S|}{2^k} \ge c/\delta^2$ for a sufficiently large constant c, we have

$$\mathbb{E}\big[|S_k|\big] > \frac{|S|}{2^k} - \frac{\delta}{4} \cdot \frac{|S|}{2^k}.$$

Since $|S_k| \le \frac{|S|}{2^k}$, applying Markov's inequality to the random variable $Z = \frac{|S|}{2^k} - |S_k|$ yields

$$\Pr\Big[|S_k| \le \frac{1}{2} \cdot \frac{|S|}{2^k}\Big] \le \frac{\delta}{2}.$$

The proof of the upper bound is similar. This time the adversary takes the largest of A_k and B_k and we have that $|S_k| \ge |S|/2^k$. Similarly to (27), we get

$$\mathbb{E}\big[|S_k|\big] \le \frac{\mathbb{E}\big[|S_{k-1}|\big]}{2} + O\left(\sqrt{\mathbb{E}\big[|S_{k-1}|\big]}\right).$$

We show by induction on k that $\mathbb{E}[|S_k|] \le \frac{3}{2} \cdot \frac{|S|}{2^k}$. This is done similarly to (28):

$$\mathbb{E}\big[|S_k|\big] \le \frac{|S|}{2^k} + \sum_{j=1}^{k} \frac{1}{2^{k-j}} \cdot O\left(\sqrt{\frac{3|S|}{2^j}}\right) = \frac{|S|}{2^k} + \sqrt{\frac{|S|}{2^k}} \sum_{j=1}^{k} \frac{O(1)}{2^{(k-j)/2}} \le \frac{|S|}{2^k} + O\left(\sqrt{\frac{|S|}{2^k}}\right).$$

Again, when $\frac{|S|}{2^k} \ge c/\delta^2$ for a sufficiently large constant c, we have $\mathbb{E}\big[|S_k|\big] < \frac{|S|}{2^k} + \frac{\delta}{4} \cdot \frac{|S|}{2^k}$ and, since $|S_k| \ge |S|/2^k$,

$$\Pr\Big[|S_k| \ge \frac{3}{2} \cdot \frac{|S|}{2^k}\Big] \le \frac{\delta}{2}. \qquad \square$$

5.2 Non-monotonicity of LTFs

Proposition 45. *If $f\colon \{-1,1\}^n \to \{-1,1\}$ is a non-constant LTF with weights $w_1, \dots, w_n$ such that $\sum_{i:w_i<0} |w_i| > \max_i w_i$, then f is not monotone.*

Proof. Let θ be the threshold of the function f, let $N = \{i \in [n] \mid w_i < 0\}$ denote the set of indices with negative weights, and let $\eta = \sum_{i \in N} |w_i|$. Let $X \in \{-1,1\}^n$ be the subset of inputs such that for every $i \in N$, $x_i = 1$.

There exists $x \in X$ such that $\theta - 2\eta \le \sum_{i\in[n]} w_i x_i < \theta$. Indeed, there exists an input $x' \in X$ with $\sum_{i\in[n]} w_i x'_i < \theta$ (otherwise f is the constant 1 function since its minimum value is attained at a point in X), and an input $x'' \in X$ with $\sum_{i\in[n]} w_i x''_i \ge \theta - 2\eta$ (otherwise f is the constant -1 function since flipping the value of any subset of coordinates of a point in X increases the value of f by at most 2η). Also, $\max_i |w_i| \le \eta$, hence, changing the value of one variable changes the value of the sum $\sum_{i\in[n]} w_i x_i$ by at most 2η.

With this choice of x, let $y \in \{-1,1\}^n$ be defined by $y_i = x_i$ for $i \in [n] \setminus N$ and $y_j = -1$ for every $j \in N$. Then $y \prec x$ since $x_j = 1$ for each $j \in N$. And $\sum_{i\in[n]} w_i y_i = \sum_{i\in[n]} w_i x_i + 2\eta \ge \theta$ so $f(y) = 1$. Therefore, this pair of inputs x and y satisfy $x \succeq y$, $f(x) = -1$, and $f(y) = 1$. Hence, f is not monotone. $\square$

In the proof of Theorem 2, we need to show that regular LTFs that are far from monotone must have a large number of reasonably large negative weights.

