Flipping out with many flips: hardness of testing k-monotonicity

Elena Grigorescu* Akash Kumar† Karl Wimmer‡

July 13, 2017

Abstract

A function $f : \{0,1\}^n \to \{0,1\}$ is said to be k-monotone if it flips 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. Our work is part of a renewed focus in understanding testability of properties characterized by freeness of arbitrary order patterns as a generalization of monotonicity. Recently, Canonne et al. (ITCS 2017) initiate the study of k-monotone functions in the area of property testing, and Newman et al. (SODA 2017) study testability of families characterized by freeness from order patterns on real-valued functions over the line $[n]$ domain.

We study k-monotone functions in the more relaxed *parametrized property testing model*, introduced by Parnas et al. (JCSS, 72(6), 2006). In this process we resolve a problem left open in previous work. Specifically, our results include the following.

1. Testing 2-monotonicity on the hypercube non-adaptively with one-sided error requires an exponential in $\sqrt{n}$ number of queries. This behavior shows a stark contrast with testing (1-)monotonicity, which only needs $\tilde{O}(\sqrt{n})$ queries (Khot et al. (FOCS 2015)). Furthermore, even the apparently easier task of distinguishing 2-monotone functions from functions that are far from being $n^{.01}$-monotone also requires an exponential number of queries.
2. On the hypercube $[n]^d$ domain, there exists a testing algorithm that makes a constant number of queries and distinguishes functions that are k-monotone from functions that are far from being $O(kd^2)$-monotone. Such a dependency is likely necessary, given the lower bound above for the hypercube.

*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.

‡Duquesne University. Email: wimmerk@duq.edu.

1 Introduction

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. Two prolific lines of research in the study of Boolean functions have recently seen ultimate results: on one hand, the study of families exhibiting algebraic symmetries (such as low-degree polynomials, and triangle-freeness), explicitly initiated in [KS08], and on the other hand, the study of functions with less symmetry, but whose values respect a monotone order relation, initiated in [GGL+00]. Extending the structural features of these classes of functions, in this work we view families of Boolean functions with less symmetry as interpolating from basic families of monotone functions to families characterized by freeness of more complex *order patterns*. This perspective is apparent in the recent works of Newman et al [NRRS17] and Canonne et al [CGG+17] (a superset of the authors) who propose the study of families exhibiting freeness from more general order patterns in the property testing setting. Here we study the property of being k-*monotone*, building towards understanding testability of freeness from arbitrary order patterns.

We first introduce some standard definitions. A *property* of Boolean functions from a discrete domain D to $\{0,1\}$ is a subset of $\{f\colon D \to \{0,1\}\}$. Given two functions $f, g\colon D \to \{0,1\}$, denote by $d(f,g)$ the (normalized) Hamming distance between them, i.e. $d(f,g) = \Pr_{x\sim D}[\, f(x) \neq g(x)\,]$. The distance of a function f to $\mathcal{P}$ is defined as $d(f,\mathcal{P}) = \min_{g\in\mathcal{P}} d(f,g)$. A q-*query tester* for $\mathcal{P}$ is a randomized algorithm $\mathcal{T}$ that takes as input $\varepsilon \in (0,1]$, and has query access to a function $f\colon D \to \{0,1\}$. After making at most $q(\varepsilon)$ queries, $\mathcal{T}$ distinguishes between the following two cases: i) if $f \in \mathcal{P}$, then $\mathcal{T}$ accepts, with probability $2/3$; ii) if $d(f,\mathcal{P}) \geq \varepsilon$, then $\mathcal{T}$ rejects w.p. $2/3$. 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 *one-sided*; 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 $d(f,\mathcal{P}) \geq \epsilon$, we say that f is ϵ-*far* from $\mathcal{P}$. We will frequently use "far" to denote "$\Omega(1)$-far".

In this work we focus on the property of being k-monotone, which we formalize next.

Definition 1.1 (**k-monotonicity**). A function $f : D \to \{0,1\}$ is k-*monotone* if there do not exist $x_1 \prec x_2 \prec \ldots \prec x_{k+1}$ in D, such that $f(x_1) = 1$ and $f(x_i) \neq f(x_{i+1})$ for all $i \in [k]$. Equivalently, a Boolean k-monotone function is free of the pattern $\langle f(x_1), f(x_2), \ldots, f(x_{k+1})\rangle = \langle 1,0,1,0,\ldots,0,1\rangle$. Note that 1-monotone functions are exactly the functions that are monotone.

