Approximating the Distance to Monotonicity of Boolean Functions[*]

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

Abstract

We design a nonadaptive algorithm that, given oracle access to a 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. The analysis of our algorithm relies on an improvement to the directed isoperimetric inequality of Khot, Minzer, and Safra (SIAM J. Comput., 2018). Furthermore, we rule out a $\mathrm{poly}(n, 1/\alpha)$-query nonadaptive algorithm that approximates the distance to monotonicity significantly better by showing that, for all constant $\kappa > 0$, every nonadaptive $n^{1/2-\kappa}$-approximation algorithm for this problem requires $2^{n^{\kappa}}$ queries. This answers a question of Seshadhri (Property Testing Review, 2014) for the case of nonadaptive algorithms. We obtain our lower bound by proving an analogous bound for erasure-resilient (and tolerant) testers. Our method also 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.

Keywords – sublinear algorithms, analysis of Boolean functions, property testing, tolerant and erasure-resilient testing.

1 Introduction

Property testing [59, 42] 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 [55] 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 $f(x) \leq f(y)$ whenever $x \preceq y$, i.e., $x_i \leq y_i$ 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 $\mathcal{P}$

---

[1]A distance approximation algorithm can be easily converted to a tolerant tester and vice versa, with at most a logarithmic increase in the query complexity. See [55] and Section 1.1.5 for a discussion of the relationship.

if $\operatorname{dist}(f, \mathcal{P}) \geq \alpha$; otherwise, it is α-close. The distance of a function f to monotonicity is denoted ε_f. We study randomized algorithms which, given oracle access to a Boolean function, output an approximation to ε_f 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 $2/3$,

$$\varepsilon_f \leq \hat{\varepsilon} \leq c \cdot \varepsilon_f.$$

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

Definition 1.1 (Hypercube, edge, i-edge, decreasing edge)**.** The points of the domain $\{0,1\}^n$ can be viewed as vertices of an n-dimensional hypercube. Two vertices $x, y \in \{0,1\}^n$ form an edge $\{x,y\}$ of the hypercube if they differ in exactly one coordinate. An edge $\{x,y\}$ is an i-edge (or an *edge along dimension i*) if x and y differ only in their i^{th} coordinates, that is, $x_i \neq y_i$, but $x_j = y_j$ for all $j \in [n] \setminus \{i\}$. An i-edge is *decreasing* with respect to a function f if $x_i < y_i$ but $f(x) > f(y)$.

Fattal and Ron [35] investigated a more general problem of approximating the distance to monotonicity of functions on the hypergrid $[t]^n$. They gave several algorithms which achieve an approximation ratio $O(n)$ in time polynomial in n and $1/\alpha$; for better approximations, they designed an algorithm with the approximation ratio n/k, for every k, but with running time exponential in k. It follows from early works on monotonicity testing that for the special case of the hypercube domain, an $O(n)$-approximation can be obtained by simply estimating the fraction ν_f of edges that are decreasing with respect to f. For a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, as shown in [33, 56, 41, 39], $\varepsilon_f/n \leq \nu_f \leq 2\varepsilon_f$. Thus, by obtaining a constant-factor approximation to the fraction of decreasing edges, one gets an $O(n)$-approximation to ε_f.

Prior to our work, no nontrivial hardness results were known for approximating the distance to monotonicity, other than the corresponding lower bounds on (standard) property testing.

1.1 Our Contributions

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].

1.1.1 Approximating the Distance to Monotonicity

We design a nonadaptive $\widetilde{O}(\sqrt{n})$-approximation algorithm for the 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 [35].

---

[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.

Theorem 1.2 (Approximation Algorithm)**.** *There is a nonadaptive (randomized) algorithm that, given a parameter $\alpha \in (0, 1/2)$ and oracle access to a Boolean function $f \colon \{0,1\}^n \to \{0,1\}$ which is α-far from monotone, makes $\mathrm{poly}(n, 1/\alpha)$ queries and returns an estimate that, with probability at least $2/3$, is an $\widetilde{O}(\sqrt{n})$-approximation to ε_f.*

Our algorithm works by estimating (in addition to the fraction ν_f of decreasing edges) the size of a particular matching from a class of matchings parameterized by subsets $S \subseteq [n]$ and consisting of decreasing edges along the dimensions in S. For every S, the size of a matching of decreasing edges, divided by 2^n, is a lower bound for ε_f. This is because any monotone function $g \colon \{0,1\}^n \to \{0,1\}$ 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. To analyze our algorithm, we use a slightly improved version of the (robust) directed isoperimetric inequality by Khot et al. [45]. Our improvements to isoperimetric inequalities of [45] are discussed in Section 1.1.2 and stated in Theorems 2.9 and 1.3. Intuitively, if the input function f is $\widetilde{O}(\varepsilon/\sqrt{n})$-close to monotone, then both the fraction of violated edges and the normalized sizes of all matchings considered by our algorithm are below some threshold $\widetilde{\Theta}(\varepsilon/\sqrt{n})$. We use Theorem 2.9 to show (in Lemma 2.8) that if f is ε-far from monotone, then either ν_f is above the threshold or our algorithm is likely to sample some set $S \subseteq [n]$ where the corresponding matching has normalized size $\widetilde{\Omega}(\varepsilon/\sqrt{n})$. This allows us to get an $\widetilde{O}(\sqrt{n})$-approximation to the distance to monotonicity.

1.1.2 Improvements in the Isoperimetric Inequalities from [45]

In Section 3, we give a sketch of the proof of slightly improved versions of the isoperimetric inequalities of Khot et al. [45, Theorems 1.6 and 1.9]. The improved version of [45, Theorem 1.6] is stated next.

For all $x \in \{0,1\}^n$, define the *negative influence* $I_f^-(x)$ to be equal to 0 if $f(x) = 0$ and equal to the number of decreasing edges incident on x if $f(x) = 1$. Note that each decreasing edge "counts" towards the endpoint with the function value 1.

Theorem 1.3 (Improvement of Theorem 1.6 from [45])**.** *For every function $f \colon \{0,1\}^n \to \{0,1\}$,*

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

The improved version of [45, Theorem 1.9] is Theorem 2.9 that appears later. It is a *robust* version of Theorem 1.3. Specifically, it holds for every 2-coloring of the edges of f. The color of each decreasing edge indicates whether it should be "counted" towards the endpoint with value 1 or the endpoint with value 0, and the negative influence is generalized accordingly to the *robust* version.

Theorems 1.6 and 1.9 in [45] state that the left-hand side of (1) (and similarly its counterpart (4) in Theorem 2.9) is $\Omega(\frac{\varepsilon_f}{\log n + \log(1/\varepsilon_f)})$. We slightly modify the proof of [45] to get a stronger lower bound of $\Omega(\varepsilon_f)$. Using the original, weaker inequality for our algorithm would result in an approximation to the distance to monotonicity within a multiplicative factor of $\sqrt{n} \cdot \mathrm{poly}(\log n, \log(1/\alpha))$. This would mean that our algorithm is an $\widetilde{O}(\sqrt{n})$-approximation only if $\varepsilon_f \geq 1/2^{\mathrm{poly}(\log(n))}$.

1.1.3 Lower Bounds for Monotonicity, Unateness, and Being a k-Junta

We show that a slightly better approximation for the distance to monotonicity, 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 1.4 (Approximation Lower Bound). *For all constant $\kappa \in (0, 1/2)$, there exist $\alpha = O(1/n^{1-\kappa})$ and $\varepsilon = \Omega(1/\sqrt{n})$ (that is, 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 1.2, answers an open question on the problem of approximating the distance to monotonicity by Seshadhri [62] 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 1.2 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 1.4, we investigate a variant of the property testing model, called *erasure-resilient* testing. This variant, proposed by Dixit et al. [32], 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 [32], 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 used 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 1.4.

Theorem 1.5 (Erasure-Resilient Lower Bound). *For all constant $\kappa \in (0, 1/2)$, there exist $\alpha = O(1/n^{1-\kappa})$ and $\varepsilon = \Omega(1/\sqrt{n})$ (that is, 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 1.5 directly implies lower bounds analogous to the one stated in Theorem 1.4 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 [50]. They showed that every

algorithm approximating the distance to unateness within a constant factor requires $\widetilde{\Omega}(n)$ queries and strengthened their lower bound to $\widetilde{\Omega}(n^{3/2})$ 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 $\widetilde{\Omega}(k^2)$ queries. Our lower bounds are exponentially larger than those obtained by Levi and Waingarten [50] and hold for larger approximation factors.

1.1.4 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 [37, 32]. For testing monotonicity of Boolean function, the celebrated nonadaptive algorithm of Khot, Minzer and Safra [45] makes $\widetilde{O}(\sqrt{n}/\varepsilon^2)$ 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 $\widetilde{O}(k^{3/2}/\varepsilon)$ queries [11]. 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 [32], for which erasure-resilient testers have essentially the same query complexity as standard testers.

1.1.5 Connection Between Tolerant Testing and Distance Approximation

Our main algorithm presented in Section 2 distinguishes functions that are $\widetilde{O}(\varepsilon/\sqrt{n})$-close to monotone from functions that are ε-far from monotone. In property testing terminology, such an algorithm is an example of a *tolerant tester*. Transforming a tolerant tester to a distance approximation algorithm can be done using standard techniques; see, for example [55, Claim 2] and [2, Section 3.3]. 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. Therefore, to obtain an algorithm that approximates the distance to monotonicity for an input function f under the assumption $\varepsilon_f \geq \alpha$, we can run the tolerant tester with ε set to $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots, \alpha$ an appropriate number of times. The guarantees of the resulting conversion from tolerant testing to distance approximation are stated in Theorem 5.1, which is a straightforward generalization of [55, Claim 2], with a small improvement in query complexity (by a factor of $\log(1/\alpha)$). This improvement applies to every tolerant tester with query complexity at least linear in $1/\varepsilon$. The details appear in Section 5.

1.2 Comparison to Potential Alternative Approaches to Approximating ε_f

Chakrabarty and Seshadhri, in a personal communication, notified us of an alternative approach towards an $O(\sqrt{n})$-approximation of ε_f via estimating the size of a maximal matching of decreasing edges. Results in [45, 22] 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 [65, 52]. However, these algorithms are adaptive. It is a compelling open problem to understand whether adaptivity can help with approximating the distance to monotonicity.

Given Theorem 1.3, a natural approach to design an algorithm for approximating the distance to monotonicity is to estimate the left-hand side of (1) by sampling points from $\{0,1\}^n$ uniformly at random. When the underlying function $f\colon \{0,1\}^n \to \{0,1\}$ is ε-far from monotone, the estimator would be $\Omega(\varepsilon)$. The problem is that this estimator could be just as high for functions that are close

to monotone. For example, consider the random function $\boldsymbol{f}\colon \{0,1\}^n \to \{0,1\}$ which is defined as follows:

$$\boldsymbol{f}(x) = \begin{cases} 1 - \mathrm{Maj}(x) & \text{with probability } \varepsilon/n; \\ \mathrm{Maj}(x) & \text{with probability } 1 - \varepsilon/n, \end{cases}$$

where $\mathrm{Maj}(\cdot)$ denotes the majority function. Then, $\varepsilon_{\boldsymbol{f}} = \Theta(\varepsilon/n)$ with high probability, yet the left-hand side of (1) is $\Omega(\varepsilon)$ with high probability.

