

K -Monotonicity is Not Testable on the Hypercube

Elena Grigorescu*

Akash Kumar†

Karl Wimmer‡

May 10, 2017

Abstract

We continue the study of k -monotone Boolean functions in the property testing model, initiated by Canonne et al. (ITCS 2017). A function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is said to be k -monotone if it alternates between 0 and 1 at most k times on every ascending chain. Such functions represent a natural generalization of (1-)monotone functions, and have been recently studied in circuit complexity, PAC learning, and cryptography.

In property testing, the fact that 1-monotonicity can be locally tested with poly n queries led to a previous conjecture that k -monotonicity can be tested with poly(n^k) queries. In this work we disprove the conjecture, and show that even 2-monotonicity requires an exponential in \sqrt{n} number of queries. Furthermore, even the apparently easier task of distinguishing 2-monotone functions from functions that are far from being $n^{0.1}$ -monotone also requires an exponential number of queries.

Our results follow from constructions of families that are hard for a canonical tester that picks a random chain and queries all points on it. Our techniques rely on a simple property of the violation graph and on probabilistic arguments necessary to understand chain tests.

*Purdue University. Email: elena-g@purdue.edu. Research supported in part by NSF CCF-1649515.

†Purdue University. Email: akumar@purdue.edu. Research supported in part by NSF CCF-1649515 and NSF CCF-1618918.

‡Duke University. Email: wimmerk@duke.edu.

1 Introduction

The model of *property testing* [BLR93, RS96, GGR98] studies the complexity of deciding if a large object satisfies a property, or is far from satisfying the property, when the algorithm has only partial access to its input, via queries. One of the most compelling feature of the model is that this notion of approximation yields surprisingly fast algorithms for many natural properties. For instance, testing if a sequence of n bits is sorted can be performed with a number of queries that is independent of n , and hence constant. Testing sortedness of functions over more general posets than lines (a.k.a., testing monotonicity) has drawn interest for almost two decades [GGL⁺00, DGL⁺99, EKK⁺00, FLN⁺02, AC06, BCGM12, FR10, CS13b, CS13a, CS14, CST14, CDST15, KMS15, BB16, CWX17], especially due to the naturalness of the property, as well as its evasiveness to tight analysis.

Analyzing monotonicity over the hypercube $\{0, 1\}^n$ domain, in particular, has proven itself a source of beautiful techniques in the quest for understanding its testing complexity. Indeed, while we have known ever since the initial work of [GGL⁺00] that an $O(n)$ -query tester exists, and that one must make $\Omega(\sqrt{n})$ queries [FLN⁺02], only a recent sequence of works establishes a tight $\tilde{O}(\sqrt{n})$ upper bound [CS13b, KMS15]. This closure has however ignited further interest in extending the techniques developed so far beyond monotonicity, to generalizations such as *unateness* [KS16, CS16b], and *k-monotonicity* [CGG⁺17].

In this work we continue the study of *k-monotonicity* over the hypercube domain. A function $f : \{0, 1\}^n \rightarrow \{0, 1\}$ is *k-monotone* if there do not exist $x_1 \prec x_2 \prec \dots \prec x_{k+1}$ in $\{0, 1\}^n$, such that $f(x_1) = 1$ and $f(x_i) \neq f(x_{i+1})$ for all $i \in [k]$. Indeed, 1-monotone functions are exactly those ones that are monotone.

The notion of *k-monotonicity* has been studied ever since the 50's in the context of circuit lower bounds [Mar57]. Indeed, *k-monotone* functions correspond to functions computable by Boolean circuits with $\log k$ negation gates, and in particular monotone functions correspond to circuits with no negation gates. In proving lower bounds for circuits with few negation gates, it has been apparent that the presence of even one negation gate leads either to failure of the common analysis techniques, or to failure of the expected results, as remarked by Jukna [Juk12].

Interest in *k-monotonicity* has been recently rekindled from multiple angles, including PAC learning, circuit complexity, cryptography and Fourier analysis [Ros15, GMOR15, GK15, LZ16]. From all these angles, *k-monotone* functions can be viewed as a gradual interpolation between the class of very structured monotone functions, and general Boolean functions. Following this observation, on the property testing front [CGG⁺17] conjecture that their lower bounds of $\Omega(n/k^2)^{k/4}$ queries should in fact be achievable up to a polynomial blowup, and pose this as the most important follow-up direction.

To build some intuition, observe that in order to test *k-monotonicity* with one-sided error, a test should only reject if it finds a *violation* in the form of a sequence $x_1 \prec x_2 \prec \dots \prec x_{k+1}$ in $\{0, 1\}^n$, such that $f(x_1) = 1$ and $f(x_i) \neq f(x_{i+1})$ for all $i \in [k]$. Hence, a canonical candidate tester suggested in [CGG⁺17] queries all points along a random chain $0^n = x_1 \prec x_2 \prec \dots \prec x_{n+1} = 1^n$ and rejects only if it finds a violation. We call such a tester the *chain tester*. The chain tester is indeed implied by all previous tests for monotonicity [GGL⁺00, CS13b, CST14, KMS15], incurring only a small polynomial blow-up in the query complexity.

However, despite the strong analogy with monotonicity that allows some common analysis techniques to extend to *k-monotonicity*, the main result of this paper disproves the conjecture from [CGG⁺17] that the chain tester works. Specifically, we show that no one-sided tester with $\text{poly}(n^k)$ queries can succeed on some carefully constructed hard families of Boolean functions.

Theorem 1.1. (Rephrasing Corollary 3.5) *Testing 2-monotonicity non-adaptively with one-sided error requires $2^{\Omega(\sqrt{n})}$ queries.*

This lower bound can be compared to the $2^{O(\sqrt{n} \cdot \log n)}$ -query chain tester provided in [CGG⁺17]. Hence, testing is essentially as hard as PAC learning, in light of the $2^{\Omega(\sqrt{n})}$ lower bound from [BCO⁺15].

While being contrary to the intuition that since monotonicity testing is easy, so should 2-monotonicity be, this result reinforces the theme discussed above in proving circuit complexity lower bounds. The class of 2-monotone functions is exactly the class of functions computable by a Boolean circuit with at most one negation gate [Mar57, BCO⁺15]. As in the circuit complexity world, allowing one negation gate significantly increases complexity.

