THEORY OF COMPUTING, Volume 16 (3), 2020, pp. 1–36
www.theoryofcomputing.org

Optimal Unateness Testers for Real-Valued Functions: Adaptivity Helps

Roksana Baleshzar[*] Deeparnab Chakrabarty[†]
Ramesh Krishnan S. Pallavoor[‡] Sofya Raskhodnikova[‡] C. Seshadhri[§]

Received June 25, 2018; Revised July 2, 2020; Published September 16, 2020

Abstract. We study the problem of testing unateness of functions $f : \{0,1\}^d \to \mathbb{R}$. A function $f : \{0,1\}^d \to \mathbb{R}$ is *unate* if for every coordinate $i \in [d]$, the function is either nonincreasing in the i^{th} coordinate or nondecreasing in the i^{th} coordinate. We give an $O((d/\varepsilon) \cdot \log(d/\varepsilon))$-query nonadaptive tester and an $O(d/\varepsilon)$-query adaptive tester and show that both testers are optimal for a fixed distance parameter ε. Previously known unateness testers worked only for Boolean functions, and their query complexity had worse dependence on the dimension d both for the adaptive and the nonadaptive case. Moreover, no lower bounds for testing unateness were known. (Concurrent work by Chen et al. (STOC'17) proved an $\Omega(d/\log^2 d)$ lower bound on the nonadaptive query complexity of testing unateness of Boolean functions.) We also generalize our results to obtain optimal unateness testers for functions $f : [n]^d \to \mathbb{R}$.

An extended abstract of this paper appeared in the Proceedings of the 44th International Colloquium on Automata, Languages, and Programming (ICALP), 2017 [4].

[*]Supported by NSF award CCF-1422975.

[†]Supported by NSF award CCF-1813053.

[‡]Supported by NSF awards CCF-1422975 and CCF-1909612.

[§]Supported by NSF awards CCF-1740850, CCF-1813165, CCF-1909790, and ARO Award W911NF1910294.

ACM Classification: F.2.2

AMS Classification: 68Q17, 68W20

Key words and phrases: property testing, unate and monotone functions, hypercube, hypergrid

DOI: 10.4086/toc.2020.v016a003

Our results establish that adaptivity helps with testing unateness of real-valued functions on domains of the form $\{0,1\}^d$ and, more generally, $[n]^d$. This stands in contrast to the situation for monotonicity testing where, as shown by Chakrabarty and Seshadhri (Theory of Computing, 2014), there is no adaptivity gap for functions $f : [n]^d \to \mathbb{R}$.

1 Introduction

We study the problem of testing whether a given real-valued function f on domain $[n]^d$, where $n, d \in \mathbb{N}$, is unate. A function $f : [n]^d \to \mathbb{R}$ is *unate* if for every coordinate $i \in [d]$, the function is either nonincreasing in the i^{th} coordinate or nondecreasing in the i^{th} coordinate. Unate functions naturally generalize monotone functions, which are nondecreasing in all coordinates, and **b**-monotone functions, which have a particular direction in each coordinate (either nonincreasing or nondecreasing), specified by a bit-vector $\mathbf{b} \in \{0,1\}^d$. More precisely, a function is **b**-monotone if it is nondecreasing in coordinates i with $\mathbf{b}_i = 0$ and nonincreasing in the other coordinates. Observe that a function f is unate iff there exists some $\mathbf{b} \in \{0,1\}^d$ for which f is **b**-monotone.

A *tester* [59, 39] for a property $\mathcal{P}$ of a function f is an algorithm that gets a distance parameter $\varepsilon \in (0,1)$ and query access to f. The (relative) distance from a function f to a property $\mathcal{P}$ is the smallest fraction of values of f that must be modified to make f satisfy $\mathcal{P}$. A function f is ε-far from $\mathcal{P}$ if the distance from f to $\mathcal{P}$ is at least ε. A tester for $\mathcal{P}$ has to accept with probability at least $2/3$ if f has property $\mathcal{P}$ and reject with probability at least $2/3$ if f is ε-far from $\mathcal{P}$. A tester has *one-sided error* if it always accepts a function satisfying $\mathcal{P}$; it has *two-sided error* otherwise. A *nonadaptive* tester makes all its queries at once, whereas an *adaptive* tester can make queries after seeing answers to previous queries.