Proposition 46. *Fix $n \geq 1$ and $\epsilon > 0$. Let $f\colon \{-1,1\}^n \to \{-1,1\}$ be a τ-regular LTF with weights $w_1, \ldots, w_n$ that is ϵ-far from monotone. Assume $\sum_{i\in[n]} w_i^2 = 1$ and $\tau \leq \frac{\epsilon}{4}$. Then the set $N = \{i \in [n] \mid w_i < 0\}$ of indices corresponding to negative weights satisfies $\sum_{i\in N} w_i^2 \geq \frac{\epsilon^2}{256\ln(8/\epsilon)}$.*

Proof. Let $g\colon \{-1,1\}^n \to \{-1,1\}$ be the LTF $g(x) = \operatorname{sgn}(\sum_{i\in[n]\setminus N} w_i x_i - \theta)$ obtained by removing the negative weights of f. Since the function g is monotone,

$$\Pr[f(x) \neq g(x)] \geq \epsilon.$$

The event $f(x) \neq g(x)$ can only occur when $|\sum_{i\in N} w_i x_i| > \left|\sum_{i\in[n]\setminus N} w_i x_i - \theta\right|$. So for any $t > 0$,

$$\Pr[f(x) \neq g(x)] \leq \Pr\left[\Big|\sum_{i\in[n]\setminus N} w_i x_i - \theta\Big| \leq t\right] + \Pr\left[\Big|\sum_{i\in N} w_i x_i\Big| > t\right].$$

Define $\eta = \sum_{i\in N} w_i^2$. If $\eta > \frac{1}{2}$, then we are done. So assume from now on that $\eta \leq \frac{1}{2}$. Fix $t = \sqrt{2\eta\ln(\frac{8}{\epsilon})}$. By Lemma 5,

$$\Pr\left[\Big|\sum_{i\in[n]\setminus N} w_i x_i - \theta\Big| \leq \sqrt{2\eta\ln(\tfrac{8}{\epsilon})}\right] \leq 2\sqrt{\frac{2\eta\ln(\frac{8}{\epsilon})}{1-\eta}} + \frac{\epsilon}{2} \leq 4\sqrt{\eta\ln(\tfrac{8}{\epsilon})} + \frac{\epsilon}{2}$$

and by the Hoeffding bound,

$$\Pr\left[\Big|\sum_{i\in N} w_i x_i\Big| > \sqrt{2\eta\ln(\tfrac{8}{\epsilon})}\right] \leq 2\mathsf{e}^{-\left(2\eta\ln(\frac{8}{\epsilon})\right)/2\eta} \leq \frac{\epsilon}{4}.$$

Putting all the inequalities together, we obtain the inequality $\frac{\epsilon}{4} \leq 4\sqrt{\eta\ln(\frac{8}{\epsilon})}$, which is satisfied if and only if $\eta \geq \frac{\epsilon^2}{256\ln(\frac{8}{\epsilon})}$. □

Corollary 47. *Fix $\epsilon > 0$ and $\tau > 0$. There exists $n_0 = n_0(\epsilon, \tau)$ such that for every $n \geq n_0$, if $f\colon \{-1,1\}^n \to \{-1,1\}$ is a $\frac{\tau}{\sqrt{n}}$-regular LTF with normalized weights $w_1, \ldots, w_n$ ($\sum_{i\in[n]} w_i^2 = 1$) that is ϵ-far from monotone, then the set $N^\dagger = \left\{i \in [n] \;\middle|\; w_i < -\frac{\epsilon}{\sqrt{512\ln(\frac{8}{\epsilon})n}}\right\}$ has cardinality $|N^\dagger| \geq \frac{\epsilon^2}{512\tau^2\ln(\frac{8}{\epsilon})} \cdot n$.*

Proof. Let n_0 be the minimal positive integer such that $\frac{\tau}{\sqrt{n_0}} < \frac{\epsilon}{4}$. Define $N = \{i \in [n] \mid w_i < 0\}$. By Proposition 46, the sum of the squares of the negative weights is bounded below by $\sum_{i\in N} w_i^2 \geq \frac{\epsilon^2}{256\ln(\frac{8}{\epsilon})}$. For every element i in $N \setminus N^\dagger$, the weight w_i satisfies $w_i^2 \leq \frac{\epsilon^2}{512\ln(\frac{8}{\epsilon})n}$ so $\sum_{i\in N\setminus N^\dagger} w_i^2 \leq |N \setminus N^\dagger| \cdot \frac{\epsilon^2}{512\ln(\frac{8}{\epsilon})n} \leq \frac{\epsilon^2}{512\ln(\frac{8}{\epsilon})}$ and

$$\sum_{i\in N^\dagger} w_i^2 = \sum_{i\in N} w_i^2 - \sum_{i\in N\setminus N^\dagger} w_i^2 \geq \frac{\epsilon^2}{256\ln(\frac{8}{\epsilon})} - \frac{\epsilon^2}{512\ln(\frac{8}{\epsilon})} = \frac{\epsilon^2}{512\ln(\frac{8}{\epsilon})}.$$

