Nearly Optimal Bounds for Sample-Based Testing and Learning of k-Monotone Functions

Hadley Black
University of California, San Diego, USA

Abstract

We study monotonicity testing of functions $f\colon \{0,1\}^d \to \{0,1\}$ using *sample-based* algorithms, which are only allowed to observe the value of f on points drawn independently from the uniform distribution. A classic result by Bshouty-Tamon (J. ACM 1996) proved that monotone functions can be learned with $\exp(\widetilde{O}(\min\{\frac{1}{\varepsilon}\sqrt{d}, d\}))$ samples and it is not hard to show that this bound extends to testing. Prior to our work the only lower bound for this problem was $\Omega(\sqrt{\exp(d)/\varepsilon})$ in the small ε parameter regime, when $\varepsilon = O(d^{-3/2})$, due to Goldreich-Goldwasser-Lehman-Ron-Samorodnitsky (Combinatorica 2000). Thus, the sample complexity of monotonicity testing was wide open for $\varepsilon \gg d^{-3/2}$. We resolve this question, obtaining a nearly tight lower bound of $\exp(\Omega(\min\{\frac{1}{\varepsilon}\sqrt{d}, d\}))$ for all ε at most a sufficiently small constant. In fact, we prove a much more general result, showing that the sample complexity of k-*monotonicity* testing and learning for functions $f\colon \{0,1\}^d \to [r]$ is $\exp(\Omega(\min\{\frac{rk}{\varepsilon}\sqrt{d}, d\}))$. For testing with one-sided error we show that the sample complexity is $\exp(\Omega(d))$.

Beyond the hypercube, we prove nearly tight bounds (up to polylog factors of $d, k, r, 1/\varepsilon$ in the exponent) of $\exp(\widetilde{\Theta}(\min\{\frac{rk}{\varepsilon}\sqrt{d}, d\}))$ on the sample complexity of testing and learning measurable k-monotone functions $f\colon \mathbb{R}^d \to [r]$ under product distributions. Our upper bound improves upon the previous bound of $\exp(\widetilde{O}(\min\{\frac{k}{\varepsilon^2}\sqrt{d}, d\}))$ by Harms-Yoshida (ICALP 2022) for Boolean functions ($r = 2$).

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms; Mathematics of computing → Probabilistic algorithms

Keywords and phrases Property testing, learning, Boolean functions, monotonicity, k-monotonicity

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2024.37

Category RANDOM

Related Version *Full Version*: https://arxiv.org/abs/2310.12375

Funding *Hadley Black*: This work was done while the author was at UCLA supported by NSF award AF:Small 2007682, NSF Award: Collaborative Research Encore 2217033.

Acknowledgements We would like to thank Eric Blais and Nathaniel Harms for helpful discussions during the early stages of this work and for their thoughtful feedback. We would also like to thank the anonymous reviewers whose comments helped significantly to improve this write up.

1 Introduction

A function $f\colon \mathcal{X} \to \mathbb{R}$ over a partial order $\mathcal{P} = (\mathcal{X}, \preceq)$ is k-*monotone* if there does not exist a chain of $k+1$ points $x_1 \prec x_2 \prec \cdots \prec x_{k+1}$ for which (a) $f(x_{i+1}) - f(x_i) < 0$ when i is odd and (b) $f(x_{i+1}) - f(x_i) > 0$ when i is even. When $k = 1$, these are the *monotone* functions, which are the non-decreasing functions with respect to $\preceq$. Monotone and k-monotone *Boolean* functions over domains $\{0,1\}^d$, $[n]^d$, and $\mathbb{R}^d$ have been the focus of a significant amount of research in property testing and computational learning theory. We give an overview of the literature in Section 1.4.

The field of property testing is concerned with the design and analysis of sub-linear time randomized algorithms for determining if a function has, or is far from having, some specific property. A key aspect in the definition of a property testing algorithm is the type of access it has to the function. Early works on property testing, e.g. [61, 44], focused on the notion of *query-based* testers, which are allowed to observe the value of the function on any point of their choosing, and since then this has become the standard model. The weaker notion of *sample-based* testers, which can only view the function on independent uniform samples, was also considered by [44] and has received some attention over the years, see e.g. [51, 3, 41, 45, 37, 38]. Sample-based algorithms are considered more natural in many settings, for example in computational learning theory, where they are the standard model. In fact, sample-based testing and learning are closely related problems; given a learning algorithm, it is always possible to design a testing algorithm with the same sample complexity, up to an additive $\mathrm{poly}(1/\varepsilon)$ factor[1].

For many fundamental properties, there is still a large gap between how much we know in the query-based vs the sample-based models. Monotonicity (and k-monotonicity) is such a property; despite a vast body of research on query-based monotonicity testing over the hypercube $\{0,1\}^d$, the only work we know of which considers this problem in the sample-based model is [43], who gave an upper bound of $O(\sqrt{2^d/\varepsilon})$ and a matching lower bound for the case when $\varepsilon = O(d^{-3/2})$ on the number of samples needed to test monotonicity of functions $f\colon \{0,1\}^d \to \{0,1\}$. The upper bound for learning monotone Boolean functions due to [23, 54] also implies a testing upper bound of $\exp(\widetilde{O}(\frac{1}{\varepsilon}\sqrt{d}))$. Thus, this question has been wide open for $\varepsilon \gg d^{-3/2}$.

Our work addresses this gap in the monotonicity testing literature, proving a lower bound which matches the learning upper bound for all ε at most some constant, up to a factor of $\log d$ in the exponent. More generally, we prove tight lower bounds for k-monotonicity testing of functions, $f\colon \{0,1\}^d \to [r]$, i.e. functions with image size at most r. To round out our results, we also give an improved learning algorithm for k-monotone functions over $\mathbb{R}^d$ under product distributions whose sample complexity matches our sample-based testing lower bound, up to poly-logarithmic factors in the exponent.

1.1 Results

Before explaining our results and the context for them, we first provide some terminology and basic notation. Given a domain $\mathcal{X}$ and a distribution μ over $\mathcal{X}$, we denote the Hamming distance between two functions $f, g\colon \mathcal{X} \to \mathbb{R}$ under μ by $d_\mu(f,g) = \mathbb{P}_{x\sim\mu}[f(x) \neq g(x)]$. We say that f is ε-far from k-monotone if $d_\mu(f,g) \geq \varepsilon$ for every k-monotone function g. The results in this paper pertain to sample-based testing and learning of k-monotone functions with respect to Hamming distance. We use the following terminology:

- The *example oracle* for f under μ, denoted by $EX(f,\mu)$, when queried, generates an example $(x, f(x))$ where x is sampled according to μ.
- A *sample-based k-monotonicity tester* under μ is a randomized algorithm which is given access to $EX(f,\mu)$ for an arbitrary input function f and satisfies the following: (a) if f is k-monotone, then the algorithm accepts with probability at least 2/3, and (b) if f is ε-far from k-monotone, then the algorithm rejects with probability at least 2/3. The tester has *one-sided error* if in case (a) it accepts with probability 1.

[1] See Lemma C.1 in the full version of the paper for a precise statement. Also, note that if the learning algorithm is *proper*, then the time complexity is also preserved. If the learning algorithm is *improper*, then there is a time complexity blow-up, but the sample complexity is still preserved.

- A *sample-based learning algorithm for k-monotone functions* under μ is a randomized algorithm which is given access to $EX(f, \mu)$ for an arbitrary k-monotone input function f and outputs a hypothesis h such that $d_\mu(f, h) \le \varepsilon$ with probability at least $1 - \delta$. If left unspecified, $\delta = 1/3$.

In all of the above definitions if μ is unspecified, then it is the uniform distribution. Testing and learning are closely related problems; any sample-based learning algorithm can be used to construct a sample-based tester with the same sample complexity. We refer to this transformation as the testing-by-learning reduction and although this is not a new idea we provide a proof in Section C in the full version of the paper for completeness.

Finally, we recall some important learning theory terminology. A learning algorithm for concept class $\mathcal{C}$ is called *proper* if it always outputs a hypothesis $h \in \mathcal{C}$, and is called *improper* if it is allowed to output arbitrary h. Given a function f, and a concept class $\mathcal{C}$, let $d(f, \mathcal{C}) = \min_{g \in \mathcal{C}} d(f, g)$. An *agnostic proper* learner is one which, given *any* f (not necessarily in $\mathcal{C}$), outputs a hypothesis $h \in \mathcal{C}$ for which $d(f, h) \le d(f, \mathcal{C}) + \varepsilon$ with probability at least $1 - \delta$.

1.1.1 Sample-Based Testing and Learning on the Hypercube

The problem of *learning* monotone Boolean functions over the hypercube $\{0,1\}^d$ was studied by [23] who proved an upper bound[2] of $\exp(O(\min\{\frac{1}{\varepsilon}\sqrt{d}\log d, d\}))$ for improper learning and very recently by [54, 55] who obtained the same upper bound for agnostic proper learning. The improper learning upper bound was extended by [17] who showed an upper bound of $\exp(O(\min\{\frac{k}{\varepsilon}\sqrt{d}\log d, d\}))$ and a nearly matching lower bound of $\exp(\Omega(\min\{\frac{k}{\varepsilon}\sqrt{d}, d\}))$ for learning k-monotone Boolean functions for any $k \ge 1$. The testing-by-learning reduction shows that their upper bound also holds for sample-based *testing*. The only prior lower bound for sample-based testing that we're aware of is $\Omega(\sqrt{2^d/\varepsilon})$ when $\varepsilon = O(d^{-3/2})$ and $k = 1$ [43, Theorem 5]. Our main result is the following much more general lower bound for this problem, which we prove in Section 3.

▶ **Theorem 1** (Testing Lower Bound). *There is an absolute constant $c > 0$ such that for all $\varepsilon \le c$, every sample-based k-monotonicity tester for functions $f\colon \{0,1\}^d \to [r]$ under the uniform distribution has sample complexity*

$$\exp\left(\Omega\left(\min\left\{\frac{rk}{\varepsilon}\sqrt{d}, d\right\}\right)\right).$$

Even for the special case of sample-based monotonicity testing of Boolean functions ($k = 1$ and $r = 2$), Theorem 1 is already a new result, which matches the upper bound for learning by [23] and is the first lower bound to hold for $\varepsilon \gg d^{-3/2}$. Moreover, our lower bound is much more general, holding for all r, k, and is optimal in all parameters, d, r, k, ε, up to a $\log d$ factor in the exponent. We show a nearly matching upper bound in Theorem 3.

We also note that the testing-by-learning reduction implies that the same lower bound holds for *learning* with samples. As we mentioned, this result was already known for Boolean

[2] We remark that any function over $\{0,1\}^d$ can be learned exactly with $O(d2^d) = \exp(O(d))$ samples by a coupon-collector argument. Combining this with the $\exp(O(\frac{1}{\varepsilon}\sqrt{d}\log d))$ upper bound by [23] yields $\exp(O(\min\{\frac{1}{\varepsilon}\sqrt{d}\log d, d\}))$. We use this slightly clunkier notation involving the min to emphasize that our upper and lower bounds are nearly matching in all parameter regimes.

functions (the $r = 2$ case) [17], but the general case of $r \geq 2$ was not known prior to our work[3].

▶ **Corollary 2** (Learning Lower Bound). *There is an absolute constant $c > 0$ such that for every $\varepsilon \leq c$, every sample-based uniform-distribution learning algorithm for k-monotone functions $f\colon \{0,1\}^d \to [r]$ has sample complexity*

$$\exp\left(\Omega\left(\min\left\{\frac{rk}{\varepsilon}\sqrt{d}, d\right\}\right)\right).$$

On the upper bound side, a relatively straightforward argument extends the learning algorithm of [17] for Boolean k-monotone functions, to k-monotone functions with image size at most r. We give a short proof in Section 1.5. This shows that our lower bounds in Theorem 1 and Corollary 2 are tight up to a factor of $\log d$ in the exponent.

▶ **Theorem 3** (Learning Upper Bound for Hypercubes). *There is a uniform-distribution learning algorithm for k-monotone functions $f\colon \{0,1\}^d \to [r]$ which achieves error at most ε with time and sample complexity*

$$\exp\left(O\left(\min\left\{\frac{rk}{\varepsilon}\sqrt{d}\log d, d\right\}\right)\right).$$

The testing-by-learning reduction again gives us the following corollary.