Testing various properties of functions, including monotonicity (see, e. g., [38, 52, 33, 34, 47, 36, 53, 35, 41, 1, 42, 5, 16, 12, 19, 22, 17, 11, 23, 24, 21, 27, 26, 45, 8, 9, 32, 49, 29, 7, 13, 25, 14, 50] and recent surveys [54, 55, 20]), the Lipschitz property [43, 22, 31, 17, 2], bounded-derivative properties [21], linearity [18, 6, 10, 44, 56], submodularity [51, 58, 60, 15], and unateness [38, 46], has been studied extensively over the past two decades. Even though unateness testing was initially discussed in the seminal paper by Goldreich et al. [38], which was one of the first papers that study property testing, relatively little was known about testing this property. All previous work on unateness testing focused on the special case of Boolean functions on domain $\{0,1\}^d$. The domain $\{0,1\}^d$ is called the *hypercube* and the more general domain $[n]^d$ is called the *hypergrid*. Goldreich et al. [38] provided an $O(d^{3/2}/\varepsilon)$-query nonadaptive tester for unateness of Boolean functions on the hypercube. More than a decade later, Khot and Shinkar [46] improved the query complexity to $O((d \log d)/\varepsilon)$, albeit with an adaptive tester.

In this paper, we improve upon both these works, and our results hold for a more general class of functions. Specifically, we show that unateness of real-valued functions on hypercubes can be tested nonadaptively with $O((d/\varepsilon)\log(d/\varepsilon))$ queries and adaptively with $O(d/\varepsilon)$ queries. More generally, we describe an $O((d/\varepsilon) \cdot (\log(d/\varepsilon) + \log n))$-query nonadaptive tester and an $O((d \log n)/\varepsilon)$-query adaptive tester for unateness of real-valued functions over hypergrids.

In contrast to the state of knowledge for unateness testing, the complexity of testing monotonicity of real-valued functions over the hypercube and the hypergrid has been resolved. For constant distance parameter ε, it is known to be $\Theta(d \log n)$. Moreover, this bound holds for all *bounded-derivative* properties [21], a large class that includes **b**-monotonicity and some properties quite different from monotonicity, such as the Lipschitz property. Amazingly, the upper bound for all these properties is achieved by the same simple and, in particular, nonadaptive tester. Even though proving lower bounds for adaptive testers has been challenging in general, a line of work, starting from Fischer [35] and including [16, 23, 21], has established that adaptivity does not help for this large class of properties. Since unateness is so closely related, it is natural to ask whether the same is true for testing unateness.

We answer this in the negative: we prove that any nonadaptive unateness tester of real-valued functions over the hypercube (for some constant distance parameter) must make $\Omega(d \log d)$ queries. More generally, it needs $\Omega(d(\log d + \log n))$ queries for the hypergrid domain. These lower bounds complement our algorithms, completing the picture for testing unateness of real-valued functions. From a property testing standpoint, our results establish that unateness is different from monotonicity and, more generally, any derivative-bounded property.

1.1 Formal statements and technical overview

Our testers are summarized in the following theorem, stated for functions over the hypergrid domains. (Recall that the hypercube is a special case of the hypergrid with $n = 2$.)

Theorem 1.1. *Consider functions $f : [n]^d \to \mathbb{R}$ and a distance parameter $\varepsilon \in (0, 1/2)$.*

1. *There is a nonadaptive unateness tester that makes*
$$O\left(\frac{d}{\varepsilon}\left(\log\frac{d}{\varepsilon} + \log n\right)\right)$$
queries.[1]

2. *There is an adaptive unateness tester that makes $O((d \log n)/\varepsilon)$ queries.*

Both testers have one-sided error.