The regularity of f guarantees that $\sum_{i\in N^\dagger} w_i^2 \leq |N^\dagger|\frac{\tau^2}{n}$ and so $|N^\dagger| \geq \frac{n}{\tau^2} \cdot \frac{\epsilon^2}{512\ln(\frac{8}{\epsilon})}$. □

5.3 Proof of Theorem 2

We modify the bisection algorithm from Algorithm 1 to make the analysis easier. Define

$$c = \frac{\epsilon^2}{512\tau^2 \ln(\frac{8}{\epsilon})} \qquad \text{and} \qquad \zeta = \frac{\epsilon}{\sqrt{512 \ln(\frac{8}{\epsilon})}}.$$

and let

$$k = \Big\lfloor \log(cn) - \max\Big\{\log(\tfrac{8\tau}{\zeta}), \kappa(\tfrac{1}{8})\Big\}\Big\rfloor,$$

where κ is as in Lemma 44.

Algorithm 2 Modified Bisection Algorithm

1: Draw $x \in \{-1,1\}^n$ uniformly at random.
2: Draw $y \in \{-1,1\}^n$ uniformly at random $8/\epsilon$ times or until $f(x) \neq f(y)$.
3: If no y satisfying the condition $f(x) \neq f(y)$ was found, **accept**.
4: Assume $f(x) = -1$ and $f(y) = 1$. Otherwise, swap x and y.
5: **for** k times **do**
6: Draw $z \in \text{Hybrid}(x, y)$ uniformly at random.
7: If $f(z) = -1$, update $x \leftarrow z$.
8: Otherwise if $f(z) = 1$, update $y \leftarrow z$.
9: **end for**
10: If $|x \triangle y| > \frac{3}{2} \cdot \frac{n}{2^k}$, **accept**.
11: Query all inputs in Hybrid(x, y). **Reject** if a non-monotone edge found; otherwise **accept**.

Consider Algorithm 2. The algorithm makes at most $1+8/\epsilon+k+2^{\frac{3}{2}\frac{n}{2^k}}$ queries. The setting of our parameters k, c, and ζ imply that $\frac{n}{2^k} \leq O(\frac{\tau}{c\zeta}) = \widetilde{O}(\tau^3/\epsilon^3)$ so that the total query complexity of the algorithm is $\log n + 2^{\widetilde{O}(\tau^3/\epsilon^3)}$. The algorithm never rejects a monotone function. It remains to show that the algorithm satisfies the soundness requirement of Theorem 2.

Fix f to be any $\frac{\tau}{\sqrt{n}}$-regular LTF that is ϵ-far from monotone. We may assume that the weights of f satisfy $\sum_i w_i^2 = 1$. By Corollary 47, the set $N = \{i \in [n] \mid w_i < -\frac{\zeta}{\sqrt{n}}\}$ has cardinality $|N| \geq cn$.

Assume $x \in \{-1,1\}^n$ is fixed, and y is uniformly sampled from $\{-1,1\}^n$. First, f is ϵ-far from a constant function, hence, an input y satisfying $f(x) \neq f(y)$ will be found with probability at least $\frac{7}{8}$. Next, let $x \triangle y$ be the set of indices where x and y differ. By Chernoff bound, the probability that $|(x \triangle y) \cap N| < cn/4$ is $o(1)$. Thus, with probability at least $\frac{3}{4}$, after Step 4, x and y satisfy $f(x) = -1$, $f(y) = 1$ and the set $S = x \triangle y$ satisfies $|S \cap N| \geq cn/4$.

