Electronic Colloquium on Computational Complexity, Report No. 163 (2019)

Approximating the Distance to Monotonicity of Boolean Functions*

Ramesh Krishnan S. Pallavoor† Sofya Raskhodnikova† Erik Waingarten‡

Abstract

We design a nonadaptive algorithm that, given a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ which is α-far from monotone, makes $\mathrm{poly}(n, 1/\alpha)$ queries and returns an estimate that, with high probability, is an $\widetilde{O}(\sqrt{n})$-approximation to the distance of f to monotonicity. Furthermore, we show that for any constant $\kappa > 0$, approximating the distance to monotonicity up to $n^{1/2-\kappa}$-factor requires 2^{n^κ} nonadaptive queries, thereby ruling out a $\mathrm{poly}(n, 1/\alpha)$-query nonadaptive algorithm for such approximations. This answers a question of Seshadhri (Property Testing Review, 2014) for the case of nonadaptive algorithms. Approximating the distance to a property is closely related to tolerantly testing that property. Our lower bound stands in contrast to standard (non-tolerant) testing of monotonicity that can be done nonadaptively with $\widetilde{O}(\sqrt{n}/\varepsilon^2)$ queries.

We obtain our lower bound by proving an analogous bound for erasure-resilient testers. An α-erasure-resilient tester for a desired property gets oracle access to a function that has at most an α fraction of values erased. The tester has to accept (with probability at least 2/3) if the erasures can be filled in to ensure that the resulting function has the property and to reject (with probability at least 2/3) if every completion of erasures results in a function that is ε-far from having the property. Our method yields the same lower bounds for unateness and being a k-junta. These lower bounds improve exponentially on the existing lower bounds for these properties.

*This work was done in part while the authors were visiting the Simons Institute for the Theory of Computing.

†Department of Computer Science, Boston University. Email: rameshkp@bu.edu, sofya@bu.edu. The work of these authors was partially supported by NSF award CCF-1909612.

‡Department of Computer Science, Columbia University. Email: eaw@cs.columbia.edu. This work is supported in part by NSF Graduate Research Fellowship (Grant No. DGE-16-44869).

ISSN 1433-8092

1 Introduction

Property testing [54, 39] was introduced to provide a formal model for studying algorithms for massive datasets. For such algorithms to achieve their full potential, they have to be robust to adversarial corruptions in the input. Tolerant property testing [50] and, equivalently[1], distance approximation, generalize the standard property testing model to allow for errors in the input.

In this work, we study the problem of approximating the distance to several properties of Boolean functions, with the focus on monotonicity. A function $f\colon \{0,1\}^n \to \{0,1\}$ is *monotone* if $f(x) \le f(y)$ whenever $x \prec y$, i.e., $x_i \le y_i$ for all $i \in [n]$. The (relative) distance between two functions over $\{0,1\}^n$ is the fraction of the domain points on which they differ. Given a function f and a set $\mathcal{P}$ (of functions with the desired property), the distance from f to $\mathcal{P}$, denoted $\mathrm{dist}(f, \mathcal{P})$, is the distance from f to the closest function in $\mathcal{P}$. Given $\alpha \in (0, 1/2)$, a function is α-far from $\mathcal{P}$ if its distance from $\mathcal{P}$ is at least α; otherwise, it is α-close. We study randomized algorithms which, given oracle access to a Boolean function, output an approximation of distance to monotonicity by making only a small number of queries. Specifically, given an input function $f\colon \{0,1\}^n \to \{0,1\}$ which is promised to be at least α-far from monotone, an algorithm that achieves a c-approximation for $c > 1$ should output a real number $\hat{\varepsilon} \in (0,1)$ that satisfies, with probability at least 2/3,

$$\mathrm{dist}(f, \textsc{Mono}) \le \hat{\varepsilon} \le c \cdot \mathrm{dist}(f, \textsc{Mono}).$$

Our goal is to understand the best approximation ratio c that can be achieved[2] in time polynomial in the dimension n and $1/\alpha$.

Fattal and Ron [32] investigated a more general problem of approximating the distance to monotonicity of functions on the hypergrid $[t]^n$. They gave several algorithms which achieved an approximation ratio $O(n)$ in time polynomial in n and $1/\alpha$; for better approximations, they designed an algorithm with approximation ratio n/k, for every k, but with running time exponential in k. For the special case of the hypercube domain, an $O(n)$-approximation can be obtained by simply estimating the number of *decreasing* edges of f, that is, edges (x, y) of the hypercube for which $x \prec y$ but $f(x) > f(y)$. This follows from early works on monotonicity testing [30, 51, 38, 36]. These early works showed that the number of decreasing edges of a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ is between $\mathrm{dist}(f, \textsc{Mono}) \cdot 2^n$ and $\mathrm{dist}(f, \textsc{Mono}) \cdot n2^n$. Thus, by obtaining a constant-factor approximation to the number of violated edges, one gets an $O(n)$-approximation to $\mathrm{dist}(f, \textsc{Mono})$. Prior to this work, no nontrivial hardness results were known for this problem, other than the corresponding lower bounds on (standard) property testing.

Our Results. All our results are on *nonadaptive* algorithms. An algorithm is *nonadaptive* if it makes all of its queries in advance, before receiving any answers; otherwise, it is *adaptive*. Nonadaptive algorithms are especially straightforward to implement and achieve maximal parallelism. Additionally, every nonadaptive algorithm that approximates the distance to monotonicity of Boolean functions can be easily converted to an algorithm for approximating the L_p-distance to monotonicity of real-valued functions [7].

[1] The query complexity of tolerant testing and distance approximation are within a logarithmic factor of each other. See [50] for a discussion of the relationship.

[2] An equivalent way of stating this type of results is to express the approximation guarantee in terms of both multiplicative and additive error, but with no lower bound on the distance. Purely multiplicative approximation would require correctly identifying inputs with the property, which generally cannot be achieved in time sublinear in the size of the input.

We design a nonadaptive $\tilde{O}(\sqrt{n})$-approximation algorithm for distance to monotonicity that runs in time polynomial in the number of dimensions, n, and $1/\alpha$. Our algorithm improves on the $O(n)$-approximation obtained by Fattal and Ron [32].

Theorem 1.1 (Approximation Algorithm)**.** *There is a nonadaptive (randomized) algorithm that, given a parameter $\alpha \in (0, 1/2)$ and oracle access to a Boolean function $f: \{0,1\}^n \to \{0,1\}$ which is α-far from monotone, makes $\operatorname{poly}(n, 1/\alpha)$ queries and returns an estimate that, with probability at least $2/3$, is an $\tilde{O}(\sqrt{n})$-approximation to $\operatorname{dist}(f, \textsc{Mono})$.*

Our algorithm works by estimating the size of a particular class of matchings parameterized by subsets $S \subseteq [n]$ and consisting of decreasing edges along the directions in S. For every S, the size of the matching, divided by 2^n, is a lower bound for $\operatorname{dist}(f, \textsc{Mono})$, because any monotone function $g: \{0,1\}^n \to \{0,1\}$ must disagree with f on at least one endpoint of each decreasing edge. The important feature of this class of matchings is that the membership of a given edge in a specified matching can be verified locally by querying f on the endpoints of the edge and their neighbors. Finally, we use a slightly improved version of the (robust) directed isoperimetric inequality by Khot et al. [42]. Our improvements to isoperimetric inequalities of [42] are stated in Theorems 2.7 and A.1 and proved in Appendix A. We use Theorem 2.7 to show that if f is ε-far from monotone, then either the algorithm samples some set $S \subseteq [n]$ where the corresponding matching has size at least $\tilde{\Omega}(\varepsilon/\sqrt{n}) \cdot 2^n$, or there exist $\varepsilon\sqrt{n} \cdot 2^n$ decreasing edges (see Lemma 2.6). In the latter case, the fact that the number of decreasing edges, divided by 2^n, is an n-approximation to the distance to monotonicity is sufficient to obtain an $\tilde{O}(\sqrt{n})$-approximation for this quantity.

Remark 1.2. *Let function $f: \{0,1\}^n \to \{0,1\}$ be ε-far from monotone. For all $x \in \{0,1\}^n$, let $I_f^-(x)$ be the number of decreasing edges incident on x. Khot et al. [42] proved that*

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \{0,1\}^n}\left[\sqrt{I_f^-(\boldsymbol{x})}\right] \geq \widetilde{\Omega}(\varepsilon).$$

(See, also, related statements in Theorems 2.7 and A.1). Hence, an algorithm that evaluates $\sqrt{I_f^-(\boldsymbol{x})}$ on a uniformly random $x \in \{0,1\}^n$ would, in expectation, get at least $\tilde{\Omega}(\varepsilon)$ on inputs f which are ε-far from monotone. The problem is in deducing an upper bound on this estimate for functions which are $\tilde{O}(\varepsilon/\sqrt{n})$-close to monotone. For example, consider the function $f: \{0,1\}^n \to \{0,1\}$ defined as follows:

$$f(x) = \begin{cases} 1 & \text{if } x = 0^n, \\ 0 & \text{if } x = 1^n, \\ \operatorname{Maj}(x) & \text{otherwise,} \end{cases}$$

where $\operatorname{Maj}(\cdot)$ is the majority function. Then, $\operatorname{dist}(f, \textsc{Mono}) = 2/2^n$, yet

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \{0,1\}^n}\left[\sqrt{I_f^-(\boldsymbol{x})}\right] \geq n/2^n.$$

The lower and the upper bound differ by a factor of $\Theta(n)$, precluding us from getting an $\tilde{O}(\sqrt{n})$-approximation.

Chakrabarty and Seshadhri, in a personal communication, notified us of an alternative approach towards a $O(\sqrt{n})$-approximation via estimating the size of a maximal matching of decreasing edges.

Results in [42, 20] imply that the size of a maximal matching is an $O(\sqrt{n})$-approximation to the distance to monotonicity, and there are sublinear time algorithms for approximating this quantity [60, 48]. However, these algorithms are adaptive.

Next, we show that a slightly better approximation, specifically, with a ratio of $n^{1/2-\kappa}$ for an arbitrarily small constant $\kappa > 0$, requires exponentially many queries in n^κ for every nonadaptive algorithm.

Theorem 1.3 (Approximation Lower Bound)**.** *Let $\kappa > 0$ be any small constant. There exist $\alpha = \text{poly}(1/n)$ and $\varepsilon = \text{poly}(1/n)$ with $\frac{\varepsilon}{\alpha} = \Omega(n^{1/2-\kappa})$, for which every nonadaptive algorithm requires more than 2^{n^κ} queries to $f\colon \{0,1\}^n \to \{0,1\}$ to distinguish functions f that are α-close to monotone from those that are ε-far from monotone with probability at least 2/3.*

This result, in combination with Theorem 1.1, answers an open question on the problem of approximating the distance to monotonicity by Seshadhri [57] for the case of nonadaptive algorithms. It is the first lower bound for this problem, and it rules out nonadaptive algorithms that achieve approximations substantially better than $\sqrt{n}$ with $\text{poly}(n, 1/\alpha)$ queries, demonstrating that Theorem 1.1 is essentially tight. This bound is exponentially larger than the corresponding lower bound in the standard property testing model and, in fact, than the running time of known algorithms for testing monotonicity. We elaborate on this point in the discussion below on separation.