Our main technical contribution is the proof that the extra $\Omega(\log d)$ is needed for nonadaptive testers. This result demonstrates a gap between adaptive and nonadaptive unateness testing.

Theorem 1.2. *Any nonadaptive unateness tester (even with two-sided error) for real-valued functions $f : \{0,1\}^d \to \mathbb{R}$ with a distance parameter $\varepsilon = 1/8$ must make $\Omega(d \log d)$ queries.*

The lower bound for adaptive testers is an easy adaptation of the monotonicity lower bound in [23]. We state this theorem for completeness and prove it in Section 4.1.

Theorem 1.3. *Any unateness tester for functions $f : [n]^d \to \mathbb{R}$ with a distance parameter $\varepsilon \in (0, 1/4)$ must make*
$$\Omega\left(\frac{d \log n}{\varepsilon} - \frac{\log 1/\varepsilon}{\varepsilon}\right)$$
queries.

[1]For many properties, when the domain is extended from the hypercube to the hypergrid, testers incur an extra multiplicative factor of $\log n$ in the query complexity. This is the case for our adaptive tester. However, the complexity of nonadaptive unateness testing (for constant ε) is $\Theta(d(\log d + \log n))$ rather than $\Theta(d \log d \log n)$.

Theorems 1.2 and 1.3 directly imply that our nonadaptive tester is optimal for constant ε, even for the hypergrid domain. The following theorem is proved in Section 4.2.

Theorem 1.4. *Any nonadaptive unateness tester (even with two-sided error) for real-valued functions* $f : [n]^d \to \mathbb{R}$ *must make* $\Omega(d(\log n + \log d))$ *queries.*

1.1.1 Overview of techniques

We first consider the hypercube domain. For each $i \in [d]$, an *i-edge* of the hypercube is a pair (x,y) of points in $\{0,1\}^d$, where $x_i = 0, y_i = 1$, and $x_j = y_j$ for all $j \in ([d] \setminus \{i\})$. Given an input function $f : \{0,1\}^d \to \mathbb{R}$, we say an i-edge (x,y) is *increasing* if $f(x) < f(y)$, *decreasing* if $f(x) > f(y)$, and *constant* if $f(x) = f(y)$.

Our nonadaptive unateness tester on the hypercube uses the work investment strategy from [11] (also refer to Section 8.2.4 of Goldreich's book [37]) to "guess" a good dimension where to look for violations of unateness (specifically, both increasing and decreasing edges). For all $i \in [d]$, let α_i be the fraction of the i-edges that are decreasing, β_i be the fraction of the i-edges that are increasing, and $\mu_i = \min(\alpha_i, \beta_i)$. The dimension reduction theorem from [21] implies that if the input function is ε-far from unate, then the average of μ_i over all dimensions is at least $\varepsilon/(4d)$. If the tester knew which dimension had $\mu_i = \Omega(\varepsilon/d)$, it could detect a violation with high probability by querying the endpoints of $O(1/\mu_i) = O(d/\varepsilon)$ uniformly random i-edges. Performing this test for every dimension would result in query complexity $\Theta(d^2/\varepsilon)$. The work investment strategy allows us to achieve query complexity $O((d/\varepsilon)\log(d/\varepsilon))$ by repeatedly choosing a uniformly random dimension and investing a specific number of queries in trying to find a violation of unateness in that dimension. It proceeds in $\Theta(\log(d/\varepsilon))$ stages, doubling, in every stage, the number of queries invested in each dimension. Intuitively, when all violations are in one dimension, a tester has to try $\Theta(d)$ dimensions before it finds the bad one, but needs only $\Theta(1/\varepsilon)$ queries in the bad dimension. In contrast, when all dimensions have $\mu_i = \varepsilon/(4d)$, only a constant number of attempts are needed to find a dimension with violations, but $\Theta(d/\varepsilon)$ queries in one dimension are required to detect a violation. (In this case, sampling $\Theta((d/\varepsilon)\log(d/\varepsilon))$ uniformly random edges would not be enough to detect a violation.) The work investment strategy allows us to interpolate between these scenarios.