▶ **Corollary 4** (Testing Upper Bound for Hypercubes). *There is a sample-based k-monotonicity tester for functions $f\colon \{0,1\}^d \to [r]$ with sample complexity*

$$\exp\left(O\left(\min\left\{\frac{rk}{\varepsilon}\sqrt{d}\log d, d\right\}\right)\right).$$

Lastly, we consider the problem of sample-based testing with *one-sided error*. For monotonicity testing of functions $f\colon \{0,1\}^d \to \{0,1\}$ with *non-adaptive queries*, we know that one-sided and two-sided error testers achieve the same query-complexity (up to polylog$(d, 1/\varepsilon)$ factors): there is a $\widetilde{O}(\sqrt{d}/\varepsilon^2)$ one-sided error upper bound due to [53] and a $\widetilde{\Omega}(\sqrt{d})$ two-sided error lower bound due to [33]. We show that the situation is quite different for *sample-based* monotonicity testing; while the sample complexity of two-sided error testers is $\exp(\widetilde{\Theta}(\min\{\frac{1}{\varepsilon}\sqrt{d}, d\}))$, one-sided error testers require $\exp(\Theta(d))$ samples for all ε.

▶ **Theorem 5** (Testing with One-Sided Error). *For every $d, r, k,$ and $\varepsilon > 0$, sample-based k-monotonicity testing of functions $f\colon \{0,1\}^d \to [r]$ with one-sided error requires* $\exp(\Theta(d))$ *samples.*

1.1.2 Sample-Based Testing and Learning in Continuous Product Spaces

Learning k-monotone Boolean-valued functions has also been studied over $\mathbb{R}^d$ with respect to product measures by [49] who gave an upper bound of $\exp(\widetilde{O}(\min\{\frac{k}{\varepsilon^2}\sqrt{d}, d\}))$ where $\widetilde{O}(\cdot)$ hides polylog factors of $d, k,$ and $1/\varepsilon$. Our next result gives an upper bound which improves the dependence on ε from $1/\varepsilon^2$ to $1/\varepsilon$ in the exponent. By the same approach we used to generalize the upper bound in Theorem 3 to arbitrary $r \geq 2$, we get the same generalization for product spaces. We obtain the following upper bound which matches our lower bound for $\{0,1\}^d$ in Theorem 1 up to polylog factors of $d, k, r,$ and $1/\varepsilon$. We say that a function $f\colon \mathbb{R}^d \to [r]$ is *measurable* if the set $f^{-1}(i)$ is measurable for every $i \in [r]$.

[3] It is possible that the techniques from [17] could be extended to provide an alternative proof of Corollary 2, but we have not checked whether this is the case.

▶ **Theorem 6** (Learning Upper Bound for Product Spaces)**.** *Given an arbitrary product measure μ, there is a learning algorithm under μ for measurable k-monotone functions $f \colon \mathbb{R}^d \to [r]$ with time and sample complexity*

$$\exp\left(\widetilde{O}\left(\min\left\{\frac{rk}{\varepsilon}\sqrt{d}, d\right\}\right)\right).$$

The $\widetilde{O}(\cdot)$ hides polylogarithmic dependencies on $d, r, k,$ and $1/\varepsilon$.

We prove Theorem 6 in Section 4. Once again the testing-by-learning reduction gives us the following corollary for sample-based testing.

▶ **Corollary 7** (Testing Upper Bound for Product Spaces)**.** *Given an arbitrary product measure μ, there is a k-monotonicity tester for measurable functions $f \colon \mathbb{R}^d \to [r]$ under μ with sample complexity*

$$\exp\left(\widetilde{O}\left(\min\left\{\frac{rk}{\varepsilon}\sqrt{d}, d\right\}\right)\right).$$

The $\widetilde{O}(\cdot)$ hides polylogarithmic dependencies on $d, r, k,$ and $1/\varepsilon$.

1.2 Proof Overviews

In this section we give an overview of our proofs for Theorem 1 and Theorem 6.

1.2.1 The Testing Lower Bound for Hypercubes

Our proof of Theorem 1 uses a family functions known as *Talagrand's random DNFs* introduced by [63] which have been used by [4] and [33] to prove lower bounds for monotonicity testing of Boolean functions $f \colon \{0,1\}^d \to \{0,1\}$ against adaptive and non-adaptive query-based testers. Very recently, they have also been used to prove lower bounds for tolerant monotonicity testing [29] and for testing convexity of sets in $\{-1,0,1\}^d$ [10].

To understand our construction, let us first consider the special case of monotonicity of Boolean functions, i.e. $k = 1$ and $r = 2$. We think of a DNF term as a point $t \in \{0,1\}^d$ which is said to be satisfied by $x \in \{0,1\}^d$ if $t \preceq x$, where $\preceq$ denotes the standard bit-wise partial order over $\{0,1\}^d$. The *width* of a term t is its Hamming weight, $|t|$, and the width of a DNF is the max width among its terms. Consider N randomly chosen terms $t^1, \ldots, t^N$ each of width $|t^j| = w$. We will see later how to choose N and w. Let $B := \{x \colon \frac{d}{2} \leq |x| \leq \frac{d}{2} + \varepsilon\sqrt{d}\}$ and for each $j \in [N]$, let

$$U_j := \{x \in B \colon t^j \preceq x \text{ and } t^{j'} \not\preceq x \text{ for all } j' \neq j\}$$

be the set of points in B which satisfy t^j and no other terms. Let $U := \bigcup_{j \in [N]} U_j$. Now observe that any two points lying in different U_j's are *incomparable* and therefore independently embedding an arbitrary monotone function into each U_j will result in a function which globally is monotone if one defines the function outside of U appropriately. Using this fact we can define two distributions $\mathcal{D}_{\texttt{yes}}$ and $\mathcal{D}_{\texttt{no}}$ as follows. Let A denote the set of points in $x \in \{0,1\}^d$ for which either $|x| > \frac{d}{2} + \varepsilon\sqrt{d}$ or $x \in B$ and $t^j, t^{j'} \preceq x$ for two different terms $j \neq j'$.

- $f \sim \mathcal{D}_{\texttt{yes}}$ is drawn by setting $f(x) = 1$ if and only if $x \in A \cup \left(\bigcup_{j \in T} U_j\right)$ where $T \subseteq [N]$ contains each $j \in [N]$ with probability $1/2$, independently. Such a function is always monotone.

- $f \sim \mathcal{D}_{\texttt{no}}$ is drawn by setting $f(x) = 1$ if and only if $x \in A \cup R$ where R contains each $x \in U$ with probability 1/2, independently. Such a function will be $\Omega(|U| \cdot 2^{-d})$-far from monotone with probability $\Omega(1)$ since its restriction with U is uniformly random.

Now, each $x \in U$ satisfies $\mathbb{P}_{f\sim\mathcal{D}_{\texttt{yes}}}[f(x) = 1] = \mathbb{P}_{f\sim\mathcal{D}_{\texttt{no}}}[f(x) = 1] = 1/2$ and for both distributions the events $f(x) = 1$ and $f(y) = 1$ are independent when x, y lie in different U_j's. Therefore, any tester will need to see at least two points from the same U_j to distinguish $\mathcal{D}_{\texttt{yes}}$ and $\mathcal{D}_{\texttt{no}}$. Roughly speaking, by birthday paradox this gives a $\Omega(\sqrt{N})$ lower bound on the number of samples. The lower bound is thus determined by the maximum number of terms N that can be used in the construction for which $|U| = \Omega(\varepsilon 2^d)$.

So how are N and w chosen? By standard concentration bounds, we have $|B| = \Omega(\varepsilon 2^d)$ and observe that a point $x \in B$ satisfies a random term with probability exactly $(|x|/d)^w$. We need U to contain a *constant fraction* of B, i.e. we need x to satisfy exactly 1 term with constant probability. The expected number of satisfied terms is $N \cdot (|x|/d)^w$ and, roughly speaking, we need this value to be $\Theta(1)$ for all $x \in B$. Applying this constraint to the case when $|x| = d/2$ forces us to pick $N \approx 2^w$. Now when $|x| = d/2 + \varepsilon\sqrt{d}$, the expected number of satisfied terms is $N \cdot 2^{-w} \cdot (1 + 2\varepsilon/\sqrt{d})^w \approx (1 + 2\varepsilon/\sqrt{d})^w$ and we are forced to choose $w \approx \sqrt{d}/\varepsilon$. The lower bound for sample-based monotonicity testing of $f\colon \{0,1\}^d \to \{0,1\}$ is then $\Omega(\sqrt{N}) \approx \exp(\Omega(\sqrt{d}/\varepsilon))$.

Let us now think about generalizing this construction to testing k-monotonicity of functions $f\colon \{0,1\}^d \to [r]$. The moral of the above argument is that the permitted number of terms is controlled by the number of distinct Hamming weights in the set B. We observe that for larger values of k and r we can partition B into $k(r-1)$ blocks as $B := B_1 \cup B_2 \cup \cdots \cup B_{k(r-1)}$ each with a window of Hamming weights of size only $\frac{\varepsilon\sqrt{d}}{k(r-1)}$. We are able to essentially repeat the above construction independently within each block wherein we can set $w \approx \frac{k(r-1)\sqrt{d}}{\varepsilon}$ and consequently $N \approx 2^{\frac{k(r-1)\sqrt{d}}{\varepsilon}}$.

For each block $i \in [k(r-1)]$, the random Talagrand DNF within block B_i is defined analogously to the above construction, except that it assigns function values from $\{i \bmod (r-1), i \bmod (r-1) + 1\}$, instead of $\{0,1\}$. See Figure 1 for an illustration. Since there are $k(r-1)$ blocks in total, the distribution $\mathcal{D}_{\texttt{yes}}$ only produces k-monotone functions. At the same time, a function $f \sim \mathcal{D}_{\texttt{no}}$ assigns uniform random $\{a, a+1\}$ values within each block $B_{m(r-1)+a}$. This results in a large number of long chains through $B_a \cup B_{(r-1)+a} \cup \cdots \cup B_{(k-1)(r-1)+a}$ which alternate between function value a and $a + 1$. Considering the union of all such chains for $a = 0, 1, \ldots, r-2$ shows that f is $\Omega(\varepsilon)$-far from k-monotone with probability $\Omega(1)$.

1.2.2 The Learning Upper Bound for Product Spaces

As we discussed in Section 1.1, it suffices to prove Theorem 6 for the case of $r = 2$, i.e. learning functions $f\colon \mathbb{R}^d \to \{\pm 1\}$ under a product measure μ. We use a downsampling technique to reduce this problem to learning a discretized proxy of f over a hypergrid $[N]^d$ where $N = \Theta(kd/\varepsilon)$ with mild label noise. This technique has been used in previous works [46, 12, 49] and our proof borrows many technical details from [49].

Next, for N which is a power of 2, we observe that a k-monotone function $f\colon [N]^d \to \{\pm 1\}$ can be viewed as a k-monotone function over the hypercube $\{\pm 1\}^{d \log N}$ by mapping each point $x \in [N]^d$ to its bit-representation. We can then leverage a result of [17] which shows that all but a ε-fraction of the mass of the Fourier coefficients of k-monotone Boolean functions $f\colon \{0,1\}^d \to \{0,1\}$ is concentrated on the terms with degree at most $\frac{k\sqrt{d}}{\varepsilon}$. We can then use the Low-Degree Algorithm introduced by [57] which was shown to work under random classification noise by [50].

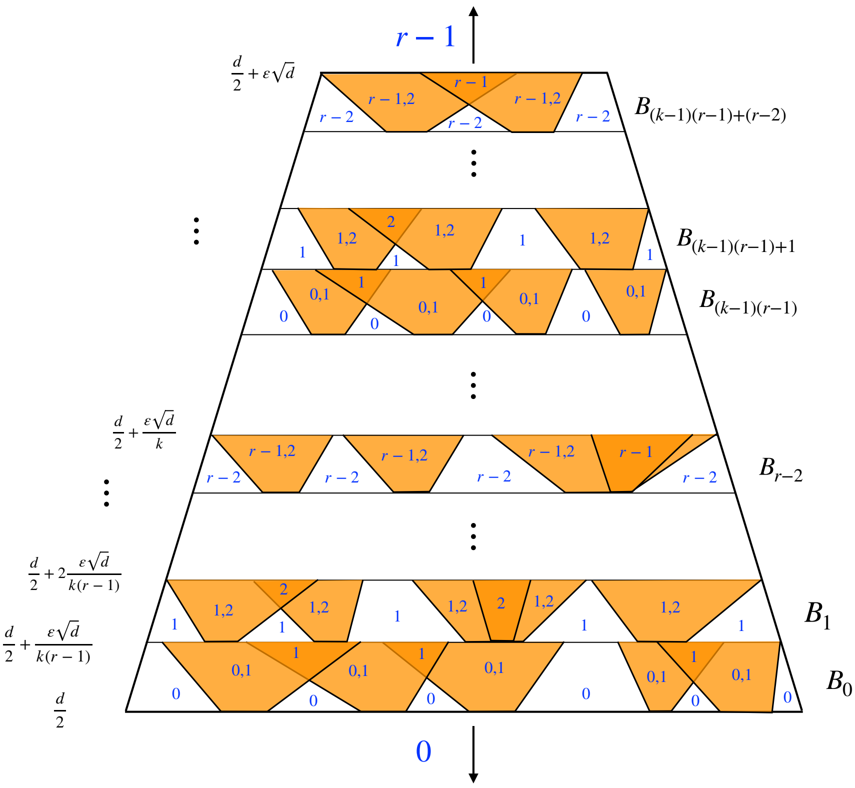

Figure 1 An illustration of the construction used in our proof of Theorem 1. The image represents the set of points in the hypercube $\{0,1\}^d$ with Hamming weight in the interval $[\frac{d}{2}, \frac{d}{2}+\varepsilon\sqrt{d})$, increasing from bottom to top. The numbers on the left denote the Hamming weight of the points lying in the adjacent horizontal line. The B_i blocks are the sets of points contained between two adjacent horizontal lines. Each orange shaded region within B_i represents the set of points satisfied by a term $t^{i,j}$. The blue numbers represent the value that functions in the support of $\mathcal{D}_{\mathsf{yes}}$ and $\mathcal{D}_{\mathsf{no}}$ can take. We have used the notation "$r-1,2$" as shorthand for $r-2, r-1$.