Finally, as thoroughly explained in [55, Section 3.2], every $q(n, \varepsilon)$-query algorithm for (standard) property testing whose queries are individually (almost) uniform exhibits some degree of tolerance. The algorithm of Khot et al. [45] for testing monotonicity of Boolean functions makes $\widetilde{O}(\sqrt{n}/\varepsilon^2)$ queries which are roughly uniformly distributed (this is implicit in Lemma 9.3 of [45]), and hence can be used to obtain an $\widetilde{O}(\sqrt{n}/\alpha)$-approximation. However, notice that the approximation factor degrades as a function of α.

1.3 Previous Work

Testing monotonicity and unateness (first studied in [41]), as well as k-juntas (first studied in [38]), are among the most widely investigated problems in property testing ([34, 33, 56, 48, 39, 1, 36, 44, 4, 55, 2, 8, 16, 13, 20, 21, 22, 15, 19, 25, 24, 45, 5, 28, 53, 9, 23, 10] study monotonicity testing, [46, 3, 28, 29, 27] study unateness testing, and [30, 11, 12, 17, 61, 26, 60] 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 $\widetilde{O}(\sqrt{n})$-query nonadaptive algorithm of [45] and lower bounds of $\widetilde{\Omega}(\sqrt{n})$ (nonadaptive) and $\widetilde{\Omega}(n^{1/3})$ (adaptive) [28] for monotonicity, and tight upper and lower bounds of $\widetilde{\Theta}(n^{2/3})$ for unateness testing [27, 28], as well as $\Theta(k \log k)$ for k-junta testing [11, 60].

Beyond the (standard) property testing, the questions of erasure-resilient and tolerant testing have also received some attention ([32, 57, 49] study the erasure-resilient model, and [43, 55, 37, 40, 2, 47, 51, 35, 18, 7, 6, 63, 14, 50, 31] study the tolerant testing model). Specifically for monotonicity, in [32], 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 [14, 31], the best algorithm with additive error of ε makes $2^k \cdot \mathrm{poly}(k, 1/\varepsilon)$ queries [31], and the best lower bound was $\Omega(k^2)$ for nonadaptive algorithms [50].

2 An Approximation Algorithm for the 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 $2/3$,

$$\varepsilon_f \leq \hat{\varepsilon} \leq \widetilde{O}(\sqrt{n}) \cdot \varepsilon_f.$$

Our main algorithm, `ApproxMono`, whose performance is summarized in Lemma 2.1, distinguishes functions that are close to monotone from those that are far. Theorem 1.2 follows directly from Lemma 2.1 and Theorem 5.1.

6

Lemma 2.1. *There exist a universal constant $c_0 \in (0,1)$ and 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 $\varepsilon_f \leq c_0 \cdot \varepsilon/\sqrt{n \log n}$, it outputs close *with probability at least $2/3$.*

   2. If $\varepsilon_f \geq \varepsilon$, it outputs far *with probability at least $2/3$.*

The algorithm `ApproxMono` is described in Algorithm 1. In the algorithm and its analysis, we use $\gamma(n, \varepsilon)$ to denote $\varepsilon/\sqrt{n \log n}$. The algorithm uses subroutines `Edge-Violations` and `Matching-Estimation`, which are described in Algorithms 2 and 3, 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 of the fraction of decreasing edges of f with an additive error less than 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[3], over $\boldsymbol{x} \sim \{0,1\}^n$, of an event (which we denote `Capture` and describe in Definition 2.5) defined with respect to $\boldsymbol{x}$, S and f up with an additive error less than δ. The high level intuition is that, as long as the estimates returned by `Matching-Estimation`(S, δ, f) and `Edge-Violations`(δ, f) are accurate, we can certify a lower bound on the distance to monotonicity of f. We prove that if f is ε-far from monotone, either the number of decreasing edges of f is large (and Step 2 of Algorithm 1 declares far) or the `Matching-Estimation` subroutine can verify a lower bound on the distance to monotonicity.

Algorithm 1: `ApproxMono`(ε, f)

   input : A parameter $\varepsilon \in (0,1/2)$; oracle access to $f\colon \{0,1\}^n \to \{0,1\}$.
   output: Either close or far.

1 Set $\hat{\boldsymbol{\nu}} \leftarrow$ `Edge-Violations`$(\gamma(n,\varepsilon)/4, f)$. // $\hat{\boldsymbol{\nu}}$ is an estimate of the fraction of decreasing edges with an additive error less than $\gamma(n,\varepsilon)/4$.
2 if $\hat{\boldsymbol{\nu}} \geq 3\gamma(n,\varepsilon)/4$ **then return** far.
3 Set $t \leftarrow 2/m$, where $m = c_1 \cdot \gamma(n,\varepsilon)/2$ with the constant c_1 dictated by Lemma 2.8.
4 foreach $d \in \{1, 2, 4, \ldots, 2^{\lfloor \log_2 n \rfloor}\}$ **do**
5   **repeat** t **times**
6   Sample $\boldsymbol{S} \subseteq [n]$ by including each $i \in [n]$ independently with probability $1/d$.
7   Set $\hat{\boldsymbol{\mu}} \leftarrow$ `Matching-Estimation`$(\boldsymbol{S}, m/4, f)$. // $\hat{\boldsymbol{\mu}}$ is an estimate of $\mu_f(\boldsymbol{S}) = \Pr_{\boldsymbol{x} \sim \{0,1\}^n}[$`Capture`$(\boldsymbol{x}, \boldsymbol{S}, f)]$ with an additive error less than $m/4$.
8   **if** $\hat{\boldsymbol{\mu}} \geq 3m/4$ **then return** far.
9 return close.

Recall the definition of an edge and a decreasing edge of the hypercube (see Definition 1.1).

Definition 2.2. For a function $f \colon \{0,1\}^n \to \{0,1\}$, let ν_f denote the probability that a uniformly random edge is decreasing. We also refer to ν_f as the fraction of decreasing edges.

---

[3]We use the convention that random variables are boldface whereas fixed quantities use standard typeface. For example, $\boldsymbol{x}$ and $\boldsymbol{S}$ are random variables whereas x and S are the corresponding fixed quantities. The notation $\sim$ stands for "sampled from".

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 ith coordinate in x.

We summarize the properties of the subroutine Edge-Violations(δ, f), which estimates ν_f, in Fact 2.3.

Algorithm 2: Edge-Violations(δ, f)

   input : A parameter $\delta > 0$; oracle access to $f \colon \{0,1\}^n \to \{0,1\}$.
   output: A real number $\hat{\boldsymbol{\nu}} \in [0,1]$ that approximates ν_f.

1 Initialize *counter* $\leftarrow 0$ and set $t \leftarrow \left\lceil \frac{2}{\delta^2} \right\rceil$.
2 repeat t **times**
3   Sample $\boldsymbol{x} \sim \{0,1\}^n$ and $\boldsymbol{i} \sim [n]$, and query $f(\boldsymbol{x}^{(i \to 0)})$ and $f(\boldsymbol{x}^{(i \to 1)})$.
4   **if** $f(\boldsymbol{x}^{(i \to 0)}) > f(\boldsymbol{x}^{(i \to 1)})$ **then** *counter* $\leftarrow$ *counter* $+ 1$.
5 return $\hat{\boldsymbol{\nu}} = $ *counter*$/t$.

Algorithm 3: Matching-Estimation(S, δ, f)

   input : A set $S \subseteq [n]$; a parameter $\delta > 0$; oracle access to $f \colon \{0,1\}^n \to \{0,1\}$.
   output: A real number $\hat{\boldsymbol{\mu}} \in [0,1]$ that approximates $\mu_f(S)$.

1 Initialize *counter* $\leftarrow 0$ and set $t \leftarrow \left\lceil \frac{2\ln(n/\varepsilon)}{\delta^2} \right\rceil$.
2 repeat t **times**
3   Sample $\boldsymbol{x} \sim \{0,1\}^n$ and query $f(\boldsymbol{x})$.
4   **foreach** $i \in S$ **do**
5   Set $\boldsymbol{y}_i \leftarrow \boldsymbol{x}^{(i)}$ and query $f(\boldsymbol{y}_i)$.
6   **foreach** $j \in S \setminus \{i\}$ **do** query $f(\boldsymbol{y}_i^{(j)})$.
7   **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, **then** *counter* $\leftarrow$ *counter* $+ 1$. // Capture($\boldsymbol{x}, S, f$)
       occurred.
8 return $\hat{\boldsymbol{\mu}} = $ *counter*$/t$.

Fact 2.3. *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 $O(1/\delta^2)$ queries, and outputs $\hat{\boldsymbol{\nu}} \in [0,1]$ which, with probability at least $24/25$, satisfies*

$$|\nu_f - \hat{\boldsymbol{\nu}}| < \delta. \tag{2}$$

Proof. By Hoeffding bound, $\Pr[|\nu_f - \hat{\boldsymbol{\nu}}| \geq \delta] \leq 2\mathrm{e}^{-2t\delta^2} \leq 2\mathrm{e}^{-4} \leq 1/25$. $\qquad \square$

Observation 2.4. *For all functions $f : \{0,1\}^n \to \{0,1\}$, we have $\nu_f \leq 2\varepsilon_f$.*

Proof. By definition, $\nu_f \cdot 2^{n-1}n$ edges in f are decreasing. To convert f to a monotone function, at least one endpoint of every decreasing edge has to be changed. Since each point in $\{0,1\}^n$ is incident on n edges of the hypercube, at least $\nu_f \cdot 2^{n-1}$ points have to be changed. Thus, $\varepsilon_f \geq \nu_f \cdot 2^{n-1}/2^n = \nu_f/2$. $\qquad \square$

Definition 2.5. For a function $f: \{0,1\}^n \to \{0,1\}$, a subset $S \subseteq [n]$, and a point $x \in \{0,1\}^n$, let $\texttt{Capture}(x, S, f)$ be the following event (see Figure 1):

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.

We denote $\Pr_{\boldsymbol{x} \sim \{0,1\}^n} [\texttt{Capture}(\boldsymbol{x}, S, f)]$ by $\mu_f(S)$.
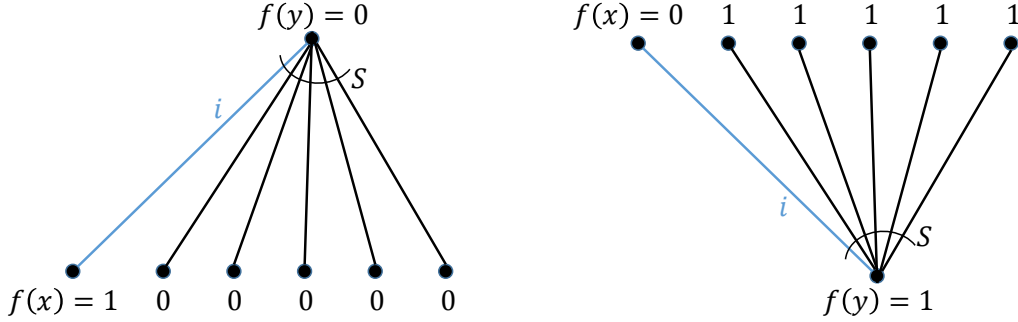

Figure 1: An illustration to Definition 2.5. Two cases are depicted, corresponding to the two possible values of $f(x)$.