To obtain Theorem 1.3, we investigate a variant of the property testing model, called *erasure-resilient* testing. This variant, proposed by Dixit et al. [29], is intended to study property testing in the presence of adversarial erasures. An erased function value is denoted by $\perp$. An α-erasure-resilient ε-tester for a desired property gets oracle access to a function $f\colon \{0,1\}^n \to \{0,1,\perp\}$ that has at most an α fraction of values erased. The tester has to accept (with probability at least 2/3) if the erasures can be filled in to ensure that the resulting function has the property and to reject (with probability at least 2/3) if every completion of erasures results in a function that is ε-far from having the property. As observed in [29], the query complexity of problems in this model lies between their complexity in the standard property testing model and the tolerant testing model. Specifically, a (standard) ε-tester that, given a parameter ε, accepts functions with the property and rejects functions that are ε-far from the property (with probability at least 2/3), is a special case of an α-erasure-resilient ε-tester with α set to 0. Importantly for us, a tolerant tester that, given $\alpha, \varepsilon \in (0, 1/2)$ with $\alpha < \varepsilon$, accepts functions that are α-close and rejects functions that are ε-far (with probability at least 2/3) can be use to get an α-erasure-resilient ε-tester. The erasure-resilient tester can be obtained by simply filling in erasures with arbitrary values and running the tolerant tester. We prove a lower bound for erasure-resilient monotonicity testing.

Our method yields lower bounds for two other properties of Boolean functions: unateness, a natural generalization of monotonicity, and being a k-junta. A Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ is *unate* if, for every variable $i \in [n]$, the function is nonincreasing or nondecreasing in that variable. A function $f\colon \{0,1\}^n \to \{0,1\}$ is a *k-junta* if it depends on at most k (out of n) variables.

We prove the following result on erasure-resilient testing which implies Theorem 1.3.

Theorem 1.4 (Erasure-Resilient Lower Bound)**.** *Let $\kappa > 0$ be a small constant. There exist $\alpha = \text{poly}(1/n)$ and $\varepsilon = \text{poly}(1/n)$ with $\frac{\varepsilon}{\alpha} = \Omega(n^{1/2-\kappa})$, for which every nonadaptive α-erasure-resilient ε-tester requires more than 2^{n^κ} queries to test monotonicity of functions $f\colon \{0,1\}^n \to \{0,1\}$. The same bound holds for testing unateness and the $n/2$-junta property.*

Theorem 1.4 directly implies lower bounds analogous to the one stated in Theorem 1.3 for unateness and being an $n/2$-junta. Lower bounds for approximating the distance to unateness and to being a k-junta have been investigated by Levi and Waingarten [46]. They showed that every algorithm approximating the distance to unateness within a constant factor requires $\tilde{\Omega}(n)$ queries and strengthened their lower bound to $\tilde{\Omega}(n^{3/2})$ queries for nonadaptive algorithms. They also showed that every nonadaptive algorithm that provides a constant approximation to the distance to being a k-junta must make $\tilde{\Omega}(k^2)$ queries. Our lower bounds are exponentially larger than those obtained by Levi and Waingarten [46] and hold for larger approximation factors.

Separation. Our lower bounds provide natural properties for which erasure-resilient property testing (and hence, distance approximation) is exponentially harder than standard property testing with nonadaptive algorithms. Previously, such strong separation was only known for artificially constructed properties based on PCPs of proximity [34, 29]. For testing monotonicity of Boolean function, the celebrated nonadaptive algorithm of Khot, Minzer and Safra [42] makes $\tilde{O}(\sqrt{n}/\varepsilon^2)$ queries. Unateness can be tested nonadaptively with $O(\frac{n}{\varepsilon}\log\frac{n}{\varepsilon})$ queries [3] whereas the property of being a k-junta can be tested nonadaptively with $\tilde{O}(k^{3/2}/\varepsilon)$ queries [9]. Our lower bound shows that, for all three properties, nonadaptive testers requires exponentially many queries when the ratio ε/α is substantially smaller than $\sqrt{n}$. This stands in contrast to examples of many properties provided in [29], for which erasure-resilient testers have essentially the same query complexity as standard testers.

1.1 Previous Work

Testing monotonicity and unateness (first studied in [38]), as well as k-juntas (first studied in [35]), are among the most widely investigated problems in property testing ([31, 30, 51, 45, 36, 1, 33, 41, 4, 50, 2, 8, 14, 11, 18, 19, 13, 17, 22, 21, 42, 5, 25, 49] study mononicity testing, [43, 3, 25, 26, 24] study unateness testing, and [27, 9, 10, 15, 56, 23, 55] study k-junta testing). Nearly all the previous work on these properties is in the standard testing model. The best bounds on the query complexity of these problems are an $\tilde{O}(\sqrt{n})$-query algorithm of [42] and lower bounds of $\tilde{\Omega}(\sqrt{n})$ (nonadaptive) and $\tilde{\Omega}(n^{1/3})$ (adaptive) [25] for monotonicity, and tight upper and lower bounds of $\tilde{\Theta}(n^{2/3})$ for unateness testing [24, 25], as well as $\Theta(k \log k)$ for k-junta testing [9, 55].

Beyond the (standard) property testing, the questions of erasure-resilient and tolerant testing have also received some attention ([29, 52] study the erasure-resilient model, and [40, 50, 34, 37, 2, 44, 47, 32, 16, 7, 6, 58, 12, 46, 28] study the tolerant testing model). Specifically for monotonicity, in [29], an erasure-resilient tester for functions on hypergrids is designed. For the special case of the hypercube domain, it runs in time $O(n/\varepsilon)$ and works when $\varepsilon/\alpha = \Omega(n)$. Using the connection between distance approximation and erasure-resilient testing, our approximation algorithm implies an erasure-resilient tester that has a less stringent restriction on ε/α, specifically, $\Omega(\sqrt{n})$. For approximating the distance to k-juntas [12, 28], the best algorithm with additive error of ε makes $2^k \cdot \text{poly}(k, 1/\varepsilon)$ queries [28], and the best lower bound was $\Omega(k^2)$ for nonadaptive algorithms [46].

2 An Approximation Algorithm for Distance to Monotonicity

This section is devoted to proving Theorem 1.1. We provide a nonadaptive algorithm that gets a parameter $\alpha > 0$ and oracle access to a function $f\colon \{0,1\}^n \to \{0,1\}$ promised to be α-far from monotone, makes $\mathrm{poly}(n, 1/\alpha)$ queries, and returns an estimate $\hat{\varepsilon} > 0$ that satisfies, with probability at least 2/3,

$$\mathrm{dist}(f, \textsc{Mono}) \leq \hat{\varepsilon} \leq \tilde{O}(\sqrt{n}) \cdot \mathrm{dist}(f, \textsc{Mono}).$$

Our main algorithm, `ApproxMono`, whose performance is summarized in Lemma 2.1, distinguishes functions that are close to monotone from those that are far. Note that the distance from any Boolean function to monotonicity is at most 1/2, since the constant-0 and constant-1 functions are monotone. As a result, Theorem 1.1 follows directly from Lemma 2.1, by running the algorithm `ApproxMono` with ε set to $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots, \alpha$ appropriate number of times. (See, for example, [2, Section 3.3] for more details on how to get an approximation algorithm from a tolerant tester.).

Lemma 2.1. *There exists a nonadaptive algorithm,* `ApproxMono`*, that gets a parameter $\varepsilon \in (0, 1/2)$ and oracle access to a function $f\colon \{0,1\}^n \to \{0,1\}$, makes $\mathrm{poly}(n, 1/\varepsilon)$ queries and outputs "close" or "far" as follows:*

1. *If $\mathrm{dist}(f, \textsc{Mono}) \leq \frac{\varepsilon}{\sqrt{n}\cdot\mathrm{poly}(\log n)}$, it outputs "close" with probability at least 2/3.*
2. *If $\mathrm{dist}(f, \textsc{Mono}) \geq \varepsilon$, it outputs "far" with probability at least 2/3.*

The main algorithm, `ApproxMono`, is described in Figure 1. The algorithm uses subroutines `Edge-Violations` and `Matching-Estimation`, described in Figures 2 and 4, respectively. The subroutine `Edge-Violations`(δ, f) gets a parameter $\delta > 0$ and oracle access to a function $f\colon \{0,1\}^n \to \{0,1\}$, and returns an estimate to the fraction of decreasing edges of f up to an additive error δ. The second subroutine, `Matching-Estimation`(S, δ, f), gets a parameter $\delta > 0$, a subset $S \subseteq [n]$ and oracle access to a function $f\colon \{0,1\}^n \to \{0,1\}$. The goal of `Matching-Estimation`$(S, \delta, f)$ is to estimate the probability, over $\boldsymbol{x} \sim \{0,1\}^n$, of an event (which we denote `Capture` and describe in Definition 2.3) defined with respect to $\boldsymbol{x}$, S and f up to an additive error δ. The high level intuition is that, as long as the estimates of `Matching-Estimation`(S, δ, f) and `Edge-Violations`(δ, f) are correct, we can certify a lower bound on the distance to monotonicity of f. We then prove that if f is ε-far from monotone, either the number of decreasing edges of f is large, (and thus line 2 declares "far"), or the `Matching-Estimation` subroutine can verify a lower bound on the distance to monotonicity.

Recall that an edge (x, y) in a hypercube is a pair of points $x, y \in \{0,1\}^n$ with $x_i = 0$ and $y_i = 1$ for some $i \in [n]$, and $x_j = y_j$ for all $j \in ([n] \setminus \{i\})$. For a function $f\colon \{0,1\}^n \to \{0,1\}$, an edge (x, y) is *decreasing* if $f(x) > f(y)$, i.e., $f(x) = 1$ and $f(y) = 0$. For a dimension $i \in [n]$, a point $x \in \{0,1\}^n$, and a bit $b \in \{0,1\}$, we use $x^{(i \to b)}$ to denote the point in $\{0,1\}^n$ whose i^{th} coordinate is b and the remaining coordinates are the same as in x. We use $x^{(i)}$ to denote the point $x^{(i \to (1-x_i))}$, where x_i is the i^{th} coordinate in x.

We summarize the properties of the subroutine `Edge-Violations`(δ, f) in Fact 2.2. It can be easily proved by an application of the Chernoff bound.

Fact 2.2. *The algorithm* `Edge-Violations` *is nonadaptive. It gets a parameter $\delta > 0$ and oracle access to a function $f\colon \{0,1\}^n \to \{0,1\}$, makes $(10 \log n)/\delta^2$ queries, and outputs $\hat{\gamma} \in [0,1]$ which, with probability at least $1 - 1/n^3$, satisfies*

$$\left| \Pr_{\substack{\boldsymbol{x} \sim \{0,1\}^n \\ \boldsymbol{i} \sim [n]}} \left[f(\boldsymbol{x}^{(\boldsymbol{i} \to 0)}) > f(\boldsymbol{x}^{(\boldsymbol{i} \to 1)}) \right] - \hat{\gamma} \right| \leq \delta.$$

Subroutine `ApproxMono` (ε, f)

Input: A parameter $\varepsilon \in (0, 1/2)$ and oracle access to a function $f\colon \{0,1\}^n \to \{0,1\}$.
Output: Either "close" or "far".