1.3 Discussion and Open Questions

Our results for sample-based testing and learning over the hypercube are tight up to a $\log d$ factor in the exponent. Our upper bound for product spaces matches the lower bound for hypercubes only up to *polylog* factors of $d, k, r, 1/\varepsilon$ in the exponent. In particular, the upper bound for product spaces goes to ∞ as any one of the parameters r, k, or $1/\varepsilon$ grow to ∞, whereas the lower bound for the hypercube can be at most $\exp(\Theta(d))$ simply because $|\{0,1\}^d| = 2^d$ and so any function $f\colon \{0,1\}^d \to \mathbb{R}$ can be learned *exactly* with $\exp(O(d))$ samples. It seems intuitive that sample-based testing and learning of k-monotone functions over $[n]^d$ should require $n^{\Omega(d)}$ samples as either of the parameters k or r approaches ∞. A corollary of such a result would be that the sample-complexity of these problems for $f\colon \mathbb{R}^d \to [r]$ grow to ∞ as k or r approach ∞. Moreover, if this is true, then k-monotonicity of functions $f\colon \mathbb{R}^d \to \mathbb{R}$ is not testable with a finite number of samples. Our results do not address this and it would be interesting to investigate this further.

▶ **Question 8.** *Is there a lower bound for sample-based k-monotonicity testing of functions $f\colon [n]^d \to [r]$ which approaches $n^{\Omega(d)}$ as r or k go to ∞?*

1.4 Related Work

Monotone functions and their generalization to k-monotone functions have been extensively studied within property testing and learning theory over the last 25 years. We highlight some of the results which are most relevant to our work. Afterwards, we discuss some selected works on sample-based property testing.

1.4.1 Sample-Based Monotonicity Testing

Sample-based monotonicity testing of Boolean functions over the hypercube, $\{0,1\}^d$, was considered by [43] (see [43, Theorems 5 and 6]) who gave an upper bound of $O(\sqrt{2^d/\varepsilon})$ and a lower bound of $\Omega(\sqrt{2^d/\varepsilon})$ for $\varepsilon = O(d^{-3/2})$. Sample-based monotonicity testing over general partial orders was studied by [42] who gave a $O(\sqrt{N/\varepsilon})$ one-sided error tester for functions $f\colon D \to \mathbb{R}$ where D is any partial order on N elements. Sample-based monotonicity testing of functions on the line $f\colon [n] \to [r]$ was studied by [58] who gave a one-sided error upper bound of $O(\sqrt{r/\varepsilon})$ and a matching lower bound of $\Omega(\sqrt{r})$ for all sample-based testers.

1.4.2 Query-Based Monotonicity Testing

Monotonicity testing has been extensively studied in the standard query model [59, 35, 43, 34, 56, 42, 47, 1, 48, 2, 39, 62, 8, 22, 36, 16, 60, 9, 26, 27, 32, 7, 19, 30, 25, 52, 4, 33, 11, 58, 12, 49, 15, 21, 14, 13, 29]. When discussing these works we treat ε as a small constant for brevity. For $f\colon \{0,1\}^d \to \{0,1\}$, the non-adaptive query complexity has been established at $\widetilde{\Theta}(\sqrt{d})$ [53, 33] with an adaptive lower bound of $\widetilde{\Omega}(d^{1/3})$ [33]. This gap for adaptive monotonicity testing of Boolean functions is still an outstanding open question. For $f\colon [n]^d \to \{0,1\}$ and $f\colon \mathbb{R}^d \to \{0,1\}$ under product measures, a recent result of [13] established a non-adaptive upper bound of $d^{1/2+o(1)}$. For functions $f\colon \{0,1\}^d \to [r]$, [15] showed upper and lower bounds of $\widetilde{\Theta}(\min(r\sqrt{d}, d))$ for non-adaptive, one-sided error testers and there is a general (adaptive) lower bound of $\Omega(\min(d, r^2))$ due to [16]. For real-valued functions $f\colon [n]^d \to \mathbb{R}$, the query complexity is known to be $\Theta(d \log n)$. The upper bound is non-adaptive [26] and the lower bound holds even for adaptive testers [28].

1.4.3 k-Monotonicity Testing

The generalization to k-monotonicity testing has also been studied in the standard query model by [46, 24]. These works show that the query-complexity of non-adaptive one-sided error k-monotonicity testing is $\exp(\widetilde{\Theta}(\sqrt{d}))$ for all $k \geq 2$, demonstrating an interesting separation between (1-)monotonicity and 2-monotonicity.

1.4.4 Learning Monotone Functions

Monotone Boolean functions $f\colon \{0,1\}^d \to \{0,1\}$ were studied in the context of learning theory by [23] who showed that they can be (improperly) learned to error ε under the uniform distribution with $\exp(\widetilde{O}(\frac{1}{\varepsilon}\sqrt{d}))$ time and samples. Very recent works [54, 55] have given *agnostic proper* learning algorithms with the same complexity.

1.4.5 Learning k-Monotone Functions

The result of [23] was generalized by [17] who gave upper and lower bounds of $\exp(\widetilde{\Theta}(\frac{k}{\varepsilon}\sqrt{d}))$ for learning k-monotone Boolean functions $f\colon \{0,1\}^d \to \{0,1\}$. For Boolean functions over hypergrids $f\colon [n]^d \to \{0,1\}$, [24] gave an upper bound of $\exp(\widetilde{O}(\min(\frac{k}{\varepsilon^2}\sqrt{d}, d)))$ where $\widetilde{O}(\cdot)$ hides polylog factors of $d, k, 1/\varepsilon$. This result was generalized to functions $f\colon \mathbb{R}^d \to \{0,1\}$ under product measures by [49].

1.4.6 Sample-Based Property Testing

The notion of sample-based property testing was first presented and briefly studied by [44]. Broader studies of sample-based testing and its relationship with query-based testing have since been given by [40, 41, 45]. A characterization of properties which are testable with a constant number of samples was given by [20].

As we mentioned, sample-based algorithms are the standard model in learning theory, and learning requires at least as many samples as testing for every class of functions. Thus, it is natural to ask, when is testing *easier* than learning in terms of sample complexity? This question is referred to as *testing vs learning* and has been studied by [51] and more recently by [18, 37, 38].

There has also been work studying models that interpolate between query-based and sample-based testers. For instance, [3] introduced the notion of *active testing*, where the tester may make queries, but only on points from a polynomial-sized batch of unlabeled samples drawn from the underlying distribution. This was inspired by the notion of *active learning* which considers learning problems under this access model.

Sample-based convexity testing of sets over various domains has also seen some recent attention [31, 5, 6, 10].

1.5 Learning Functions with Bounded Image Size: Proof of Theorem 3

In this section we give a short proof showing that the learning algorithm of [17] can be extended in a relatively straightforward manner to functions $f\colon \{0,1\}^d \to [r]$ by increasing the sample-complexity by a factor of r in the exponent.

Proof of Theorem 3. [17, Theorem 1.4] proved this result for the case of $r = 2$. In particular, they show that there is a sample-based learning algorithm which given an arbitrary k-monotone Boolean function f, outputs h such that $\mathbb{P}_h[d(f,h) > \varepsilon] < \delta$ using $\ln(1/\delta) \cdot \exp\left(O\left(\min\left\{\frac{rk}{\varepsilon}\sqrt{d}\log d, d\right\}\right)\right)$ queries[4] to the example oracle, $EX(f)$. We will make use of this result.

For each $t \in [r]$, let $f_t\colon \{0,1\}^d \to \{0,1\}$ denote the thresholded Boolean function defined as $f_t(x) := \mathbf{1}(f(x) \geq t)$. Observe that for all $x \in \{0,1\}^d$ we have $f(x) = \text{argmax}_t\{f_t(x) = 1\}$. Thus, for each $t \in [r]$, run the learning algorithm of [17] with error parameters set to $\varepsilon' := \varepsilon/r$ and $\delta = 1/3r$ to obtain a hypothesis h_t. We have $\mathbb{P}[d(h_t, f_t) > \varepsilon/r] < 1/3r$. By a union bound, with probability at least 2/3, every $t \in [r]$ satisfies $d(h_t, f_t) \leq \varepsilon/r$. Moreover, if this holds then by another union bound we have $\mathbb{P}_x[\exists t \in [r] \colon h_t(x) \neq f(x)] \leq \varepsilon$. Thus, the hypothesis $h(x) := \text{argmax}_t\{h_t(x) = 1\}$ satisfies $d(h,f) \leq \varepsilon$. The number of samples used is $\ln(1/\delta) \cdot \exp(O(\min\{\frac{k}{\varepsilon'}\sqrt{d}\log d, d\})) = \exp(O(\min\{\frac{rk}{\varepsilon}\sqrt{d}\log d, d\}))$ and this completes the proof. ◀

2 Preliminaries on k-Monotonicity

We use the notation $[n] := \{0, 1, \ldots, n-1\}$.

▶ **Definition 9.** *Given a poset $\mathcal{P} = (\mathcal{X}, \preceq)$ and a function $f\colon \mathcal{X} \to \mathbb{R}$, an m-alternating chain is a sequence of points $x_1 \prec x_2 \prec \cdots \prec x_m$ such that for all $i \in \{1, \ldots, m-1\}$,*

1. *$f(x_{i+1}) - f(x_i) < 0$ when i is odd, and*
2. *$f(x_{i+1}) - f(x_i) > 0$ when i is even.*

[4] Their result (Thm 1.4 of [17]) is stated for constant δ, but can be easily extended to arbitrary δ with the stated query complexity by replacing Thm 3.1 in their proof with the Low-Degree Algorithm stated for general δ.

▶ **Definition 10** (k-monotonicity)**.** *For a poset $\mathcal{P} = (\mathcal{X}, \preceq)$, a function $f\colon \mathcal{X} \to \mathbb{R}$ is called k-monotone if it does not have any $(k+1)$-alternating chains.*

Let $\mathcal{M}_{\mathcal{P},k}$ denote the set of all k-monotone functions $f\colon \mathcal{X} \to \mathbb{R}$ over the poset $\mathcal{P} = (\mathcal{X}, \preceq)$. The Hamming distance between two functions $f, g\colon \mathcal{X} \to \mathbb{R}$ is $d(f,g) = |\mathcal{X}|^{-1} \cdot |\{x \in \mathcal{X}\colon f(x) \neq g(x)\}|$. The distance to k-monotonicity of f is denoted by $\varepsilon(f, \mathcal{M}_{\mathcal{P},k}) := \min_{g \in \mathcal{M}_{\mathcal{P},k}} d(f,g)$. The following claim is our main tool for lower bounding the distance to k-monotonicity.