Let x_ℓ and y_ℓ denote the value of x and y after the ℓth iteration of the loop in Algorithm 2. In particular, x_0 and y_0 are the inputs x and y after Step 4 as in the previous paragraph. Denote $S_\ell = x_\ell \triangle y_\ell$ and $N_\ell = N \cap S_\ell$. The sets $S_0, S_1, S_2, \ldots$ and the sets $N_0, N_1, N_2, \ldots$ are randomized bisection processes with the initial sets S and $N \cap S$, respectively. By Lemma 44, with probability at least $\frac{1}{4}$, the sets S_k and N_k satisfy

$$|S_k| < \frac{3}{2} \cdot \frac{|S|}{2^k} \leq O\Big(\frac{n}{cn\frac{\zeta}{\tau}}\Big) = O\Big(\frac{\tau}{c\zeta}\Big)$$

and

$$|N_k| > \frac{1}{2} \cdot \frac{|N \cap S|}{2^k} \geq \frac{1}{2} \cdot \frac{cn/4}{cn\frac{\zeta}{8\tau}} \geq \frac{\tau}{\zeta}.$$

When these bounds are satisfied, the algorithm does not accept in Step 10 and the sum of the weights with coordinates in N_k satisfies

$$\sum_{i\in N_k} |w_i| \geq |N_k| \cdot \min_{i\in N_k} |w_i| > \frac{\tau}{\zeta} \cdot \frac{\zeta}{\sqrt{n}} = \frac{\tau}{\sqrt{n}} \geq \max_j |w_j|.$$

Let $f_{x,y}$ denote the function f restricted to the set Hybrid(x, y), where x and y are as in Steps 10 and 11 of the algorithm. This function is non-constant since since $f_{x,y}(x) = -1$ and $f_{x,y}(y) = 1$. Then by Proposition 45, $f_{x,y}$ is a non-monotone LTF on $|S_k| < \frac{3}{2} \cdot \frac{n}{2^k}$ variables and the algorithm rejects in Step 11.

5.4 Truncated functions

In Section 5.3, we showed that the modified bisection algorithm efficiently ϵ-tests regular LTFs for monotonicity, as specified by Theorem 2. However, in the actual lower bounds by Fischer *et al.* [18], and Chen *et al.* [13, 11], *truncated* LTFs are used, as in Definition 7. In this section, we show that if the bisection algorithm can test some class of functions for monotonicity, then it can also test the truncated version of the same class with a modest slow-down.

Towards this goal, we argue that with probability $\Omega_\epsilon(1)$, the bisection algorithm only queries inputs in the middle layers of the cube, i.e., satisfying $\frac{n}{2} - \delta\sqrt{n} \leq |x| \leq \frac{n}{2} + \delta\sqrt{n}$ in the notation of Definition 7. It is easy to modify the parameters of Algorithm 2 in Section 5.3 so that the algorithm uses $O_{\epsilon,\tau,p}(1) + \log n$ queries, always accepts a monotone function, and rejects a non-monotone $\frac{\tau}{\sqrt{n}}$-regular LTF with probability at least $1 - p$. Combining the two statements, we get an algorithm that ϵ-tests truncated $\frac{\tau}{\sqrt{n}}$-regular LTFs for monotonicity in $O_{\epsilon,\tau}(\log n)$ queries.

Consider the performance of Algorithm 2 on a function of the form Truncate$_\delta(f)$. The algorithm does not know the value of δ, but it knows ϵ, the distance from a non-monotone function Truncate$_\delta(f)$ to the closest monotone function. By Lemma 5, there exists $\beta = \beta(\epsilon) > 0$ such that with probability at least $\frac{\epsilon}{2}$, the input y found on Step 2 of the algorithm satisfies

$$\frac{n}{2} - (\delta - \beta)\sqrt{n} \leq |y| \leq \frac{n}{2} + (\delta - \beta)\sqrt{n}.$$

The input x also satisfies the same estimates with probability $\Omega_\epsilon(1)$.

Let x_ℓ, y_ℓ and S_ℓ be as in Section 5.3. Let also z_ℓ denote the input z on the $(\ell+1)$st iteration of the loop, so that either $x_{\ell+1}$ or $y_{\ell+1}$ equals z_ℓ. We consider those executions of the algorithm, in which

$$\max\left\{\left|\big|z_\ell \cap S_\ell \cap x_\ell\big| - \frac{|S_\ell \cap x_\ell|}{2}\right|,\ \left|\big|z_\ell \cap S_\ell \setminus x_\ell\big| - \frac{|S_\ell \setminus x_\ell|}{2}\right|\right\} \leq \frac{\beta(1-\alpha)}{4}\alpha^\ell\sqrt{n} \tag{29}$$

for all $\ell \leq k$, where $k = \lceil \log(4\sqrt{n}/\beta) \rceil$ and $\alpha = 0.9$ (or any other constant $\frac{1}{\sqrt{2}} < \alpha < 1$).