1. Let $\hat{\gamma} \leftarrow$ `Edge-Violations`$(\varepsilon/(2\sqrt{n}), f)$ be an estimate to the fraction of decreasing edges up to an additive error $\varepsilon/(2\sqrt{n})$.
2. If $\hat{\gamma} \geq 3\varepsilon/(2\sqrt{n})$, output "far".
3. For each $d = 2^h$, where $h \in \{0, 1, \dots, \log_2 n\}$, repeat the following $t = \sqrt{n} \cdot \mathrm{poly}(\log n)/\varepsilon$ times:
   (a) Sample $\boldsymbol{S} \subseteq [n]$ by including each $i \in [n]$ independently with probability $1/d$.
   (b) Let $\hat{\xi} \leftarrow$ `Matching-Estimation`$(\boldsymbol{S}, 1/(4t), f)$ be an estimate to
   $$\Pr_{\boldsymbol{x} \sim \{0,1\}^n} [\texttt{Capture}(\boldsymbol{x}, \boldsymbol{S}, f) = 1]$$
   up to an additive error $1/(4t)$.
   (c) If $\hat{\xi} \geq 3/(4t)$, output "far".
4. If the procedure has not yet produced an output, output "close".

Figure 1: Description of the `ApproxMono` subroutine.

Subroutine `Edge-Violations` (δ, f)

Input: A parameter $\delta > 0$ and oracle access to a function $f\colon \{0,1\}^n \to \{0,1\}$.
Output: A real number $\hat{\gamma} \in [0, 1]$.

1. Initialize counter $c \leftarrow 0$.
2. Set $t \leftarrow \left\lceil \frac{10 \log n}{\delta^2} \right\rceil$.
3. Repeat the following steps t times:
   (a) Sample $\boldsymbol{x} \sim \{0,1\}^n$ and $\boldsymbol{i} \sim [n]$, both uniformly at random, and query $f(\boldsymbol{x}^{(\boldsymbol{i} \to 0)})$ and $f(\boldsymbol{x}^{(\boldsymbol{i} \to 1)})$.
   (b) If $f(\boldsymbol{x}^{(\boldsymbol{i} \to 0)}) > f(\boldsymbol{x}^{(\boldsymbol{i} \to 1)})$, i.e., the edge $(\boldsymbol{x}^{(\boldsymbol{i} \to 0)}, \boldsymbol{x}^{(\boldsymbol{i} \to 1)})$ is decreasing, update $c \leftarrow c + 1$.
4. Output $\hat{\gamma} = c/t$.

Figure 2: Description of the `Edge-Violations` subroutine.

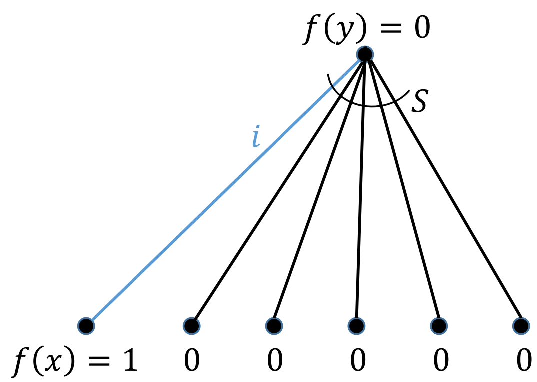

Figure 3: An illustration to Definition 2.3.

Definition 2.3. *For a function $f\colon \{0,1\}^n \to \{0,1\}$, a subset $S \subseteq [n]$, and a point $x \in \{0,1\}^n$, let* $\texttt{Capture}(x, S, f) \in \{0,1\}$ *be the indicator of the following event (see Figure 3):*

1. *There exists an index $i \in S$ such that (x, y) is a decreasing edge in f, where $y = x^{(i)}$.*
2. *For all $j \in S \setminus \{i\}$, the edge $(y, y^{(j)})$ is a nondecreasing edge in f.*

Given Definition 2.3, we summarize the properties of subroutine $\texttt{Matching-Estimation}(S, \delta, f)$ in Fact 2.4. As Fact 2.2, it can be easily proved by an application of the Chernoff bound.

Fact 2.4. *The algorithm* $\texttt{Matching-Estimation}$ *is nonadaptive. It gets a set $S \subseteq [n]$, a parameter $\delta > 0$ and oracle access to a function $f\colon \{0,1\}^n \to \{0,1\}$, makes $O(|S|^2 \log(n/\varepsilon)/\delta^2)$ queries, and outputs $\hat{\xi} \in [0,1]$ which, with probability at least $1 - (\varepsilon/n)^3$, satisfies*

$$\left| \Pr_{\boldsymbol{x} \sim \{0,1\}^n} [\texttt{Capture}(\boldsymbol{x}, S, f) = 1] - \hat{\xi} \right| \leq \delta.$$

Lemma 2.1 follows from Lemmas 2.5 and 2.6.

Lemma 2.5. *For a function $f\colon \{0,1\}^n \to \{0,1\}$ and a set $S \subseteq [n]$,*

$$\Pr_{\boldsymbol{x} \sim \{0,1\}^n} [\texttt{Capture}(\boldsymbol{x}, S, f) = 1] \leq 2 \cdot \mathrm{dist}(f, \textsc{Mono}).$$

Proof. Let $X = \{x \in \{0,1\}^n : \texttt{Capture}(x, S, f) = 1\}$. For each $x \in X$, let $y_x = x^{(i)}$ for a dimension $i \in S$ be the point for which (x, y_x) is decreasing and, for all $j \in S \setminus \{i\}$, the edge $(y_x, y_x^{(j)})$ is nondecreasing. Consider the set of decreasing edges of f given by $E_X = \{\{x, y_x\} : x \in X\}$. If x_1, x_2 from X are distinct, then $y_{x_1} \neq y_{x_2}$, because otherwise y_{x_1} would violate Item 2 in Definition 2.3. Thus, E_X is a matching. Each edge is added to E_X at most twice (once for each endpoint), so $|E_X| \geq |X|/2$. Since we have a matching of at least $|X|/2$ decreasing edges, at least $|X|/2$ values of f must be changed to make it monotone. □

Lemma 2.6. *Let $f\colon \{0,1\}^n \to \{0,1\}$ be ε-far from monotone, with fewer than $\varepsilon\sqrt{n} \cdot 2^n$ decreasing edges. Then, for some $d = 2^h$ where $h \in \{0, \ldots, \log_2 n\}$,*

$$\mathop{\mathbb{E}}_{\substack{\boldsymbol{S} \subseteq [n] \\ i \in \boldsymbol{S} \text{ w.p. } 1/d}} \left[\Pr_{\boldsymbol{x} \sim \{0,1\}^n} [\texttt{Capture}(\boldsymbol{x}, \boldsymbol{S}, f) = 1] \right] \geq \frac{\varepsilon}{\sqrt{n} \cdot \mathrm{poly}(\log n)}.$$

Subroutine `Matching-Estimation`(S, δ, f)

Input: A set $S \subseteq [n]$, a parameter $\delta > 0$, and oracle access to a function $f\colon \{0,1\}^n \to \{0,1\}$.
Output: A real number $\hat{\xi} \in [0,1]$.

1. Initialize counter $c \leftarrow 0$.
2. Set $t \leftarrow \left\lceil \frac{10 \log(n/\varepsilon)}{\delta^2} \right\rceil$.
3. Repeat the following steps t times:
   (a) Sample $\boldsymbol{x} \sim \{0,1\}^n$ and query $f(\boldsymbol{x})$; for all $i \in S$, let $\boldsymbol{y}_i = \boldsymbol{x}^{(i)}$ and query $f(\boldsymbol{y}_i)$ and $f(\boldsymbol{y}_i^{(j)})$ for all $j \in S \setminus \{i\}$.
   (b) If, for some $i \in S$, the edge $(\boldsymbol{x}, \boldsymbol{y}_i)$ is decreasing and, for all $j \in S \setminus \{i\}$, the edge $(\boldsymbol{y}_i, \boldsymbol{y}_i^{(j)})$ is nondecreasing, update $c \leftarrow c + 1$.
4. Output $\hat{\xi} = c/t$.

Figure 4: Description of the `Matching-Estimation` subroutine.

The proof of Lemma 2.6 appears in Section 2.1. We use Lemmas 2.5 and 2.6 to complete the proof of Lemma 2.1.

Proof of Lemma 2.1. By a union bound over the invocation of `Edge-Violations`$(\varepsilon/(2\sqrt{n}), f)$ and at most $t(\log_2 n + 1) \leq n/\varepsilon$ invocations of `Matching-Estimation`$(\boldsymbol{S}, 1/(2t), f)$, we get that, with probability at least $3/4$, all outputs produced by these subroutines satisfy the conclusions of Facts 2.2 and 2.4.

First, we prove the contrapositive of part 1 of Lemma 2.1. Suppose that `ApproxMono`(ε, f) outputs "far". Since the total number of edges in the hypercube $\{0,1\}^n$ is $n2^{n-1}$, if the output was produced by line 2, the number of decreasing edges in f is at least $(\varepsilon/\sqrt{n}) \cdot n2^{n-1} = (\varepsilon\sqrt{n}/2) \cdot 2^n$. The number of decreasing edges in f divided by $n2^n$ is a lower bound on the distance to monotonicity[3]. Hence, $\mathrm{dist}(f, \textsc{Mono}) \geq \varepsilon/(2\sqrt{n})$. Otherwise, `ApproxMono`$(\varepsilon, f)$ outputs "far" in line 3(c), but then Lemma 2.5 implies $\mathrm{dist}(f, \textsc{Mono}) \geq 1/(4t) = \varepsilon/(\sqrt{n} \cdot \mathrm{poly}(\log n))$, completing the proof of part 1.

Next, we prove part 2 of the Lemma 2.1. Suppose that $f\colon \{0,1\}^n \to \{0,1\}$ is ε-far from monotone. If the number of decreasing edges in f is at least $\varepsilon\sqrt{n} \cdot 2^n$, then line 2 outputs "far". Otherwise, by Lemma 2.6, there exists some $d = 2^h$ with $h \in \{0, \ldots, \log_2 n\}$ for which,

$$\mathop{\mathbb{E}}_{\boldsymbol{S} \subseteq [n]} \left[\Pr_{\boldsymbol{x} \sim \{0,1\}^n} [\texttt{Capture}(\boldsymbol{x}, \boldsymbol{S}, f) = 1] \right] \geq \frac{\varepsilon}{\sqrt{n} \cdot \mathrm{poly}(\log n)} \stackrel{\text{def}}{=} \mu, \tag{1}$$

where $\boldsymbol{S} \subseteq [n]$ is sampled by including each $i \in [n]$ independently with probability $1/d$. Let $\beta \in (0,1)$ be the probability over the draw of $\boldsymbol{S} \subseteq [n]$ that $\Pr_{\boldsymbol{x} \sim \{0,1\}^n} [\texttt{Capture}(\boldsymbol{x}, \boldsymbol{S}, f) = 1] \geq \mu/2$.