▷ Claim 11. Let $f\colon \mathcal{X} \to \mathbb{R}$ and $k' \geq 3k$ be an integer. Let $\mathcal{C} \subset \mathcal{X}^{k'}$ be a collection of disjoint k'-alternating chains for f. Then

$$\varepsilon(f, \mathcal{M}_{\mathcal{P},k}) \geq \frac{1}{3|\mathcal{X}|} \cdot \left| \bigcup_{C \in \mathcal{C}} C \right|.$$

Proof. Observe that every k-monotone function $g \in \mathcal{M}_{\mathcal{P},k}$ has the following property: for every $C = (x_1, x_2, \ldots, x_{k'}) \in \mathcal{C}$, the sequence

$$\big(1, g(x_2) - g(x_1), g(x_3) - g(x_2), \ldots, g(x_{k'}) - g(x_{k'-1})\big)$$

changes sign at most $k-1$ times, whereas the sequence

$$\big(1, f(x_2) - f(x_1), f(x_3) - f(x_2), \ldots, f(x_{k'}) - f(x_{k'-1})\big)$$

changes sign exactly $k'-1$ times. We have prepended a 1 so that the first sign change occurs as soon as the function value decreases. Now, changing $f(x_i)$ can only reduce the number of times the sequence changes sign by at most 2 and so $|\{i\colon f(x_i) \neq g(x_i)\}| \geq \frac{k'-k}{2}$. Summing over all chains in $\mathcal{C}$ and normalizing yields

$$d(f,g) \geq \frac{k'-k}{2} \cdot \frac{|\mathcal{C}|}{|\mathcal{X}|} \geq \frac{k'}{3} \cdot \frac{|\mathcal{C}|}{|\mathcal{X}|} \geq \frac{1}{3|\mathcal{X}|} \cdot \left| \bigcup_{C \in \mathcal{C}} C \right|$$

where the second inequality follows from $k \leq k'/3$ and the third inequality is due to the fact that the chains in $\mathcal{C}$ are all disjoint and each of size k'. This completes the proof since this inequality holds for all $g \in \mathcal{M}_{\mathcal{P},k}$. ◁

We use the notation $\mathcal{M}_{r,k}$ to denote the set of all k-monotone functions $f\colon \{0,1\}^d \to [r]$ over the hypercube whose image has at most r distinct values.

3 Lower Bound for Sample-Based Testers

In this section we prove Theorem 1, our lower bound on the sample-complexity of testing k-monotonicity of functions $f\colon \{0,1\}^d \to [r]$. We refer the reader to Section 1.2.1 for a discussion of our main ideas and a proof sketch for the special case of $k = 1$ and $r = 2$, i.e. *monotone Boolean* functions. Our proof follows the standard approach of defining a pair of distributions $\mathcal{D}_{\texttt{yes}}, \mathcal{D}_{\texttt{no}}$ over functions $f\colon \{0,1\}^d \to [r]$ which satisfy the following:

- $\mathcal{D}_{\texttt{yes}}$ is supported over k-monotone functions.
- Functions drawn from $\mathcal{D}_{\texttt{no}}$ are typically $\Omega(\varepsilon)$-far from k-monotone: $\mathbb{P}_{f \sim \mathcal{D}_{\texttt{no}}}[\varepsilon(f, \mathcal{M}_{r,k}) = \Omega(\varepsilon)] = \Omega(1)$.
- The distributions over labeled examples from $\mathcal{D}_{\texttt{yes}}$ and $\mathcal{D}_{\texttt{no}}$ are close in TV-distance.

Our construction uses a generalized version of a family functions known as random Talagrand DNFs, which were used by [4] and [33] to prove lower bounds for testing monotonicity of Boolean functions with adaptive and non-adaptive queries.

Let r, k satisfy $rk \le \frac{\varepsilon\sqrt{d}}{24300}$. For convenience, we will assume that $\frac{k(r-1)}{\varepsilon}$ and $\sqrt{d}$ are integers and that $\frac{k(r-1)}{\varepsilon}$ divides $\sqrt{d}$. Let $L_\ell := \left\{x \in \{0,1\}^d : |x| = \ell\right\}$ denote the ℓ'th Hamming level of the hypercube. We partition $\bigcup_{\ell\in[0,\varepsilon\sqrt{d})} L_{d/2+\ell}$ into $k(r-1)$ blocks as follows. For each $i \in [k(r-1)]$, define

$$B_i = \bigcup_{\ell = i\cdot\frac{\varepsilon\sqrt{d}}{k(r-1)}}^{(i+1)\cdot\frac{\varepsilon\sqrt{d}}{k(r-1)}-1} L_{\frac{d}{2}+\ell}.$$

The idea of our proof is to define a random DNF within each B_i. The *width* of each DNF will be set to $w := \frac{(r-1)k\sqrt{d}}{2\varepsilon}$ and for each i, the number of terms in the DNF within B_i will be set to $N_i := 2^w \cdot e^{-i} = 2^{\frac{(r-1)k\sqrt{d}}{2\varepsilon}(1-o(1))}$. The DNF defined over B_i will assign function values from $\{i \bmod (r-1), i \bmod (r-1)+1\}$. The terms in each DNF will be chosen randomly from the following distribution. We think of terms as points $t \in \{0,1\}^d$ in the hypercube where another point x *satisfies* t if $t \preceq x$, i.e. $t_i = 1$ implies $x_i = 1$.

▶ **Definition 12** (Term distribution). *A term $t \in \{0,1\}^d$ is sampled from the distribution $\mathcal{D}_{\texttt{term}}$ as follows. Form a (multi)-set $S \subseteq [d]$ by choosing w independent uniform samples from $[d]$. For each $a \in [d]$, let $t_a := \mathbf{1}(a \in S)$.*

3.1 The Distributions $\mathcal{D}_{\texttt{yes}}$ and $\mathcal{D}_{\texttt{no}}$

We now define the yes and no distributions over functions $f : \{0,1\}^d \to [r]$. For each $i \in [k(r-1)]$, choose terms $t^{i,1}, \dots, t^{i,N_i}$ i.i.d. from $\mathcal{D}_{\texttt{term}}$ and let $\boldsymbol{t} = \{t^{i,j} : i \in [k(r-1)],\ j \in [N_i]\}$ denote the random set of all terms. Now, for each $i \in [k(r-1)]$ and $j \in [N_i]$, define the set

$$U_{i,j} = \left\{x \in B_i : x \succeq t^{i,j} \text{ and } x \not\succeq t^{i,j'} \text{ for all } j' \neq j\right\} \tag{1}$$

of all points in the i'th block that satisfy the j'th term *uniquely*. Let $U_i = \bigcup_{j\in[N_i]} U_{i,j}$ denote the set of points in B_i that satisfy a unique term. The following claim is key to our result and motivates our choice of w and N_i. We defer its proof to Section 3.2.

▷ Claim 13. For any $i \in [k(r-1)]$, $j \in [N_i]$, and $x \in B_i$, we have

$$\frac{1}{45N_i} \le \mathbb{P}_{\mathbf{t}}[x \in U_{i,j}] \le \frac{3}{N_i}.$$

As a corollary, we have $\mathbb{P}_{\mathbf{t}}[x \in U_i] \ge 1/45$.

Functions drawn from $\mathcal{D}_{\texttt{yes}}$ are generated as follows. For each $i \in [k(r-1)]$ choose a uniform random assignment

$$\boldsymbol{\phi}_i : [N_i] \to \{i \bmod (r-1), i \bmod (r-1)+1\} \text{ and let } \boldsymbol{\phi} = (\boldsymbol{\phi}_i : i \in [k(r-1)]).$$

For every $x \in B_i$ define

$$f_{\boldsymbol{t},\boldsymbol{\phi}}(x) = \begin{cases} i \bmod (r-1), & \text{if } \forall j \in [N_i],\ x \not\succeq t^{i,j} \\ i \bmod (r-1)+1, & \text{if } \exists j \neq j' \in [N_i],\ x \succeq t^{i,j}, t^{i,j'} \\ \boldsymbol{\phi}_i(j), & \text{if } x \in U_{i,j}. \end{cases}$$

Functions drawn $\mathcal{D}_{\texttt{no}}$ are generated as follows. For each $i \in [k(r-1)]$ choose a uniform random function

$$\boldsymbol{r}_i \colon U_i \to \{i \bmod (r-1), i \bmod (r-1)+1\} \text{ and let } \boldsymbol{r} = (\boldsymbol{r}_i \colon i \in [k(r-1)]).$$

For each $x \in B_i$ define

$$f_{\boldsymbol{t},\boldsymbol{r}}(x) = \begin{cases} i \bmod (r-1), & \text{if } \forall j \in [N_i],\ x \not\succeq t^{i,j} \\ i \bmod (r-1)+1, & \text{if } \exists j \neq j' \in [N_i],\ x \succeq t^{i,j}, t^{i,j'} \\ \boldsymbol{r}_i(x), & \text{if } x \in U_i. \end{cases}$$

For x not belonging to any B_i: if $|x| < \frac{d}{2}$, then both the yes and no distributions assign value 0 and if $|x| \geq \frac{d}{2} + \varepsilon\sqrt{d}$, then both the yes and no distributions assign value $r-1$.

In summary, a function $f_{\boldsymbol{t},\boldsymbol{\phi}} \sim \mathcal{D}_{\texttt{yes}}$ assigns the same random value $\boldsymbol{\phi}_i(j) \in \{i \bmod (r-1), i \bmod (r-1)+1\}$ to all points in $U_{i,j}$, which results in a k-monotone function, whereas a function $f_{\boldsymbol{t},\boldsymbol{r}} \sim \mathcal{D}_{\texttt{no}}$ assigns an i.i.d. uniform random $\{i \bmod (r-1), i \bmod (r-1)+1\}$-value to each point in U_i, resulting in a function that is far from being k-monotone. By construction, to detect any difference between these cases a tester will need to sample at least two points from the same $U_{i,j}$. Theorem 1 follows immediately from the following three lemmas.

▶ **Lemma 14.** *Every function in the support of $\mathcal{D}_{yes}$ is k-monotone.*

Proof. Consider any $f_{\mathbf{t},\boldsymbol{\phi}}(x) \in \text{supp}(\mathcal{D}_{\texttt{yes}})$. For each $a \in [k]$, consider the union of $r-1$ blocks formed by

$$Y_a := B_{a(r-1)} \cup B_{a(r-1)+1} \cup \cdots \cup B_{(a+1)(r-1)-1}.$$

Recall that if $|x| < d/2$, then $f_{\mathbf{t},\boldsymbol{\phi}}(x) = 0$ and if $|x| \geq d/2 + \varepsilon\sqrt{d}$, then $f_{\mathbf{t},\boldsymbol{\phi}}(x) = r-1$. If $d/2 \leq |x| < d/2 + \varepsilon\sqrt{d}$, then $x \in \bigcup_{a\in[k]} Y_a$. Therefore, it suffices to show that for any pair of comparable points $x \prec y \in Y_a$, we have $f_{\mathbf{t},\boldsymbol{\phi}}(x) \leq f_{\mathbf{t},\boldsymbol{\phi}}(y)$. Firstly, observe that by construction all points $z \in B_{a(r-1)+b}$ have function value $f_{\mathbf{t},\boldsymbol{\phi}}(z) \in \{b, b+1\}$. Since $x \prec y$, if x and y are in different blocks, then $x \in B_{a(r-1)+b}$ and $y \in B_{a(r-1)+b'}$ where $b < b'$ and so the inequality is satisfied. Therefore, we may assume $x, y \in B_{a(r-1)+b}$ are in the same block. Since $x \prec y$, if $t \prec x$ for some term $t \in \text{supp}(\mathcal{D}_{\texttt{term}})$, then $t \prec y$ as well. I.e. the set of terms in $B_{a(r-1)+b}$ satisfied by y is a superset of the set of terms in $B_{a(r-1)+b}$ satisfied by x. By construction, this implies $f_{\mathbf{t},\boldsymbol{\phi}}(x) \leq f_{\mathbf{t},\boldsymbol{\phi}}(y)$. ◀

▶ **Lemma 15.** *For $f_{\boldsymbol{t},\boldsymbol{r}} \sim \mathcal{D}_{no}$, we have $\mathbb{P}_{\boldsymbol{t},\boldsymbol{r}}[\varepsilon(f_{\boldsymbol{t},\boldsymbol{r}}, \mathcal{M}_{r,k}) = \Omega(\varepsilon)] = \Omega(1)$.*

We prove Lemma 15 in Section 3.4.

▶ **Lemma 16.** *Given a collection of points $\boldsymbol{x} = (x_1, \ldots, x_s) \in (\{0,1\}^d)^s$ and a function $f \colon \{0,1\}^d \to [r]$, let $(\boldsymbol{x}, f(\boldsymbol{x})) = ((x_1, f(x_1)), \ldots, (x_s, f(x_s))))$ denote the corresponding collection of labelled examples. Let $\mathcal{E}_{yes}$ and $\mathcal{E}_{no}$ denote the distributions over $(\boldsymbol{x}, f(\boldsymbol{x}))$ when $\boldsymbol{x}$ consists of s i.i.d. uniform samples and $f \sim \mathcal{D}_{yes}$ and $f \sim \mathcal{D}_{no}$, respectively. If $s \leq 2^{\frac{(r-1)k\sqrt{d}}{5\varepsilon}}$, then the total variation distance between $\mathcal{E}_{yes}$ and $\mathcal{E}_{no}$ is $o(1)$.*

We prove Lemma 16 in Section 3.3.