Testing monotonicity has drawn interest in the theoretical computer science community for almost two decades (e.g., [GGL+00, DGL+99, EKK+00, FLN+02, Fis04, BRW05, AC06, HK08, BGJ+09, BCGM12, FR10, BBM12, CS13b, CS13a, CS14, BRY14, CST14, CDST15, KMS15, BB16, CWX17]), especially due to the naturalness of the property, as well as its evasiveness to tight analysis.

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]. In these areas, as k increases, k-monotone functions are viewed as robust classes sitting between the very structured monotone functions, and general Boolean functions. Following these developments, Canonne et al [CGG+17] recently initiate the study of k-monotonicity in the property testing model. Here we continue this study and reveal a stark difference from testing monotonicity while disproving a conjecture from [CGG+17]. Our study leads to open questions that might have implications to monotonicity testing.

1.1 Testing 2-monotonicity on the hypercube

Recall that a function $f : \{0,1\}^n \to \{0,1\}$ is 2-monotone if it is free from $\langle f(x_1), f(x_2), f(x_3)\rangle = \langle 1, 0, 1\rangle$, for $x_1 \prec x_2 \prec x_3$. How much more difficult can one-sided testing of this pattern be than testing freeness from $\langle f(x_1), f(x_2)\rangle = \langle 1, 0\rangle$ (i.e., monotonicity)?

The first proof of the $O(n)$-query one-sided non-adaptive tester for monotonicity 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 (i.e., $x_1 \prec x_2$, with $\langle f(x_1), f(x_2)\rangle = \langle 1, 0\rangle$, and there is no x_3 such that $x_1 \prec x_3 \prec x_2$). 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 'block 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 $\sqrt{n}$.

This non-locality of a violation seems to be a reason why testing k-monotonicity (non-adaptively, with one-sided error) turns out to be very different than just testing monotonicity. Initial lower bounds from [CGG+17] only show a separation for $k \geq 3$. Indeed, they show that testing k-monotonicity non-adaptively with one sided error requires $\Omega(n/k^2)^{k/4}$ queries, which for $k \geq 3$ beats the recent tester for monotonicity of $\tilde{O}(\sqrt{n})$ query complexity [CS13a, KMS15]. However, no such separation was known for $k = 2$. Furthermore, they conjecture that k-monotonicity gets gradually more difficult to test as k increases, and that k-monotonicity should be testable with $\Theta(n^{O(k)})$ queries. In this work we show that this is not the case, and in fact even testing 2-monotonicity itself requires an exponential number of queries.

Theorem 1.2. *(Rephrasing Corollary 2.4) Testing 2-monotonicity non-adaptively with one-sided error requires $2^{\Omega(\sqrt{n})}$ queries.*

For comparison, [CGG+17] obtains a tester for 2-monotonicity with $2^{O(\sqrt{n}\cdot\log n)}$ queries, and [BCO+15] obtains a $2^{\Omega(\sqrt{n})}$ lower bound for PAC learning. Hence, testing 2-monotonicity is a natural property for which testing is essentially as hard as PAC learning.

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.

A canonical test for one-sided testing As alluded to above, in order to test k-monotonicity with one-sided error, a tester should only reject if it finds a *violation* in the form of a sequence $x_1 \prec x_2 \prec \ldots \prec x_{k+1}$ 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 and rejects only if it finds a violation.

Definition 1.3. We define the (basic) *chain tester* to be the algorithm that picks a *uniformly random chain* $\mathbf{Z} = \langle 0^n \prec \mathbf{z}_1 \prec \mathbf{z}_2 \prec \cdots \mathbf{z}_{n-1} \prec 1^n \rangle$ 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 denote by a chain tester an algorithm that picks multiple chains (possibly from a joint distribution).

All previous tests for monotonicity (e.g., [GGL+00, CS13b, CST14, KMS15]) imply a chain tester incurring only a small polynomial blow-up in the query complexity.

As expected, a chain tester is indeed implied by any non-adaptive one-sided algorithm for k-monotonicity.

Theorem 1.4. *(Corollary to Theorem 2.2) Any non-adaptive one-sided q-query tester for k-monotonicity over $\{0,1\}^n$ implies an $O\left(q^{k+1}n\right)$-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 $p = \Omega(1/q^{k+1})$.*

To prove our lower bounds, we create families for which the chain tester fails. We first remark that the chain tester does very well on the 'block functions' described above that are far from being 2-monotone. Indeed, every chain from 0^n to 1^n will uncover a violation to 2-monotonicity! In our constructions, 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). The violation graph is to some extent the only handle we have on arguing structural properties of functions that are far from being k-monotone. 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.