[3] In order to see this, suppose that f has m decreasing edges. Consider any monotone function $g\colon \{0,1\}^n \to \{0,1\}$. Note that each point in $\{0,1\}^n$ is incident on n edges of the hypercube. Hence, if f and g differ on less than m/n points, then there are less than m edges for which f and g differ on at least one endpoint. Since f has m decreasing edges, there exists a decreasing edge in f for which g and f agree on both endpoints of the edge, contradicting the fact that g is monotone.

Then,

$$\mathop{\mathbb{E}}_{\boldsymbol{S}\subseteq[n]}\left[\Pr_{\boldsymbol{x}\sim\{0,1\}^n}\left[\texttt{Capture}(\boldsymbol{x},\boldsymbol{S},f)=1\right]\right]\le\beta+(1-\beta)\cdot\frac{\mu}{2}. \tag{2}$$

Using (1) and (2), we get $\beta \ge \mu/2$. Since $t = \sqrt{n}\cdot\mathrm{poly}(\log n)/\varepsilon$ is high enough so that $t \ge (2/\mu)\cdot\log n$, we have that, with probability at least $1-1/n$, there exists some $\boldsymbol{S}\subseteq[n]$ sampled in line 3(a) of $\texttt{ApproxMono}(\varepsilon, f)$ such that $\Pr_{\boldsymbol{x}\sim\{0,1\}^n}[\texttt{Capture}(\boldsymbol{x},\boldsymbol{S},f)=1]\ge\mu/2\ge 1/t$. When this occurs, line 3(c) outputs "far". □

2.1 Proof of Lemma 2.6

To prove Lemma 2.6, we use the main (robust) directed isoperimetric inequality of [42] as the starting point. We use notation from [42]. For a function $f\colon\{0,1\}^n\to\{0,1\}$, let $\mathcal{S}_f^-$ denote the set of decreasing edges of f. Let the function $I_f^-\colon\{0,1\}^n\to\{0,1,\dots,n\}$ map each point $x\in\{0,1\}^n$ to the number of decreasing edges of f incident on x. For an arbitrary coloring of $\mathcal{S}_f^-$ into red and blue edges, $\mathrm{col}\colon\mathcal{S}_f^-\to\{\mathrm{red},\mathrm{blue}\}$, let $I_{f,\mathrm{red}}^-, I_{f,\mathrm{blue}}^-\colon\{0,1\}^n\to\{0,\dots,n\}$ be the functions given by:

$$I_{f,\mathrm{red}}^-(x)=\begin{cases}0 & \text{if } f(x)=0;\\ |\{\{x,y\}\in\mathcal{S}_f^-:\mathrm{col}(x,y)=\mathrm{red}\}| & \text{if } f(x)=1;\end{cases}$$
$$I_{f,\mathrm{blue}}^-(x)=\begin{cases}|\{\{x,y\}\in\mathcal{S}_f^-:\mathrm{col}(x,y)=\mathrm{blue}\}| & \text{if } f(x)=0;\\ 0 & \text{if } f(x)=1.\end{cases}$$

We crucially rely on the main theorem of [42], which is stated next, with a minor improvement in the bound. We obtain the improvement in Appendix A.

Theorem 2.7 (Close to Theorem 1.9 from [42])**.** *Let $f\colon\{0,1\}^n\to\{0,1\}$ be ε-far from monotone. Then, for any coloring of $\mathcal{S}_f^-$ into* red *and* blue*,*

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\{0,1\}^n}\left[\sqrt{I_{f,\mathrm{red}}^-(\boldsymbol{x})}\right]+\mathop{\mathbb{E}}_{\boldsymbol{y}\sim\{0,1\}^n}\left[\sqrt{I_{f,\mathrm{blue}}^-(\boldsymbol{y})}\right]\ge\Omega(\varepsilon). \tag{3}$$

To prove Lemma 2.6, consider a function $f\colon\{0,1\}^n\to\{0,1\}$ which is ε-far from monotone with $|\mathcal{S}_f^-|<\varepsilon\sqrt{n}\cdot 2^n$. Consider the coloring of $\mathcal{S}_f^-$ given by:

$$\mathrm{col}(x,y)=\begin{cases}\mathrm{red} & \text{if } f(x)=1 \text{ and } I_f^-(x)\ge I_f^-(y);\\ \mathrm{blue} & \text{if } f(x)=1 \text{ and } I_f^-(x)<I_f^-(y).\end{cases}$$

In this coloring, each decreasing edge in f is counted in (3) towards its endpoint adjacent to a higher number of decreasing edges. For each $d=2^h$, where $h\in\{0,\dots,\lfloor\log_2 n\rfloor\}$, define the subsets of points

$$H_{d,\mathrm{blue}}=\{x\in\{0,1\}^n : d\le I_f^-(x)<2d \text{ and } f(x)=0\},$$
$$H_{d,\mathrm{red}}=\{x\in\{0,1\}^n : d\le I_f^-(x)<2d \text{ and } f(x)=1\}.$$

The sets $(H_{d,\mathrm{red}}, H_{d,\mathrm{blue}} : d=2^h, h\in\{0,\dots,\lfloor\log_2 n\rfloor\})$ partition the endpoints of decreasing edges. By (3), there exist $d^*=2^{h^*}$, for some $h^*\in\{0,\dots,\lfloor\log_2 n\rfloor\}$, and a color $b^*\in\{\mathrm{red},\mathrm{blue}\}$ such that

$$\frac{1}{2^n}\sum_{x\in H_{d^*,b^*}}\sqrt{I_{f,b^*}^-(x)}\ge\Omega\left(\frac{\varepsilon}{\log n}\right). \tag{4}$$

Fix such d^* and b^*. Let

$$H'_{d^*,b^*} = \{x \in H_{d^*,b^*} : I^-_{f,b^*}(x) > 0\}$$

be the subset of points in H_{d^*,b^*} which are endpoints of decreasing edges colored b^*. Note that (4) still holds if the summation is changed to be over H'_{d^*,b^*} instead of H_{d^*,b^*}. We further partition H'_{d^*,b^*} into $\log_2 d^* + 1$ sets, $(H_{d^*,b^*,s} : s = 2^q, q \in \{0, \ldots, \log_2 d^*\})$, where $H_{d^*,b^*,s} = \{x \in H_{d^*,b^*} : s \leq I^-_{f,b^*}(x) < 2s\}$. By (4) and an argument similar to the one used in deriving (4), there exists some $s^* = 2^{q^*}$ for some $q^* \in \{0, \ldots, \log_2 d^*\}$ satisfying

$$\frac{|H_{d^*,b^*,s^*}|}{2^n} \cdot \sqrt{s^*} \geq \Omega\left(\frac{\varepsilon}{\log n \cdot \log d^*}\right) \geq \Omega\left(\frac{\varepsilon}{\log^2 n}\right), \tag{5}$$

where we used the fact that $d^* \leq n$. Each $x \in H_{d^*,b^*,s^*}$ is an endpoint of at least d^* decreasing edges of f. Moreover, the sets of decreasing edges incident on different points in H_{d^*,b^*,s^*} are disjoint. Consequently, by the bound on the number of decreasing edges in the statement of Lemma 2.6,

$$d^*|H_{d^*,b^*,s^*}| \leq |\mathcal{S}^-_f| < \varepsilon\sqrt{n} \cdot 2^n,$$

implying $|H_{d^*,b^*,s^*}|/2^n < \varepsilon\sqrt{n}/d^*$. Together with (5), this gives

$$\frac{\varepsilon\sqrt{n}}{d^*} \cdot \sqrt{s^*} \geq \Omega\left(\frac{\varepsilon}{\log^2 n}\right);$$
$$\Rightarrow \frac{\sqrt{s^*}}{d^*} \geq \Omega\left(\frac{1}{\sqrt{n} \cdot \log^2 n}\right). \tag{6}$$

Next, we show that for each $x \in H_{d^*,b^*,s^*}$, the probability that `Capture` happens is sufficiently large.

Claim 2.8. *For each $x \in H_{d^*,b^*,s^*}$, the probability*

$$\Pr_{\substack{\boldsymbol{S} \subseteq [n] \\ i \in \boldsymbol{S} \text{ w.p. } 1/d^*}} [\mathtt{Capture}(x, \boldsymbol{S}, f) = 1] = \Omega\left(\frac{s^*}{d^*}\right).$$

Proof. Consider the case when $d^* = 1$. Then $s^* = 1$. Fix an arbitrary $x \in H_{d^*,b^*,s^*}$. Then

$$I^-_f(x) = I^-_{f,b^*}(x) = 1,$$

that is, the only edge incident on x is colored b^*. Call this edge $\{x, y\}$. Since $\mathrm{col}(x, y) = b^*$, by definition of coloring, $I^-_f(y) \leq I^-_f(x) = 1$. Therefore, x and y are not endpoints of any decreasing edges other than the edge $\{x, y\}$. Note that $\boldsymbol{S} = [n]$, since each $i \in [n]$ is in $\boldsymbol{S}$ with probability $1/d^* = 1$. By Definition 2.3, $\mathtt{Capture}(x, \boldsymbol{S}, f) = 1$ since $\{x, y\}$ is a decreasing edge along a dimension in $\boldsymbol{S}$, and all other edges incident on y are nondecreasing. Hence,

$$\Pr_{\boldsymbol{S}=[n]}[\mathtt{Capture}(x, \boldsymbol{S}, f) = 1] = 1 = \Omega\left(\frac{s^*}{d^*}\right),$$

concluding the proof for the case $d^* = 1$.