We summarize the properties of subroutine $\texttt{Matching-Estimation}(S, \delta, f)$, which estimates $\mu_f(S)$, in Fact 2.6. Like Fact 2.3, it can be easily proved by an application of Hoeffding bound.

Fact 2.6. *The algorithm* $\texttt{Matching-Estimation}$ *is nonadaptive. It gets a set $S \subseteq [n]$, a parameter $\delta > 0$ and oracle access to a function $f: \{0,1\}^n \to \{0,1\}$, makes $O(|S|^2 \log(n/\varepsilon)/\delta^2)$ queries, and outputs $\hat{\boldsymbol{\mu}} \in [0,1]$ which, with probability at least $1 - (\varepsilon/n)^3$, satisfies*

$$|\mu_f(S) - \hat{\boldsymbol{\mu}}| < \delta. \tag{3}$$

Lemma 2.1 follows from Lemmas 2.7 and 2.8.

Lemma 2.7. *For all functions $f: \{0,1\}^n \to \{0,1\}$ and sets $S \subseteq [n]$,*

$$\mu_f(S) \leq 2 \cdot \varepsilon_f.$$

Proof. Let $X = \{x \in \{0,1\}^n : \text{the event } \texttt{Capture}(x, S, f) \text{ occurs}\}$. For each $x \in X$, let $y_x = x^{(i)}$ for a dimension $i \in S$ be a 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 2.5. 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, we must change f on at least $|X|/2 = \mu_f \cdot 2^n/2$ points to make it monotone. $\qquad\square$

Lemma 2.8 (Key Lemma). *There exists a constant $c_1 \in (0,1)$ such that the following holds. Let $f: \{0,1\}^n \to \{0,1\}$ be ε-far from monotone, with $\nu_f < \gamma(n, \varepsilon)$. Let m_f be the maximum over $d \in \{1, 2, 4, \ldots, 2^{\lfloor \log_2 n \rfloor}\}$ of*

$$\mathop{\mathbb{E}}_{\substack{\boldsymbol{S} \subseteq [n] \\ i \in \boldsymbol{S} \text{ w.p. } 1/d}} [\mu_f(\boldsymbol{S})].$$

Then $m_f > c_1 \cdot \gamma(n, \varepsilon)$.

9

The proof of Lemma 2.8 appears in Section 2.1. We use Observation 2.4 and Lemmas 2.7 and 2.8 to complete the proof of Lemma 2.1.

Proof of Lemma 2.1. We set the constant $c_0 = c_1/8$ in Lemma 2.1, where the constant $c_1 \in (0, 1)$ is from Lemma 2.8.

Let `CorrectEstimates` be the event that all invocations of the subroutines in Algorithm 1 produce outputs within the error bounds specified in (2) and (3). By Facts 2.3 and 2.6 and by a union bound over the invocation of `Edge-Violations` and at most $t(\log_2 n + 1) = O(n/\varepsilon)$ invocations of `Matching-Estimation`, this event occurs with probability at least $5/6$.

First, we prove part 1 of Lemma 2.1. Suppose $\varepsilon_f \leq c_0 \cdot \gamma(n, \varepsilon)$. Then, by Observation 2.4, $\nu_f \leq 2c_0 \cdot \gamma(n, \varepsilon) < \gamma(n, \varepsilon)/2$, since $c_0 \leq 1/8$. By Lemma 2.7 and our choice of c_0, for all sets $S \subseteq [n]$, we have $\mu_f(S) \leq 2c_0 \cdot \gamma(n, \varepsilon) = c_1 \cdot \gamma(n, \varepsilon)/4 = m/2$, where m is the parameter from Step 3 of Algorithm 1. When event `CorrectEstimates` occurs, the estimate $\hat{\nu}$ in Step 2 is less than $3\gamma(n, \varepsilon)/4$ and all the estimates $\hat{\mu}$ in Step 8 are less than $3m/4$. In this case, Algorithm 1 outputs `close`. This happens with probability at least $3/4$, so the proof of part 1 is complete.

Next, we prove part 2 of Lemma 2.1. Suppose that $f \colon \{0,1\}^n \to \{0,1\}$ is ε-far from monotone. If $\nu_f \geq \gamma(n, \varepsilon)$ then, whenever `CorrectEstimates` occurs, Step 2 outputs `far`. Now, assume $\nu_f < \gamma(n, \varepsilon)$. By Lemma 2.8, there exists some $d \in \{1, 2, 4, \dots, 2^{\lfloor \log_2 n \rfloor}\}$ for which $\mathbb{E}_{S \subseteq [n]}[\mu_f(S)] = m_f \geq 2m$, where $S \subseteq [n]$ is sampled by including each $i \in [n]$ independently with probability $1/d$. By the reverse Markov's inequality, since $\mu_f(S) \leq 1$ always holds, we get

$$\Pr_{S}[\mu_f(S) \geq m] \geq \frac{m_f - m}{1 - m} \geq m.$$

Step 6 of `ApproxMono`(ε, f) fails to sample some $S \subset [n]$ such that $\mu_f(S) \geq m$ with probability at most $(1 - m)^t \leq \mathrm{e}^{-mt} \leq \mathrm{e}^{-2} \leq 1/6$. Since `CorrectEstimates` occurs with probability at least $5/6$, Step 8 outputs `far` with probability at least $5/6 - 1/6 = 2/3$. $\qquad\square$

2.1 Proof of Lemma 2.8

We start by outlining some of the ideas from our proof of Lemma 2.8 at a high level. In the proof, we attribute each decreasing edge to its endpoint adjacent to a larger number of decreasing edges. We partition all the hypercube vertices into $\log^2 n$ buckets $B_{d,s}$, where the bucket $B_{d,s}$ contains each vertex with d to $2d$ adjacent decreasing edges and s to $2s$ adjacent decreasing edges attributed to it. Importantly, we show that, for each vertex $x \in B_{d,s}$, when each coordinate of $[n]$ is included in S independently with probability $1/d$, the probability of the event $\textit{Capture}(x, S, f)$ is $\Omega(s/d)$. Then we apply (a variant of) the Cauchy-Schwartz inequality, the directed (robust) isoperimetric inequality, and the upper bound on the fraction ν_f of decreasing edges (from the premise of Lemma 2.8) to get a lower bound on

$$\sum_{d \in \{1, 2, 4, \dots, 2^{\lfloor \log_2 n \rfloor}\}} \mathbb{E}_{\substack{S \subseteq [n] \\ i \in S \text{ w.p. } 1/d}} [\mu_f(S)].$$

We conclude that, by averaging, there exists a setting of d for which one of the terms in the sum is large.

A crucial tool in our proof of Lemma 2.8 is the main (robust) directed isoperimetric inequality of Khot et al. [45]. We use notation consistent with [45]. 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 $U_f^-\colon \{0,1\}^n \to \{0,1,\dots,n\}$ map each point $x \in \{0,1\}^n$ to the number of decreasing edges[4] 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 $I_{f,\mathrm{col}}^-\colon \{0,1\}^n \to \{0,\dots,n\}$ be the function given by:

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

That is, each decreasing edge is counted towards the lower endpoint if it is red and towards the higher endpoint if it is blue.

We crucially rely on the main theorem of [45], which is stated next, with a minor improvement in the bound. We obtain the improvement in Section 3.

Theorem 2.9 (Improvement of Theorem 1.9 from [45]). *Let $f\colon \{0,1\}^n \to \{0,1\}$ be ε-far from monotone. Then, for every coloring* col *of $\mathcal{S}_f^-$ into* red *and* blue,

$$\mathop{\mathbb{E}}_{\boldsymbol{x} \sim \{0,1\}^n} \left[\sqrt{I_{f,\mathrm{col}}^-(\boldsymbol{x})} \right] = \Omega(\varepsilon). \tag{4}$$

To prove Lemma 2.8, consider a function $f\colon \{0,1\}^n \to \{0,1\}$ which is ε-far from monotone with $\nu_f < \gamma(n,\varepsilon)$. Consider the coloring of $\mathcal{S}_f^-$ that colors each edge $\{x,y\}$ with $x \prec y$ as follows:

$$\mathrm{col}(\{x,y\}) = \begin{cases} \mathrm{red} & \text{if } U_f^-(x) \geq U_f^-(y); \\ \mathrm{blue} & \text{if } U_f^-(x) < U_f^-(y). \end{cases}$$

In this coloring, each decreasing edge in f is counted in (4) towards its endpoint incident on a higher number of decreasing edges. (If there is a tie, it is counted towards the lower endpoint.)

Without loss of generality, suppose that the red edges contribute at least as much as the blue edges to the Talagrand objective (the case where the blue edges contribute more is symmetric). In other words, we break up the left-hand side of (4) into two terms and assume one is greater than or equal to the other:

$$\sum_{x : f(x)=1} \left[\sqrt{I_{f,\mathrm{col}}^-(x)} \right] \geq \sum_{x : f(x)=0} \left[\sqrt{I_{f,\mathrm{col}}^-(x)} \right].$$

Then, by (4),

$$\frac{1}{2^n} \sum_{x : f(x)=1} \left[\sqrt{I_{f,\mathrm{col}}^-(x)} \right] \geq \frac{1}{2} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \{0,1\}^n} \left[\sqrt{I_{f,\mathrm{col}}^-(\boldsymbol{x})} \right] = \Omega(\varepsilon).$$

We partition the points $x \in \{0,1\}^n$ with $f(x) = 1$ and $I_{f,\mathrm{col}}^-(x) > 0$ into buckets $B_{d,s}$ indexed by $d, s \in \{1, 2, 4, \dots, 2^{\lfloor \log_2 n \rfloor}\}$, where $d \geq s$. The bucket $B_{d,s}$ is defined by:

$$B_{d,s} = \{x \in \{0,1\}^n : d \leq U_f^-(x) < 2d \text{ and } s \leq I_{f,\mathrm{col}}^-(x) < 2s \text{ and } f(x) = 1\}.$$

---

[4]Note that U_f^- is the "undirected" version of I_f^-. Specifically,

$$I_f^-(x) = \begin{cases} U_f^-(x) & \text{if } f(x) = 1, \\ 0 & \text{if } f(x) = 0. \end{cases}$$

11

That is, vertices in $B_{d,s}$ are incident on between d and $2d$ decreasing edges and between s and $2s$ red edges. By the definition of the buckets, $I^-_{f,\text{col}}(x) \leq 2s$ for every $x \in B_{d,s}$, implying

$$\sum_{d,s:d\geq s} |B_{d,s}| \cdot \sqrt{s} = \sum_{d,s:d\geq s} \sum_{x \in B_{d,s}} \sqrt{s} \geq \frac{1}{\sqrt{2}} \sum_{x:f(x)=1} \sqrt{I^-_{f,\text{col}}(x)} = \Omega(\varepsilon \cdot 2^n). \tag{5}$$

Each $x \in B_{d,s}$ is an endpoint of at least d decreasing edges of f. Moreover, the sets of decreasing edges incident on different points x with $f(x) = 1$ are disjoint. Consequently, by the bound on the fraction of decreasing edges in the statement of Lemma 2.8,

$$\sum_{d,s:d\geq s} |B_{d,s}| \cdot d < \gamma(n,\varepsilon) \cdot 2^{n-1} n = \varepsilon \sqrt{n/\log n} \cdot 2^{n-1}. \tag{6}$$

Next, we show that for each bucket $B_{d,s}$ and each $x \in B_{d,s}$, the probability that the event $\texttt{Capture}(x, \boldsymbol{S}, f)$ occurs is sufficiently large when $\boldsymbol{S}$ is chosen appropriately.