One further natural question is approximating the “monotonicity” of a function. For a concrete problem, suppose we are promised that the unknown function f is either 2-monotone or far from, say, $n^{0.01}$ -monotone. That is, either f changes value at most twice on every chain, or a constant fraction of points of $\{0, 1\}^n$ need to be changed so that f changes value at most $n^{0.01}$ times on every chain. This promise problem can only require fewer queries, and intuitively, it should be much fewer.

However, we show that intuition is incorrect: even the apparently much easier task of distinguishing between functions that are 2-monotone and functions that are far from being $n^{0.01}$ -monotone requires an exponential number of queries.

Theorem 1.2. (Rephrasing Corollary 3.4) *Testing non-adaptively with one-sided error whether a function is 2-monotone or far from being $n^{0.01}$ -monotone requires $2^{\Omega(n^{0.48})}$ queries.*

These results are essentially consequences of a more general statement.

Theorem 1.3. (Rephrasing Corollary 3.3) *Given $2 \leq k \leq g(k, n) = o(\sqrt{n})$, testing non-adaptively with one-sided error whether a function is k -monotone or far from being $g(k, n)$ -monotone requires $2^{\Omega\left(\frac{\sqrt{n}}{(k+1)(g(k, n)/k)^2}\right)}$ queries.*

1.1 Overview of the proofs

We first observe that the chain tester is indeed a canonical tester for k -monotonicity, namely any non-adaptive tester making q queries can be transformed into a chain tester making $O(q^{k+1}n)$ queries (See Theorem 3.2).

Thus, the most intriguing question that we solve becomes: can one construct a function that is far from every $n^{0.01}$ -monotone function (not only far from every 2-monotone function), yet only a tiny fraction of chains in $\{0, 1\}^n$ contain a violation to 2-monotonicity? Our construction manages to overcome these two conflicting goals by carefully hiding the violations inside a region of the cube that the chain tester can easily miss.

The proof of the $O(n)$ -query tester from [GGL⁺00] relies on the following structural theorem: if f is far from monotone, then there are many *edges* that contain a violation to monotonicity. For k -monotonicity with $k \geq 2$, there is no such result: since all violations require at least 3 points, the violations can all be spread across many levels of the cube. For example, consider a totally symmetric function $f(x)$ such that $f(x) = 1$ if $|x|$ is between $n/2 - 2\sqrt{n}$ and $n/2 - \sqrt{n}$, or between $n/2 + \sqrt{n}$ and $n/2 + 2\sqrt{n}$, and $f(x) = 0$ otherwise. This function is far from 2-monotone, yet any triple of points that witnesses this fact will contain a pair of points whose Hamming distance is at least \sqrt{n} .

At the same time, the chain tester does very well on such a function, since every chain from 0^n to 1^n will uncover a violation to 2-monotonicity! We get around this by hiding such functions with “long” violations on a small set of coordinates, while still making sure it comprises a constant fraction of the cube. We show that a random chain is unlikely to visit enough of these coordinates to find a violation.

Proving that these functions are far from k -monotone amounts to understanding the structure of the violation hypergraph (i.e., the hypergraph whose vertices are elements of $\{0, 1\}^n$, and whose edges are the tuples that witness a violation). A large matching (edges with disjoint sets of vertices) in this hypergraph implies that the function is far from k -monotone. Indeed, such families can be shown to have a large matching.

1.2 Discussion and open problems

While non-adaptive, one-sided testing is inherently hard for k -monotonicity, it may be the case that efficient 2-sided and/or adaptive tests exist. The question of whether these variants help in testing monotonicity have been lurking for a long time, yet it has witnessed no progress so far. In fact, it would be quite exciting if these variants do help in testing k -monotonicity, or even in distinguishing 2-monotonicity from being far from n^{01} -monotonicity. All these questions are also relevant to testing other domains, and in particular the hypergrid $[n]^d$. Preliminary arguments using coarsening similar to [CGG⁺17] show that distinguishing k -monotone functions from functions that are far from $g(k, n)$ -monotone does not need a dependence on n , but the exponential dependence on d still remains. Thus, removing this dependence on d on the hypergrid domains could lead to similar improved complexity over the hypercube.

1.3 Related work

The previous work on testing k -monotonicity [CGG⁺17] reveals connections with testing surface area [KR00, BBY12, KNOW14, Nee14], as well as estimating support of distributions [CR14]. Besides the relevant extensive literature on monotonicity testing mentioned before, another recent direction that generalizes monotonicity is testing *unateness*. Namely, a unate function is monotone on each full chain, however, edge-disjoint chains may be monotone in different directions. Initiated in [GGL⁺00], query-efficient unateness testers were obtained in [GGL⁺00, CS16b], and recently, [CWX17] shows a matching one-sided non-adaptive $\tilde{\Omega}(n)$ bound.

A trickle of recent work in cryptography focuses on understanding how many negation gates are needed to compute cryptographic primitives, such as one-way permutations, small bias generators, hard core bits, and extractors [GMOR15, GK15, LZ16].

Yet another extension of monotonicity on the specific domain of a line, considers the point of view that monotonicity is a property defined by freeness of a specific order pattern in the function: indeed, the function is free of tuples $x_1 \prec x_2$ with $f(x_1) = 1 \geq f(x_2) = 0$. [NRRS17] extends this view to real-valued inputs, where a sequence can follow more complex order patterns.