Now, consider the case when $d^* \geq 2$. For $x \in \{0,1\}^n$, let $D^-_f(x) = \{i \in [n] : \{x, x^{(i)}\} \in \mathcal{S}^-_f\}$ denote the set of dimensions along which the edge incident on x is decreasing in f, and let

$E_f^-(x) = \{i \in D_f^-(x) : I_f^-(x) \geq I_f^-(x^{(i)})\}$ be the set of dimensions along which the other endpoint is adjacent to no more decreasing edges than x. For each $x \in H_{d^*,b^*,s^*}$, we have $|D_f^-(x)| < 2d^*$ and $|E_f^-(x)| \geq s^*$. If we sample $\boldsymbol{S} \subseteq [n]$ by including each index $i \in [n]$ independently with probability $1/d^*$, then, for each $x \in H_{d^*,b^*,s^*}$, the probability that $\texttt{Capture}(x, \boldsymbol{S}, f) = 1$ is at least the probability that there exists a unique $i \in \boldsymbol{S}$ such that $y = x^{(i)}$ satisfies $\{x, y\} \in \mathcal{S}_f^-$ with $I_f^-(x) \geq I_f^-(y)$, and all other decreasing edges of f incident on y are along dimensions in $[n] \setminus \boldsymbol{S}$. Hence, for each $x \in H_{d^*,b^*,s^*}$, the probability

$$\begin{aligned}
\Pr_{\boldsymbol{S}\subseteq[n]}[\texttt{Capture}(x, \boldsymbol{S}, f) = 1] &\geq \sum_{i \in E_f^-(x)} \Bigg(\Pr[i \in \boldsymbol{S}] \cdot \prod_{\substack{j \in (D_f^-(x) \cup \\ D_f^-(x^{(i)}))\setminus\{i\}}} \Pr[j \notin \boldsymbol{S}]\Bigg) \\
&\geq s^* \cdot \frac{1}{d^*} \cdot \left(1 - \frac{1}{d^*}\right)^{4d^*} \\
&= \Omega\left(\frac{s^*}{d^*}\right),
\end{aligned}$$

where we used $|E_f^-(x)| \geq s^*$ and

$$|D_f^-(x^{(i)})| \leq I_f^-(x) = |D_f^-(x)| < 2d^*$$

to get the second inequality and $(1 - 1/d^*)^{d^*} \geq 1/4$ for all $d^* \geq 2$ to get the final equality. □

This concludes the proof of Lemma 2.6, since

$$\begin{aligned}
\mathop{\mathbb{E}}_{\substack{\boldsymbol{S}\subseteq[n] \\ i\in\boldsymbol{S} \text{ w.p. } 1/d^*}} \left[\Pr_{\boldsymbol{x}\sim\{0,1\}^n}[\texttt{Capture}(\boldsymbol{x}, \boldsymbol{S}, f) = 1]\right] &\geq \frac{1}{2^n} \sum_{x \in H_{d^*,b^*,s^*}} \Pr_{\substack{\boldsymbol{S}\subseteq[n] \\ i\in\boldsymbol{S} \text{ w.p. } 1/d^*}}[\texttt{Capture}(x, \boldsymbol{S}, f) = 1] \\
&\geq \frac{|H_{d^*,b^*,s^*}|}{2^n} \cdot \Omega\left(\frac{s^*}{d^*}\right) \\
&\geq \Omega\left(\frac{\varepsilon}{\log^2 n} \cdot \frac{\sqrt{s^*}}{d^*}\right) \\
&\geq \Omega\left(\frac{\varepsilon}{\sqrt{n} \cdot \log^4 n}\right),
\end{aligned}$$

where we used Claim 2.8, (5) and (6) to get the second, third and fourth inequalities, respectively.

3 A Nonadaptive Lower Bound for Erasure-Resilient Testers

In this section, we prove Theorem 1.4 that gives a lower bound on the query complexity of erasure-resilient testers of monotonicity, unateness and the k-junta property. We prove the lower bound by constructing two distributions $\mathcal{D}^+$ and $\mathcal{D}^-$ on input functions $f\colon \{0,1\}^n \to \{0, 1, \perp\}$ that are hard to distinguish for any nonadaptive tester and then applying Yao's Minimax principle [59].

Recall that $\perp$ denotes an erased function value. We say that a function $f\colon \{0,1\}^n \to \{0, 1, \perp\}$ is *α-erased* if at most an α fraction of its values are erased. If α is not specified, we call such a function *partially erased*. A *completion* of a partially erased function $f\colon \{0,1\}^n \to \{0, 1, \perp\}$ is a function $f' \colon \{0,1\}^n \to \{0,1\}$ that agrees with f on all nonerased values, that is, for all $x \in \{0,1\}^n$, if $f(x) \neq \perp$ then $f'(x) = f(x)$. A partially erased function f is monotone (or, more generally, has property $\mathcal{P}$) if there exists a monotone completion of f (respectively, a completion of f that has $\mathcal{P}$). A partially erased function is ε-far from monotone (or, more generally, from having property $\mathcal{P}$) if every completion of f is ε-far from monotone (respectively, from having property $\mathcal{P}$).

Proof of Theorem 1.4. We start by defining distributions $\mathcal{D}^+$ and $\mathcal{D}^-$ on α-erased functions. Later, we show that $\mathcal{D}^+$ is over monotone functions whereas $\mathcal{D}^-$ is over functions that are ε-far from monotone. Interestingly, the same distributions work to prove our lower bounds for unateness and k-juntas: all functions in the support of $\mathcal{D}^+$ are unate (because they are monotone) and also $n/2$-juntas. We will also show that all functions in the support of $\mathcal{D}^-$ are ε-far from unate and ε-far from $n/2$-juntas. The core of the argument is demonstrating that the two distributions are hard to distinguish for nonadaptive testers that make too few queries.

For every $x \in \{0,1\}^n$, let $|x|$ denote the Hamming weight of x, and let x_S denote the vector $x \in \{0,1\}^n$ restricted to the dimensions in the set $S \subseteq [n]$.

Let n be a multiple of 4. We first describe a collection of random variables used for defining the distributions $\mathcal{D}^+$ and $\mathcal{D}^-$.

- **The set $\boldsymbol{M}$ of *control* dimensions.** The set $\boldsymbol{M}$ is a uniformly random subset of $[n]$ of size $n/2$. We use $\overline{\boldsymbol{M}}$ to denote the set of remaining dimensions, $[n] \setminus \boldsymbol{M}$.
- **The subcube partition set $\boldsymbol{P}_M$ and *action* subcubes**. For a fixed set $M \subset [n]$ of size $n/2$, let $\{0,1\}^M$ denote the restriction of $\{0,1\}^n$ to the dimensions in M. Let the set $\Psi_M = \left\{x_M \in \{0,1\}^M : |x_M| = \frac{n}{4}\right\}$ denote the set of all "prefixes" of x which lie in the middle layer of the subcube $\{0,1\}^M$. The subcube partition set $\boldsymbol{P}_M$ is a uniformly random subset of Ψ_M of size $|\Psi_M|/2$. Each $z \in \Psi_M$ corresponds to a subcube of the form $\{0,1\}^{\overline{M}}$ with the vertex set comprised of points x with $x_M = z$. All such subcubes are called *action* subcubes.
- **The functions g_{M,P_M}.** For a fixed setting of $M \subset [n]$ of size $n/2$ and a set of action subcubes $P_M \subset \Psi_M$, the function $g_{M,P_M} \colon \{0,1\}^n \to \{0,1,\perp,0^\star,1^\star\}$:

$$g_{M,P_M}(x) = \begin{cases} 0 & \text{if } |x_M| < \frac{n}{4}; \\ 1 & \text{if } |x_M| > \frac{n}{4}; \\ \perp & \text{if } |x_M| = \frac{n}{4} \text{ and } |x_{\overline{M}}| \in \left[\frac{n}{4} - n^\kappa, \frac{n}{4} + n^\kappa\right]; \\ 0^\star & \text{if } x_M \in P_M \text{ and } |x_{\overline{M}}| \notin \left[\frac{n}{4} - n^\kappa, \frac{n}{4} + n^\kappa\right]; \\ 1^\star & \text{otherwise.} \end{cases}$$

Functions from $\mathcal{D}^+$ and $\mathcal{D}^-$ are sampled by first letting $\boldsymbol{M} \subset [n]$ be a random set of control dimensions and then letting $\boldsymbol{P_M}$ be a random subcube partition set. A function $\boldsymbol{f}$ sampled from $\mathcal{D}^+$ and $\mathcal{D}^-$ will be identical to $g_{\boldsymbol{M},\boldsymbol{P_M}}$ on points $x \in \{0,1\}^n$ for which $g_{\boldsymbol{M},\boldsymbol{P_M}}(x) \in \{0,1,\perp\}$, but differ on the remaining values (see Figures 5 and 6). Specifically, for functions in $\mathcal{D}^+$, the values $0^\star$ and $1^\star$ are replaced with 0 and 1, respectively. That is, $\boldsymbol{f} \sim \mathcal{D}^+$ is defined by sampling $\boldsymbol{M}$ and $\boldsymbol{P_M}$, and letting:

$$\boldsymbol{f}(x) = \begin{cases} g_{\boldsymbol{M},\boldsymbol{P_M}}(x) & \text{if } g_{\boldsymbol{M},\boldsymbol{P_M}}(x) \in \{0,1,\perp\}; \\ 0 & \text{if } g_{\boldsymbol{M},\boldsymbol{P_M}}(x) = 0^\star; \\ 1 & \text{if } g_{\boldsymbol{M},\boldsymbol{P_M}}(x) = 1^\star. \end{cases}$$

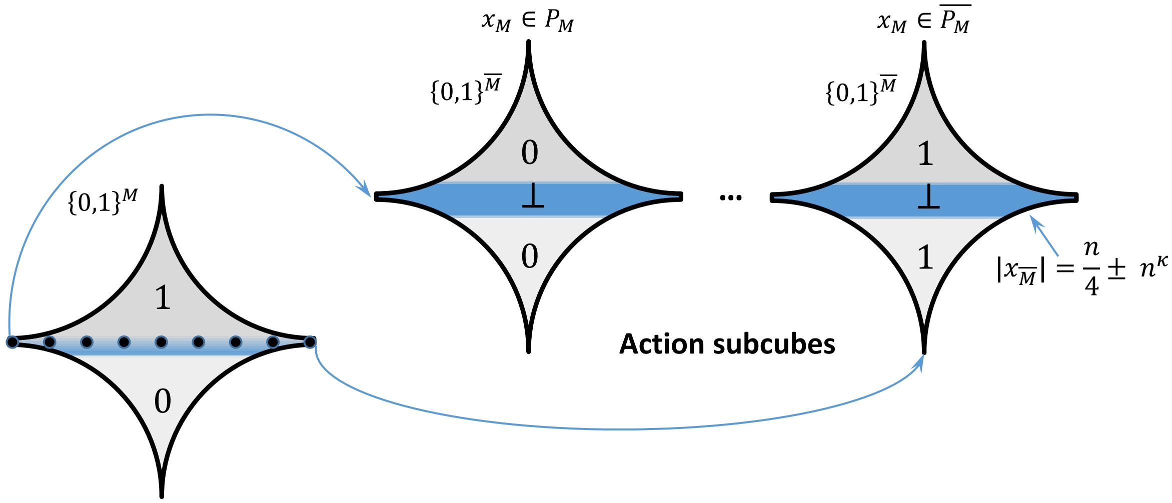

Figure 5: Functions $\boldsymbol{f} \sim \mathcal{D}^+$ defined with respect to control dimensions M and the subcube partition set P_M.