Claim 2.10. *For all $d, s \in \{1, 2, 4, \ldots, 2^{\lfloor \log_2 n \rfloor}\}$, where $d \geq s$, and all $x \in B_{d,s}$,*

$$\Pr_{\substack{\boldsymbol{S} \subseteq [n] \\ i \in \boldsymbol{S} \text{ w.p. } 1/d}} [\texttt{Capture}(x, \boldsymbol{S}, f)] \geq \frac{1}{e^4} \cdot \frac{s}{d}.$$

Proof. Fix $d \geq s$ and an arbitrary $x \in B_{d,s}$.

First, consider the case when $d = 1$. Then $s = 1$ and, consequently,

$$U^-_f(x) = I^-_{f,\text{col}}(x) = 1,$$

that is, the only decreasing edge incident on x is colored red. Call this edge $\{x, y\}$. Since $\text{col}(x, y) = \text{red}$, by definition of the coloring, $U^-_f(y) \leq U^-_f(x) = 1$. 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 2.5, $\texttt{Capture}(x, \boldsymbol{S}, f)$ occurs since $\{x, y\}$ is a decreasing edge along a dimension in $\boldsymbol{S}$, and all other edges incident on y are nondecreasing. Hence,

$$\Pr_{\boldsymbol{S}=[n]} [\texttt{Capture}(x, \boldsymbol{S}, f)] = 1,$$

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 edges incident on x are decreasing in f, and let $E^-_f(x) = \{i \in D^-_f(x) : U^-_f(x) \geq U^-_f(x^{(i)})\}$ be the set of dimensions along which the other endpoint is adjacent to no more decreasing edges than x. By the definition of the buckets, $|D^-_f(x)| \leq 2d-1$ and $s \leq |E^-_f(x)| \leq 2s-1$. If we sample $\boldsymbol{S} \subseteq [n]$ by including each index $i \in [n]$ independently with probability $1/d$, then the probability of $\texttt{Capture}(x, \boldsymbol{S}, f)$ 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 $U^-_f(x) \geq U^-_f(y)$, and all other decreasing edges of f incident on y are along dimensions in $[n] \setminus \boldsymbol{S}$. This probability, in turn, is at least the probability of the union over $i \in E^-_f(x)$ of the following events: for each $i \in E^-_f(x)$, the corresponding event is that $i \in \boldsymbol{S}$, but all other dimensions in $E^-_f(x) \cup D^-_f(x^{(i)})$ are not in $\boldsymbol{S}$. Since these events are disjoint,

$$\Pr_{\boldsymbol{S} \subseteq [n]} [\texttt{Capture}(x, \boldsymbol{S}, f)] \geq \sum_{i \in E^-_f(x)} \left(\Pr[i \in \boldsymbol{S}] \cdot \prod_{j \in (E^-_f(x) \cup D^-_f(x^{(i)})) \setminus \{i\}} \Pr[j \notin \boldsymbol{S}] \right),$$

12

$$\geq s \cdot \frac{1}{d} \cdot \left(1 - \frac{1}{d}\right)^{(2s-2)+(2d-2)} \geq s \cdot \frac{1}{d} \cdot \left(1 - \frac{1}{d}\right)^{4(d-1)} \geq \frac{s}{e^4 d},$$

where we used $s \leq |E_f^-(x)| \leq 2s - 1$ and

$$|D_f^-(x^{(i)})| \leq U_f^-(x) = |D_f^-(x)| \leq 2d - 1$$

to get the second inequality, $d \geq s$ to get the third inequality, and $(1 - 1/d)^{d-1} \geq 1/e$ for all $d \geq 2$ to get the final inequality. This concludes the proof of Claim 2.10. $\square$

By Claim 2.10, for all $d \in \{1, 2, 4, \ldots, 2^{\lfloor \log_2 n \rfloor}\}$,

$$
\begin{aligned}
\mathop{\mathbb{E}}_{\substack{\boldsymbol{S} \subseteq [n] \\ i \in \boldsymbol{S} \text{ w.p. } 1/d}} [\mu_f(\boldsymbol{S})] &= \mathop{\mathbb{E}}_{\substack{\boldsymbol{S} \subseteq [n] \\ i \in \boldsymbol{S} \text{ w.p. } 1/d}} \left[\mathop{\Pr}_{\boldsymbol{x} \sim \{0,1\}^n} [\texttt{Capture}(\boldsymbol{x}, \boldsymbol{S}, f)] \right] \\
&\geq \frac{1}{2^n} \sum_{x \in \bigcup_s B_{d,s}} \mathop{\Pr}_{\substack{\boldsymbol{S} \subseteq [n] \\ i \in \boldsymbol{S} \text{ w.p. } 1/d}} [\texttt{Capture}(x, \boldsymbol{S}, f)] \\
&= \frac{1}{2^n} \sum_{s \in \{1,2,4,\ldots,d\}} |B_{d,s}| \cdot \frac{s}{e^4 d} .
\end{aligned}
\tag{7}
$$

To conclude the proof of Lemma 2.8, we apply Titu's Lemma, which states that for all positive real numbers $a_1, \ldots, a_k$ and $b_1, \ldots, b_k$,

$$\frac{(\sum_{i=1}^k a_i)^2}{\sum_{i=1}^k b_i} \leq \sum_{i=1}^k \frac{a_i^2}{b_i} .$$

Titu's lemma follows directly from the Cauchy-Schwartz inequality:

$$\left(\sum_{i=1}^k a_i\right)^2 = \left(\sum_{i=1}^k \frac{a_i}{\sqrt{b_i}} \cdot \sqrt{b_i}\right)^2 \leq \left(\sum_{i=1}^k \frac{a_i^2}{b_i}\right)\left(\sum_{i=1}^k b_i\right) .$$

By (7) and Titu's Lemma,

$$
\begin{aligned}
\sum_{d \in \{1,2,4,\ldots,2^{\lfloor \log_2 n \rfloor}\}} \mathop{\mathbb{E}}_{\substack{\boldsymbol{S} \subseteq [n] \\ i \in \boldsymbol{S} \text{ w.p. } 1/d}} [\mu_f(\boldsymbol{S})] &\geq \frac{1}{2^n} \sum_{d,s:d \geq s} |B_{d,s}| \cdot \frac{s}{e^4 d} \\
&= \frac{1}{e^4} \cdot \frac{1}{2^n} \sum_{\substack{d,s:d \geq s \\ |B_{d,s}| \neq 0}} \frac{(|B_{d,s}|\sqrt{s})^2}{|B_{d,s}|d} \geq \frac{1}{e^4} \cdot \frac{1}{2^n} \frac{(\sum_{d,s:d \geq s} |B_{d,s}|\sqrt{s})^2}{\sum_{d,s:d \geq s} |B_{d,s}|d} \\
&= \frac{1}{2^n} \cdot \frac{\Omega((\varepsilon \cdot 2^n)^2)}{\varepsilon\sqrt{n/\log n} \cdot 2^{n-1}} = \Omega\left(\frac{\varepsilon \sqrt{\log n}}{\sqrt{n}}\right),
\end{aligned}
$$

where we used (5) and (6) to get the final line. By an averaging argument, since the summation on the left-hand side above has $O(\log n)$ terms, at least one of the terms is $\Omega(\varepsilon/(\sqrt{n \log n}))$, completing the proof of Lemma 2.8.

3 Improvements to the Isoperimetric Inequalities from [45]

In this section, we prove Theorems 2.9 and 1.3.

First, we 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.

In Proposition 3.1 used in the proof of [45, Theorem 1.6] and stated below, we fix a subset $S \subseteq [n]$ and sample a uniformly random $\boldsymbol{z} \sim \{0,1\}^{\overline{S}}$. Then we consider $f(\cdot, \boldsymbol{z})$, a random restriction of f.

Proposition 3.1. *For a function $f \colon \{0,1\}^n \to \{0,1\}$ and a set $S \subseteq [n]$,*

$$\underset{\boldsymbol{x} \sim \{0,1\}^n}{\mathbb{E}} \left[\sqrt{I_f^-(\boldsymbol{x})} \right] \geq \underset{\boldsymbol{z} \sim \{0,1\}^{\overline{S}}}{\mathbb{E}} \left[\underset{\boldsymbol{w} \sim \{0,1\}^S}{\mathbb{E}} \left[\sqrt{I_{f(\cdot, \boldsymbol{z})}^-(\boldsymbol{w})} \right] \right].$$

Intuitively, for the restricted function $f(\cdot, \boldsymbol{z})$, we count the decreasing edges *only along dimensions in S*. To strengthen Proposition 3.1, instead of fixing S, we sample S according to the following distribution: For a real number $p \in [0,1]$, let $\mathcal{S}(p)$ denote the distribution on subsets $S \subseteq [n]$, where each $i \in [n]$ is included in S independently with probability p. We improve Proposition 3.1 to the following.

Proposition 3.2. *For a function $f \colon \{0,1\}^n \to \{0,1\}$ and a parameter $p \in [0,1]$,*

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

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. Let

$$A_f^-(x) = \begin{cases} D_f^-(x) & \text{if } f(x) = 1, \\ 0 & \text{if } f(x) = 0. \end{cases}$$

Note that $|A_f^-(x)| = I_f^-(x)$ for all $x \in \{0,1\}^n$. Hence,

$$
\begin{aligned}
\underset{\substack{\boldsymbol{S} \sim \mathcal{S}(p) \\ \boldsymbol{z} \sim \{0,1\}^{\overline{\boldsymbol{S}}}}}{\mathbb{E}} \left[\underset{\boldsymbol{w} \sim \{0,1\}^{\boldsymbol{S}}}{\mathbb{E}} \left[\sqrt{I_{f(\cdot, \boldsymbol{z})}^-(\boldsymbol{w})} \right] \right] &= \underset{\substack{\boldsymbol{S} \sim \mathcal{S}(p) \\ \boldsymbol{x} \sim \{0,1\}^n}}{\mathbb{E}} \left[\sqrt{|A_f^-(\boldsymbol{x}) \cap \boldsymbol{S}|} \right] \\
&\leq \underset{\boldsymbol{x} \sim \{0,1\}^n}{\mathbb{E}} \left[\sqrt{\underset{\boldsymbol{S} \sim \mathcal{S}(p)}{\mathbb{E}} \left[|A_f^-(x) \cap \boldsymbol{S}| \right]} \right] \\
&= \underset{\boldsymbol{x} \sim \{0,1\}^n}{\mathbb{E}} \left[\sqrt{I_f^-(x) \cdot p} \right] \\
&= \sqrt{p} \cdot \underset{\boldsymbol{x} \sim \{0,1\}^n}{\mathbb{E}} \left[\sqrt{I_f^-(\boldsymbol{x})} \right],
\end{aligned}
$$

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

Similarly, we have the analogous proposition for the proof of the robust version of the Talagrand objective (Theorem 1.9 of [45]).