In fact, our results work in a larger context of parametrized testing, and our lower bounds hold even for apparently much easier tasks, as we describe next.

1.2 Parametrized monotonicity testing on the hypercube

The notion of k-monotonicity allows us to propose the natural problem of approximating the "monotonicity" of a function. For a concrete example, 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 $n^{.01}$-monotone requires an exponential number of queries.

Theorem 1.5. *(Rephrasing Corollary 2.3) 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.*

This is also the setting of parametrized property testing introduced by Parnas et al [PRR06]. In this setting, a property is parametrized by an integer k and denoted $\mathcal{P} = \{\mathcal{P}_k\}_k$. A (k_1, k_2)-*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, must accept if $f \in \mathcal{P}_{k_1}$, with probability at least 2/3; and reject if $d(f, \mathcal{P}_{k_2}) > \varepsilon$, with probability at least 2/3.

Hence, we are interested in $(k, g(k,n))$-testing for parametrized monotonicity, denoted $\{\mathcal{M}_k\}_k$.

Theorem 1.6. *Given $2 \le k \le g(k,n) = o(\sqrt{n})$, $(k, g(k,n))$-testing $\{\mathcal{M}_k\}_k$ (on the hypercube) non-adaptively with one-sided error requires $2^{\Omega\left(\frac{\sqrt{n}}{(k+1)(g(k,n)/k)^2}\right)}$ queries.*

Parametrized monotonicity testing may provide a new angle for approaching yet unanswered questions in the goal of understanding the role of adaptivity in testing monotonicity. The current strong lower bounds for testing monotonicity adaptively [CST14, CDST15, BB16] come polynomially close to the one-sided error upper bounds [KMS15], yet the question of whether adaptive algorithms can beat the current lower bounds still remain open.

1.3 Parametrized monotonicity testing on the hypergrid

We next study parametrized monotonicity testing of Boolean functions over the hypergrid $[n]^d$ domain. For these domains [CGG+17] shows that testing k-monotonicity non-adaptively with two-sided error can be performed with a number of queries independent of n, but *exponential* in the dimension d. More explicitly, $q(n,d,\varepsilon,k) = \min\left(\tilde{O}\left(\frac{1}{\varepsilon^2}(5kd/\varepsilon)^d\right), 2^{\tilde{O}\left(k\sqrt{d}/\varepsilon^2\right)}\right)$.

Our algorithmic results show a contrast to the hypercube case. We obtain a tester with *constant* query complexity (independent of k, n, d), albeit trading off for the (k_1, k_2) parameters that the tester needs to distinguish between.

Theorem 1.7. *There is a non-adaptive two-sided tester using* $\mathrm{poly}(1/\varepsilon)$ *samples for $(k, 2kd^2/\varepsilon)$-testing of $\{\mathcal{M}_k\}_k$ on the grid $[n]^d$.*

Since two-sided versus one-sided testing for monotonicity on the hypercube is an unresolved problem, and in light of our exponential bounds for the $\{0,1\}^n$ domain, the dependence in d in the $g(n,d,k) = 2kd^2$ function appears necessary. Achieving a sublinear dependence on d while controlling the query complexity would require new ideas that apply to the hypercube, and our subject to our lower bound.

1.4 Related work and the larger perspective

As previously mentioned, [NRRS17] proposes studying more general order patterns for real-valued functions defined over the line domain $[n]$. They view a pattern of length k as a permutation $\pi : [k] \to [k]$. A function $f : [n] \to \mathbb{R}$ if π-free if there does not exist indices $i_1 < i_2 < \ldots < i_k$ such that $f(i_x) > f(i_y)$ whenever $\pi(x) < \pi(y)$. This is a more fine-grained notion than k-monotonicity; for example, $(2,1)$-freeness describes monotonicity, and the intersection of $(2,1,3)$-freeness and $(3,1,2)$-freeness describes 2-monotonicity over the reals. The authors of [NRRS17] obtain several results for non-adaptive one-sided testing on the line, essentially distinguishing monotone and non-monotone patterns. They further raise the question of characterizing the testing complexity as a function of the structure of the pattern. Their lower bounds are $\Omega(\sqrt{n})$; this is quite strong given that the domain size is n, and this bound is likely not tight for most patterns. Transferring this question to our setting, we may ask:

Question 1.8. *On the hypercube, do all order patterns (longer than some constant length) require* $\exp(O(\sqrt{n}))$ *queries to test? Can this lower bound be strengthened to* $\exp(O(n))$ *for most patterns? Can these patterns be characterized?*