For functions in $\mathcal{D}^-$, the value $0^\star$ is replaced with the majority function, denoted $\mathrm{Maj}(\cdot)$, evaluated on the bits indexed by $\overline{M}$, whereas $1^\star$ is replaced with the antimajority of those bits. That is, $\boldsymbol{f} \sim \mathcal{D}^-$ is defined by sampling $\boldsymbol{M}$ and $\boldsymbol{P_M}$, and letting:

$$\boldsymbol{f}(x) = \begin{cases} g_{\boldsymbol{M},\boldsymbol{P_M}}(x) & \text{if } g_{\boldsymbol{M},\boldsymbol{P_M}}(x) \in \{0,1,\perp\}; \\ \mathrm{Maj}(x_{\overline{\boldsymbol{M}}}) & \text{if } g_{\boldsymbol{M},\boldsymbol{P_M}}(x) = 0^\star; \\ 1 - \mathrm{Maj}(x_{\overline{\boldsymbol{M}}}) & \text{if } g_{\boldsymbol{M},\boldsymbol{P_M}}(x) = 1^\star. \end{cases}$$

Lemma 3.1. *There is an $\alpha = O(1/n^{1-\kappa})$, for which every function in the support of the distributions $\mathcal{D}^+$ and $\mathcal{D}^-$ is α-erased.*

Proof. A function $f\colon \{0,1\}^n \to \{0,1\}$ in the support of $\mathcal{D}^+$ (and $\mathcal{D}^-$) defined with respect to control dimensions M and subcube partition set P_M is erased in the middle $2n^\kappa + 1$ layers of every action subcube $\{0,1\}^{\overline{M}}$ whenever x_M is in the middle layer of the subcube $\{0,1\}^M$. Since $|M| = |\overline{M}| = n/2$, the number of points we erase is at most

$$\binom{n/2}{n/4} \cdot \binom{n/2}{n/4}(2n^\kappa + 1) = O\left(\left(\frac{2^{n/2}}{\sqrt{n/2}}\right)^2 n^\kappa\right) = O\left(\frac{2^n}{n^{1-\kappa}}\right).$$

Thus, the fraction of erasures in the constructed functions is $O(1/n^{1-\kappa})$. □

Lemma 3.2. *Every $\boldsymbol{f} \sim \mathcal{D}^+$ is monotone, unate, and $n/2$-junta whereas every $\boldsymbol{f} \sim \mathcal{D}^-$ has distance at least $\varepsilon = \Omega\left(\frac{1}{\sqrt{n}}\right)$ from monotonicity, unateness, and being an $n/2$-junta.*

Proof. Consider a partially erased function f in the support of $\mathcal{D}^+$. For all pairs x, y with $x \prec y$ for which the function values are not erased, $f(x) \leq f(y)$. Therefore, as shown in [36], f can be completed to a monotone function. Thus, f is monotone and, consequently, unate. Finally, we will

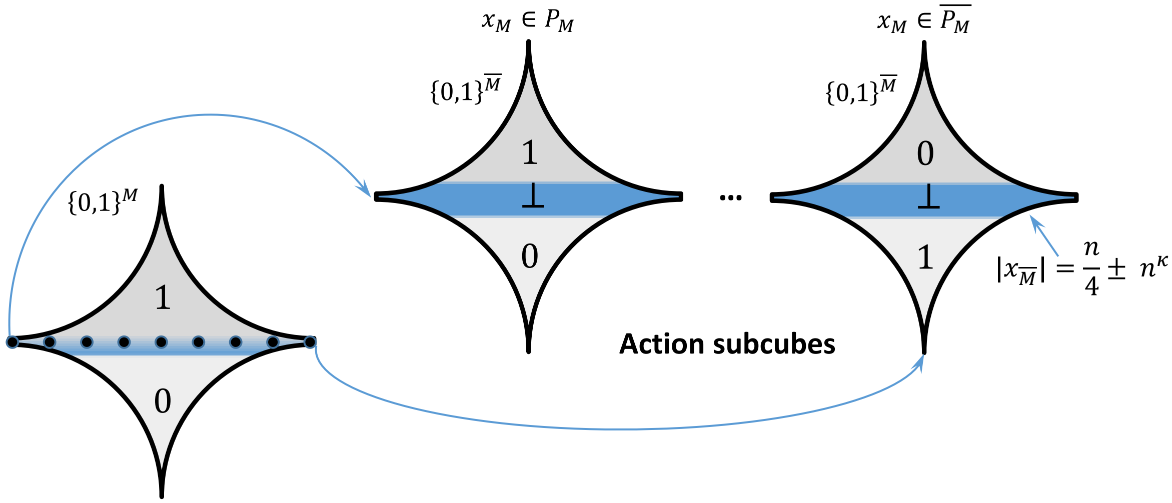

Figure 6: Functions $\boldsymbol{f} \sim \mathcal{D}^-$ defined with respect to control dimensions M and the subcube partition set P_M.

show that f can be completed to an $n/2$-junta. Let M and P_M be the set of control dimensions and the subcube partition set used in defining f, respectively. Define a completion $f' \colon \{0,1\}^n \to \{0,1\}$ of f as follows. For all $x \in \{0,1\}^n$ with $f(x) = \perp$,

$$f'(x) = \begin{cases} 0 & \text{if } x_M \in P_M, \\ 1 & \text{otherwise.} \end{cases}$$

Then f' only depends on coordinates in M. Hence, f can be completed to an $n/2$-junta.

Now consider a partially erased function f in the support of $\mathcal{D}^-$. Let M and P_M be the set of control dimensions and the subcube partition set used in defining f, respectively. By standard arguments (see, e.g., [36, Lemma 22]), in each action subcube $\{0,1\}^{\overline{M}}$, there is a perfect matching between points x on which $f(x) = 1$ and points y on which $f(y) = 0$, where each pair (x, y) in the matching has comparable x, y. Specifically, if $x_M \in P_M$ for this action subcube (that is, $g_{M,P_M}(x) = g_{M,P_M}(y) = 0^\star$), then $y \prec x$, and the function f is increasing on the pair (x, y) in the subset of dimensions of $\overline{M}$ on which x and y differ. If $x_M \notin P_M$ for this action subcube (that is, $g_{M,P_M}(x) = g_{M,P_M}(y) = 1^\star$), then $x \prec y$, and the function f is decreasing on the pair (x, y) in the dimensions on which x and y differ. This matching contains all nonerased points of the action subcube, and in half of the action subcubes at least half of the points in the matching need to be changed to make the function monotone. Since $\Theta(1/\sqrt{n})$ fraction of points participates in the action subcubes, the distance to monotonicity is $\Omega(1/\sqrt{n})$.

Moreover, we can pair up action subcubes in which g got assigned $0^\star$ values with those in which g got assigned $1^\star$ values. Consider the corresponding matchings for both action subcubes. Suppose (x, y) is a matched pair in an action subcube with $0^\star$ values, and (x', y') is the corresponding matched pair in the action subcube with $1^\star$ values, that is $x_{\overline{M}} = x'_{\overline{M}}$ and $y_{\overline{M}} = y'_{\overline{M}}$. Then the function f has to change on at least one of the four points x, x', y, y' to become unate, since a unate function has to be consistently either nondecreasing or nonincreasing in each dimension. Therefore, a constant fraction of all points participating in the action subcubes must be changed to make f unate. So, the distance from f to unateness is also $\Omega(1/\sqrt{n})$.

Finally, we prove that all functions f in the support of $\mathcal{D}^-$ are ε-far from being $n/2$-juntas for $\varepsilon = \Omega(\frac{1}{\sqrt{n}})$.

For a dimension $i \in [n]$, an edge (x, y) of a hypercube $\{0,1\}^n$ is called an *i-pair* if x and y differ in only their i-th bits, that is, $x_i \neq y_i$, but $x_j = y_j$ for all $j \in [n] \setminus \{i\}$. We say that a function f is independent of a variable $i \in [n]$ if $f(x) = f(y)$ for all i-pairs (x, y). Observe that f is an $n/2$-junta iff it is independent of $n/2$ variables.

Next, for each $i \in M$, we show that f is ε-far from being independent of i. Fix $i \in M$. We construct a large set $\mathcal{M}_i$ of nonconstant i-edges (x, y), that is, i-edges satisfying $f(x) \neq f(y)$. At least one of $f(x)$ and $f(y)$ for each such edge has to change to make f independent of i. Since $\mathcal{M}_i$ is a matching, $|\mathcal{M}_i|/2^n$ is a lower bound on the distance from f to functions that do not depend on variable i.

Recall that the set $\Psi_M = \{x_M \in \{0,1\}^M : |x_M| = \frac{n}{4}\}$, the set of all "prefixes" of x that lie in the middle layer of the subcube $\{0,1\}^M$. We define, for every dimension $i \in M$,

$$\mathcal{M}_i = \{(x, y) \mid (x, y) \text{ is an } i\text{-edge}, \ x_M \in \Psi_M, \text{ and } f(x) = x_i\}.$$

Note that $x_i \neq y_i$ and, by construction of functions g in the definition of $\mathcal{D}^-$, we have $f(y) = g(y) = y_i$. Therefore, $f(x) \neq f(y)$ for all i-edges $(x, y) \in \mathcal{M}_i$. For each $x_M \in \Psi_M$, more than 1/3 of the points x in the corresponding action subcube are assigned $f(x) = 0$, and the same holds for $f(x) = 1$. Since each action subcube has $2^{n/2}$ points, the size of $\mathcal{M}_i$ is at least $\frac{1}{3} \cdot 2^{n/2} \cdot |\Psi_M| = \frac{1}{3} \cdot 2^{n/2} \cdot \binom{n/2}{n/4} = \Omega(\frac{2^n}{\sqrt{n}})$. That is, the distance from f to being independent of variable i is at least ε, where $\varepsilon = \Omega(\frac{1}{\sqrt{n}})$.

Thus, if we change less than an ε fraction of values of f, we cannot eliminate the dependence on any of the $n/2$ variables in M. The only remaining possibility to make f an $n/2$-junta with fewer than $\varepsilon \cdot 2^n$ modifications is to eliminate the dependence on all variables in $\overline{M}$. This can happen only if the modified function becomes constant on all the action subcubes, which again requires changing at least $\frac{1}{3} \cdot 2^{n/2} \cdot |\Psi_M|$ values of f. Thus, f is ε-far from the set of $n/2$-juntas, where $\varepsilon = \Omega(\frac{1}{\sqrt{n}})$. □

Next, we show that the distributions $\mathcal{D}^+$ and $\mathcal{D}^-$ are hard to distinguish for nonadaptive testers. For two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ and a constant δ, let $\mathcal{D}_1 \approx_\delta \mathcal{D}_2$ denote that the statistical distance between $\mathcal{D}_1$ and $\mathcal{D}_2$ at most δ.

Consider a deterministic tester that makes q queries. Let $a_1 \dots a_q(f)$ be the answers to the queries on input f. Define $\mathcal{D}^+$-view to be the distribution on $a_1 \dots a_q(\boldsymbol{f})$ when $\boldsymbol{f} \sim \mathcal{D}^+$. Similarly, define $\mathcal{D}^-$-view. We use the version of Yao's principle stated in [53] that asserts that to prove a lower bound q on the worst-case query complexity of a randomized algorithm, it is enough to give two distributions $\mathcal{D}^+$ and $\mathcal{D}^-$, on positive and negative instances, respectively, for which the statistical distance between $\mathcal{D}^+$-view and $\mathcal{D}^-$-view is less than 1/3.

Lemma 3.3. *For a deterministic tester making $q \leq 2^{n^\kappa}$ queries, $\mathcal{D}^+$-view $\approx_{2/7}$ $\mathcal{D}^-$-view.*

Proof. The key point is the following: the only way a tester can distinguish the two distributions is by querying a pair of points $x, y \in \{0,1\}^n$ that fall in the same action subcube, but in different nonerased layers – one below erasures, the other above erasures. If it queries no such pair, then its view (that is, the distribution on the answers it receives) is identical for the two cases: $\boldsymbol{f} \sim \mathcal{D}^+$ and $\boldsymbol{f} \sim \mathcal{D}^-$. Observe that any such x and y must have weights that differ by at least $2n^\kappa + 2$. Consequently, x and y differ on at least $2n^\kappa + 2$ bits.