Proposition 3.3. *For a function* $f \colon \{0,1\}^n \to \{0,1\}$, *a coloring* $\mathrm{col} \colon \mathcal{S}_f^- \to \{\mathrm{red}, \mathrm{blue}\}$, *and a parameter* $p \in [0,1]$,

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

Khot et al. [45] established a connection between the Talagrand objective and the function obtained after applying the *switch* operator, which was defined in [33, 41] (see also [56, Definition 4]).

Definition 3.4 (Switch operator)**.** The *switch* operator with a parameter $i \in [n]$ applied to a function $f \colon \{0,1\}^n \to \{0,1\}$ returns a function $\mathbb{S}_i[f] \colon \{0,1\}^n \to \{0,1\}$ defined as follows:

$$\mathbb{S}_i[f](x) = \begin{cases} \min(f(x^{(i \to 0)}), f(x^{(i \to 1)})) & \text{if } x = x^{(i \to 0)}; \\ \max(f(x^{(i \to 0)}), f(x^{(i \to 1)})) & \text{if } x = x^{(i \to 1)}. \end{cases}$$

The definition of the switch operator $\mathbb{S}$ can be extended to an ordered set $T = (i_1, \ldots, i_\ell)$, where $i_1, \ldots, i_\ell \in [n]$, by applying the switch operator along the dimensions in T in order:

$$\mathbb{S}_T[f] = \mathbb{S}_{i_\ell}[\mathbb{S}_{i_{\ell-1}}[\cdots \mathbb{S}_{i_2}[\mathbb{S}_{i_1}[f]] \cdots]].$$

Recall from Definition 1.1 that, for $i \in [n]$, an i-edge is a hypercube edge whose endpoints differ in the ith coordinate. Intuitively, the switch operator "repairs" each decreasing i-edge by switching the function values on its endpoints. Note that the switch operator has no effect on all nondecreasing i-edges. Dodis et al. [33] proved that, for every permutation ρ of $[n]$, the function $\mathbb{S}_\rho[f]$ is monotone. Consequently, $\mathrm{dist}(f, \mathbb{S}_\rho[f]) \geq \varepsilon_f$. Fattal and Ron [35] observed that $\mathrm{dist}(f, \mathbb{S}_\rho[f]) \leq 2 \cdot \varepsilon_f$. (See [45, Section 3.1] for a discussion of this.)

For an ordered set T and a vector $\pi \in \{Y, N\}^{|T|}$, let $T \star \pi$ denote the ordered set consisting of only the elements in T whose corresponding position in π is Y. For example, if $T = (5, 1, 7, 4, 2, 9)$ and $\pi = (Y, Y, N, Y, N, N)$, then $T \star \pi = (5, 1, 4)$. For any finite set S, let $\mathcal{P}(S)$ denote the uniform distribution supported on the set of all permutations of S.

The following theorem is implicit in [45, Section 4].

Theorem 3.5. *For a function* $f \colon \{0,1\}^n \to \{0,1\}$, *there exists a constant* $C > 0$ *such that*

$$C \cdot \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \{0,1\}^n} \left[\sqrt{I_f^-(\boldsymbol{x})} \right] \geq \mathop{\mathbb{E}}_{\boldsymbol{\lambda} \sim \mathcal{P}([n])} [\mathrm{dist}(f, \mathbb{S}_{\boldsymbol{\lambda}}[f])] - \mathop{\mathbb{E}}_{\substack{\boldsymbol{\rho} \sim \mathcal{P}([n]) \\ \boldsymbol{\pi} \sim \{Y,N\}^n}} [\mathrm{dist}(f, \mathbb{S}_{\boldsymbol{\rho} \star \boldsymbol{\pi}}[f])].$$

Consider the quantity $\mathbb{E}_{\boldsymbol{w} \sim \{0,1\}^{\boldsymbol{S}}} \left[\sqrt{I_{f(\cdot,\boldsymbol{z})}^-(\boldsymbol{w})} \right]$ in the statement of Proposition 3.2. From Theorem 3.5, we get

$$C \cdot \mathop{\mathbb{E}}_{\boldsymbol{w} \sim \{0,1\}^{\boldsymbol{S}}} \left[\sqrt{I_{f(\cdot,\boldsymbol{z})}^-(\boldsymbol{w})} \right] \geq \mathop{\mathbb{E}}_{\boldsymbol{\lambda} \sim \mathcal{P}(\boldsymbol{S})} [\mathrm{dist}(f(\cdot,\boldsymbol{z}), \mathbb{S}_{\boldsymbol{\lambda}}[f(\cdot,\boldsymbol{z})])] - \mathop{\mathbb{E}}_{\substack{\boldsymbol{\rho} \sim \mathcal{P}(\boldsymbol{S}) \\ \boldsymbol{\pi} \sim \{Y,N\}^{|\boldsymbol{S}|}}} [\mathrm{dist}(f(\cdot,\boldsymbol{z}), \mathbb{S}_{\boldsymbol{\rho} \star \boldsymbol{\pi}}[f(\cdot,\boldsymbol{z})])].$$

Taking the expectation over $\boldsymbol{S} \sim \mathcal{S}(p)$, we get

15

$$C \cdot \mathop{\mathbb{E}}_{\substack{S \sim \mathcal{S}(p) \\ z \sim \{0,1\}^{\overline{S}}}} \left[\mathop{\mathbb{E}}_{w \sim \{0,1\}^S} \left[\sqrt{I_{f(\cdot, z)}^-(w)} \right] \right] \geq \mathop{\mathbb{E}}_{\substack{S \sim \mathcal{S}(p) \\ z \sim \{0,1\}^{\overline{S}}}} \left[\mathop{\mathbb{E}}_{\lambda \sim \mathcal{P}(S)} [\mathrm{dist}(f(\cdot, z), \mathbb{S}_\lambda[f(\cdot, z)])] \right]$$

$$- \mathop{\mathbb{E}}_{\substack{S \sim \mathcal{S}(p) \\ z \sim \{0,1\}^{\overline{S}}}} \left[\mathop{\mathbb{E}}_{\substack{\rho \sim \mathcal{P}(S) \\ \pi \sim \{Y,N\}^{|S|}}} [\mathrm{dist}(f(\cdot, z), \mathbb{S}_{\rho \star \pi}[f(\cdot, z)])] \right]. \quad (8)$$

For ease of notation, let $\Psi_f(p)$ denote the first term on the right-hand side of (8):

$$\Psi_f(p) = \mathop{\mathbb{E}}_{\substack{S \sim \mathcal{S}(p) \\ z \sim \{0,1\}^{\overline{S}}}} \left[\mathop{\mathbb{E}}_{\lambda \sim \mathcal{P}(S)} [\mathrm{dist}(f(\cdot, z), \mathbb{S}_\lambda[f(\cdot, z)])] \right].$$

We can also express the second term on the right-hand side of (8) using this notation:

$$\mathop{\mathbb{E}}_{\substack{S \sim \mathcal{S}(p) \\ z \sim \{0,1\}^{\overline{S}}}} \left[\mathop{\mathbb{E}}_{\substack{\rho \sim \mathcal{P}(S) \\ \pi \sim \{Y,N\}^{|S|}}} [\mathrm{dist}(f(\cdot, z), \mathbb{S}_{\rho \star \pi}[f(\cdot, z)])] \right] = \Psi_f(p/2),$$

since the probability that an arbitrary $i \in [n]$ appears in $\rho \star \pi$ is equal to the product of the probability that $i \in S$ and the probability that the position where i appears in π has Y. By (8) and Proposition 3.2, for every $p \in [0, 1]$, we get

$$\Psi_f(p) - \Psi_f(p/2) \leq C \cdot \mathop{\mathbb{E}}_{\substack{S \sim \mathcal{S}(p) \\ z \sim \{0,1\}^{\overline{S}}}} \left[\mathop{\mathbb{E}}_{w \sim \{0,1\}^S} \left[\sqrt{I_{f(\cdot, z)}^-(w)} \right] \right]$$

$$\leq C \cdot \sqrt{p} \cdot \mathop{\mathbb{E}}_{x \sim \{0,1\}^n} \left[\sqrt{I_f^-(x)} \right]. \quad (9)$$

It follows from the analysis of Dodis et al. [33] that $\Psi_f(1) \geq \varepsilon_f$. Also note that $\Psi_f(0) = 0$. We apply (9) for $p = 1, \frac{1}{2}, \frac{1}{4} \ldots$ and consider the resulting telescoping sum:

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

$$\leq C \cdot \sum_{i=0}^{\infty} 2^{-i/2} \cdot \mathop{\mathbb{E}}_{x \sim \{0,1\}^n} \left[\sqrt{I_f^-(x)} \right]$$

$$\leq \frac{\sqrt{2}}{\sqrt{2} - 1} C \cdot \mathop{\mathbb{E}}_{x \sim \{0,1\}^n} \left[\sqrt{I_f^-(x)} \right],$$

completing the proof of Theorem 1.3. Similarly, Proposition 3.3 implies Theorem 2.9.

4 A Nonadaptive Lower Bound for Erasure-Resilient Testers

In this section, we prove Theorem 1.5 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 $f : \{0, 1\}^n \rightarrow \{0, 1, \bot\}$ that are hard to distinguish for every nonadaptive tester and then applying Yao's Minimax principle [64].

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 has property $\mathcal{P}$ if there exists a completion of f that has $\mathcal{P}$. A partially erased function is ε-far from having property $\mathcal{P}$ if every completion of f is ε-far from having property $\mathcal{P}$.

Proof of Theorem 1.5. 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 most of the probability mass of $\mathcal{D}^-$ is on 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 whereas $\mathcal{D}^-$ is mostly supported on functions that 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.

Definition 4.1. For a set $S \subseteq [n]$, let $\overline{S}$ denote the complement set $[n] \setminus S$, and let $\{0,1\}^S$ denote the restriction of $\{0,1\}^n$ to the dimensions in S. 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 objects needed for defining the distributions $\mathcal{D}^+$ and $\mathcal{D}^-$. As before, we use the convention that random variables are boldface whereas fixed quantities use standard typeface.

- **The set M of *control dimensions* and the *control substring* x_M.** We partition $[n]$ into two sets of size $n/2$: the set M of *control dimensions* and the set $\overline{M}$. We call x_M the *control substring* of x. The random variable $\boldsymbol{M}$ is a uniformly random subset of $[n]$ of size $n/2$.

- **The *subcube partition set* P_M and *action* subcubes.** Let $\Psi_M = \left\{ x_M \in \{0,1\}^M : |x_M| = \frac{n}{4} \right\}$ denote the set of all control substrings of x which lie in the middle layer of the subcube $\{0,1\}^M$. 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 *subcube partition set* P_M is a subset of Ψ_M. The random variable $\boldsymbol{P_M}$ is a uniformly random subset of Ψ_M, that is, each $x_M \in \Psi_M$ is included in $\boldsymbol{P_M}$ independently with probability $1/2$.