2 Preliminaries

The *weight* of an element x in $\{0, 1\}^n$, denoted $|x|$, is the number of non-zero entries of x . Given two functions $f, g: \{0, 1\}^n \rightarrow \{0, 1\}$, denote by $d(f, g)$ for the (normalized) Hamming distance between

them, i.e.

$$d(f, g) = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \mathbb{1}_{\{f(x) \neq g(x)\}} = \Pr_{x \sim \{0,1\}^n} [f(x) \neq g(x)]$$

where $x \sim \{0,1\}^n$ refers to x being drawn from the uniform distribution on $\{0,1\}^n$. A *property* of boolean functions from $\{0,1\}^n$ to $\{0,1\}$ is a subset of all the boolean functions over the binary hypercube. We define the distance of a function f to \mathcal{P} as the minimum distance of f to any $g \in \mathcal{P}$:

$$d(f, \mathcal{P}) = \min_{g \in \mathcal{P}} d(f, g).$$

Property testing. We recall the standard definition of testing algorithms, as well as some terminology:

Definition 2.1. Let \mathcal{P} be a property of functions from $\{0,1\}^n$ to $\{0,1\}$. A *q-query testing algorithm* for \mathcal{P} is a randomized algorithm \mathcal{T} which takes as input $\varepsilon \in (0,1]$ as well as query access to a function $f: \{0,1\}^n \rightarrow \{0,1\}$. After making at most $q(\varepsilon)$ queries to the function, \mathcal{T} either outputs ACCEPT or REJECT, such that the following holds:

- if $f \in \mathcal{P}$, then \mathcal{T} outputs ACCEPT with probability at least $2/3$; (Completeness)
- if $d(f, \mathcal{P}) \geq \varepsilon$, then \mathcal{T} outputs REJECT with probability at least $2/3$; (Soundness)

where the probability is taken over the algorithm’s randomness. If the algorithm only errs in the second case but accepts any function $f \in \mathcal{P}$ with probability 1, it is said to be a *one-sided* tester; otherwise, it is said to be *two-sided*. Moreover, if the queries made to the function can only depend on the internal randomness of the algorithm, but not on the values obtained during previous queries, it is said to be *non-adaptive*; otherwise, it is *adaptive*. The maximum number of queries made to f in the worst case is the *query complexity* of the testing algorithm.

When $d(f, \mathcal{P}) \geq \varepsilon$, we say that f is ε -far from \mathcal{P} . In this document, we will frequently use “far” to denote “ $\Omega(1)$ -far”.

The property of interest in this work is k -monotonicity. As mentioned earlier, a function $f: \{0,1\}^n \rightarrow \{0,1\}$ is *k-monotone* if there do not exist $x_1 \prec x_2 \prec \dots \prec x_{k+1}$ in $\{0,1\}^n$, such that $f(x_1) = 1$ and $f(x_i) \neq f(x_{i+1})$ for all $i \in [k]$. We denote the set of k -monotone functions as \mathcal{M}_k . A *violation* of k -monotonicity is a sequence $x_1 \prec x_2 \prec \dots \prec x_{k+1}$ in $\{0,1\}^n$, such that $f(x_1) = 1$ and $f(x_i) \neq f(x_{i+1})$ for all $i \in [k]$.

Following Parnas, Ron, and Rubinfeld [PRR06], we also consider the following relaxation version of property testers, for parameterized properties:

Definition 2.2. Let $\mathcal{P} = \{\mathcal{P}_k\}_k$ be a parameterized family of properties. A (k, ℓ) -tester for the family \mathcal{P} is a randomized algorithm which, on input a proximity parameter $\varepsilon \in (0,1)$ and oracle access to an unknown function f , satisfies the following.

- if $f \in \mathcal{P}_k$, then the algorithm returns ACCEPT with probability at least $2/3$;
- if $d(f, \mathcal{P}_\ell) > \varepsilon$, then the algorithm returns REJECT with probability at least $2/3$.

As mentioned, we are also interested in $(k, g(k, n))$ -testing for $\{\mathcal{M}_k\}_k$. We sometimes denote the problem of distinguishing k -monotone functions from functions that are far from being $g(k, n)$ -monotone as the $(k, g(k, n))$ -problem.

We define the (basic) *chain tester* to be the algorithm that picks a *uniformly random chain* $\mathbf{Z} = \langle 0^n \prec \mathbf{z}_1 \prec \mathbf{z}_2 \prec \dots \prec \mathbf{z}_{n-1} \prec 1^n \rangle$ of comparable points from $\{0,1\}^n$, and queries f at all these

points. The chain tester rejects if \mathbf{Z} reveals a violation to k -monotonicity, otherwise it accepts. We also sometimes call a chain tester an algorithm that picks multiple chains (possibly dependently of each other).

3 Lower Bounds over the Hypercube

We prove all of our results in this section. We first show in [Theorem 3.1](#) that the basic chain tester detects a violation with negligibly small probability, and hence the $(k, g(k, n))$ -problem is hard for chain testers.

Theorem 3.1. *Given $2 \leq k \leq g(k, n) = o(\sqrt{n})$, there exist $C > 0$, and a collection \mathcal{F} of Boolean functions, such that*

- (i) *every $f \in \mathcal{F}$ is $\Omega(1)$ -far from being $g(k, n)$ -monotone, and*
- (ii) *the probability that a uniformly random chain in $\{0, 1\}^n$ detects a violation to k -monotonicity for f is $o\left(\exp\left(-C \frac{\sqrt{n}}{(g(k, n)/k)^2}\right)\right)$.*

We then show that any other non-adaptive, one-sided tester gives rise to a chain tester, with only a small blowup in the query complexity.

Theorem 3.2. *Any non-adaptive one-sided q -query $(k, g(k, n))$ -tester implies an $O(q^{k+1}n)$ -query tester that queries points on a distribution over chains, and succeeds with constant probability. In particular, if p is the success probability of the basic chain tester, then $p = \Omega(1/q^{k+1})$.*

[Theorem 3.1](#) and [Theorem 3.2](#) imply the following general consequence.

Corollary 3.3. *Given $2 \leq k \leq g(k, n) = o(\sqrt{n})$, testing non-adaptively with one-sided error whether a function is k -monotone or far from being $g(k, n)$ -monotone requires $\exp\left(\Omega\left(\frac{\sqrt{n}}{(k+1)(g(k, n)/k)^2}\right)\right)$ queries.*

Instantiating k and $g(k, n)$ in [Corollary 3.3](#), we obtain the following immediate corollaries.

Corollary 3.4. *Any non-adaptive one-sided $(2, n^{0.01})$ -tester for the $(2, n^{0.01})$ -problem requires $\exp(\Omega(n^{0.48}))$ queries.*

Corollary 3.5. *Let $2 \leq k = o(\sqrt{n})$. Then any non-adaptive, one-sided tester for k -monotonicity requires $\Omega\left(\exp\left(\frac{\sqrt{n}}{k+1}\right)\right)$ queries.*

Note that the lower bound of [Corollary 3.5](#) is $> \exp(\sqrt[4]{n})$ for $2 \leq k \leq \sqrt[4]{n}$. Using the previous lower bound of $\Omega\left(\left(\frac{n}{k^2}\right)^{k/4}\right)$ for any $2 \leq k = o(\sqrt{n})$ from [\[CGG+17\]](#), we obtain the following immediate consequence.