Let $T = \{i \in [n] \mid x_i \neq y_i\}$ denote the set of all coordinates on which the points x and y differ. Then $|T| \geq 2n^\kappa + 2$. Observe that $x_{\boldsymbol{M}} = y_{\boldsymbol{M}}$ iff $T \cap \boldsymbol{M} = \emptyset$. Since $\boldsymbol{M}$ is a uniformly random subset of $[n]$ of size $n/2$, the probability

$$\begin{aligned}
\Pr_{\boldsymbol{M}}[T \cap \boldsymbol{M} = \emptyset] = \frac{\binom{n-|T|}{n/2}}{\binom{n}{n/2}} &= \frac{\frac{(n-|T|)!}{(n/2)!\cdot(n/2-|T|)!}}{\frac{n!}{(n/2)!\cdot(n/2)!}} \\
&= \frac{(n/2)!}{(n/2-|T|)!} \cdot \frac{(n-|T|)!}{n!} \\
&= \frac{n/2 \cdot (n/2-1) \cdots (n/2-|T|+1)}{n \cdot (n-1) \cdots (n-|T|+1)} \\
&\leq 2^{-|T|}.
\end{aligned}$$

Let BAD be the event that one of the $\binom{q}{2}$ pairs of points the tester queries ends up in the same action subcube, on different sides of erasures, as discussed above. Then, by a union bound,

$$\Pr[\mathsf{BAD}] < \frac{q^2}{2} \cdot 2^{-|T|} \leq \frac{1}{2} \cdot 2^{2n^\kappa} \cdot 2^{-2n^\kappa - 2} = \frac{1}{8}.$$

By the discussion above, conditioned on BAD not occurring, the view of the tester is the same for both distributions:

$$\mathcal{D}^+\text{-view}|_{\overline{\mathsf{BAD}}} = \mathcal{D}^-\text{-view}|_{\overline{\mathsf{BAD}}}.$$

Conditioning on $\overline{\mathsf{BAD}}$ does not significantly change the view distributions. We use the following claim [53, Claim 4] to formalize this statement.

Claim 3.4 ([53])**.** *Let E be an event that happens with probability at least $\delta = 1 - 1/a$ under the distribution $\mathcal{D}$ and let $\mathcal{B}$ denote distribution $\mathcal{D}|_E$. Then $\mathcal{B} \approx_{\delta'} \mathcal{D}$ where $\delta' = 1/(a-1)$.*

Applying Claim 3.4 twice, we get

$$\mathcal{D}^+\text{-view} \approx_{1/7} \mathcal{D}^+\text{-view}|_{\overline{\mathsf{BAD}}} = \mathcal{D}^-\text{-view}|_{\overline{\mathsf{BAD}}} \approx_{1/7} \mathcal{D}^-\text{-view}.$$

This completes the proof of Lemma 3.3. □

Theorem 1.4 follows by Yao's Principle. □

Acknowledgments. We thank Deeparnab Chakrabarty and C. Seshadhri for useful discussions and, in particular, for mentioning an adaptive algorithm for approximating the distance to monotonicity up to a factor of $O(\sqrt{n})$.

References

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

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

[3] Roksana Baleshzar, Deeparnab Chakrabarty, Ramesh Krishnan S. Pallavoor, Sofya Raskhodnikova, and C. Seshadhri. Optimal unateness testers for real-valued functions: Adaptivity helps. In *Proceedings of International Colloquium on Automata, Languages and Processing (ICALP)*, pages 5:1–5:14, 2017.

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

[5] Aleksandrs Belovs and Eric Blais. A polynomial lower bound for testing monotonicity. In *Proceedings of ACM Symposium on Theory of Computing (STOC)*, pages 1021–1032, 2016.

[6] Piotr Berman, Meiram Murzabulatov, and Sofya Raskhodnikova. Tolerant testers of image properties. In *Proceedings of International Colloquium on Automata, Languages and Processing (ICALP)*, pages 90:1–90:14, 2016.

[7] Piotr Berman, Sofya Raskhodnikova, and Grigory Yaroslavtsev. L_p-testing. In *Proceedings of ACM Symposium on Theory of Computing (STOC)*, pages 164–173, 2014.

[8] Arnab Bhattacharyya, Elena Grigorescu, Kyomin Jung, Sofya Raskhodnikova, and David P. Woodruff. Transitive-closure spanners. *SIAM J. Comput.*, 41(6):1380–1425, 2012.

[9] Eric Blais. Improved bounds for testing juntas. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, pages 317–330, 2008.

[10] Eric Blais. Testing juntas nearly optimally. In *Proceedings of ACM Symposium on Theory of Computing (STOC)*, pages 151–158, 2009.

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

[12] Eric Blais, Clément L. Canonne, Talya Eden, Amit Levi, and Dana Ron. Tolerant junta testing and the connection to submodular optimization and function isomorphism. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2113–2132, 2018.

[13] Eric Blais, Sofya Raskhodnikova, and Grigory Yaroslavtsev. Lower bounds for testing properties of functions over hypergrid domains. In *IEEE 29th Conference on Computational Complexity, CCC*, pages 309–320, 2014.

[14] 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.

[15] Harry Buhrman, David García-Soriano, Arie Matsliah, and Ronald de Wolf. The non-adaptive query complexity of testing k-parities. *Chicago J. Theor. Comput. Sci.*, 2013, 2013.

[16] Andrea Campagna, Alan Guo, and Ronitt Rubinfeld. Local reconstructors and tolerant testers for connectivity and diameter. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, pages 411–424. 2013.

[17] Deeparnab Chakrabarty, Kashyap Dixit, Madhav Jha, and C. Seshadhri. Property testing on product distributions: Optimal testers for bounded derivative properties. *ACM Trans. Algorithms*, 13(2):20:1–20:30, 2017.

[18] Deeparnab Chakrabarty and C. Seshadhri. Optimal bounds for monotonicity and Lipschitz testing over hypercubes and hypergrids. In *Proceedings of ACM Symposium on Theory of Computing (STOC)*, pages 419–428, 2013.

[19] Deeparnab Chakrabarty and C. Seshadhri. An optimal lower bound for monotonicity testing over hypergrids. *Theory of Computing*, 10:453–464, 2014.

[20] Deeparnab Chakrabarty and C. Seshadhri. An $o(n)$ monotonicity tester for Boolean functions over the hypercube. *SIAM J. Comput.*, 45(2):461–472, 2016.

[21] Xi Chen, Anindya De, Rocco A. Servedio, and Li-Yang Tan. Boolean function monotonicity testing requires (almost) $n^{1/2}$ non-adaptive queries. In *Proceedings of ACM Symposium on Theory of Computing (STOC)*, pages 519–528, 2015.

[22] Xi Chen, Rocco A. Servedio, and Li-Yang Tan. New algorithms and lower bounds for monotonicity testing. In *Proceedings of IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 286–295, 2014.

[23] Xi Chen, Rocco A. Servedio, Li-Yang Tan, Erik Waingarten, and Jinyu Xie. Settling the query complexity of non-adaptive junta testing. In *32nd Computational Complexity Conference, CCC*, pages 26:1–26:19, 2017.

[24] Xi Chen and Erik Waingarten. Testing unateness nearly optimally. In *Proceedings of ACM Symposium on Theory of Computing (STOC)*, pages 547–558, 2019.

[25] Xi Chen, Erik Waingarten, and Jinyu Xie. Beyond Talagrand functions: new lower bounds for testing monotonicity and unateness. In *Proceedings of ACM Symposium on Theory of Computing (STOC)*, pages 523–536, 2017.

[26] Xi Chen, Erik Waingarten, and Jinyu Xie. Boolean unateness testing with $\widetilde{O}(n^{3/4})$ adaptive queries. In *Proceedings of IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 868–879, 2017.

[27] Hana Chockler and Dan Gutfreund. A lower bound for testing juntas. *Inf. Process. Lett.*, 90(6):301–305, 2004.

[28] Anindya De, Elchanan Mossel, and Joe Neeman. Junta correlation is testable. In *Proceedings of IEEE Symposium on Foundations of Computer Science (FOCS)*, 2019. To appear.

[29] Kashyap Dixit, Sofya Raskhodnikova, Abhradeep Thakurta, and Nithin M. Varma. Erasure-resilient property testing. *SIAM J. Comput.*, 47(2):295–329, 2018.

[30] Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved testing algorithms for monotonicity. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, pages 97–108, 1999.

[31] Funda Ergün, Sampath Kannan, Ravi Kumar, Ronitt Rubinfeld, and Mahesh Viswanathan. Spot-checkers. *J. Comput. Syst. Sci.*, 60(3):717–751, 2000.

[32] Shahar Fattal and Dana Ron. Approximating the distance to monotonicity in high dimensions. *ACM Trans. Algorithms*, 6(3):52:1–52:37, 2010.

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

[34] Eldar Fischer and Lance Fortnow. Tolerant versus intolerant testing for Boolean properties. *Theory of Computing*, 2(9):173–183, 2006.

[35] Eldar Fischer, Guy Kindler, Dana Ron, Shmuel Safra, and Alex Samorodnitsky. Testing juntas. *J. Comput. Syst. Sci.*, 68(4):753–787, 2004.

[36] Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *Proceedings of ACM Symposium on Theory of Computing (STOC)*, pages 474–483, 2002.

[37] Eldar Fischer and Ilan Newman. Testing versus estimation of graph properties. *SIAM J. Comput.*, 37(2):482–501, 2007.

[38] Oded Goldreich, Shafi Goldwasser, Eric Lehman, Dana Ron, and Alex Samorodnitsky. Testing monotonicity. *Combinatorica*, 20(3):301–337, 2000.

[39] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998.

[40] Venkatesan Guruswami and Atri Rudra. Tolerant locally testable codes. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, pages 306–317, 2005.

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

[42] Subhash Khot, Dor Minzer, and Muli Safra. On monotonicity testing and Boolean isoperimetric-type theorems. *SIAM J. Comput.*, 47(6):2238–2276, 2018.

[43] Subhash Khot and Igor Shinkar. An $\widetilde{O}(n)$ queries adaptive tester for unateness. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, pages 37:1–37:7, 2016.

[44] Swastik Kopparty and Shubhangi Saraf. Tolerant linearity testing and locally testable codes. In *Proceedings of Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, pages 601–614. 2009.

[45] Eric Lehman and Dana Ron. On disjoint chains of subsets. *J. Comb. Theory, Ser. A*, 94(2):399–404, 2001.

[46] Amit Levi and Erik Waingarten. Lower bounds for tolerant junta and unateness testing via rejection sampling of graphs. In *Proceedings of Innovations in Theoretical Computer Science (ITCS)*, pages 52:1–52:20, 2019.

[47] Sharon Marko and Dana Ron. Approximating the distance to properties in bounded-degree and general sparse graphs. *ACM Trans. Algorithms*, 5(2):22:1–22:28, 2009.

[48] Krzysztof Onak, Dana Ron, Michal Rosen, and Ronitt Rubinfeld. A near-optimal sublinear-time algorithm for approximating the minimum vertex cover size. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1123–1131, 2012.