- **The functions g_{M,P_M}.** Recall the parameter $\kappa \in (0,1/2)$ from Theorem 1.5. For a fixed setting of $M \subset [n]$ of size $n/2$ and a subcube partition set $P_M \subseteq \Psi_M$, we define the function $g_{M,P_M}\colon \{0,1\}^n \to \{0,1,\perp,0^\star,1^\star\}$ as follows:

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

To sample a function from $\mathcal{D}^+$ and $\mathcal{D}^-$, we pick a uniformly random subset $\boldsymbol{M} \subset [n]$ of size $n/2$ of control dimensions and a uniformly random subcube partition set $\boldsymbol{P_M} \subseteq \Psi_{\boldsymbol{M}}$. A function sampled

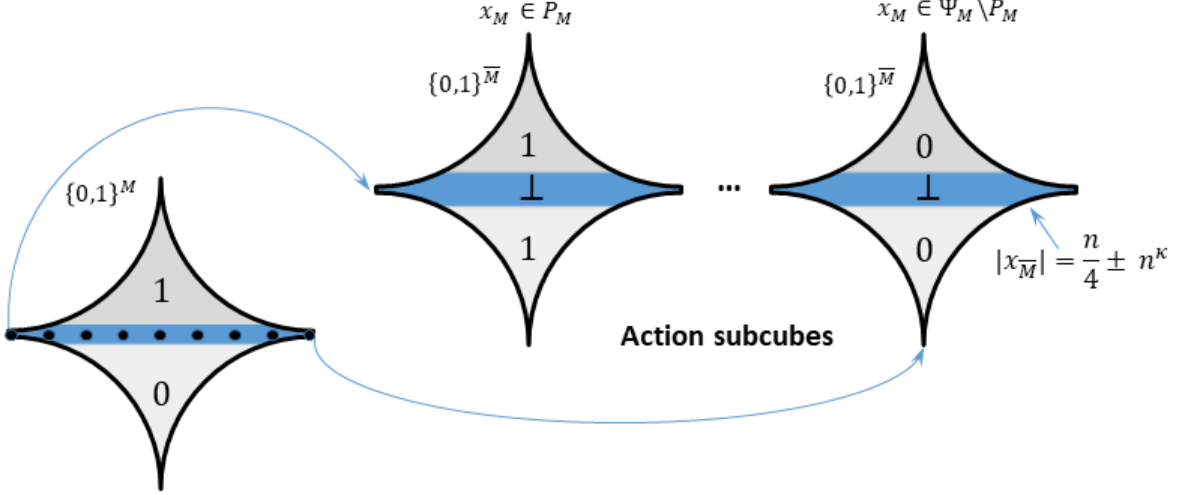Figure 2: Functions $\boldsymbol{f}^+ \sim \mathcal{D}^+$ defined with respect to control dimensions M and the subcube partition set P_M.

from $\mathcal{D}^+$ and $\mathcal{D}^-$ is identical to $g_{M,\boldsymbol{P_M}}$ on points $x \in \{0,1\}^n$ for which $g_{M,\boldsymbol{P_M}}(x) \in \{0,1,\perp\}$, but differs on the remaining values (see Figures 2 and 3). 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

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

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

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

Lemma 4.2. *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. Consider a partially erased function $f\colon \{0,1\}^n \to \{0,1\}$ in the support of $\mathcal{D}^+$ or $\mathcal{D}^-$. Let M be the set of control dimensions used in defining f. The function f is erased in the middle $2n^\kappa + 1$ layers of every action subcube $\{0,1\}^{\overline{M}}$ for which the control substring x_M is in the middle layer of the subcube $\{0,1\}^M$. Since $|M| = |\overline{M}| = n/2$, the number of erased points 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 $O(1/n^{1-\kappa})$. □

18

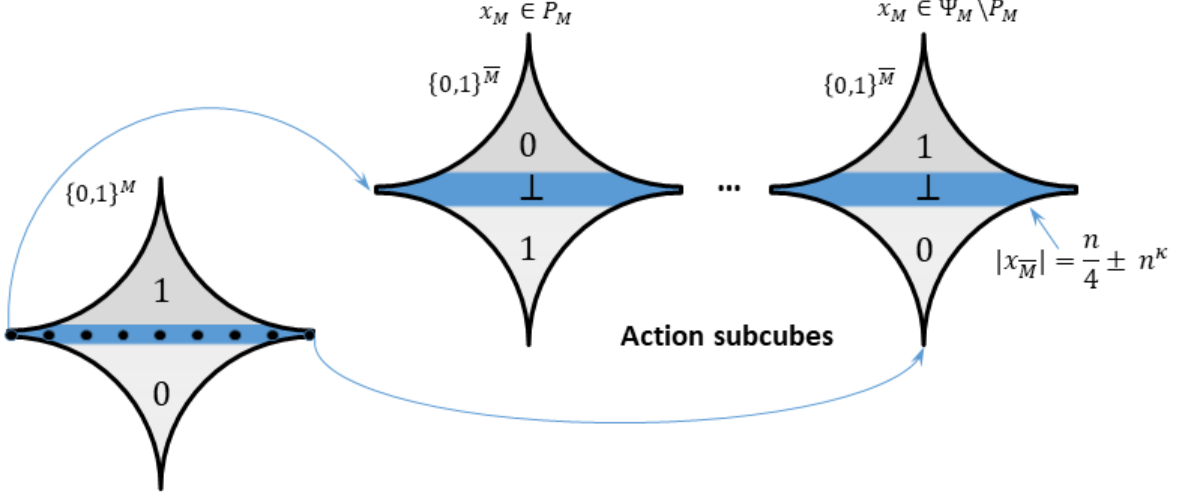Figure 3: Functions $\boldsymbol{f}^- \sim \mathcal{D}^-$ defined with respect to control dimensions M and the subcube partition set P_M.

4.1 Distance to Monotonicity, Unateness, and Being a Junta

In this section, we analyze the distance to monotonicity, unateness, and being an $n/2$-junta for functions in the support of our hard distributions, $\mathcal{D}^+$ and $\mathcal{D}^-$.

Lemma 4.3. *Every function f^+ in the support of $\mathcal{D}^+$ is monotone, unate, and an $n/2$-junta.*

Proof. Consider a partially erased function f^+ in the support of $\mathcal{D}^+$. We show that f^+ can be completed to a monotone $n/2$-junta. Let M be the set of control dimensions and P_M be the subcube partition set used in defining f^+. Define a completion $f' \colon \{0,1\}^n \to \{0,1\}$ of f^+ as follows. For all $x \in \{0,1\}^n$ with $f^+(x) = \perp$,

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

Since $|M| = n/2$ and f' depends only on coordinates in M, it is an $n/2$-junta.

To prove that f' is monotone, we show it is the disjunction of monotone functions. Let $h :$ $\{0,1\}^n \to \{0,1\}$ be the indicator for $|x_M| > n/4$, that is, $h(x) = 1$ iff $|x_M| > n/4$. For every $y \in P_M$, let $h_y : \{0,1\}^n \to \{0,1\}$ be the indicator for $x_M = y_M$. Functions h and h_y for all $y \in P_M$ are monotone, and f' is the disjunction of these functions. Hence, f^+ can be completed to a monotone $n/2$-junta. Since monotone functions are unate, Lemma 4.5 holds. $\qquad\square$

Next we define an event called `FAR`, observe that it happens with high probability, and show that functions distributed according to $\mathcal{D}^-$ conditioned on `FAR` are far from the three properties we are considering.

Definition 4.4 (Event `FAR` and distribution $\hat{\mathcal{D}}^-$)**.** Let `FAR` be the event that

$$\frac{|\Psi_M|}{3} \le |P_M| \le \frac{2|\Psi_M|}{3}.$$

19

Define distribution $\hat{\mathcal{D}}^- = \mathcal{D}^-|_{\texttt{FAR}}$.

Recall that each element of Ψ_M is included in $\boldsymbol{P}_M$ independently with probability $1/2$. Therefore, for sufficiently large n, by a Hoeffding bound,

$$\Pr[\overline{\texttt{FAR}}] \leq 2e^{2|\Psi_M| \cdot \frac{1}{6^2}} \leq \frac{1}{30} \tag{10}$$

over the draw of $\boldsymbol{f}^- \sim \mathcal{D}^-$.

Lemma 4.5. *Every function f^- in the support of $\hat{\mathcal{D}}^-$ 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 $\hat{\mathcal{D}}^-$, and let M and P_M be the set of control dimensions and the subcube partition set used in defining f^-, respectively. We start by proving that f^- is far from monotone. Even though it is technically not necessary, because it will follow from the fact that f^- is far from unate, we choose to do it for ease of presentation. After that, we prove Claim 4.8 that states that f^- is far from being an $n/2$-junta. Finally, we build on ideas in these two proofs to show in Claim 4.9 that f^- is far from unate.

Definition 4.6 (Decreasing pair). *A pair of points $\{x, y\}$ is decreasing with respect to a function f if $x \prec y$, but $f(x) = 1$ and $f(y) = 0$.*

First, we prove that f^- is ε-far from monotone for some $\varepsilon = \Omega(1/\sqrt{n})$ by showing that there exists a large matching of decreasing pairs with respect to f^-. Consider all action subcubes $\{0,1\}^{\overline{M}}$ for which $x_M \in P_M$. By definition, each such subcube is $n/2$-dimensional, and hence contains $2^{n/2}$ points. Since $\texttt{FAR}$ holds, there are at least $\binom{n/2}{n/4} \cdot \frac{1}{3} = \Omega(\frac{2^{n/2}}{\sqrt{n}})$ such subcubes. By standard arguments (see[5], e.g., [39, Lemma 22]), in each such action subcube, there is a matching of size $\Omega(2^{n/2})$ consisting of decreasing pairs with respect to f^-. The values of at least half of the points in the matching need to be changed to make the function f^- monotone. Moreover, at least two thirds of the points in the action subcube are nonerased, and this matching contains all of them. Since $\Omega(1/\sqrt{n})$ fraction of points participates in action subcubes with $x_M \in P_M$, and the values of at least a third of these points need to be changed to make f^- monotone, the distance from f^- to monotonicity is $\Omega(1/\sqrt{n})$.

Next, recall from Definition 1.1 that, for $i \in [n]$, a hypercube edge whose endpoints differ in the ith coordinate is called an i-edge.

Definition 4.7 (A constant edge, independence of a variable). *An edge $\{x, y\}$ is constant with respect to a function f if $f(x) = f(y)$. An edge that is neither decreasing nor constant is called increasing. A function f is independent of a variable $i \in [n]$ if all i-edges are constant with respect to f.*

Claim 4.8. *Every function f^- in the support of $\hat{\mathcal{D}}^-$ has distance at least $\varepsilon = \Omega\left(\frac{1}{\sqrt{n}}\right)$ from being an $n/2$-junta.*

---

[5] Fischer et al. [39, Lemma 22] prove that *trimmed anti-oligarchy* functions are far from monotone by applying Hall's Theorem to argue that there is a large matching of decreasing edges. Given $B \subseteq [n]$, the *trimmed anti-oligarchy* function $f_B(x)$ is 1 if the Hamming weight of x is large, 0 if it is small, and is equal to $1 - Maj(x_B)$, otherwise. In particular, when $B = [n]$, this function is equal to $1 - Maj(x)$ on a constant fraction of points x (located in the middle layers of the hypercube). The functions on the actions subcubes under consideration in our lower are also anti-majorities on a constant fraction of all points x (located in all but the middle layers of the subcubes).