3.2 Proof of Claim 13

Proof. Recall $w = \frac{(r-1)k\sqrt{d}}{2\varepsilon}$, $N_i = 2^w \cdot e^{-i}$, the definition of $\mathcal{D}_{\texttt{term}}$ from Definition 12, and the definition of $U_{i,j}$ from Equation (1). Since $x \in B_i$ we have $|x| = \frac{d}{2} + \ell$ where $\frac{i\varepsilon\sqrt{d}}{k(r-1)} \leq \ell < \frac{(i+1)\varepsilon\sqrt{d}}{k(r-1)}$. Note that $\mathbb{P}_{t\sim\mathcal{D}_{\texttt{term}}}[t \preceq x] = (|x|/d)^w$ since $t \preceq x$ iff the non-zero coordinates of t are a subset of the non-zero coordinates of x. Therefore, we have

$$\mathbb{P}_{\mathbf{t}}[x \in U_{i,j}] = \mathbb{P}_{t^{i,j}}[t^{i,j} \preceq x] \cdot \prod_{j' \in [N_i]\setminus\{j\}} \mathbb{P}_{t^{i,j'}}[t^{i,j'} \not\preceq x] = (|x|/d)^w \left(1 - (|x|/d)^w\right)^{N_i - 1}.$$

Note that the first term is upper bounded as

$$(|x|/d)^w \leq \left(\frac{\frac{d}{2} + \frac{(i+1)\cdot\varepsilon\sqrt{d}}{k(r-1)}}{d}\right)^w = \frac{1}{2^w}\left(1 + \frac{2\varepsilon}{k(r-1)\sqrt{d}} \cdot (i+1)\right)^w \leq \frac{e^{i+1+o(1)}}{2^w} \leq \frac{e^{1+o(1)}}{N_i}$$

and this immediately implies the upper bound on $\mathbb{P}_{\mathbf{t}}[x \in U_{i,j}]$. We can also lower bound this quantity by

$$(|x|/d)^w \geq \left(\frac{\frac{d}{2} + \frac{i\cdot\varepsilon\sqrt{d}}{k(r-1)}}{d}\right)^w = \frac{1}{2^w}\left(1 + \frac{2\varepsilon}{k(r-1)\sqrt{d}} \cdot i\right)^w \geq \frac{e^{i-o(1)}}{2^w} \geq \frac{1}{e^{o(1)}N_i}.$$

Now, combining our upper and lower bounds on $(|x|/d)^w$ yields

$$\mathbb{P}_{\mathbf{t}}[x \in U_{i,j}] \geq \frac{1}{e^{o(1)}N_i}\left(1 - \frac{e^{1+o(1)}}{N_i}\right)^{N_i} \geq \frac{1}{e^{o(1)}N_i} e^{-(1+o(1))\cdot e^{1+o(1)}} \geq \frac{1}{e^{e+1}N_i} \geq \frac{1}{45N_i}.$$

◁

3.3 $\mathcal{D}_{\texttt{yes}}$ and $\mathcal{D}_{\texttt{no}}$ are Hard to Distinguish: Proof of Lemma 16

Proof. Recall the definition of the set $U_{i,j}$ in Equation (1). For $a \neq b \in [s]$, let E_{ab} denote the event that x_a and x_b belong to the same $U_{i,j}$ for some $i \in [k(r-1)]$ and $j \in [N_i]$. Observe that conditioned on $\overline{\vee_{a,b} E_{ab}}$, the distributions $\mathcal{E}_{yes}$ and $\mathcal{E}_{no}$ are identical. Let $x, y \in \{0,1\}^d$ denote two i.i.d. uniform samples. We have

$$\begin{aligned}\mathbb{P}[E_{ab}] &= \mathbb{P}_{x,y,\mathbf{t}}\left[\bigvee_{i,j}(x \in U_{i,j} \wedge y \in U_{i,j})\right] \\ &= \sum_{i,j} \mathbb{P}_{x,y,\mathbf{t}}\left[x \in U_{i,j} \wedge y \in U_{i,j}\right] = \sum_{i,j} \mathbb{P}_{x,\mathbf{t}}[x \in U_{i,j}]^2 \end{aligned} \qquad (2)$$

where the first step holds since the $U_{i,j}$'s are disjoint and the second step holds by independence of x and y. Now, for a fixed $i \in [k(r-1)]$ and $j \in [N_i]$ we have the following: by Claim 13, for $x \in B_i$ we have $\mathbb{P}_{\mathbf{t}}[x \in U_{i,j}] \leq \frac{3}{N_i}$ and for $x \notin B_i$ we have $\mathbb{P}_{\mathbf{t}}[x \in U_{i,j}] = 0$. Therefore $\mathbb{P}_{x,\mathbf{t}}[x \in U_{i,j}] \leq \frac{3}{N_i}$. Therefore, the RHS of Equation (2) is bounded as

$$\sum_{i,j} \mathbb{P}_{x,\mathbf{t}}[x \in U_{i,j}]^2 = \sum_i N_i \cdot \mathbb{P}_{x,\mathbf{t}}[x \in U_{i,j}]^2 \leq \sum_i \frac{9}{N_i} \leq rk \cdot \frac{9}{N_{k(r-1)-1}}$$

since the N_i's are decreasing with respect to i. Therefore,

$$d_{TV}(\mathcal{E}_{\texttt{yes}}, \mathcal{E}_{\texttt{no}}) \leq \mathbb{P}_{\boldsymbol{x},\boldsymbol{t}}\left[\bigvee_{a,b\in[s]} E_{ab}\right] \leq s^2 \cdot rk \cdot \frac{9}{N_{k(r-1)-1}} = o(1)$$

since $N_{k(r-1)-1} = 2^{\frac{(r-1)k\sqrt{d}}{2\varepsilon}(1-o(1))} = \omega(s^2 \cdot rk)$. ◀

3.4 Functions Drawn from $\mathcal{D}_{\texttt{no}}$ are Far from k-Monotone: Proof of Lemma 15

Proof. We will use Claim 11, restated below for the special case of r-valued functions over the hypercube. Recall that $\mathcal{M}_{r,k}$ is the set of k-monotone functions $f\colon \{0,1\}^d \to [r]$.

$\triangleright$ Claim 17. Let $f\colon \{0,1\}^d \to [r]$ and $k' \geq 3k$ be an integer. Let $\mathcal{C} \subset (\{0,1\}^d)^{k'}$ be a collection of disjoint k'-alternating chains for f. Then

$$\varepsilon(f, \mathcal{M}_{r,k}) \geq \frac{1}{3 \cdot 2^d} \cdot \left|\bigcup_{C \in \mathcal{C}} C\right|.$$

From the above claim, we can lower bound the distance to k-monotonicity of f by showing that it contains a collection of disjoint k'-alternating chains where $k' \geq 3k$ whose union makes up an $\Omega(\varepsilon)$-fraction of the hypercube.

Recall $U_i = U_{i,1} \cup \cdots \cup U_{i,N_i} \subseteq B_i$ and note that $f_{\mathbf{t},\boldsymbol{r}} \sim \mathcal{D}_{\texttt{no}}$ takes values only from $\{i \bmod (r-1), i \bmod (r-1)+1\}$ in B_i. In particular, for $a \in \{0,1,\ldots,r-2\}$, let

$$X_a = B_a \cup B_{(r-1)+a} \cup B_{2(r-1)+a} \cup \cdots \cup B_{(k-1)(r-1)+a} = \bigcup_{i \in [k]} B_{i(r-1)+a} \tag{3}$$

and note that all points $x \in X_a$ are assigned value $f_{\mathbf{t},\boldsymbol{r}}(x) \in \{a, a+1\}$. Moreover, this value is chosen uniformly at random when $x \in \bigcup_{i \in [k]} U_{i(r-1)+a}$, which occurs with probability $\geq 1/45$ by Claim 13. Let $k'' := \frac{\varepsilon\sqrt{d}}{r-1}$ and recall that we are assuming $rk \leq \frac{\varepsilon\sqrt{d}}{24300}$ and so $k'' \geq 24300k$. We first show there exists a large collection $\mathcal{C}_a$ of length-k'' disjoint chains in X_a for all $a \in \{0,1,\ldots,r-2\}$.

$\triangleright$ Claim 18. For every $a \in \{0,1,\ldots,r-2\}$, there exists a collection of vertex disjoint chains $\mathcal{C}_a \subset (X_a)^{k''}$ in X_a of length k'' of size $|\mathcal{C}_a| \geq \Omega(\frac{2^d}{\sqrt{d}})$.

Proof. We start by showing that there is a large matching in the transitive closure of the hypercube from $L_{\frac{d}{2}}$ to $L_{\frac{d}{2}+\varepsilon\sqrt{d}-1}$. Consider the bipartite graph (U,V,E) where $U := L_{\frac{d}{2}}$, $V := L_{\frac{d}{2}+\varepsilon\sqrt{d}-1}$, and $E := \{(x,y) \in U \times V : x \prec y\}$. Observe that vertices in U have degree exactly $\Delta := \binom{\frac{d}{2}}{\varepsilon\sqrt{d}-1}$ while vertices in V have degree exactly $\binom{\frac{d}{2}+\varepsilon\sqrt{d}-1}{\varepsilon\sqrt{d}-1} \geq \Delta$. Note also that $|V| = \binom{d}{\frac{d}{2}+\varepsilon\sqrt{d}-1} \geq \Omega(\frac{2^d}{\sqrt{d}})$ by Stirling's approximation. We now use the following claim from [10].

$\triangleright$ Claim 19 (Claim 5.10 of [10]). Let (U,V,E) be a bipartite graph and $\Delta > 0$ be such that (a) each vertex $x \in U$ has degree exactly Δ and (b) each vertex $y \in V$ has degree at least Δ. Then there exists a matching $M \subseteq E$ in (U,V,E) of size $|M| \geq \frac{1}{2}|V|$.

By the above claim and the previous observations, there exist subsets $S \subseteq L_{\frac{d}{2}}$ and $T \subseteq L_{\frac{d}{2}+\varepsilon\sqrt{d}-1}$ of size $|S| = |T| = \Omega(\frac{2^d}{\sqrt{d}})$ and a bijection $\phi\colon S \to T$ satisfying $x \prec \phi(x)$ for all $x \in S$. We now use the following routing theorem due to Lehman and Ron to obtain a collection of disjoint chains from S to T.

▶ **Theorem 20** (Lehman-Ron, [56]). *Let $a < b$ and $S \subseteq L_a$, $T \subseteq L_b$ where $m := |S| = |T|$. Moreover, suppose there is a bijection $\phi\colon S \to T$ satisfying $x \prec \phi(x)$ for all $x \in S$. Then there exist m vertex disjoint paths from S to T in the hypercube.*

Now, invoking the above theorem on our bijection $\phi\colon S \to T$ yields a collection P of $|P| \geq \Omega(\frac{2^d}{\sqrt{d}})$ vertex disjoint paths from $L_{\frac{d}{2}}$ to $L_{\frac{d}{2}+\varepsilon\sqrt{d}-1}$. For each $a \in \{0, 1, \ldots, r-2\}$, let $\mathcal{C}_a$ denote the collection of chains formed by taking a path in P and including only the vertices from X_a (recall Equation (3)). Note that the resulting chains in $\mathcal{C}_a$ are of length $k'' = \frac{\varepsilon\sqrt{d}}{r-1}$. This completes the proof of Claim 18. ◁

From Claim 18, we have $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_{r-2}$ where each $\mathcal{C}_a \subset (X_a)^{k''}$ is a collection of vertex disjoint chains of length $k'' \geq 24300k$ of size $|\mathcal{C}_a| \geq \Omega(\frac{2^d}{\sqrt{d}})$. Fix a chain $C = (x_1, x_2, \ldots, x_{k''}) \in \mathcal{C}_a$. Let $A(C)$ be the random variable which denotes the max-length alternating sub-chain (recall Definition 9) of C over a random $f_{\mathbf{t},\boldsymbol{r}} \sim \mathcal{D}_{\texttt{no}}$. Fix x_j in the chain and suppose $x_j \in B_i \subseteq X_a$. By Claim 13, $\mathbb{P}_{\mathbf{t}}[x_j \in U_i] \geq 1/45$. Moreover, conditioned on $x_j \in U_i$, $f_{\mathbf{t},\boldsymbol{r}}(x_j)$ is chosen from $\{a, a+1\}$ uniformly at random. Thus, any step of the sequence

$$(1, f_{\mathbf{t},\boldsymbol{r}}(x_2) - f_{\mathbf{t},\boldsymbol{r}}(x_1), f_{\mathbf{t},\boldsymbol{r}}(x_3) - f_{\mathbf{t},\boldsymbol{r}}(x_2), \ldots, f_{\mathbf{t},\boldsymbol{r}}(x_{k''}) - f_{\mathbf{t},\boldsymbol{r}}(x_{k''-1}))$$

is non-zero *and* differs in sign from the previous non-zero step with probability at least $1/90$ and so $\mathrm{E}[A(C)] \geq k''/90$. I.e., $0 \leq \mathrm{E}[k'' - A(C)] < k''(1 - \frac{1}{90})$. Thus, using Markov's inequality we have

$$\mathbb{P}\left[A(C) < \frac{k''}{8100}\right] = \mathbb{P}\left[k'' - A(C) > k''\left(1 - \frac{1}{90}\right)\left(1 + \frac{1}{90}\right)\right] \leq \frac{1}{(1+\frac{1}{90})} = 1 - \frac{1}{91}. \quad (4)$$