Corollary 3.6. *Any non-adaptive, one sided tester for k -monotonicity, for $2 \leq k = o(\sqrt{n})$, requires $\Omega(\exp(\sqrt[4]{n}))$ queries.*

3.1 Proof of [Theorem 3.1](#)

We first recall some standard useful facts.

Fact 3.7. *The maximum value of $\binom{n}{t}$ occurs when $t = \lfloor n/2 \rfloor$, and this maximum value is less than $2 \cdot 2^n / \sqrt{n}$.*

Fact 3.8. *There exists a constant $C > 0$, such that for every $\varepsilon > 0$, the number of points of $\{0, 1\}^n$ that with weight outside the middle levels $[\frac{n}{2} - \sqrt{n} \log \frac{C}{\varepsilon}, \frac{n}{2} + \sqrt{n} \log \frac{C}{\varepsilon}]$ is at most $\varepsilon 2^{n-1}$.*

In [Section 3.1.1](#), we define our hard family, and show that every function in this family is indeed far from $g(k, n)$ -monotonicity. In [Section 3.1.2](#) we show that this family is hard for the chain tester.

The hard family hides instances of a *balanced blocks* function, which was previously used in [\[CGG⁺17\]](#) towards proving that testing k -monotonicity is at least as hard as testing monotonicity, even with adaptive, two-sided queries.

Definition 3.9 (Balanced Blocks function). For every n and $\ell \leq o(\sqrt{n})$, let us partition the vertex set of the Hamming cube into ℓ blocks B_1, B_2, \dots, B_ℓ where every block B_i consists of *all* points in *consecutive levels* of the Hamming cube, such that all of the blocks have *roughly* the same size, i.e., for every $i \in \ell$, we have

$$\left(1 - \frac{\ell}{\sqrt{n}}\right) \frac{2^n}{\ell} \leq |B_i| \leq \left(1 + \frac{\ell}{\sqrt{n}}\right) \frac{2^n}{\ell}.$$

Then the corresponding *balanced blocks function with ℓ blocks*, denoted¹ $\text{BB}(n, \ell): \{0, 1\}^n \rightarrow \{0, 1\}$, is defined to be the blockwise constant function which takes value 1 on all of B_1 and whose value alternates on consecutive blocks.

Note that $\text{BB}(n, \ell)$ is a totally symmetric function: it is unchanged under permutations of its inputs. Thus, we can partition $\{0, 1, \dots, n\}$ into ℓ intervals I_1, I_2, \dots, I_ℓ , such that I_i is the set of Hamming levels that B_i contains.

[\[CGG⁺17\]](#) shows that Balanced Blocks functions satisfy a useful property that we soon recall in [Claim 3.11](#), after making an important definition.

Definition 3.10 (Violation hypergraph ([\[CGG⁺17\]](#))). Given a function $f: \{0, 1\}^n \rightarrow \{0, 1\}$, the *violation hypergraph* of f is $H_{\text{viol}}(f) = (\{0, 1\}^n, E(H_{\text{viol}}))$ where $(x_1, x_2, \dots, x_{\ell+1}) \in E(H_{\text{viol}})$ if the ordered $(\ell + 1)$ -tuple $x_1 \prec x_2 \prec \dots \prec x_{\ell+1}$ (which is a $(\ell + 1)$ -uniform hyperedge) forms a violation to ℓ -monotonicity in f . A collection M_h of pairwise disjoint $(\ell + 1)$ -uniform hyperedges of the violation hypergraph is said to form a *violated matching*.

Claim 3.11 ([\[CGG⁺17, Claim 3.8\]](#)). *Let $h \stackrel{\text{def}}{=} \text{BB}(n, \ell)$. Then h is $(\ell - 1)$ -monotone and not $(\ell - 2)$ -monotone. Furthermore, the violation graph of h with respect to $(\ell - 2)$ -monotonicity contains a violated matching of size at least $\frac{(1-o(1))2^n}{\ell}$, where every edge of the matching $y_1 \preceq \dots \preceq y_{\ell-1}$ has $h(y_1) = 1$ and $h(y_i) \neq h(y_{i+1})$ for $1 \leq i < \ell - 1$.*

This machinery allows us to deduce the following.

¹We will arbitrarily fix a function that satisfies these conditions.

Claim 3.12. Let $h \stackrel{\text{def}}{=} \text{BB}(n, 3k)$. Then $d(h, \mathcal{M}_k) \geq \Omega(1)$.

Proof of Claim 3.12. By Claim 3.11, $H_{\text{viol}}(h)$ contains a matching M_h of $(3k - 1)$ -tuples of size $\frac{(1-o(1))2^n}{3k}$, and for every tuple in the matching $y_1 \preceq \cdots \preceq y_{3k-1}$ we have $h(y_1) = 1$ and $h(y_i) \neq h(y_{i+1})$ for $1 \leq i < 3k - 1$. We see that any k -monotone function close to h must have at most k flips within any such tuple, by definition. It follows that any k -monotone function differs from h in at least $k - 1$ vertices in every tuple of M_h . Thus, the Hamming distance between h and any k -monotone function is at least

$$\frac{(1 - o(1))2^n}{3k} \cdot (k - 1) \geq \frac{2^n}{5}.$$

□

We are now ready to describe the hard family.

3.1.1 The Hard Family

In what follows, let $s \stackrel{\text{def}}{=} g(k, n)$. Also, let $r \stackrel{\text{def}}{=} \frac{g(k, n)}{k}$. We now describe the hard instance for (k, s) -testing.