The description of a function family as being free from a pattern is common in property testing of Boolean functions, especially in the study of families that exhibit 'affine-invariance'. In that line of work, the Boolean domain $\{0,1\}^n$ is viewed as a vector space over the field of two elements, and now one may perform algebraic operations over the domain. For instance, the property of being 'triangle-free' is defined as freeness from the pattern $\langle f(x), f(y), f(z)\rangle = \langle 1,1,1\rangle$ whenever $x+y+z=0$. An almost complete characterization of when such a property is testable with a constant number of queries has recently been obtained as a result of sustained interest in this quest [Gre05, KSV11, Sha10, BCSX11, BFL13, BFH+13, Yos14, BGS15].

Monotonicity and k-monotonicity however exhibit less symmetric structure, but these families are still invariant under permutations of the variables. Invariance under permutations of the variables is also maintained by properties defined by freeness of order patterns. While this symmetry is not as strong as to imply constant query testers in general, it is however crucial in the design of algorithms for such properties. For example, we use this invariance in showing the reduction to a canonical tester in Theorem 1.4.

The notion of k-monotonicity can be easily extended to real-valued functions. In fact, this extension has already implied 'tolerant' monotonicity testers for families consisting of functions $f\colon [n]^d \to [0,1]$ [CGG+17]. Furthermore, [CGG+17] reveals connections between testing k-monotonicity and testing surface area [KR00, BBBY12, 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, KS16, BCP+17], and lower bounds in [CWX17, BCP+17].

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].

Organization We proceed with the proofs of our lower bounds for the hypercube domain in Section 2 and with our algorithmic results for the hypergrid in Section 3.

2 Lower Bounds over the Hypercube

We prove all of our results in this section. We first show in Theorem 2.1 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 2.1. *Given* $2 \le k \le g(k,n) = o(\sqrt{n})$*, 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* $o\Big(\exp\Big(-C\frac{\sqrt{n}}{(g(k,n)/k)^2}\Big)\Big)$.

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 2.2. *Any non-adaptive one-sided q-query $(k, g(k,n))$-tester for Boolean monotonicity over $\{0,1\}^n$ implies an $O\left(q^{k+1}n\right)$-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 2.1 and Theorem 2.2 imply Theorem 1.6.

Instantiating k and $g(k,n)$ in Theorem 1.6, we obtain the following immediate corollaries.

Corollary 2.3. *Any non-adaptive one-sided $(2, n^{0.01})$-tester for parametrized monotonicity requires $\exp(\Omega(n^{0.48}))$ queries.*

Corollary 2.4. *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 2.4 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 2.5. *Any non-adaptive, one sided tester for k-monotonicity, for $2 \leq k = o(\sqrt{n})$, requires $\Omega(\exp(\sqrt[4]{n}))$ queries.*

2.1 Proof of Theorem 2.1

Define the *weight* of an element x in $\{0,1\}^n$, denoted $|x|$, to be the number of non-zero entries of x.

We first recall some standard useful facts.

Fact 2.6. *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 2.7. *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 2.1.1, we define our hard family, and show that every function in this family is indeed far from $g(k,n)$-monotonicity. In Section 2.1.2 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 2.8 (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 $B_1, B_2, \ldots, B_\ell$ 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*, denoted[1] $\mathrm{BB}(n,\ell)\colon \{0,1\}^n \to \{0,1\}$, 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 $\mathrm{BB}(n,\ell)$ is a totally symmetric function: it is unchanged under permutations of its inputs. Thus, we can partition $\{0,1,\ldots,n\}$ into ℓ intervals $I_1, I_2, \ldots, I_\ell$, 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 2.10, after making an important definition.

Definition 2.9 (Violation hypergraph ([CGG+17]))**.** Given a function $f\colon \{0,1\}^n \to \{0,1\}$, the *violation hypergraph of f* is $H_{\mathrm{viol}}(f) = (\{0,1\}^n, E(H_{\mathrm{viol}}))$ where $(x_1, x_2, \cdots, x_{\ell+1}) \in E(H_{\mathrm{viol}})$ if the ordered $(\ell+1)$-tuple $x_1 \prec x_2 \prec \ldots \prec x_{\ell+1}$ (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 2.10 ([CGG+17, Claim 3.8])**.** *Let $h \overset{\mathrm{def}}{=} \mathrm{BB}(n,\ell)$. 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 $y_1 \preceq \cdots \preceq y_{\ell-1}$ 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.