Now, let $\mathcal{C} = \mathcal{C}_0 \cup \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_{r-2}$ and let $Z := |\{C \in \mathcal{C} \colon A(C) \geq \frac{k''}{8100}\}|$. By Equation (4) we have $\mathrm{E}[Z] \geq |\mathcal{C}|/91$ and $0 \leq \mathrm{E}[|\mathcal{C}| - Z] \leq |\mathcal{C}|(1 - \frac{1}{91})$. Again using Markov's inequality, we have

$$\mathbb{P}\left[Z < \frac{|\mathcal{C}|}{8281}\right] = \mathbb{P}\left[|\mathcal{C}| - Z > |\mathcal{C}|\left(1 - \frac{1}{91}\right)\left(1 + \frac{1}{91}\right)\right] \leq \frac{1}{(1+\frac{1}{91})} = 1 - \frac{1}{92}. \quad (5)$$

Now, for $C \in \mathcal{C}$ such that $A(C) \geq k''/8100$, let C' be any $(k''/8100)$-alternating sub-chain of C. Let $\mathcal{C}' = \{C' \colon C \in \mathcal{C} \text{ such that } A(C) \geq k''/8100\}$ which is a collection of disjoint $(k''/8100)$-alternating chains for $f_{\mathbf{t},\boldsymbol{r}}$. Now, recall that $k'' \geq 24300k$ and so $k''/8100 \geq 3k$. Thus, if $Z \geq |\mathcal{C}|/8281$, then $|\mathcal{C}'| \geq |\mathcal{C}|/8281$ and so by Claim 17 we have

$$\varepsilon(f_{\mathbf{t},\boldsymbol{r}}, \mathcal{M}_{r,k}) \geq \frac{1}{3 \cdot 2^d}\left|\bigcup_{C' \in \mathcal{C}'} C'\right| \geq \frac{1}{3 \cdot 2^d} \cdot |\mathcal{C}'| \cdot \frac{k''}{8100} \geq \frac{k'' \cdot |\mathcal{C}|}{201{,}228{,}300 \cdot 2^d} \quad (6)$$

By Claim 18 we have $|\mathcal{C}| \geq (r-1) \cdot \Omega(\frac{2^d}{\sqrt{d}})$ and recall that $k'' = \frac{\varepsilon\sqrt{d}}{r-1}$. Thus, the RHS of Equation (6) is $\Omega(\varepsilon)$. In conclusion,

$$\mathbb{P}_{\mathbf{t},\boldsymbol{r}}\left[\varepsilon(f_{\mathbf{t},\boldsymbol{r}}, \mathcal{M}_{r,k}) \geq \Omega(\varepsilon)\right] \geq \mathbb{P}\left[Z \geq \frac{|\mathcal{C}|}{8281}\right] \geq \frac{1}{92}$$

by Equation (5) and this completes the proof of Lemma 15. ◀

4 Learning Upper Bound over Product Spaces

In this section we prove Theorem 6, our upper bound for learning measurable k-monotone functions in $\mathbb{R}^d$. We restate the theorem below without any hidden logarithmic factors and for the case of $r = 2$. The theorem for general $r \geq 2$ can then be obtained by replacing ε with ε/r and δ by $1/3r$ following the same approach we used to prove Theorem 3 in Section 1.5.

► **Theorem 21.** *Given an arbitrary product measure μ, there is a learning algorithm under μ which learns any measurable k-monotone function $f\colon \mathbb{R}^d \to \{\pm 1\}$ to error ε with probability $1-\delta$ with time and sample complexity*

$$\ln\left(\frac{1}{\delta}\right)\cdot \min\left\{(d\log(dk/\varepsilon))^{O\left(\frac{k}{\varepsilon}\sqrt{d\log(dk/\varepsilon)}\right)}, \left(\frac{dk}{\varepsilon}\right)^{O(d)}\right\} \tag{7}$$

Our proof uses downsampling to reduce our learning problem over $\mathbb{R}^d$ to learning over a hypergrid, $[N]^d$, under the uniform distribution with mild label noise. In Section 4.1 we synthesize the results from [49] which we borrow for our proof. In Section 4.2 we give two learning results for hypergrids whose time complexities correspond to the two arguments inside the min expression in Equation (7). In Section 4.3 we describe the learning algorithm and prove its correctness.

Throughout this section, let $\mu = \prod_{i=1}^d \mu_i$ be any product measure over $\mathbb{R}^d$ and let N be a power of two satisfying $8kd/\varepsilon \le N \le 16kd/\varepsilon$.

4.1 Reduction to Hypergrids via Downsampling

The idea of downsampling is to construct a grid-partition of $\mathbb{R}^d$ into N^d blocks such that (a) the measure of each block under μ is roughly N^{-d}, and (b) the function f we're trying to learn is constant on most of the blocks. Roughly speaking, this allows us to learn f under μ by learning a proxy for f over $[N]^d$ under the uniform distribution. The value of N needed to achieve this depends on what [49] call the "block boundary size" of the function. Formally, the downsampling procedure constructs query access to maps $\mathsf{block}\colon \mathbb{R}^d \to [N]^d$ and $\mathsf{blockpoint}\colon [N]^d \to \mathbb{R}^d$ which have various good properties which we will spell out in the rest of this section. One should think of block as mapping each point $x \in \mathbb{R}^d$ to the block of the grid-partition that x belongs to and $\mathsf{blockpoint}$ as mapping each block to some specific point contained in the block. See [49, Def 2.1] for a formal definition. Given these maps and a function $f\colon \mathbb{R}^d \to \{\pm 1\}$ we define the function $f^{\mathsf{block}}\colon [N]^d \to \{\pm 1\}$ as $f^{\mathsf{block}}(z) = f(\mathsf{blockpoint}(z))$. We let $\mathsf{block}(\mu)$ denote the distribution over $[N]^d$ induced by sampling $x \sim \mu$ and then taking $\mathsf{block}(x)$.

► **Proposition 22** (Downsampling, [49])**.** *Let $f\colon \mathbb{R}^d \to \{0,1\}$ be a k-monotone function and $N, Q \in \mathbb{Z}^+$. Using*

$$m := O\left(\frac{NQ^2d^2}{\min(\delta,\varepsilon)^2}\ln\left(\frac{Nd}{\delta}\right)\right)$$

samples from $\mu = \mu_1 \times \cdots \times \mu_d$, there is a downsampling procedure that constructs query access to maps $\mathsf{block}\colon \mathbb{R}^d \to [N]^d$ and $\mathsf{blockpoint}\colon [N]^d \to \mathbb{R}^d$ such that with probability at least $1-\delta$ over the random samples, the following two conditions are satisfied:

1. $\left\|\mathsf{block}(\mu) - \mathsf{unif}([N]^d)\right\|_{TV} \le \frac{\delta}{Q}$.
2. $\mathbb{P}_{x\sim\mu}\left[f(x) \ne f^{\mathsf{block}}(\mathsf{block}(x))\right] \le \varepsilon$.

The total running time and number of samples is $O(m)$.

Proof. [49, Prop. 2.5] shows that there is a randomized procedure using m samples from μ and $O(m)$ time which constructs the maps block and $\mathsf{blockpoint}$ such that with probability 1, we get

$$\mathbb{P}_{x\sim\mu}\left[f(x) \ne f^{\mathsf{block}}(\mathsf{block}(x))\right] \le N^{-d}\cdot \mathsf{bbs}(f,N) + \left\|\mathsf{block}(\mu) - \mathsf{unif}([N]^d)\right\|_{\mathrm{TV}} \tag{8}$$

where $\mathsf{bbs}(f, N)$ is the N-block boundary size of f [49, Def. 2.4], which is at most kdN^{d-1} when f is k-monotone [49, Lemma 7.1]. Thus, the first of the two quantities in the RHS is at most kd/N which is at most $\varepsilon/8$ using our definition of N. Then, [49, Lemma 2.7] states that

$$\mathbb{P}\left[\left\|\mathsf{block}(\mu) - \mathsf{unif}([N]^d)\right\|_{\mathrm{TV}} > \beta\right] \leq 4Nd \cdot \exp\left(-\frac{\beta^2 m}{18Nd^2}\right) \tag{9}$$

and so invoking this lemma with $\beta := \min(\delta/4Q, \varepsilon/8)$ and $m := \frac{18Nd^2}{\beta^2} \ln\left(\frac{16Nd}{\delta}\right)$ completes the proof. ◀

4.2 Learning over Hypergrids

For a function $f : \mathcal{X} \to \{\pm 1\}$ and a measure μ over $\mathcal{X}$, recall that the *example oracle* for f under μ, denoted by $EX(f, \mu)$, when queried, generates an example, $(x, f(x))$, where x is sampled from μ. Given a *noise parameter* η, the *noisy example oracle* $EX^\eta(f, \mu)$, when queried, samples x from μ, returns the true example $(x, f(x))$ with probability $1 - \eta$, and returns the corrupted example $(x, -f(x))$ with probability η. This is referred to as *random classification noise* (RCN).

We prove the following two upper bounds for learning over hypergrids under RCN. The bound in Lemma 23 is relatively straightforward to prove using coupon collector arguments plus some additional work to handle the label noise. We give a proof in the appendix (see Section B in the full version of the paper).

▶ **Lemma 23** (Coupon Collecting Learner)**.** *Let $\varepsilon, \delta \in (0, 1)$, $\eta \in (0, 1/2)$, and $N \in \mathbb{Z}^+$. There is an algorithm which, given any k-monotone function $f : [N]^d \to \{\pm 1\}$, uses at most*

$$\widetilde{O}\left(\frac{1}{(1-2\eta)^2}\left(\log\frac{1}{\varepsilon} + \log\frac{1}{\delta}\right)\right) \cdot N^{O(d)}$$

examples from $EX^\eta(f, \mathsf{unif}([N]^d))$ and returns $h : [N]^d \to \{\pm 1\}$, satisfying $\mathbb{P}_h[d(f, h) \leq \varepsilon] \geq 1 - \delta$.

▶ **Lemma 24** (Hypercube Mapping Learner)**.** *Let $\varepsilon, \delta \in (0, 1)$, $\eta \in (0, 1/2)$, and $N \in \mathbb{Z}^+$ be a power of two. There is an algorithm which, given any k-monotone function $f : [N]^d \to \{\pm 1\}$, uses at most*

$$O\left(\frac{1}{\varepsilon^2(1-2\eta)^2} + \log\frac{1}{\delta}\right)(d\log N)^{O\left(\frac{k}{\varepsilon}\sqrt{d\log N}\right)}$$

examples from $EX^\eta(f, \mathsf{unif}([N]^d))$ and returns $h : [N]^d \to \{\pm 1\}$, satisfying $\mathbb{P}_h[d(f, h) \leq \varepsilon] \geq 1 - \delta$.

Proof. Let $b : [N] \to \{\pm 1\}^{\log N}$ denote the bijection which maps each element of $[N]$ to its bit representation. Let $\boldsymbol{b} : [N]^d \to \{\pm 1\}^{d \log N}$ be defined as $\boldsymbol{b}(x) = (b(x_1), \ldots, b(x_d))$. Given $f : [N]^d \to \{\pm 1\}$ define the function $f^{\mathsf{cube}} : \{\pm 1\}^{d \log N} \to \{\pm 1\}$ as $f^{\mathsf{cube}}(z) = f(\boldsymbol{b}^{-1}(z))$.

▶ **Observation 25.** *If f is k-monotone over $[N]^d$, then f^{cube} is k-monotone over $\{\pm 1\}^{d \log N}$.*

Proof. Observe that if $\boldsymbol{b}(x) \prec \boldsymbol{b}(y)$ in $\{\pm 1\}^{d \log N}$, then $x \prec y$ in $[N]^d$. Thus, if $\boldsymbol{b}(x_1) \prec \cdots \prec \boldsymbol{b}(x_m)$ is an m-alternating chain for f^{cube}, then $x_1 \prec \cdots \prec x_m$ is an m-alternating chain for f. Therefore, if f^{cube} is not k-monotone, then neither is f. ◀

Now, given Observation 25 and the bijection $\boldsymbol{b}\colon [N]^d \to \{\pm 1\}^{d \log N}$, it suffices to provide a learning algorithm for f^{cube}. This is achieved using the Low-Degree Algorithm introduced by [57] which was shown by [50] to be robust to classification noise. Formally, we use the following theorem, which we prove in the appendix for the sake of completeness (see Section A in the full version of the paper).