Consider the partition of the set of indices in $[n]$ into two different sets, L and R , with sizes $|L| = n_L = n \cdot (1 - \frac{1}{625r^2})$ and $|R| = n_R = \frac{n}{625r^2}$, respectively.² We will write $z \in \{0, 1\}^n$ as $z = (x, y)$, where $x \in \{0, 1\}^{|L|}$ and $y \in \{0, 1\}^{|R|}$. We define $\text{MID}_L \stackrel{\text{def}}{=} \{i : |i - \frac{n_L}{2}| \leq \frac{\sqrt{n_L}}{100}\}$, to denote the set of “balanced” inputs restricted to the set of indices in L .

Assuming k is even³, we define $f_{n_L} : \{0, 1\}^n \rightarrow \{0, 1\}$ by

$$f_{n_L}(x, y) \stackrel{\text{def}}{=} \begin{cases} 0, & \text{if } |x| \notin \text{MID}_L \\ \text{BB}(n_R, 3s)(y), & \text{otherwise i.e., } |x| \in \text{MID}_L, \end{cases}$$

where $x \in \{0, 1\}^{|L|}$ and $y \in \{0, 1\}^{|R|}$.

For $x \in \{0, 1\}^{|L|}$, let us denote by H_x the restriction of the hypercube $\{0, 1\}^n$ to the points (x, y) , with $y \in \{0, 1\}^{|R|}$. Note that for $x \in \text{MID}_L$, the restriction of the function to H_x is a copy of balanced blocks function on $n_R = n/625r^2$ variables with $3s$ blocks.

Claim 3.13. The function $f = f_{n_L}$ is $\Omega(1)$ -far from being s -monotone.

Proof. By Fact 3.8, picking $\varepsilon = \frac{1}{3}$, say, it follows that for a constant fraction of the points $x \in \{0, 1\}^{n_L}$, the function f restricted to the cube H_x is a balanced blocks function on $3s$ blocks. By Claim 3.11, there is a matching of violations to $(3s - 2)$ -monotonicity within the violation graph on H_x , of size at least $\Omega(1) \cdot \frac{2^{n_R}}{3s}$. It follows that there is a matching of violations to $(3s - 2)$ -monotonicity on the whole domain $\{0, 1\}^n$ of size at least $\Omega(1) \frac{2^n}{3s}$. As in the proof of Claim 3.12, to produce a function that is s -monotone, one must change the value of f in at least s points of each matched hyper-edge. It follows that f is $\Omega(1)$ -far from being s -monotone.

□

Our hard family of functions is the orbit of the function f_{n_L} under all the permutations of the variables.

²For the sake of presentation, we ignore integrality issues where possible.

³For odd k , the function is defined almost analogously – the only difference is that $f_{n_L}(x, y) = 1$ whenever $|x| > \frac{n_L}{2} + \frac{\sqrt{n_L}}{100}$. We make this assumption throughout the paper.

Definition 3.14. The family $\mathcal{F}_{k,s}$, parameterized by k and s is defined as follows. Setting $n_L = (1 - \frac{k^2}{625s^2})n$, we define

$$\mathcal{F}_{k,s} \stackrel{\text{def}}{=} \{ f_{n_L} \circ \pi_\sigma : \sigma \in S_n \}$$

where $\pi_\sigma : \{0, 1\}^n \rightarrow \{0, 1\}^n$ is a permutation that sends the string $\{(a_1, a_2, \dots, a_n)\}$ to $\{a_{\sigma(1)}, a_{\sigma(2)}, \dots, a_{\sigma(n)}\}$ for a permutation $\sigma : [n] \rightarrow [n]$. We omit the parameters k and s if it is clear from the context.

We now observe that these functions are indeed far from being s -monotone. This follows since s -monotonicity is closed under permutation of the variables, and by [Claim 3.13](#).

Claim 3.15. *Every $f \in \mathcal{F}_{k,s}$, f is $\Omega(1)$ -far from s -monotone.*

Therefore, we proved the distance property from [Theorem 3.1](#)

3.1.2 The Hard Family vs. the Chain Tester

Recalling that the basic chain tester picks a uniformly random chain in $\{0, 1\}^n$, note that the distribution of the queries chosen by the chain tester is unchanged over permutations of the variables. Thus, it suffices to analyze the probability that the chain tester discovers a violation to k -monotonicity on f_{n_L} . We will show that this probability is very small if the quantity s/k is small.

Claim 3.16. *There exists a constant $C > 0$, such that the probability that a random chain reveals a violation to k -monotonicity in f_{n_L} is at most $\exp(-C \frac{k^2}{s^2} \sqrt{n})$.*

Let Z be a fixed chain $0^n = z_0 \prec z_1 \prec z_2 \prec \dots \prec z_n = 1^n$ in $\{0, 1\}^n$. Note that $f_{n_L}(z_i) = f_{n_L}(x_i, y_i) = 0$ if $|x_i| \notin \text{MID}_L$. Thus, if there is a violation to k -monotonicity in Z , then it can be found among points in Z where $|x_i| \in \text{MID}_L$. Thus, a chain can only exhibit a violation on points $z_i = (x_i, y_i)$ where $n_L/2 - \sqrt{n_L}/100 \leq |x_i| \leq n_L/2 + \sqrt{n_L}/100$. By definition, regardless of the exact structure of x_i in this interval, $f_{n_L}(x_i, y_i) = \text{BB}(n_R, 3s)(y_i)$. Since BB is a totally symmetric function, to determine if Z exhibits a violation, it is enough to analyze the set

$$V(Z) \stackrel{\text{def}}{=} \{j : \text{there exists } z_i = (x_i, y_i) \in Z \text{ such that } |x_i| \in \text{MID}_L \text{ and } |y_i| = j\}.$$

We remark that for every chain Z , $V(Z)$ is a set of consecutive integers.

Claim 3.17. *Suppose Z contains a violation to k -monotonicity. Then $|V(Z)| \geq k\sqrt{n_R}/(16s)$.*

Proof. By [Fact 3.7](#), the maximum value of $\Pr_{\mathbf{y} \sim \{0,1\}^{n_R}}[|\mathbf{y}| = t]$ over values of t is $2/\sqrt{n_R}$. Since $\text{BB}(n_R, 3s)$ has $3s$ blocks, the number of consecutive levels of I_i in any block B_j must satisfy

$$(2/\sqrt{n_R})|I_i| \geq \frac{1}{3s}(1 - o(1)) \geq \frac{1}{4s},$$

so $|I_i| \geq \sqrt{n_R}/(8s)$. To see a violation to k -monotonicity, the chain Z must contain points from each Hamming level in $k - 1$ complete blocks, so this requires $|V(Z)| \geq (k - 1)\sqrt{n_R}/(8s) \geq k\sqrt{n_R}/(16s)$, as claimed. □

We will show that that $|V(\mathbf{Z})|$ reaching this value is very unlikely for a random chain \mathbf{Z} . Let \mathbf{Z} be a random chain $0^n \prec \mathbf{z}_1 \prec \dots \prec \mathbf{z}_{n-1} \prec 1^n$.