Claim 2.11. *Let $h \overset{\mathrm{def}}{=} \mathrm{BB}(n, 3k)$. Then $d(h, \mathcal{M}_k) \geq \Omega(1)$.*

Proof of Claim 2.11. By Claim 2.10, $H_{\mathrm{viol}}(h)$ 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.

2.1.1 The Hard Family

In what follows, let $s \overset{\mathrm{def}}{=} g(k,n)$. Also, let $r \overset{\mathrm{def}}{=} \frac{g(k,n)}{k}$. 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 $z = (x,y)$,

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

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

where $x \in \{0,1\}^{|L|}$ and $y \in \{0,1\}^{|R|}$. We define $\mathrm{MID}_L \stackrel{\text{def}}{=} \{i :|\ i - \frac{n_L}{2}\ | \leq \frac{\sqrt{n_L}}{100}\}$, to denote the set of "balanced" inputs restricted to the set of indices in L.

Assuming k is even[3], we define $f_{n_L} : \{0,1\}^n \to \{0,1\}$ by

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

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 $x \in \mathrm{MID}_L$, 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 2.12. *The function $f = f_{n_L}$ is $\Omega(1)$-far from being s-monotone.*

Proof. By Fact 2.7, 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 2.10, 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 the 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 2.11, 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.

Definition 2.13. 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 $\pi_\sigma : \{0,1\}^n \to \{0,1\}^n$ is a permutation that sends the string $\{(a_1, a_2, \ldots, a_n)\}$ to $\{a_{\sigma(1)}, a_{\sigma(2)}, \ldots, a_\sigma(n)\}$ for a permutation $\sigma : [n] \to [n]$. 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 2.12.

Claim 2.14. *Every $f \in \mathcal{F}_{k,s}$, f is $\Omega(1)$-far from s-monotone.*

Therefore, we proved the distance property from Theorem 2.1

[3] For 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.

2.1.2 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 2.15. *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* $\exp\left(-C\frac{k^2}{s^2}\sqrt{n}\right)$.

Let Z be a fixed chain $0^n = z_0 \prec z_1 \prec z_2 \prec \cdots \prec z_n = 1^n$ in $\{0,1\}^n$. Note that $f_{n_L}(z_i) = f_{n_L}(x_i, y_i) = 0$ if $|x_i| \notin \mathrm{MID}_L$. Thus, if there is a violation to k-monotonicity in Z, then it can be found among points in Z where $|x_i| \in \mathrm{MID}_L$. 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, $f_{n_L}(x_i, y_i) = \mathrm{BB}(n_R, 3s)(y_i)$. Since BB is a totally symmetric function, to determine if Z exhibits a violation, it is enough to analyze the set

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

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

Claim 2.16. *Suppose Z contains a violation to k-monotonicity. Then $|V(Z)| \geq k\sqrt{n_R}/(16s)$.*

Proof. By Fact 2.6, 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 $BB(n_R, 3s)$ 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 $0^n \prec \mathbf{z}_1 \prec \cdots \prec \mathbf{z}_{n-1} \prec 1^n$.

Proof of Claim 2.15. The proof follows from Claim 2.16 and the following claim.

Claim 2.17. *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 $\mathbf{z}_{j-1} \prec \mathbf{z}_j \prec \cdots \prec \mathbf{z}_w \prec \mathbf{z}_{w+1}$. By construction, we have $f(\mathbf{z}_\ell) = 0$ if $\ell \leq j-1$ or $\ell \geq w+1$. Further, $V(\mathbf{Z}) = \{|\mathbf{y}_j|, |\mathbf{y}_{j+1}|, \ldots, |\mathbf{y}_{w-1}|, |\mathbf{y}_w|\}$, 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 2.16, 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 2.18 (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 2.19. $\frac{4}{15} \cdot t \leq \mathbb{E}[\mathbf{X}] \leq \frac{4}{5} \cdot t$.

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.

□

□

2.2 Proof of Theorem 2.2

We show that given a q-query non-adaptive, one-sided (k,s)-tester, one can obtain a $O\left(q^{k+1}n\right)$-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 $q_1 \prec q_2 \prec \cdots \prec q_{k+1}$, 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 $\boldsymbol{\sigma}\colon [n] \to [n]$ and then applies the queries of T' to the function $f \circ \pi_{\boldsymbol{\sigma}}$ (where $\pi_{\boldsymbol{\sigma}}$ is defined as in Definition 2.13). This means that if T' queries q, T'' queries $\pi_{\boldsymbol{\sigma}}(q)$. Then T'' ignores what T' answers and only rejects if it finds a violation on any one of the chains.