▶ **Theorem 26** (Low-Degree Algorithm with Classification Noise). *Let $\varepsilon, \delta \in (0,1)$ and $\eta \in (0,1/2)$. Suppose $\mathcal{C}$ is a concept class of Boolean functions over $\{\pm 1\}^d$ such that for some fixed positive integer τ, all $f \in \mathcal{C}$ satisfy $\sum_{S \subseteq [d]\colon |S| > \tau} \widehat{f}(S)^2 \le \varepsilon/2$. Then there is an algorithm $\mathcal{A}$ which, on any input $f \in \mathcal{C}$, uses at most*

$$O\left(\left(\frac{1}{\varepsilon^2(1-2\eta)^2} + \log\frac{1}{\delta}\right) \cdot d^{\tau}\right)$$

examples from $EX^{\eta}(f, \mathsf{unif}(\{\pm 1\}^d))$ and returns a hypothesis $h\colon \{\pm 1\}^d \to \{\pm 1\}$ where $\mathbb{P}_h[d(f,h) \le \varepsilon] \ge 1 - \delta$.

We use the following Fourier concentration lemma due to [17] for k-monotone Boolean functions.

▶ **Lemma 27** ([17]). *If $f\colon \{\pm 1\}^d \to \{\pm 1\}$ is k-monotone, then $\sum_{S\colon |S| > \frac{k\sqrt{d}}{\varepsilon}} \widehat{f}(S)^2 \le \varepsilon$.*

By Lemma 27, we can invoke Theorem 26 with $\tau = \frac{k\sqrt{d \log N}}{\varepsilon}$, concluding the proof of Lemma 24. ◀

4.3 Putting it Together: Proof of Theorem 21

Proof. We now have all the tools to define the algorithm and prove its correctness.

Algorithm 1 Learning algorithm for k-monotone functions under product measure μ.

Input: $\varepsilon, \delta \in (0,1)$ and access to examples from $EX(f,\mu)$ where $f\colon \mathbb{R}^d \to \{\pm 1\}$ is k-monotone;

1. Let N be a power of 2 such that $\frac{8kd}{\varepsilon} \le N \le \frac{16kd}{\varepsilon}$. Let $\mathcal{A}$ denote the learning algorithm for k-monotone functions $g\colon [N]^d \to \{\pm 1\}$ which has the smaller sample-complexity among the algorithms guaranteed by Lemma 23 and Lemma 24. Let Q be the sample-complexity of $\mathcal{A}$;
2. Run the downsampling procedure of Proposition 22 to obtain the maps block, $\mathsf{blockpoint}$, and access to the corresponding function $f^{\mathsf{block}}\colon [N]^d \to \{\pm 1\}$;
3. Obtain a set of Q examples $S \in (\mathbb{R}^d \times \{\pm 1\})^Q$ from $(EX(f,\mu))^Q$;
4. Let $S^{\mathsf{block}} = \{(\mathsf{block}(x), f(x))\colon (x, f(x)) \in S\} \in ([N]^d \times \{\pm 1\})^Q$;
5. Run $\mathcal{A}$ using the sample S^{block}, which returns a hypothesis $h^{\mathsf{block}}\colon [N]^d \to \{\pm 1\}$ for f^{block};

Return the hypothesis $h\colon \mathbb{R}^d \to \{\pm 1\}$ for $f\colon \mathbb{R}^d \to \{\pm 1\}$ defined as $h(x) = h^{\mathsf{block}}(\mathsf{block}(x))$

Recall that given maps $\mathsf{block}\colon \mathbb{R}^d \to [N]^d$, $\mathsf{blockpoint}\colon [N]^d \to \mathbb{R}^d$, and a function $f\colon \mathbb{R}^d \to \{\pm 1\}$ we define the function $f^{\mathsf{block}}\colon [N]^d \to \{\pm 1\}$ as $f^{\mathsf{block}}(z) = f(\mathsf{blockpoint}(z))$. Recall that $\mathsf{block}(\mu)$ is the distribution over $\mathsf{block}(x) \in [N]^d$ when $x \sim \mu$. By Proposition 22, step (2) of Alg. 1 results in the following items being satisfied with probability at least $1-\delta$.

1. $\left\|\mathsf{block}(\mu) - \mathsf{unif}([N]^d)\right\|_{\mathrm{TV}} \le \frac{\delta}{Q}$.
2. $\mathbb{P}_{x\sim\mu}\left[f(x) \ne f^{\mathsf{block}}(\mathsf{block}(x))\right] \le \varepsilon$.

Firstly, by item (2), an example $(\mathsf{block}(x), f(x))$ where $x \sim \mu$, is equivalent to an example $(z, b) \sim EX^\eta(f^{\mathsf{block}}, \mathsf{block}(\mu))$ for some $\eta \le \varepsilon$. I.e. the set $S^{\mathsf{block}} \in ([N]^d \times \{\pm 1\})^Q$ from step (4) of Alg. 1 is distributed according to $(EX^\eta(f^{\mathsf{block}}, \mathsf{block}(\mu)))^Q$. Now, as stated, Lemma 23 and Lemma 24 only hold when $\mathcal{A}$ is given a sample from $(EX^\eta(f^{\mathsf{block}}, \mathsf{unif}([N]^d)))^Q$. However, the following claim shows that since $\mathsf{block}(\mu)$ and $\mathsf{unif}([N]^d))$ are sufficiently close (item (1) above), the guarantees on $\mathcal{A}$ from Lemma 23 and Lemma 24 also hold when $\mathcal{A}$ is given a sample from $(EX^\eta(f^{\mathsf{block}}, \mathsf{block}(\mu)))^Q$.

▷ Claim 28. Let $\mathcal{C} \colon \mathcal{X} \to \{\pm 1\}$ be a concept class and let $\mathcal{A}$ be an algorithm which given any $f \in \mathcal{C}$, $\varepsilon, \delta \in (0, 1)$, and $\eta \in [0, 1/2)$ uses a sample from $(EX^\eta(f, \mathsf{unif}([N]^d)))^Q$ and produces h satisfying $\mathbb{P}_{x \sim \mathsf{unif}([N]^d)}[h(x) \ne f(x)] \le \varepsilon$ with probability at least $1 - \delta$. If $\mathcal{D}$ is a distribution over $[N]^d$ with $\left\|\mathcal{D} - \mathsf{unif}([N]^d)\right\|_{TV} \le \gamma$, then given a sample from $(EX^\eta(f, \mathcal{D}))^Q$, $\mathcal{A}$ produces h satisfying $\mathbb{P}_{x \sim \mathcal{D}}[h(x) \ne f(x)] \le \varepsilon + \gamma$ with probability at least $1 - (\delta + \gamma Q)$.

Using Claim 28 and item (1) above, if step (2) of Alg. 1 succeeds, then with probability at least $1 - 2\delta$, step (5) produces h^{block} such that $\mathbb{P}_{z \sim \mathsf{block}(\mu)}[h^{\mathsf{block}}(z) \ne f^{\mathsf{block}}(z)] \le 2\varepsilon$. By the triangle inequality and using our definition of h in the return statement of Alg. 1, we have

$$\begin{aligned}
&\mathbb{P}_{x \sim \mu}[h(x) \ne f(x)] \\
&\le \mathbb{P}_{x \sim \mu}[f(x) \ne f^{\mathsf{block}}(\mathsf{block}(x))] + \mathbb{P}_{x \sim \mu}[f^{\mathsf{block}}(\mathsf{block}(x)) \ne h^{\mathsf{block}}(\mathsf{block}(x))] \\
&= \mathbb{P}_{x \sim \mu}[f(x) \ne f^{\mathsf{block}}(\mathsf{block}(x))] + \mathbb{P}_{z \sim \mathsf{block}(\mu)}[f^{\mathsf{block}}(z) \ne h^{\mathsf{block}}(z)]. \qquad (10)
\end{aligned}$$

The first term in the RHS is at most ε by item (2) above and the second term is at most 2ε as we argued in the previous paragraph. Finally, adding up the failure probabilities of steps (2) and (5), we conclude that Alg. 1 produces h satisfying $\mathbb{P}_{x \sim \mu}[h(x) \ne f(x)] \le 3\varepsilon$ with probability at least $1 - 3\delta$. ◀

4.3.1 Proof of Claim 28

Proof. It is a well-known fact that for two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$, the TV-distance between the corresponding product distributions satisfies $\left\|\mathcal{D}_1^Q - \mathcal{D}_2^Q\right\|_{TV} \le Q \left\|\mathcal{D}_1 - \mathcal{D}_2\right\|_{TV}$ and thus we have

$$\left\|\mathcal{D}^Q - \mathsf{unif}([N]^d)^Q\right\|_{TV} \le \gamma Q$$

Given a set of Q examples $S \in ([N]^d \times \{\pm 1\})^Q$, let $E(S)$ denote the event that the algorithm $\mathcal{A}$ fails to produce a hypothesis with error at most ε, after sampling S. First, note the distribution over labels for the distributions are the same, and therefore

$$\begin{aligned}
&\mathbb{P}_{S \sim (EX^\eta(f, \mathcal{D}))^Q}[E(S)] - \mathbb{P}_{S \sim (EX^\eta(f, \mathsf{unif}([N]^d)))^Q}[E(S)] \\
&= \mathbb{P}_{S \sim \mathcal{D}^Q}[E(S)] - \mathbb{P}_{S \sim \mathsf{unif}([N]^d)^Q}[E(S)]. \qquad (11)
\end{aligned}$$

Using the definition of TV-distance we have

$$\mathbb{P}_{S \sim \mathcal{D}^Q}[E(S)] - \mathbb{P}_{S \sim \mathsf{unif}([N]^d)^Q}[E(S)] \le \left\|\mathcal{D}^Q - \mathsf{unif}([N]^d)^Q\right\|_{TV} \le \gamma Q \qquad (12)$$

and therefore

$$\mathbb{P}_{S \sim (EX^\eta(f, \mathcal{D}))^Q}[E(S)] \le \mathbb{P}_{S \sim (EX^\eta(f, \mathsf{unif}([N]^d)))^Q}[E(S)] + \gamma Q \le \delta + \gamma Q \qquad (13)$$

where we used $\mathbb{P}_{S\sim(EX^\eta(f,\mathsf{unif}([N]^d)))^Q}[E(S)] \le \delta$ by the assumption in the statement of the claim. Now, conditioned on $\neg E(S)$, we have that $\mathcal{A}$ produces h satisfying $\mathbb{P}_{x\sim\mathsf{unif}([N]^d)}[h(x) \neq f(x)] \le \varepsilon$. Again using our bound on the TV-distance, we have

$$\mathbb{P}_{x\sim\mathcal{D}}[h(x) \neq f(x)] - \mathbb{P}_{x\sim\mathsf{unif}([N]^d)}[h(x) \neq f(x)] \le \left\|\mathcal{D} - \mathsf{unif}([N]^d)\right\|_{TV} \le \gamma$$

and so $\mathbb{P}_{x\sim\mathcal{D}}[h(x) \neq f(x)] \le \varepsilon + \gamma$. ◁

5 Sample-Based Testing with One-Sided Error

In this section we prove Theorem 5, our upper and lower bound on sample-based testing with one-sided error over the hypercube.

Proof of Theorem 5. By a coupon-collecting argument, there is an $O(d \cdot 2^d)$ sample upper bound for *exactly learning* any function over $\{0,1\}^d$ under the uniform distribution and therefore the upper bound is trivial.

It suffices to prove the lower bound for the case of $r = 2$ and $k = 1$, i.e. for testing monotonicity of Boolean functions. We will need the following fact.

▶ **Fact 29.** *Let $A \subset \{0,1\}^d$ be any anti-chain and let $\ell\colon A \to \{0,1\}$ be any labelling of A. Then there exists a monotone function $f\colon \{0,1\}^d \to \{0,1\}$ such that $f(x) = \ell(x)$ for all $x \in A$. I.e. A* shatters *the class of monotone functions.*

Now, let T be any monotonicity tester with one-sided error and let $S \subseteq \{0,1\}^d$ denote a set of s i.i.d. uniform samples. Since T has one-sided error, if the input function is monotone, then T must accept. In other words, for T to reject it must be sure without a doubt that the input function is not monotone. By Fact 29 for T to be sure the input function is not monotone, it must be that S is *not* an anti-chain. Let $f\colon \{0,1\}^d \to \{0,1\}$ be any function which is ε-far from monotone. Since T is a valid tester, it rejects f with probability at least $2/3$. By the above argument we have

$$2/3 \le \mathbb{P}_S[T \text{ rejects } f] \le \mathbb{P}_S[S \text{ is not an anti-chain}] \le s^2 \cdot \mathbb{P}_{x,y\sim\{0,1\}^d}[x \preceq y] \tag{14}$$

where the last inequality is by a union bound over all pairs of samples. We then have

$$\mathbb{P}_{x,y\sim\{0,1\}^d}[x \preceq y] = \mathbb{P}_{x,y\sim\{0,1\}^d}[x_i \le y_i,\ \forall i \in [d]] = \prod_{i=1}^{d} \mathbb{P}_{x_i,y_i\sim\{0,1\}}[x_i \le y_i] = (3/4)^d. \tag{15}$$

Thus, combining Equation (14) and Equation (15) yields $s \ge \sqrt{\frac{2}{3}(\frac{4}{3})^d} = \exp(\Omega(d))$. ◀

References

1 Nir Ailon and Bernard Chazelle. Information theory in property testing and monotonicity testing in higher dimension. *Information and Computation*, 204(11):1704–1717, 2006.