Proof. Observe that f^- is an $n/2$-junta iff it is independent of at least $n/2$ 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 increasing i-edges $\{x, y\}$. 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 $\Psi_M = \{x_M \in \{0,1\}^M : |x_M| = \frac{n}{4}\}$ is the set of all control substrings of x that lie in the middle layer of the subcube $\{0,1\}^M$. We define, for every dimension $i \in M$,

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

Note that $x_i \neq y_i$ and, by construction of functions g_{M,P_M}, we have $f^-(y) = g_{M,P_M}(y) = y_i$. Therefore, $f^-(x) \neq f^-(y)$ for all i-edges $\{x, y\} \in \mathcal{M}_i$. (Note that all edges in $\mathcal{M}_i$ are increasing. This fact will be used in the proof of Claim 4.9.) For each $x_M \in \Psi_M$, more than a third 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 $\varepsilon = \Omega(\frac{1}{\sqrt{n}})$. $\square$

Claim 4.9. *Every function f^- in the support of $\hat{\mathcal{D}}^-$ has distance at least $\varepsilon = \Omega\left(\frac{1}{\sqrt{n}}\right)$ from unateness.*

Proof. To show that f^- is ε-far from unate, let $r \in \{0,1\}^n$ be an arbitrary assignment of directions for each variable in $[n]$ (that is, $r_i = 0$ signifies that the variable i is monotone non-decreasing, and $r_i = 1$ signifies that the variable i is monotone non-increasing). We use $\oplus$ to represent bit-wise XOR of two vectors. It suffices to show that the function $g \colon \{0,1\}^n \to \{0,1\}$, given by $g(x) = f^-(x \oplus r)$, is ε-far from monotone.

Notice that if $r_i = 1$ for some $i \in M$, the function g is $\Omega(1/\sqrt{n})$-far from monotone. To see this, recall that we constructed a matching $\mathcal{M}_i$ of $\Omega(2^n/\sqrt{n})$ increasing i-edges in f^- in the proof of Claim 4.8. If $r_i = 1$, these i-edges are decreasing with respect to g, so the function is $\Omega(1/\sqrt{n})$-far from monotone. Henceforth, we assume $r_i = 0$ for all $i \in M$.

Let R be the subset of dimensions $\{i \in \overline{M} : r_i = 1\}$. First consider the case when $|R| \leq n/4$. Recall that each action subcube contains $2^{n/2}$ points. Consider all action subcubes $\{0,1\}^{\overline{M}}$ for which $x_M \in P_M$. Since FAR holds, there are $\Omega(\frac{2^{n/2}}{\sqrt{n}})$ such subcubes, as we argued before. We show that each of them contains a matching of $\Omega(2^{n/2})$ decreasing pairs with respect to g, thus proving that g is $\Omega(1/\sqrt{n})$-far from monotone.

We further partition action subcubes into smaller subcubes, each of which contains all points $x \in \{0,1\}^n$ with the same x_R (and the same control substring x_M). Let $\mathcal{C}$ be the set of all such subcubes with $x_M \in P_M$ and the Hamming weight $|x_R|$ in the range $\left[\frac{|R|}{2} - \sqrt{n}, \frac{|R|}{2} + \sqrt{n}\right]$. Since a constant fraction of $z \in \{0,1\}^R$ is in the specified range, $\mathcal{C}$ contains $\Omega(2^{|R|})$ smaller subcubes

21

for each action subcube we are considering. It remains to show that each subcube in $\mathcal{C}$ contains a matching of $\Omega(2^{n/2-|R|})$ decreasing pairs with respect to g.

By definition of f^-, the value of $f^-(y)$ depends only on the Hamming weight of $y_{\overline{M}}$ for all points y with $y_M \in P_M$. In particular, $f^-(y) = 0$ if $|y_{\overline{M}}| \geq \frac{n}{4} + n^{\kappa}$ and $f^-(y) = 1$ if $|y_{\overline{M}}| \leq \frac{n}{4} - n^{\kappa}$. That is, for every x with $x_M \in P_M$,

$$g(x) = f^-(x \oplus r) = \begin{cases} 0 \text{ if } |(x \oplus r)_{\overline{M}}| \geq \frac{n}{4} + n^{\kappa}; \\ 1 \text{ if } |(x \oplus r)_{\overline{M}}| \leq \frac{n}{4} - n^{\kappa}. \end{cases}$$

Recall that $R \subseteq \overline{M}$. Since $(x \oplus r)_i = 1 - x_i$ for all $i \in R$, and $(x \oplus r)_i = x_i$ for all $i \notin R$, we get

$$|(x \oplus r)_{\overline{M}}| = |(x \oplus r)_R| + |(x \oplus r)_{\overline{M} \backslash R}| = |R| - |x_R| + |x_{\overline{M} \backslash R}|.$$

Fix a subcube in $\mathcal{C}$ and note that it is of the form $\{0,1\}^{\overline{M} \backslash R}$. Recall that $|\overline{M}| = n/2$, so the subcube contains $2^{n/2-|R|}$ points. The quantity $|R| - |x_R|$ is the same for all points x in the subcube. Moreover, this quantity is in the range $\left[\frac{|R|}{2} - \sqrt{n}, \frac{|R|}{2} + \sqrt{n}\right]$. We claim that g evaluates to 1 on the points in the bottom layers of the subcube, and that it evaluates to 0 on the points in the top layers. Specifically, if $|x_{\overline{M} \backslash R}| \leq \frac{|\overline{M} \backslash R|}{2} - \sqrt{n} - n^{\kappa}$, which holds for a constant fraction of points x in the subcube, then

$$|(x \oplus r)_{\overline{M}}| = |R| - |x_R| + |x_{\overline{M} \backslash R}| \leq \frac{|R|}{2} + \sqrt{n} + \frac{|\overline{M} \backslash R|}{2} - \sqrt{n} - n^{\kappa} = \frac{|\overline{M}|}{2} - n^{\kappa} = \frac{n}{4} - n^{\kappa}.$$

That is, in this case, $g(x) = 0$. Similarly, if $|y_{\overline{M} \backslash R}| \geq \frac{|\overline{M} \backslash R|}{2} + \sqrt{n} + n^{\kappa}$, which also holds for a constant fraction of points in the subcube, then

$$|(y \oplus r)_{\overline{M}}| \geq \frac{|R|}{2} - \sqrt{n} + \frac{|\overline{M} \backslash R|}{2} + \sqrt{n} + n^{\kappa} = \frac{n}{4} + n^{\kappa}.$$

That is, in this case, $g(y) = 0$. Similarly to the case of monotonicity, by a standard argument [39, Lemma 22], there is a matching of decreasing pairs $\{x, y\}$ with respect to g that matches all the points x and y described above and, consequently has size $\Omega(2^{n-|R|}/\sqrt{n})$. We obtained the desired bound on the matching size and, therefore, on the distance to unateness.

The argument for the case when $|R| > n/4$ follows similarly by symmetry. We consider action subcubes for which the control substrings are not in P_M. For each such subcube, f^- evaluates to a majority on $x_{\overline{M}}$, and variables in R are flipped in g, allowing us to construct a large matching of decreasing pairs with respect to g. $\qquad\square$

This completes the proof of Lemma 4.5. $\qquad\square$

4.2 Indistinguishability of the Hard Distributions

Next, we show that the distributions $\mathcal{D}^+$ and $\hat{\mathcal{D}}^-$ are hard to distinguish for nonadaptive testers. Consider a deterministic tester that makes q queries. Let $\boldsymbol{A}_q(f)$ be the sequence of query-answer pairs $(x, f(x))$ obtained by the tester on input f. For every distribution $\mathcal{D}$ on input functions, define $\mathcal{D}$-view to be the distribution on $\boldsymbol{A}(\boldsymbol{f}^+)$ when $\boldsymbol{f}^+ \sim \mathcal{D}$. We use the version of Yao's principle stated in [58] 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 $\hat{\mathcal{D}}^-$, on positive and negative instances, respectively, for which the statistical distance between $\mathcal{D}^+$-view and $\hat{\mathcal{D}}^-$-view is less than $1/3$.

We start by analyzing how a tester can distinguish $\mathcal{D}^+$ and $\mathcal{D}^-$. The key point is the only way to do it 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 the erased layers, the other above the erased layers. If the tester queries no such pair, then its view (that is, the distribution on the sequence of query-answer pairs it obtains) is identical for the two cases: $\boldsymbol{f}^+ \sim \mathcal{D}^+$ and $\boldsymbol{f}^- \sim \mathcal{D}^-$.

Let $\mathtt{BAD}$ 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. Next, we show that conditioned on $\mathtt{BAD}$ not occurring, the view of the tester is the same for both distributions.

Claim 4.10. $\mathcal{D}^+\text{-}view|_{\overline{\mathtt{BAD}}} = \mathcal{D}^-\text{-}view|_{\overline{\mathtt{BAD}}}$.

Proof. Consider an arbitrary fixed set of control dimensions $M \subset [n]$ of size $n/2$, and let Ψ_M be the set of control substrings. Partition the set of queries $x_1, \ldots, x_q \in \{0,1\}^n$ with nonerased values according to their control substrings in Ψ_M; namely, for $z \in \Psi_M$, let $Q_z \subset \{x_1, \ldots, x_q\}$ be the set of queries x for which $x_M = z$ and $x_{\overline{M}}$ does not lie in the middle $2n^\kappa$ layers of its action subcube. This partition depends only on M, and not on the set $\boldsymbol{P}_M$.

Suppose, furthermore, that any two queries $x, y \in \{0,1\}^n$ falling in the same action subcube (i.e., from the same part Q_z) are either both above the erased layers or both below the erased layers, and notice that this implies $\boldsymbol{f}(x) = \boldsymbol{f}(y)$ whenever $x, y \in Q_z$, for $\boldsymbol{f}$ sampled from $\mathcal{D}^+$ as well as $\mathcal{D}^-$. We will show that the distribution over answers to queries for a fixed M (and a random $\boldsymbol{P}_M$) is exactly the same for $\mathcal{D}^+$ and $\mathcal{D}^-$. In both cases, the answer to a query $x \in \{0,1\}^n$ is a function of the control substring x_M and is independent for two queries $x, y \in \{0,1\}^n$ with different control substrings. In fact, for each control substring $z \in \Psi_M$, the value of every $\boldsymbol{f}(x)$ with $x_M = z$ is a uniformly random bit.