[49] Ramesh Krishnan S. Pallavoor, Sofya Raskhodnikova, and Nithin M. Varma. Parameterized property testing of functions. *ACM Trans. on Computation Theory*, 9(4):17:1–17:19, 2018.

[50] Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *J. Comput. Syst. Sci.*, 72(6):1012–1042, 2006.

[51] Sofya Raskhodnikova. Monotonicity testing. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1999.

[52] Sofya Raskhodnikova, Noga Ron-Zewi, and Nithin M. Varma. Erasures vs. errors in local decoding and property testing. In *Proceedings of Innovations in Theoretical Computer Science (ITCS)*, pages 63:1–63:21, 2019.

[53] Sofya Raskhodnikova and Adam D. Smith. A note on adaptivity in testing properties of bounded degree graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 13(089), 2006.

[54] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.

[55] Mert Saglam. Near log-convexity of measured heat in (discrete) time and consequences. In *Proceedings of IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 967–978, 2018.

[56] Rocco A. Servedio, Li-Yang Tan, and John Wright. Adaptivity helps for testing juntas. In *30th Conference on Computational Complexity, CCC*, pages 264–279, 2015.

[57] C. Seshadhri. Property testing review: Open problem for February 2014: Better approximations for the distance to monotonicity. `https://ptreview.sublinear.info/?p=250`, February 2014.

[58] Roei Tell. A note on tolerant testing with one-sided error. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:32, 2016.

[59] Andrew Chi-Chih Yao. Probabilistic computations: Toward a unified measure of complexity (extended abstract). In *Proceedings of IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 222–227, 1977.

[60] Yuichi Yoshida, Masaki Yamamoto, and Hiro Ito. An improved constant-time approximation algorithm for maximum matchings. In *Proceedings of ACM Symposium on Theory of Computing (STOC)*, pages 225–234, 2009.

A Removing the Logarithmic Dependence from Isoperimetric Inequalities in [42]

In this section, we give a sketch of the proof of slightly improved versions of the isoperimetric inequalities of Khot et al. [42, Theorems 1.6 and 1.9]. The improved version of [42, Theorem 1.9] is Theorem 2.7 and the improved version of [42, Theorem 1.6] is stated next.

Theorem A.1 (Close to Theorem 1.6 from [42])**.** *Let $f\colon \{0,1\}^n \to \{0,1\}$ be ε-far from monotone. Then,*

$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\{0,1\}^n}\left[\sqrt{I_f^-(\boldsymbol{x})}\right] \geq \Omega(\varepsilon). \tag{7}$$

The statements of Theorems 1.6 and 1.9 in [42] show that the left hand sides of (3) and (7) are at least $\Omega(\frac{\varepsilon}{\log n+\log(1/\varepsilon)})$. We slightly modify the proof of [42] to get a stronger lower bound of $\Omega(\varepsilon)$. Using the original, weaker inequality for our algorithm would result in an approximation to the distance to monotonicity within a multiplicative factor of $\sqrt{n}\cdot\mathrm{poly}(\log n, \log(1/\varepsilon))$. This would mean that our algorithm is an $\tilde{O}(\sqrt{n})$-approximation only if $\varepsilon \geq 1/2^{\mathrm{poly}(\log(n))}$.

To prove Theorems 2.7 and A.1, we first set up some notation. For a function $f\colon \{0,1\}^n \to \{0,1\}$, a set $S \subseteq [n]$, and a string $z \in \{0,1\}^{\overline{S}}$, let $f(\cdot, z)\colon \{0,1\}^S \to \{0,1\}$ denote the function f restricted to the subcube $\{0,1\}^S$ and obtained from f by setting the input bits in $\{0,1\}^{\overline{S}}$ to z. For a real number $p \in (0,1]$, let $\mathcal{S}(p)$ denote the distribution on subsets $\boldsymbol{S} \subseteq [n]$, where each $i \in [n]$ is included in $\boldsymbol{S}$ with probability p independently at random.

In the following proposition, used in the proof of [42, Theorem 1.6], we consider the following experiment: We sample a subset $\boldsymbol{S} \sim \mathcal{S}(p)$ and a uniformly random $\boldsymbol{z} \sim \{0,1\}^{\overline{\boldsymbol{S}}}$. Then we consider $f(\cdot, \boldsymbol{z})$, a random restriction of f.

Proposition A.2. *For a function $f\colon \{0,1\}^n \to \{0,1\}$ and a parameter $p \in [0,1]$,*

$$\mathop{\mathbb{E}}_{\substack{\boldsymbol{S}\sim\mathcal{S}(p)\\ \boldsymbol{z}\sim\{0,1\}^{\overline{\boldsymbol{S}}}}}\left[\mathop{\mathbb{E}}_{\boldsymbol{w}\sim\{0,1\}^{\boldsymbol{S}}}\left[\sqrt{I^-_{f(\cdot,\boldsymbol{z})}(\boldsymbol{w})}\right]\right] \leq \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\{0,1\}^n}\left[\sqrt{I_f^-(\boldsymbol{x})}\right].$$

In other words, we consider the restricted function $f(\cdot, \boldsymbol{z})$, and count the decreasing edges *only along dimensions in* $\boldsymbol{S}$. We improve Proposition A.2 to the following.

Proposition A.3. *For a function $f\colon \{0,1\}^n \to \{0,1\}$ and a parameter $p \in [0,1]$,*

$$\mathop{\mathbb{E}}_{\substack{\boldsymbol{S}\sim\mathcal{S}(p)\\ \boldsymbol{z}\sim\{0,1\}^{\overline{\boldsymbol{S}}}}}\left[\mathop{\mathbb{E}}_{\boldsymbol{w}\sim\{0,1\}^{\boldsymbol{S}}}\left[\sqrt{I^-_{f(\cdot,\boldsymbol{z})}(\boldsymbol{w})}\right]\right] \leq \sqrt{p}\cdot\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\{0,1\}^n}\left[\sqrt{I_f^-(\boldsymbol{x})}\right].$$

Proof. Recall that for $x \in \{0,1\}^n$, the set $D_f^-(x)$ denotes the subset of dimensions along which the edges incident on x are decreasing in f. Note that $|D_f^-(x)| = I_f^-(x)$ for all $x \in \{0,1\}^n$. Hence, we have

$$\mathop{\mathbb{E}}_{\substack{\boldsymbol{S}\sim\mathcal{S}(p)\\ \boldsymbol{z}\sim\{0,1\}^{\overline{\boldsymbol{S}}}}}\left[\mathop{\mathbb{E}}_{\boldsymbol{w}\sim\{0,1\}^{\boldsymbol{S}}}\left[\sqrt{I^-_{f(\cdot,\boldsymbol{z})}(\boldsymbol{w})}\right]\right] = \mathop{\mathbb{E}}_{\substack{\boldsymbol{S}\sim\mathcal{S}(p)\\ \boldsymbol{x}\sim\{0,1\}^n}}\left[\sqrt{|D_f^-(\boldsymbol{x})\cap\boldsymbol{S}|}\right]$$

$$= \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \mathop{\mathbb{E}}_{\boldsymbol{S} \sim \mathcal{S}(p)} \left[\sqrt{|D_f^-(x) \cap \boldsymbol{S}|} \right]$$
$$\leq \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \sqrt{\mathop{\mathbb{E}}_{\boldsymbol{S} \sim \mathcal{S}(p)} \left[|D_f^-(x) \cap \boldsymbol{S}| \right]}$$
$$= \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \sqrt{I_f^-(x) \cdot p}$$
$$= \sqrt{p} \cdot \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \{0,1\}^n} \left[\sqrt{I_f^-(\boldsymbol{x})} \right],$$

where we used Jensen's inequality and the fact that the transformation $\phi(t) = \sqrt{t}$ is concave to derive the inequality. □

Similarly, we have the analogous proposition for the proof of the robust version of the Talagrand objective (Theorem 1.9 of [42]).

Proposition A.4. *For a function $f \colon \{0,1\}^n \to \{0,1\}$ and a parameter $p \in [0,1]$,*

$$\mathop{\mathbb{E}}_{\substack{\boldsymbol{S} \sim \mathcal{S}(p) \\ \boldsymbol{z} \sim \{0,1\}^{\overline{\boldsymbol{S}}}}} \left[\mathop{\mathbb{E}}_{\boldsymbol{w} \sim \{0,1\}^{\boldsymbol{S}}} \left[\sqrt{I^-_{f(\cdot,\boldsymbol{z}),\mathrm{red}}(\boldsymbol{w})} \right] + \mathop{\mathbb{E}}_{\boldsymbol{u} \sim \{0,1\}^{\boldsymbol{S}}} \left[\sqrt{I^-_{f(\cdot,\boldsymbol{z}),\mathrm{blue}}(\boldsymbol{u})} \right] \right]$$
$$\leq \sqrt{p} \cdot \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \{0,1\}^n} \left[\sqrt{I^-_{f,\mathrm{red}}(\boldsymbol{x})} \right] + \sqrt{p} \cdot \mathop{\mathbb{E}}_{\boldsymbol{y} \sim \{0,1\}^n} \left[\sqrt{I^-_{f,\mathrm{blue}}(\boldsymbol{y})} \right].$$

Now we are ready to complete the proof of Theorems 2.7 and A.1. Let $\Psi_f(p)$ denote the expected distance between f' and f where $f' \colon \{0,1\}^n \to \{0,1\}$ is constructed from f by the following random process:

1. Initialize f' to f.
2. Sample a subset $\boldsymbol{S} \sim \mathcal{S}(p)$, then order the elements in $\boldsymbol{S}$ according to a uniformly random permutation.
3. For each dimension $i \in \boldsymbol{S}$ according to the ordering, modify f' by switching the function values on the endpoints of all decreasing i-edges (from (1,0) to (0,1)).

Using Proposition A.3 and following the argument from Section 4.2.2 of [42], we conclude that every $p \in [0,1]$ satisfies, for a constant C,

$$\Psi_f(p) - \Psi_f(p/2) \leq C \cdot \sqrt{p} \cdot \mathop{\mathbb{E}}_{\boldsymbol{x}} \left[\sqrt{I_f^-(\boldsymbol{x})} \right].$$

It follows from the analysis of Dodis et al. [30] that $\Psi_f(1) \geq \varepsilon$. Also note that $\Psi_f(0) = 0$. Therefore, by the telescoping argument for $p = 1, \frac{1}{2}, \frac{1}{4} \ldots$,

$$\varepsilon \leq \Psi_f(1) - \Psi_f(0) = \sum_{i=0}^{\infty} \left(\Psi_f(2^{-i}) - \Psi_f(2^{-i-1}) \right)$$

$$\leq C \cdot \sum_{i=0}^{\infty} 2^{-i/2} \cdot \mathop{\mathbb{E}}_{\boldsymbol{x}}\left[\sqrt{I_f^-(\boldsymbol{x})}\right]$$
$$\leq 4C \cdot \mathop{\mathbb{E}}_{\boldsymbol{x}}\left[\sqrt{I_f^-(\boldsymbol{x})}\right],$$

completing the proof of Theorem A.1. Similarly, Proposition A.4 implies Theorem 2.7.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il