2 Nir Ailon, Bernard Chazelle, Seshadhri Comandur, and Ding Liu. Estimating the distance to a monotone function. *Random Structures Algorithms*, 31(3):371–383, 2007.

3 Maria-Florina Balcan, Eric Blais, Avrim Blum, and Liu Yang. Active property testing. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS*, 2012. doi:10.1109/FOCS.2012.64.

4 Aleksandrs Belovs and Eric Blais. A polynomial lower bound for testing monotonicity. In *Proceedings, ACM Symposium on Theory of Computing (STOC)*, 2016.

5 Piotr Berman, Meiram Murzabulatov, and Sofya Raskhodnikova. The power and limitations of uniform samples in testing properties of figures. *Algorithmica*, 81(3):1247–1266, 2019. doi:10.1007/s00453-018-0467-9.

6 Piotr Berman, Meiram Murzabulatov, and Sofya Raskhodnikova. Testing convexity of figures under the uniform distribution. *Random Struct. Algorithms*, 54(3):413–443, 2019. doi:10.1002/rsa.20797.

7 Piotr Berman, Sofya Raskhodnikova, and Grigory Yaroslavtsev. L_p-testing. In *Proceedings, ACM Symposium on Theory of Computing (STOC)*, 2014.

8 Arnab Bhattacharyya. A note on the distance to monotonicity of boolean functions. Technical Report 012, Electronic Colloquium on Computational Complexity (ECCC), 2008.

9 Arnab Bhattacharyya, Elena Grigorescu, Madhav Jha, Kyoming Jung, Sofya Raskhodnikova, and David Woodruff. Lower bounds for local monotonicity reconstruction from transitive-closure spanners. *SIAM Journal on Discrete Mathematics (SIDMA)*, 26(2):618–646, 2012.

10 Hadley Black, Eric Blais, and Nathaniel Harms. Testing and learning convex sets in the ternary hypercube. In *15th Innovations in Theoretical Computer Science Conference, ITCS*, 2024. doi:10.4230/LIPICS.ITCS.2024.15.

11 Hadley Black, Deeparnab Chakrabarty, and C. Seshadhri. A $o(d) \cdot \mathrm{polylog}(n)$ monotonicity tester for Boolean functions over the hypergrid $[n]^d$. In *Proceedings, ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2018.

12 Hadley Black, Deeparnab Chakrabarty, and C. Seshadhri. Domain reduction for monotonicity testing: A $o(d)$ tester for boolean functions in d-dimensions. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA*, 2020. doi:10.1137/1.9781611975994.122.

13 Hadley Black, Deeparnab Chakrabarty, and C. Seshadhri. A $d^{1/2+o(1)}$ monotonicity tester for boolean functions on d-dimensional hypergrids. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, 2023. doi:10.1109/FOCS57990.2023.00110.

14 Hadley Black, Deeparnab Chakrabarty, and C. Seshadhri. Directed isoperimetric theorems for boolean functions on the hypergrid and an $\widetilde{O}(n\sqrt{d})$ monotonicity tester. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC*, 2023. doi:10.1145/3564246.3585167.

15 Hadley Black, Iden Kalemaj, and Sofya Raskhodnikova. Isoperimetric inequalities for real-valued functions with applications to monotonicity testing. *Random Structures & Algorithms*, 2024.

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

17 Eric Blais, Clément L. Canonne, Igor Carboni Oliveira, Rocco A. Servedio, and Li-Yang Tan. Learning circuits with few negations. In *RANDOM*, 2015. doi:10.4230/LIPIcs.APPROX-RANDOM.2015.512.

18 Eric Blais, Renato Ferreira Pinto Jr, and Nathaniel Harms. Vc dimension and distribution-free sample-based testing. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 504–517, 2021.

19 Eric Blais, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Lower bounds for testing properties of functions over hypergrid domains. In *Proceedings, IEEE Conference on Computational Complexity (CCC)*, 2014.

20 Eric Blais and Yuichi Yoshida. A characterization of constant-sample testable properties. *Random Struct. Algorithms*, 55(1):73–88, 2019. doi:10.1002/rsa.20807.

21 Mark Braverman, Subhash Khot, Guy Kindler, and Dor Minzer. Improved monotonicity testers via hypercube embeddings. In *14th Innovations in Theoretical Computer Science Conference, ITCS*, 2023. doi:10.4230/LIPIcs.ITCS.2023.25.

22 Jop Briët, Sourav Chakraborty, David García Soriano, and Ari Matsliah. Monotonicity testing and shortest-path routing on the cube. *Combinatorica*, 32(1):35–53, 2012.

23 Nader H. Bshouty and Christino Tamon. On the fourier spectrum of monotone functions. *J. ACM*, 43(4):747–770, 1996. doi:10.1145/234533.234564.

24 Clément L. Canonne, Elena Grigorescu, Siyao Guo, Akash Kumar, and Karl Wimmer. Testing k-monotonicity: The rise and fall of boolean functions. *Theory Comput.*, 15:1–55, 2019. doi:10.4086/toc.2019.v015a001.

25 Deeparnab Chakrabarty, Kashyap Dixit, Madhav Jha, and C. Seshadhri. Property testing on product distributions: Optimal testers for bounded derivative properties. In *Proceedings, ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2015.

26 Deeparnab Chakrabarty and C. Seshadhri. Optimal bounds for monotonicity and Lipschitz testing over hypercubes and hypergrids. In *Proceedings, ACM Symposium on Theory of Computing (STOC)*, 2013.

27 Deeparnab Chakrabarty and C. Seshadhri. An $o(n)$ monotonicity tester for Boolean functions over the hypercube. *SIAM Journal on Computing (SICOMP)*, 45(2):461–472, 2014.

28 Deeparnab Chakrabarty and C. Seshadhri. An optimal lower bound for monotonicity testing over hypergrids. *Theory of Computing*, 10:453–464, 2014. doi:10.4086/toc.2014.v010a017.

29 Xi Chen, Anindya De, Yuhao Li, Shivam Nadimpalli, and Rocco A. Servedio. Mildly exponential lower bounds on tolerant testers for monotonicity, unateness, and juntas. In *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA*, 2024. doi:10.1137/1.9781611977912.151.

30 Xi Chen, Anindya De, Rocco A. Servedio, and Li-Yang Tan. Boolean function monotonicity testing requires (almost) $O(n^{1/2})$ non-adaptive queries. In *Proceedings, ACM Symposium on Theory of Computing (STOC)*, 2015.

31 Xi Chen, Adam Freilich, Rocco A. Servedio, and Timothy Sun. Sample-based high-dimensional convexity testing. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, 2017. doi:10.4230/LIPICS.APPROX-RANDOM.2017.37.

32 Xi Chen, Rocco A. Servedio, and Li-Yang. Tan. New algorithms and lower bounds for monotonicity testing. In *Proceedings, IEEE Symposium on Foundations of Computer Science (FOCS)*, 2014.

33 Xi Chen, Erik Waingarten, and Jinyu Xie. Beyond talagrand: New lower bounds for testing monotonicity and unateness. In *Proceedings, ACM Symposium on Theory of Computing (STOC)*, 2017.

34 Yevgeny Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved testing algorithms for monotonicity. *Proceedings, International Workshop on Randomization and Computation (RANDOM)*, 1999.

35 Funda Ergun, Sampath Kannan, Ravi Kumar, Ronitt Rubinfeld, and Mahesh Viswanathan. Spot-checkers. *J. Comput. System Sci.*, 60(3):717–751, 2000.

36 Shahar Fattal and Dana Ron. Approximating the distance to monotonicity in high dimensions. *ACM Trans. on Algorithms (TALG)*, 6(3), 2010.

37 Renato Ferreira Pinto Jr and Nathaniel Harms. Distribution testing under the parity trace, 2023. arXiv:2304.01374.

38 Renato Ferreira Pinto Jr and Nathaniel Harms. Distribution testing with a confused collector. In *15th Innovations in Theoretical Computer Science Conference, ITCS*, 2024. doi:10.4230/LIPICS.ITCS.2024.47.

39 Eldar Fischer. On the strength of comparisons in property testing. *Information and Computation*, 189(1):107–116, 2004.

40 Eldar Fischer, Yonatan Goldhirsh, and Oded Lachish. Partial tests, universal tests and decomposability. In *Innovations in Theoretical Computer Science, ITCS*. ACM, 2014. doi:10.1145/2554797.2554841.

41 Eldar Fischer, Oded Lachish, and Yadu Vasudev. Trading query complexity for sample-based testing and multi-testing scalability. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS*. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.75.

42 Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, and Ronitt Rubinfeld. Monotonicity testing over general poset domains. *Proceedings, ACM Symposium on Theory of Computing (STOC)*, 2002.

43 Oded Goldreich, Shafi Goldwasser, Eric Lehman, Dana Ron, and Alex Samorodnitsky. Testing monotonicity. *Combinatorica*, 20:301–337, 2000.

44 Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998.

45 Oded Goldreich and Dana Ron. On sample-based testers. *ACM Trans. Comput. Theory*, 8(2):7:1–7:54, 2016. doi:10.1145/2898355.

46 Elena Grigorescu, Akash Kumar, and Karl Wimmer. Flipping out with many flips: Hardness of testing k-monotonicity. *SIAM J. Discret. Math.*, 33(4):2111–2125, 2019. doi:10.1137/18M1217978.

47 Shirley Halevy and Eyal Kushilevitz. Distribution-free property testing. *Proceedings, International Workshop on Randomization and Computation (RANDOM)*, 2003.

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

49 Nathaniel Harms and Yuichi Yoshida. Downsampling for testing and learning in product distributions. In *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022*, 2022. doi:10.4230/LIPIcs.ICALP.2022.71.

50 Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. *J. ACM*, 45(6):983–1006, 1998. doi:10.1145/293347.293351.

51 Michael J. Kearns and Dana Ron. Testing problems with sublearning sample complexity. *J. Comput. Syst. Sci.*, 61(3):428–456, 2000. doi:10.1006/jcss.1999.1656.

52 Subhash Khot, Dor Minzer, and Muli Safra. On monotonicity testing and Boolean isoperimetric type theorems. In *Proceedings, IEEE Symposium on Foundations of Computer Science (FOCS)*, 2015.

53 Subhash Khot, Dor Minzer, and Muli Safra. On monotonicity testing and boolean isoperimetric-type theorems. *SIAM J. Comput.*, 47(6):2238–2276, 2018. doi:10.1137/16M1065872.

54 Jane Lange, Ronitt Rubinfeld, and Arsen Vasilyan. Properly learning monotone functions via local correction. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS*, 2022. doi:10.1109/FOCS54457.2022.00015.

55 Jane Lange and Arsen Vasilyan. Agnostic proper learning of monotone functions: beyond the black-box correction barrier. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, 2023. doi:10.1109/FOCS57990.2023.00068.

56 Eric Lehman and Dana Ron. On disjoint chains of subsets. *Journal of Combinatorial Theory, Series A*, 94(2):399–404, 2001.

57 Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, fourier transform, and learnability. *J. ACM*, 40(3):607–620, 1993. doi:10.1145/174130.174138.

58 Ramesh Krishnan S. Pallavoor, Sofya Raskhodnikova, and Nithin Varma. Parameterized property testing of functions. *ACM Trans. Comput. Theory*, 9(4):17:1–17:19, 2018. doi:10.1145/3155296.

59 Sofya Raskhodnikova. Monotonicity testing. *Masters Thesis, MIT*, 1999.

60 Dana Ron, Ronitt Rubinfeld, Muli Safra, and Omri Weinstein. Approximating the Influence of Monotone Boolean Functions in $O(\sqrt{n})$ Query Complexity. In *Proceedings, International Workshop on Randomization and Computation (RANDOM)*, 2011.

61 R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal of Computing*, 25:647–668, 1996.

62 Michael E. Saks and C. Seshadhri. Parallel monotonicity reconstruction. In *Proceedings, ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2008.

63 Michel Talagrand. How much are increasing sets positively correlated? *Comb.*, 16(2):243–258, 1996. doi:10.1007/BF01844850.