[4] We 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.

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

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 $q_1, q_2, \cdots q_{k+1}$ with $q_1 \prec q_2 \prec \cdots \prec q_{k+1}$. Then $\pi_{\boldsymbol{\sigma}}(q_i)$ is a uniformly random point on its weight level, and $\pi_{\boldsymbol{\sigma}}(q_1) \prec \pi_{\boldsymbol{\sigma}}(q_2) \prec \ldots \prec \pi_{\boldsymbol{\sigma}}(q_{k+1})$. 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 of 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.

3 Upper Bounds over the Hypergrid

In this section we prove Theorem 1.7. In this section, we consider Boolean functions over the hypergrid $[n]^d$. For convenience, we will define $[n] = \{0, 1, 2, \ldots, n-1\}$. Assuming m divides n, we define $\mathcal{B}_{m,n} : [n]^d \to [m]^d$ be the mapping such that $\mathcal{B}_{m,n}(y)_i = \lfloor y_i/m \rfloor$ for $1 \leq i \leq d$. For $x \in [m]^d$, we define $\mathcal{B}^{-1}_{m,n}(x)$ to be the inverse image of x under $\mathcal{B}_{m,n}$. Specifically, $\mathcal{B}^{-1}_{m,n}(x)$ is the set of points of the form $m \cdot x + [n/m]^d$, using the standard definitions of scalar multiplication and coordinatewise addition. That is, $\mathcal{B}^{-1}_{m,n}$ is a "coset" of $[n/m]^d$ points in $[n]^d$. We will call these cosets *blocks*, and we will say that $h : [n]^d \to \{0,1\}$ is an m-block function if it is constant on $\mathcal{B}^{-1}_{m,n}(x)$ for every $x \in [m]^d$. For readability, we will often suppress the dependence on m and n.

Claim 3.1 (Claim 7.1, [CGG+17])**.** *Suppose $f : [n]^d \to \{0,1\}$ is k-monotone. Then there is an m-block function $h : [n]^d \to \{0,1\}$ such that $d(f,h) < kd/m$.*

Claim 3.2. *Suppose $h : [n]^d \to \{0,1\}$ is an m-block function. Then h is $((m-1)d-1)$-monotone.*

Proof. Without loss of generality, we assume that $h(0^d) = 0$. Suppose for the sake of contradiction that h is an m-block function such that h contains a violation to $((m-1)d-1)$-monotonicity. Equivalently, there exists $y_0 \prec y_1 \prec \cdots \prec y_{(m-1)d-1}$ in $[n]^d$ such that $h(y_0) = 1$ and $h(y_i) \neq h(y_{i+1})$ for $0 \leq i \leq (m-1)d-2$. Since h is constant on each block, we have no two y_i's are in the same block. Thus the set $\{\mathcal{B}_{m,n}(y_i) : 0 \leq i \leq (m-1)d-1\}$ contains $(m-1)d$ distinct vectors in $[m]^d$ that can be totally ordered. This implies that 0^d and $(m-1)^d$ (this is a vector of d coordinates, all of which are $m-1$) are in this set. Clearly, 0^d is the "smallest" vector in this total order, and it follows from our definition of violation that $h(0^d) = 1$. This is a contradiction, since we assumed at the outset that $h(0^d) = 0$. Thus h does not contain a violation to $((m-1)d-1)$-monotonicity, and h is $((m-1)d-1)$-monotone. ☐

We define the m-block-coarsening of a function $f : [n]^d \to \{0,1\}$ to be the m-block function $h : [n]^d \to \{0,1\}$ such that $d(f,h)$ is as small as possible. We would like to use query access to f

to get query access to h, but it is not guaranteed we can do this. Rather, for each $x \in [m]^d$, we randomly select a set $\mathbf{S}_x$ of points in the block $\mathcal{B}^{-1}(x)$, where we choose $|\mathbf{S}_x|$ to be large enough such that

$$\left| \Pr_{\mathbf{z} \sim \mathbf{S}_{\mathcal{B}(y)}}[f(\mathbf{z}) = 0] - \Pr_{\mathbf{z} \sim \mathcal{B}^{-1}(\mathcal{B}(y))}[f(\mathbf{z}) = 0] \right| \le 1/9$$

with high probability. Our target query complexity is independent of m, so we can *not* necessarily query points from each $\mathbf{S}_x$; these points merely allow us to talk about a specific (randomly chosen) function. We denote by $\boldsymbol{h}' : [n]^d \to \{0,1\}$ the m-block function such that

$$\boldsymbol{h}'(y) = \operatorname*{argmax}_{b \in \{0,1\}} \Pr_{\mathbf{z} \sim \mathbf{S}_{\mathcal{B}(y)}}[f(\mathbf{z}) = b],$$

breaking ties arbitrarily.