We first show that (29) is satisfied for all $\ell \leq k$ with probability $\Omega_\epsilon(1)$. By induction, for each ℓ,

$$|S_\ell| \leq \frac{n}{2^\ell} + \frac{\beta(1-\alpha)}{2}\sum_{i=0}^{\ell-1}\frac{\alpha^i}{2^{\ell-i-1}}\sqrt{n} \leq \frac{n}{2^\ell} + \frac{\beta(1-\alpha)}{2\alpha-1}\alpha^\ell\sqrt{n} \leq 2 \cdot \frac{n}{2^\ell}, \tag{30}$$

where the last inequality holds if n is large enough. By the Chernoff-Hoeffding bound, the probability that (29) is satisfied for all $\ell \leq k$ is at least

$$\prod_{\ell=0}^{k}\left(1 - 2\exp\Big(-2\frac{\frac{\beta^2(1-\alpha)^2}{16}\alpha^{2\ell}n}{2 \cdot \frac{n}{2^\ell}}\Big)\right)^2 \geq \left(\prod_{\ell=0}^{\infty}\left(1 - 2\mathsf{e}^{-\Omega_\epsilon\left((2\alpha^2)^\ell\right)}\right)\right)^2,$$

and the infinite product converges.

By induction again, for each $\ell \le k$,

$$\Big||z_\ell| - \frac{n}{2}\Big| < (\delta-\beta)\sqrt{n} + \frac{\beta(1-\alpha)}{2}\sum_{\ell=0}^{+\infty}\alpha^\ell\sqrt{n} = \big(\delta - \frac{\beta}{2}\big)\sqrt{n}.$$

Also, by (30), $|S_k| \le \frac{\beta}{2}\sqrt{n}$. Hence, all the inputs queried by the algorithm after the kth iteration of the loop are also in the middle layers of the cube.

Acknowledgements

The authors thank Oded Goldreich, Sofya Raskhodnikova, Rocco Servedio, Li-Yang Tan, Erik Waingarten, and the anonymous referees for extremely valuable feedback on earlier versions of this manuscript.

E.B. also offers special thanks to Sofya Raskhodnikova for first introducing him to the problem of monotonicity testing, for sharing her idea of using an adaptive bisection algorithm for the problem, and for many interesting discussions on the problem over the years.

During this research, A.B. was supported by FP7 FET Proactive project QALGO. Part of this work was completed while A.B. was at CWI, Amsterdam. E.B. is supported by an NSERC Discovery grant.

References

[1] A. Belovs and E. Blais. Quantum algorithm for monotonicity testing on the hypercube. *Theory of Computing*, 11(16):403–412, 2015. `arXiv:1503.02868`.

[2] A. Belovs and E. Blais. A polynomial lower bound for testing monotonicity. In *Proc. of 48th ACM STOC*, pages 1021–1032, 2016. `arXiv:1511.05053`.

[3] P. Berman, S. Raskhodnikova, and G. Yaroslavtsev. L_p-testing. In *Proc. of 46th ACM STOC*, pages 164–173, 2014.

[4] A. Bhattacharyya, E. Grigorescu, K. Jung, S. Raskhodnikova, and D. P. Woodruff. Transitive-closure spanners. *SIAM Journal on Computing*, 41(6):1380–1425, 2012.

[5] E. Blais, J. Brody, and K. Matulef. Property testing lower bounds via communication complexity. *Computational Complexity*, 21(2):311–358, 2012. Earlier: *CCC'11*, `ECCC:2011/045`.

[6] A. Bogdanov and M. Safra. Hardness amplification for errorless heuristics. In *Proc. of 48th IEEE FOCS*, pages 418–426, 2007.

[7] J. Briët, S. Chakraborty, D. García-Soriano, and A. Matsliah. Monotonicity testing and shortest-path routing on the cube. *Combinatorica*, 32(1):35–53, 2012. Earlier: *RANDOM'10*, `ECCC:2010/048`.

[8] C. Canonne. Open problem for february 2015. Property Testing Review (Blog post), 2015.