Proof of Claim 3.16. The proof follows from Claim 3.17 and the following claim.

Claim 3.18. *Let \mathbf{Z} be a random chain. Then $\Pr[|V(\mathbf{Z})| \geq k\sqrt{n_R}/(20s)] \leq \exp\left(-\frac{0.00009}{r^2}\sqrt{n}\right)$.*

Proof. Let j be the smallest index such that $\mathbf{z}_j = (\mathbf{x}_j, \mathbf{y}_j)$ satisfies $|\mathbf{x}_j| = n_L/2 - \sqrt{n_L}/100$. This is the index where the chain enters the region where it could find violations.

Let w be the largest index such that $\mathbf{z}_w = (\mathbf{z}_w, \mathbf{y}_w)$ satisfies $|\mathbf{x}_w| = n_L/2 + \sqrt{n_L}/100$. If \mathbf{Z} contains a violation to k -monotonicity, then it must occur on the subchain $\mathbf{z}_{j-1} \prec \mathbf{z}_j \prec \dots \prec \mathbf{z}_w \prec \mathbf{z}_{w+1}$. By construction, we have $f(\mathbf{z}_\ell) = 0$ if $\ell \leq j - 1$ or $\ell \geq w + 1$. Further, $V(\mathbf{Z}) = \{|\mathbf{y}_j|, |\mathbf{y}_{j+1}|, \dots, |\mathbf{y}_{w-1}|, |\mathbf{y}_w|\}$, and $|V(\mathbf{Z})| = |\mathbf{y}_w| - |\mathbf{y}_j| + 1$. Thus, to prove the claim, it satisfies to analyze $|\mathbf{y}_w| - |\mathbf{y}_j|$. Note $w - j$ is exactly $\sqrt{n_L}/50 + |V(\mathbf{Z})| - 1$; this accounts for $\sqrt{n_L}/50$ flips of variables in L and $|V(\mathbf{Z})| - 1$ flips of variables in R . Informally, we want to show that the ratio of the number of variables flipped in L to number of variables flipped in R is, with very high probability, too small for the chain tester to succeed in finding a violation to k -monotonicity.

To simplify our analysis, we will not work directly with w . Instead, define j as above, but consider $\mathbf{z}_{j'} = (\mathbf{x}_{j'}, \mathbf{y}_{j'})$, where $j' = j + \sqrt{n}/3$. We will show that, with high probability, $j' \geq w$, and $|\mathbf{y}_{j'}| - |\mathbf{y}_j|$ (and thus $|\mathbf{y}_w| - |\mathbf{y}_j|$) is small.

We claim that the value of $|\mathbf{y}_{j'}| - |\mathbf{y}_j|$ is a random variable with a (random) hypergeometric distribution. Indeed, to draw a random variable distributed as this difference, we construct the following experiment that simulates the behavior of a random chain with respect to the function f :

- The chain tester picks $\sqrt{n}/3$ coordinates from the $n - |\mathbf{z}_j|$ coordinates set to 0.
- $n_R - |\mathbf{y}_j|$ of these coordinates are “successes” for the chain tester, which correspond to flipping variables in R , and
- $n_L - |\mathbf{x}_j|$ of these coordinates are “failures” for the chain tester, which correspond to flipping variables in L .

Let $H(u, N, t, i)$ denote the probability of seeing exactly i successes in t independent samples, drawn uniformly and without replacement, from a population of N objects containing exactly u successes.

The chain tester is most likely to see successes in the above experiment if $|\mathbf{y}_j| = 0$; we will assume that this happens. As seen in the proof of Claim 3.17, in order to successfully reject f , the chain must witness at least $\frac{k\sqrt{n_R}}{16s}$ successes. Let $t = \frac{k\sqrt{n_R}}{20s}$.

Note that if the number of successes is $i < t$, then the number of failures is at least $\frac{\sqrt{n}}{3} - t \geq \frac{\sqrt{n}}{3} - \frac{\sqrt{n}}{500} > \frac{\sqrt{n_L}}{50}$, and so in this case we have $|V(\mathbf{Z})| < t$; this corresponds to the chain missing a complete balanced block. It follows that the proof reduces to upper bounding the quantity

$$\Pr[|V(\mathbf{Z})| \geq t] = \sum_{i \geq t} H(n_R, n_L/2 + \sqrt{n_L}/100 + n_R, \sqrt{n}/3, i).$$

We analyze the above quantity using a Chernoff bound for hypergeometric random variables, where $\mathbf{X} = |\mathbf{y}_{j'}| - |\mathbf{y}_j|$.

Theorem 3.19 (Theorem 1.17 in [Doe11]). *Let \mathbf{X} be a hypergeometrically distributed random variable. Then*

$$\Pr[\mathbf{X} \geq \frac{5}{4} \cdot \mathbb{E}[\mathbf{X}]] \leq \exp(-\mathbb{E}[\mathbf{X}]/48).$$

We use the following claim.