Claim 3.3. *We have $d(f, \boldsymbol{h}') \le \frac{5}{4} d(f,h)$.*

Proof. Let B be a block such that h and $\boldsymbol{h}'$ disagree. Then the wrong bit was estimated to be the majority value of f when restricted to B. By our construction of $\boldsymbol{h}'$, we must have

$$\left| \Pr_{\mathbf{y} \sim B}[f(\mathbf{y}) = h(\mathbf{y})] - \Pr_{\mathbf{y} \sim B}[f(\mathbf{y}) = \boldsymbol{h}'(\mathbf{y})] \right| \le 1/9$$

since exactly one of $h(\mathbf{y})$ and $\boldsymbol{h}'(\mathbf{y})$ is 0 and the other is 1. It follows that $\Pr_{\mathbf{y} \sim B}[f(\mathbf{y}) \neq h(\mathbf{y})] \ge 4/9$ and $\Pr_{\mathbf{y} \sim B}[f(\mathbf{y}) \neq \boldsymbol{h}'(\mathbf{y})] \le 5/9$. Combining these inequalities, we get $\Pr_{\mathbf{y} \sim B}[f(\mathbf{y}) \neq \boldsymbol{h}'(\mathbf{y})] \le \frac{5}{4} \Pr_{\mathbf{y} \sim B}[f(\mathbf{y}) \neq h(\mathbf{y})]$.

Clearly, if h and $\boldsymbol{h}'$ do not disagree on B, then the previous probabilities are equal and the inequality holds. Thus, for all blocks B, $\Pr_{\mathbf{y} \sim B}[f(\mathbf{y}) \neq \boldsymbol{h}'(\mathbf{y})] \le \frac{5}{4} \Pr_{\mathbf{y} \sim B}[f(\mathbf{y}) \neq h(\mathbf{y})]$. The claim follows by taking the expected value of each side over a uniformly chosen block.

□

Proof of Theorem 1.7. In our tester, we set $m = (2kd^2/\varepsilon + 1)/d + 1$, and the tester simply estimates $d(f, \boldsymbol{h}')$ to within $\pm\varepsilon/8$, which can be done with $\widetilde{O}(1/\varepsilon^2)$ queries. By Claim 3.1, if f is k-monotone, then there is an m-block function h such that

$$d(f,h) < kd/m = kd/((2kd^2/\varepsilon + 1)/d + 1) < kd^2/(2kd^2/\varepsilon) = \varepsilon/2,$$

and it follows that $d(f, \boldsymbol{h}') \le \frac{5}{4} d(f,h) \le \frac{5}{4}(\varepsilon/2) = 5\varepsilon/8$. By Claim 3.2, if f is far from $2kd^2/\varepsilon$-monotone, then it is ε-far from every m-block function, as for our setting of m, we have $((m-1)d-1) = 2kd^2/\varepsilon$. It follows that $d(f, \boldsymbol{h}') \ge \varepsilon$. Thus, the tester correctly accepts if the estimate of $d(f, \boldsymbol{h}')$ is at most $3\varepsilon/4$ and correctly rejects if this estimate is at least $7\varepsilon/8$.

□

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, 1.2

[BBBY12] Maria-Florina Balcan, Eric Blais, Avrim Blum, and Liu Yang. Active property testing. In *FOCS*, pages 21–30. IEEE Computer Society, 2012. 1.4

[BBM12] Eric Blais, Joshua Brody, and Kevin Matulef. Property testing lower bounds via communication complexity. *Computational Complexity*, 21(2):311–358, 2012. 1

[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. 1.1

[BCP+17] Roksana Baleshzar, Deeparnab Chakrabarty, Ramesh Krishnan S. Pallavoor, Sofya Raskhodnikova, and C. Seshadhri. Optimal unateness testers for real-valued functions: Adaptivity helps. *CoRR*, abs/1703.05199, 2017. 1.4

[BCSX11] Arnab Bhattacharyya, Victor Chen, Madhu Sudan, and Ning Xie. Testing linear-invariant non-linear properties. *Theory of Computing*, 7(1):75–99, 2011. 1.4

[BFH+13] Arnab Bhattacharyya, Eldar Fischer, Hamed Hatami, Pooya Hatami, and Shachar Lovett. Every locally characterized affine-invariant property is testable. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 429–436, 2013. 1.4