[9] D. Chakrabarty and C. Seshadhri. A $o(n)$ monotonicity tester for Boolean functions over the hypercube. In *Proc. of 45th ACM STOC*, pages 411–418, 2013. `arXiv:1302.4536`.

[10] D. Chakrabarty and C. Seshadhri. An optimal lower bound for monotonicity testing over hypergrids. *Theory of Computing*, 10:453–464, 2014. Earlier: *RANDOM'13*, ECCC:2013/062.

[11] X. Chen, A. De, R. A. Servedio, and L.-Y. Tan. Boolean function monotonicity testing requires (almost) $n^{1/2}$ non-adaptive queries. In *Proc. of 47th ACM STOC*, pages 519–528, 2015. arXiv:1412.5657.

[12] X. Chen, R. A. Servedio, L. Tan, and E. Waingarten. Adaptivity is exponentially powerful for testing monotonicity of halfspaces. In *Proc. of 21st RANDOM*, volume 81 of *LIPIcs*, pages 38:1–38:21. Dagstuhl, 2017. arXiv:1706.05556.

[13] X. Chen, R. A. Servedio, and L.-Y. Tan. New algorithms and lower bounds for monotonicity testing. In *Proc. of 55th IEEE FOCS*, pages 286–295, 2014. arXiv:1412.5655.

[14] X. Chen, E. Waingarten, and J. Xie. Beyond Talagrand functions: New lower bounds for testing monotonicity and unateness. In *Proc. of 49th ACM STOC*, pages 523–536, 2017. arXiv:1702.06997.

[15] I. Diakonikolas and R. A. Servedio. Improved approximation of linear threshold functions. *Computational Complexity*, 22(3):623–677, 2013. Earlier: *CCC'09*, arXiv:0910.3719.

[16] Y. Dodis, O. Goldreich, E. Lehman, S. Raskhodnikova, D. Ron, and A. Samorodnitsky. Improved testing algorithms for monotonicity. In *Proc. of 3rd RANDOM*, pages 97–108. Springer, 1999. ECCC:1999/017.

[17] E. Fischer. The art of uninformed decisions: A primer to property testing. *Bulletin of EATCS*, 75:97–126, 2001.

[18] E. Fischer, E. Lehman, I. Newman, S. Raskhodnikova, R. Rubinfeld, and A. Samorodnitsky. Monotonicity testing over general poset domains. In *Proc. of 34th ACM STOC*, pages 474–483, 2002.

[19] O. Goldreich, S. Goldwasser, E. Lehman, and D. Ron. Testing monotonicity. In *Proc. of 39th IEEE FOCS*, pages 426–435, 1998.

[20] O. Goldreich, S. Goldwasser, E. Lehman, D. Ron, and A. Samorodnitsky. Testing monotonicity. *Combinatorica*, 20(3):301–337, 2000. Earlier: *FOCS'98*.

[21] T. E. Harris. A lower bound for the critical probability in a certain percolation process. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 56, pages 13–20. Cambridge University Press, 1960.

[22] S. Khot, D. Minzer, and M. Safra. On monotonicity testing and Boolean isoperimetric type theorems. In *Proc. of 56th IEEE FOCS*, pages 52–58, 2015. ECCC:2015/011.

[23] D. J. Kleitman. Families of non-disjoint subsets. *Journal of Combinatorial Theory*, 1:153–155, 1966.

[24] H. K. Lee. Learning talagrand DNF formulas. In *Proc. of 23rd COLT*, pages 310–311, 2010.

[25] K. Matulef, R. O'Donnell, R. Rubinfeld, and R. A. Servedio. Testing halfspaces. *SIAM Journal on Computing*, 39(5):2004–2047, 2010. Earlier: *SODA'09*, ECCC:2007/128.

[26] E. Mossel and R. O'Donnell. On the noise sensitivity of monotone functions. *Random Structures and Algorithms*, 23(3):333–350, 2003.

[27] R. O'Donnell and R. A. Servedio. The chow parameters problem. *SIAM Journal on Computing*, 40(1):165–199, 2011. Earlier: *STOC'08*.

[28] R. O'Donnell and K. Wimmer. Approximation by DNF: examples and counterexamples. In *Proc. of 34th ICALP*, pages 195–206, 2007.

[29] A. A. Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106(2):385–390, 1992. Earlier: *ICALP'90*.

[30] M. Talagrand. How much are increasing sets positively correlated? *Combinatorica*, 16(2):243–258, 1996.