In order to see why, notice that for $\boldsymbol{f}^+ \sim \mathcal{D}^+$, every $x \in Q_z$ has value $\boldsymbol{f}^+(x) = 1$ if and only if $z \in \boldsymbol{P}_M$, which occurs with probability $1/2$. For $\boldsymbol{f}^- \sim \mathcal{D}^-$, we have two cases. The first case is when $x \in Q_z$ has $|x_{\overline{M}}| \leq n/4 - n^\kappa$; then, $\boldsymbol{f}^-(x) = 1$ if and only if $z \in \boldsymbol{P}_M$, which occurs with probability $1/2$. The second case is when $x \in Q_z$ has $|x_{\overline{M}}| \geq n/4 + n^\kappa$; then, $\boldsymbol{f}^+(x) = 0$ if and only if $z \notin \boldsymbol{P}_M$, which occurs with probability $1/2$. $\square$

Claim 4.11. *For a deterministic nonadaptive tester making $q \leq 2^{n^\kappa}$ queries,*

$$\Pr[\mathtt{BAD}] < 1/8$$

over the draw of $\boldsymbol{M} \subset [n]$ of size $n/2$.

Proof. We want to bound the probability over the draw of $\boldsymbol{M}$ that any two queries $x, y \in \{0,1\}^n$ which have the same control substring are either both above the erased layers, or both below the erased layers in their action subcube. In particular, observe that the Hamming weights of any such x and y must differ by at least $2n^\kappa + 2$. Consequently, x and y differ on at least $2n^\kappa + 2$ bits. Next, we upper bound the probability that two queries $x, y \in \{0,1\}^n$ which differ on at least $2n^\kappa + 2$ dimensions have the same setting of coordinates in $\boldsymbol{M}$.

Let $T = \{i \in [n] : x_i \neq y_i\}$ 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

23

of $[n]$ of size $n/2$,

$$\Pr_{\boldsymbol{M}}[T \cap \boldsymbol{M} = \emptyset] = \frac{\binom{n-|T|}{n/2}}{\binom{n}{n/2}} = \frac{\frac{(n-|T|)!}{(n/2)!\cdot(n/2-|T|)!}}{\frac{n!}{(n/2)!\cdot(n/2)!}}$$

$$= \frac{(n/2)!}{(n/2-|T|)!} \cdot \frac{(n-|T|)!}{n!}$$

$$= \frac{n/2 \cdot (n/2-1) \cdots (n/2-|T|+1)}{n \cdot (n-1) \cdots (n-|T|+1)} \leq 2^{-|T|}.$$

Let $\mathtt{BAD}$ 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,

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

$\square$

Definition 4.12 (Notation for statistical distance). 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 $\mathcal{D}_2$ is at most δ.

Lemma 4.13. *For a deterministic nonadaptive tester making $q \leq 2^{n^\kappa}$ queries,*

$$\mathcal{D}^+\text{-view} \approx_{9/28} \hat{\mathcal{D}}^-\text{-view}.$$

Proof. By Claim 4.10, conditioned on $\mathtt{BAD}$ not occurring, the view of the tester is the same for distributions $\mathcal{D}^+$ and $\mathcal{D}^-$:

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

Conditioning on $\overline{\mathtt{BAD}}$ does not significantly change the view distributions. We use the following claim [58, Claim 4] to formalize this statement.

Claim 4.14 ([58]). *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 Claims 4.11 and 4.14 twice, we get

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

Similarly, recalling that $\hat{\mathcal{D}}^- = \mathcal{D}^-|_{\mathtt{FAR}}$, by (10) and Claim 4.14, we get

$$\mathcal{D}^-\text{-view} \approx_{1/29} \mathcal{D}^-\text{-view}|_{\mathtt{FAR}} = \hat{\mathcal{D}}^-\text{-view}.$$

Since $1/7 + 1/7 + 1/29 < 9/28$, this completes the proof of Lemma 4.13. $\square$

Theorem 1.5 follows by Yao's Principle. $\square$

5 From Tolerant Testing to Distance Approximation

The following theorem shows how to convert a tolerant tester to a distance approximation algorithm.

Theorem 5.1. *Consider input objects whose size is measured with respect to a parameter n, and let $\mathcal{P}$ be a property of these objects. Let $g(n)$ be a polynomial in n such that $g(n) = \omega(1)$. Let $\mathcal{A}$ be a tolerant tester of the property $\mathcal{P}$ that gets a parameter $\varepsilon \in (0, 1/2)$ and oracle access to an input object f, makes $Q_{\mathcal{A}}(n, \varepsilon)$ queries, and outputs close or far as follows:*

1. *If $\mathrm{dist}(f, \mathcal{P}) \leq \frac{\varepsilon}{g(n)}$, then $\mathcal{A}(\varepsilon, f)$ outputs close with probability at least $2/3$;*

2. *If $\mathrm{dist}(f, \mathcal{P}) \geq \varepsilon$, then $\mathcal{A}(\varepsilon, f)$ outputs far with probability at least $2/3$.*

Suppose $Q_{\mathcal{A}}(n, \varepsilon)$ has at least linear dependence on $1/\varepsilon$. Then, there exists a distance approximation algorithm $\mathcal{B}$ for the property $\mathcal{P}$ that, given a parameter $\alpha \in (0, 1/2)$ and oracle access to an input object f such that $\mathrm{dist}(f, \mathcal{P}) \geq \alpha$, returns an estimate that, with probability at least $2/3$, is a $2g(n)$-approximation to $\mathrm{dist}(f, \mathcal{P})$. The query complexity of $\mathcal{B}$ is $O(Q_{\mathcal{A}}(n, \alpha) \cdot \log\log(1/\alpha))$.

Proof. The distance approximation algorithm $\mathcal{B}$ is described in Algorithm 4.

Algorithm 4: Distance Approximation Algorithm $\mathcal{B}(\alpha, f)$

> **input** : minimum distance parameter $\alpha \in (0, 1/2)$; oracle access to the input object f.
> **output:** $2g(n)$ approximation $\hat{\varepsilon}$ to $\mathrm{dist}(f, \mathcal{P})$.

1 **for** $i = 1$ to $\left\lfloor \log \frac{1}{\alpha} \right\rfloor$ **do**
2   Set $c \leftarrow 0$ and $t \leftarrow \lceil 18 \ln(3 \log(1/\alpha)) \rceil$.
3   **repeat** t **times**
4   Run $\mathcal{A}(2^{-i}, f)$. If it outputs far, $c \leftarrow c + 1$.
5   **if** $c \geq t/2$ **then return** $\hat{\varepsilon} = 2^{-i+1}$.
6 **return** $\hat{\varepsilon} = \alpha$.

For each $i \in \left[\left\lfloor \log \frac{1}{\alpha} \right\rfloor\right]$, let E_i denote the event that c calculated by the algorithm in the ith iteration of the **for** loop satisfies $c \geq t/2$ when $\mathrm{dist}(f, \mathcal{P}) \geq 2^{-i}/g(n)$ and $c < t/2$ when $\mathrm{dist}(f, \mathcal{P}) \leq 2^{-i}$. Event $\overline{E_i}$ occurs when $\mathcal{A}$ errs in at least $t/2$ of its runs in Step 4. In each run, $\mathcal{A}$ errs with probability at most $1/3$. By Hoeffding bound, for each $i \in [\lfloor \log(1/\alpha) \rfloor]$, the probability $\Pr[\overline{E_i}] < 1/(3 \log(1/\alpha))$. Let $E = \bigcap_{i \in [\lfloor \log(1/\alpha) \rfloor]} E_i$. By the union bound, $\Pr[\overline{E}] < 1/3$, which implies that $\Pr[E] > 2/3$. For the rest of the proof, condition on the event E happening. If $\hat{\varepsilon} = 2^{-i+1}$, then it implies that for all $j < i$, the algorithm determined that $\mathrm{dist}(f, \mathcal{P}) \leq 2^{-j}$. Substituting $j = i - 1$ yields $\mathrm{dist}(f, \mathcal{P}) \leq 2^{-i+1} = \hat{\varepsilon}$. Similarly, in the ith iteration, the algorithm determined that $\mathrm{dist}(f, \mathcal{P}) \geq 2^{-i}/g(n) = \hat{\varepsilon}/(2g(n))$. Hence, with probability at least $2/3$, the output $\hat{\varepsilon}$ of the algorithm $\mathcal{B}$ satisfies $\mathrm{dist}(f, \mathcal{P}) \leq \hat{\varepsilon} \leq 2g(n) \cdot \mathrm{dist}(f, \mathcal{P})$, completing the proof of the approximation guarantee.

The query complexity of $\mathcal{B}$ is at most $t \cdot \sum_{i=1}^{\lfloor \log(1/\alpha) \rfloor} Q_{\mathcal{A}}(n, 2^{-i})$. Since $Q_{\mathcal{A}}(n, \varepsilon)$ has at least linear dependence on $1/\varepsilon$, the sum is dominated by the last term. Therefore, the query complexity is $O(Q_{\mathcal{A}}(n, \alpha) \cdot \log\log(1/\alpha))$. $\qquad\square$

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. *Theory of Computing*, 16(3):1–36, 2020.

[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] Hadley Black, Deeparnab Chakrabarty, and C. Seshadhri. A $o(d) \cdot$ polylog n monotonicity tester for Boolean functions over the hypergrid $[n]^d$. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2133–2151, 2018.

[10] 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 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1975–1994, 2020.

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

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

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

[14] Eric Blais, Clément L. Canonne, Talya Eden, Amit Levi, and Dana Ron. Tolerant junta testing and the connection to submodular optimization and function isomorphism. *ACM Trans. on Computation Theory*, 11(4):24:1–24:33, 2019.

[15] 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.

[16] 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.

[17] 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(6), 2013.

[18] 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.

[19] 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.

[20] 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.

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

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

[23] Deeparnab Chakrabarty and C. Seshadhri. Adaptive Boolean monotonicity testing in total influence time. In *Proceedings of Innovations in Theoretical Computer Science (ITCS)*, pages 20:1–20:7, 2019.

[24] 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.

[25] 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.

[26] Xi Chen, Rocco A. Servedio, Li-Yang Tan, Erik Waingarten, and Jinyu Xie. Settling the query complexity of non-adaptive junta testing. *J. ACM*, 65(6):40:1–40:18, 2018. `doi:10.1145/3213772`.

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

[28] 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.

[29] 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.

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

[31] Anindya De, Elchanan Mossel, and Joe Neeman. Junta correlation is testable. In *Proceedings of IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1549–1563, 2019.

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

[33] 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.

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

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

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

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

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

[39] 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.

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

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

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

[43] 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.

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

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

[46] 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.

[47] 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.

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

[49] Amit Levi, Ramesh Krishnan S. Pallavoor, Sofya Raskhodnikova, and Nithin Varma. Erasure-resilient sublinear-time graph algorithms. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 80:1–80:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. `doi:10.4230/LIPIcs.ITCS.2021.80`.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] Ramesh Krishnan S. Pallavoor, Sofya Raskhodnikova, and Erik Waingarten. Approximating the distance to monotonicity of Boolean functions. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1995–2009, 2020.

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

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

[57] 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.

[58] 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.

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

[60] 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.

[61] 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.

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

[63] Roei Tell. A note on tolerant testing with one-sided error. In *Computational Complexity and Property Testing - On the Interplay Between Randomness and Computation*, volume 12050 of *Lecture Notes in Computer Science*, pages 173–177. Springer, 2020.

[64] 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.

[65] 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.