[BFL13] Arnab Bhattacharyya, Eldar Fischer, and Shachar Lovett. Testing low complexity affine-invariant properties. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1337–1355, 2013. 1.4

[BGJ+09] Arnab Bhattacharyya, Elena Grigorescu, Kyomin Jung, Sofya Raskhodnikova, and David P. Woodruff. Transitive-closure spanners. In *SODA*, pages 932–941. SIAM, 2009. 1

[BGS15] Arnab Bhattacharyya, Elena Grigorescu, and Asaf Shapira. A unified framework for testing linear-invariant properties. *Random Struct. Algorithms*, 46(2):232–260, 2015. 1.4

[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

[BRW05] Tŭgkan Batu, Ronitt Rubinfeld, and Patrick White. Fast approximate PCPs for multidimensional bin-packing problems. *Inf. Comput.*, 196(1):42–56, 2005. 1

[BRY14] Piotr Berman, Sofya Raskhodnikova, and Grigory Yaroslavtsev. L_p-testing. In *STOC*, pages 164–173. ACM, 2014. 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, 1.2

[CGG+17] Clément L. Canonne, Elena Grigorescu, Siyao Guo, Akash Kumar, and Karl Wimmer. Testing k-monotonicity. In *ITCS*, page (to appear), 2017. 1, 1, 1.1, 1.1, 1.1, 1.3, 1.4, 2, 2.1, 2.1, 2.9, 2.10, 3.1

[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. 1.4

[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, 1.1

[CS13b] Deeparnab Chakrabarty and C. Seshadhri. Optimal bounds for monotonicity and Lipschitz testing over hypercubes and hypergrids. In *STOC*, pages 419–428, 2013. 1, 1.1

[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

[CS16b] Deeparnab Chakrabarty and C. Seshadhri. A $\widetilde{O}(n)$ non-adaptive tester for unateness. *CoRR*, 2016. 1.4

[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, 1.1, 1.2

[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, 1.4

[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. 2.18

[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

[Fis04] Eldar Fischer. On the strength of comparisons in property testing. *Inf. Comput.*, 189(1):107–116, 2004. 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, 1.1, 1.4

[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, 1.4

[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, 1.4

[Gre05] Ben Green. A Szemerédi-type regularity lemma in abelian groups. *Geometric and Functional Analysis*, 15(2):340–376, 2005. 1.4

[HK08] Shirley Halevy and Eyal Kushilevitz. Testing monotonicity over graph products. *Random Struct. Algorithms*, 33(1):44–67, 2008. 1

[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, 1.1, 1.1, 1.2

[KNOW14] Pravesh Kothari, Amir Nayyeri, Ryan O'Donnell, and Chenggang Wu. Testing surface area. In *SODA*, pages 1204–1214. SIAM, 2014. 1.4

[KR00] Michael J. Kearns and Dana Ron. Testing problems with sublearning sample complexity. *J. Comput. Syst. Sci.*, 61(3):428–456, 2000. 1.4

[KS08] Tali Kaufman and Madhu Sudan. Algebraic property testing: the role of invariance. pages 403–412. ACM, 2008. 1

[KS16] Subhash Khot and Igor Shinkar. An ~o(n) queries adaptive tester for unateness. In *APPROX/RANDOM 2016, Paris, France*, pages 37:1–37:7, 2016. 1.4

[KSV11] Daniel Král', Oriol Serra, and Lluís Vena. A removal lemma for systems of linear equations over finite fields. *Israel Journal of Mathematics*, pages 1–15, 2011. 1.4

[LZ16] Chengyu Lin and Shengyu Zhang. Sensitivity conjecture and log-rank conjecture for functions with small alternating numbers. *CoRR*, abs/1602.06627, 2016. 1, 1.4

[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, 1.1

[Mar58] A. A. Markov. On the inversion complexity of a system of functions. *Journal of the ACM*, 5(4):331–334, October 1958. 3

[Nee14] Joe Neeman. Testing surface area with arbitrary accuracy. In *STOC*, pages 393–397. ACM, 2014. 1.4

[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, 1.4

[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. 1.2

[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

[Sha10] Asaf Shapira. Green's conjecture and testing linear invariant properties. In *Property Testing - Current Research and Surveys [outgrow of a workshop at the Institute for Computer Science (ITCS) at Tsinghua University, January 2010]*, pages 355–358, 2010. 1.4

[Yos14] Yuichi Yoshida. A characterization of locally testable affine-invariant properties via decomposition theorems. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 154–163, 2014. 1.4