Claim 3.20. $\frac{4}{15} \cdot t \leq \mathbb{E}[\mathbf{X}] \leq \frac{4}{5} \cdot t.$

Proof. Standard facts about the hypergeometric distribution imply that

$$\mathbb{E}[\mathbf{X}] = \frac{\sqrt{n}}{3} \cdot \frac{n_R}{n_L/2 + \sqrt{n_L}/100 + n_R}.$$

Recall that $r = s/k \geq 1$, $n_L = n(1 - 1/(625r^2)) > 2n/3$, $n_R = n/(625r^2)$, and $t = \frac{k\sqrt{n_R}}{20s} = \frac{\sqrt{n_R}}{20r} = \frac{\sqrt{n}}{500r^2}$. It follows that

$$n_L/2 + \sqrt{n_L}/100 + n_R > n_L/2 > n/3.$$

Therefore

$$\mathbb{E}[\mathbf{X}] < \frac{\sqrt{n}}{3} \cdot \frac{3n_R}{n} = \sqrt{n} \cdot \frac{1}{625r^2} = \frac{4}{5} \frac{\sqrt{n}}{500r^2} = \frac{4}{5} \cdot t.$$

Since $n_L/2 + \sqrt{n_L}/100 + n_R < n$,

$$\mathbb{E}[\mathbf{X}] > \frac{\sqrt{n}}{3} \cdot \frac{n_R}{n} = \frac{\sqrt{n}}{3} \cdot \frac{1}{625r^2} = \frac{4}{15} \frac{\sqrt{n}}{500r^2} = \frac{4}{15} \cdot t.$$

□

It now follows that

$$\begin{aligned} \Pr[\mathbf{X} \geq t] &= \Pr[|V(\mathbf{Z})| \geq t] = \sum_{i \geq t} H(n_R, n_L/2 + \sqrt{n_L}/100, \sqrt{n}/3, i) \\ &= \Pr \left[X \geq \mathbb{E}[\mathbf{X}] \cdot \frac{t}{\mathbb{E}[\mathbf{X}]} \right] \\ &\leq \Pr \left[X \geq \mathbb{E}[\mathbf{X}] \cdot \frac{5}{4} \right] \\ &\leq \exp(-\mathbb{E}[\mathbf{X}]/48) \\ &\leq \exp(-t/180) = \exp\left(\frac{\sqrt{n}}{90000r^2}\right), \end{aligned}$$

which concludes the proof.

□

□

3.2 Proof of Theorem 3.2

We show that given a q -query non-adaptive, one-sided (k, s) -tester, one can obtain a $O(q^{k+1}n)$ -query (k, s) -tester that only queries values on a distribution over random chains.

Let T be a q -query non-adaptive, one-sided (k, s) -monotonicity tester. Therefore, T accepts functions that are k -monotone, and rejects functions that are ε -far from being s -monotone with probability $2/3$.

Define a tester T' that on input a function f does the following: it first gets the queries of T , then for each $(k+1)$ -tuple $q_1 \prec q_2 \prec \dots \prec q_{k+1}$, T' queries an entire uniformly random chain from 0^n to 1^n , conditioned on containing these $k+1$ points. Therefore, T' is also one-sided, makes $O(\binom{q}{k+1}n) = O(q^{k+1}n)$ queries, and its success probability is no less than the success probability of T ⁴.

Define T'' that on input f picks a random permutation $\sigma: [n] \rightarrow [n]$ and then applies the queries of T' to the function $f \circ \pi_\sigma$ (where π_σ is defined as in Definition 3.14). This means that if T' queries q , T'' queries $\pi_\sigma(q)$. Then T'' ignores what T' answers and only rejects if it finds a violation on any one of the chains.

Note that if f is k -monotone, then so is $f \circ \pi_\sigma$, and if f is ε -far from being s -monotone, then so is $f \circ \pi_\sigma$.

Therefore, T'' is one-sided, non-adaptive, and makes $O(q^{k+1}n)$ queries. Since T' is one-sided, it can only reject if it finds a $(k+1)$ -tuple forming a violation to k -monotonicity. So if T' rejects, so does T'' , and it follows that the success probability of T'' is at least the success probability of T' , which is at least $2/3$.

We now claim that the queries of T'' are distributed as $O(\binom{q}{k+1})$ uniformly random chains. While the marginal distribution for each individual chain is the uniform distribution over chains, the joint distribution over these chains is not necessarily independent. Suppose T' queries points q_1, q_2, \dots, q_{k+1} with $q_1 \prec q_2 \prec \dots \prec q_{k+1}$. Then $\pi_\sigma(q_i)$ is a uniformly random point on its weight level, and $\pi_\sigma(q_1) \prec \pi_\sigma(q_2) \prec \dots \prec \pi_\sigma(q_{k+1})$. It follows that a chain chosen uniformly at random conditioned on passing through these points is a uniformly random chain in $\{0, 1\}^n$.

Let p be the success probability of the basic chain tester that picks a uniformly random chain in $\{0, 1\}^n$ and rejects only if it finds a violation to k -monotonicity. Taking a union bound over the chains chosen by T'' , the success probability of T'' is at most $p \cdot \binom{q}{k+1} \leq p \cdot q^{k+1}$. It follows that $p \cdot q^{k+1} \geq 2/3$, from which it easily follows that $p = \Omega(q^{1/(k+1)})$, concluding the proof.

Acknowledgments. We thank our collaborators Clément Canonne and Siyao Guo, who have gracefully refused to co-author this paper.

References

- [AC06] Nir Ailon and Bernard Chazelle. Information theory in property testing and monotonicity testing in higher dimension. *Inf. Comput.*, 204(11):1704–1717, 2006. 1
- [BB16] Aleksandrs Belovs and Eric Blais. A polynomial lower bound for testing monotonicity. In *STOC*, pages 1021–1032. ACM, 2016. 1

⁴We assume that every query made by T belongs to at least one $(k+1)$ -tuple. Queries that do not are of no help to T , since these queries can not be part of a violation to k -monotonicity discovered by T , and we are assuming that T is non-adaptive and one-sided.

- [BBBY12] Maria-Florina Balcan, Eric Blais, Avrim Blum, and Liu Yang. Active property testing. In *FOCS*, pages 21–30. IEEE Computer Society, 2012. [1.3](#)
- [BCGM12] Jop Briët, Sourav Chakraborty, David García-Soriano, and Arie Matsliah. Monotonicity testing and shortest-path routing on the cube. *Combinatorica*, 32(1):35–53, 2012. [1](#)
- [BCO⁺15] Eric Blais, Clément L. Canonne, Igor Carboni Oliveira, Rocco A. Servedio, and Li-Yang Tan. Learning circuits with few negations. In *APPROX-RANDOM*, volume 40 of *LIPICs*, pages 512–527. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. [1](#)
- [BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. Comput. Syst. Sci.*, 47(3):549–595, 1993. [1](#)
- [CDST15] Xi Chen, Anindya De, Rocco A. Servedio, and Li-Yang Tan. Boolean function monotonicity testing requires (almost) $n^{1/2}$ non-adaptive queries. In *STOC*, pages 519–528. ACM, 2015. [1](#)
- [CGG⁺17] Clément L. Canonne, Elena Grigorescu, Siyao Guo, Akash Kumar, and Karl Wimmer. Testing k-monotonicity. 2017. [1](#), [1](#), [1.2](#), [1.3](#), [3](#), [3.1](#), [3.1](#), [3.10](#), [3.11](#)
- [CR14] Clément L. Canonne and Ronitt Rubinfeld. Testing probability distributions underlying aggregated data. In *ICALP (1)*, volume 8572 of *Lecture Notes in Computer Science*, pages 283–295. Springer, 2014. [1.3](#)
- [CS13a] Deeparnab Chakrabarty and C. Seshadhri. An $o(n)$ monotonicity tester for boolean functions over the hypercube. In *STOC*, pages 411–418. ACM, 2013. Journal version as [\[CS16a\]](#). [1](#)
- [CS13b] Deeparnab Chakrabarty and C. Seshadhri. Optimal bounds for monotonicity and Lipschitz testing over hypercubes and hypergrids. In *STOC*, pages 419–428, 2013. [1](#)
- [CS14] Deeparnab Chakrabarty and C. Seshadhri. An optimal lower bound for monotonicity testing over hypergrids. *Theory of Computing*, 10:453–464, 2014. [1](#)
- [CS16a] Deeparnab Chakrabarty and C. Seshadhri. An $o(n)$ Monotonicity Tester for Boolean Functions over the Hypercube. *SIAM J. Comput.*, 45(2):461–472, 2016. [3.2](#)
- [CS16b] Deeparnab Chakrabarty and C. Seshadhri. A $\widetilde{O}(n)$ non-adaptive tester for unateness. *CoRR*, 2016. [1](#), [1.3](#)
- [CST14] Xi Chen, Rocco A. Servedio, and Li-Yang Tan. New algorithms and lower bounds for monotonicity testing. In *FOCS*, pages 286–295. IEEE Computer Society, 2014. [1](#)
- [CWX17] Xi Chen, Erik Waingarten, and Jinyu Xie. Beyond talagrand functions: New lower bounds for testing monotonicity and unateness. *CoRR*, abs/1702.06997, 2017. [1](#), [1.3](#)
- [DGL⁺99] Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved testing algorithms for monotonicity. In *RANDOM-APPROX*, volume 1671 of *Lecture Notes in Computer Science*, pages 97–108. Springer, 1999. [1](#)

- [Doe11] Benjamin Doerr. *Theory of Randomized Search Heuristics: Foundations and Recent Developments*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2011. 3.19
- [EKK⁺00] Funda Ergün, Sampath Kannan, Ravi Kumar, Ronitt Rubinfeld, and Mahesh Viswanathan. Spot-checkers. *J. Comput. Syst. Sci.*, 60(3):717–751, 2000. 1
- [FLN⁺02] Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 474–483, 2002. 1
- [FR10] Shahar Fattal and Dana Ron. Approximating the distance to monotonicity in high dimensions. *ACM Trans. Algorithms*, 6(3), 2010. 1
- [GGL⁺00] Oded Goldreich, Shafi Goldwasser, Eric Lehman, Dana Ron, and Alex Samorodnitsky. Testing monotonicity. *Combinatorica*, 20(3):301–337, 2000. 1, 1.1, 1.3
- [GGR98] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998. 1
- [GK15] Siyao Guo and Ilan Komargodski. Negation-limited formulas. In *APPROX-RANDOM*, volume 40 of *LIPICs*, pages 850–866. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015. 1, 1.3
- [GMOR15] Siyao Guo, Tal Malkin, Igor Carboni Oliveira, and Alon Rosen. The power of negations in cryptography. In *TCC (1)*, volume 9014 of *Lecture Notes in Computer Science*, pages 36–65. Springer, 2015. 1, 1.3
- [Juk12] Stasys Jukna. *Boolean Function Complexity*. Springer, 2012. 1
- [KMS15] Subhash Khot, Dor Minzer, and Muli Safra. On monotonicity testing and Boolean isoperimetric type theorems. In *FOCS*, pages 52–58. IEEE Computer Society, 2015. 1
- [KNOW14] Pravesh Kothari, Amir Nayyeri, Ryan O’Donnell, and Chenggang Wu. Testing surface area. In *SODA*, pages 1204–1214. SIAM, 2014. 1.3
- [KR00] Michael J. Kearns and Dana Ron. Testing problems with sublearning sample complexity. *J. Comput. Syst. Sci.*, 61(3):428–456, 2000. 1.3
- [KS16] Subhash Khot and Igor Shinkar. An $\tilde{o}(n)$ queries adaptive tester for unateness. In *APPROX/RANDOM 2016, Paris, France*, pages 37:1–37:7, 2016. 1
- [LZ16] Chengyu Lin and Shengyu Zhang. Sensitivity conjecture and log-rank conjecture for functions with small alternating numbers. *CoRR*, abs/1602.06627, 2016. 1, 1.3
- [Mar57] A. A. Markov. On the inversion complexity of systems of functions. *Doklady Akademii Nauk SSSR*, 116:917–919, 1957. English translation in [Mar58]. 1, 1
- [Mar58] A. A. Markov. On the inversion complexity of a system of functions. *Journal of the ACM*, 5(4):331–334, October 1958. 3.2

- [Nee14] Joe Neeman. Testing surface area with arbitrary accuracy. In *STOC*, pages 393–397. ACM, 2014. **1.3**
- [NRRS17] Ilan Newman, Yuri Rabinovich, Deepak Rajendraprasad, and Christian Sohler. Testing for forbidden order patterns in an array. In *SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1582–1597, 2017. **1.3**
- [PRR06] Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *Journal of Computer and System Sciences*, 72(6):1012–1042, 2006. **2**
- [Ros15] Benjamin Rossman. Correlation bounds against monotone NC^1 . In *Conference on Computational Complexity (CCC)*, 2015. **1**
- [RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM-J-COMPUT*, 25(2):252–271, April 1996. **1**