With adaptivity, this search through $\Theta(\log(d/\varepsilon))$ different scenarios is not required. A pair of queries in each dimension detects dimensions with many non-constant edges, and the algorithm focuses on finding violations in those dimensions. This leads to the query complexity of $O(d/\varepsilon)$, removing the $\log(d/\varepsilon)$ factor.

It is relatively easy to extend (both adaptive and nonadaptive) testers from hypercubes to hypergrids by incurring an extra factor of $\log n$ in the query complexity. The role of i-edges is now played by *i-lines*. An i-line is a set of n domain points that differ only in coordinate i. The domain $[n]$ is called a line. Monotonicity on the line (a.k.a. sortedness) was one of the first properties studied in the context of property testing [34]. What we need is a nonadaptive tester for sortedness that has 1-sided error and is, in addition, *proximity oblivious*: that is, it rejects a function $f : [n] \to \mathbb{R}$ with probability proportional to the distance from f to monotonicity. Proximity-oblivious testers (POTs) were defined by Goldreich and Ron [40]. There are several POTs for sortedness that make $O(\log n)$ queries: the *tree tester* [34], the spanners-based tester [12, 54], and the *power of 2* tester [22]. Such testers can be easily modified to work for unateness on the line and to output not just an accept/reject decision, but whether they found a pair of points on which the function is strictly increasing and, similarly, whether they found a pair of points on which the function is strictly decreasing (see Section 2.3 for details). To generalize our unateness testers from the hypercube to the hypergrid domains, instead of sampling a random i-edge, we sample a random i-line ℓ and run a modified POT for unateness on the restriction $f_{|\ell}$ of function f to the line ℓ. This "direct generalization" is optimal for adaptive testers, but, interestingly, not for nonadaptive testers.

Intuitively, for nonadaptive testers, the "direct generalization" works with only $O((d\log n)/\varepsilon)$ queries when there are enough lines that are, cumulatively, sufficiently far from unate. However, it could happen that a function on the hypergrid is far from unate, but does not have any lines that are far from unate: the distance from unateness arises because some lines are far from monotone and other lines in the same dimension are far from *antitone* (that is, far from nonincreasing). We prove that each function f on the line that is ε-far from monotone, but is not $\varepsilon/2$-far from unate, is strictly decreasing on an $\varepsilon/4$ fraction of pairs in $[n]$. Symmetrically, if a function is far from antitone but close to unate, it is strictly increasing on a large fraction of pairs. The dimension reduction allows us to use this statement to show that if the "direct generalization" does not work well, then, intuitively, the average dimension has many pairs on which f is increasing and many pairs on which f is decreasing. We again use the work investment strategy to get a tester for this case that has the same complexity as our nonadaptive tester for the hypercube. The resulting nonadaptive complexity (for constant ε) is $O(d(\log d + \log n))$, which we show is optimal.

The nonadaptive lower bound. Our most significant finding is the $\log d$ gap in the query complexity between adaptive and nonadaptive testing of unateness. By techniques from previous work [35, 23], it suffices to prove lower bounds for *comparison-based* testers, i. e., testers that can only perform comparisons of the function values at queried points, but cannot use the values themselves. Our main technical contribution is the $\Omega(d\log d)$ lower bound for nonadaptive comparison-based testers of unateness on hypercube domains.

Note that nonadaptivity must be critical in our lower bound construction, since we obtained an $O(d)$-query adaptive tester for unateness. Another challenge in proving the lower bound is the existence of a *single, universal* nonadaptive $O(d)$-tester for all $\mathbf{b}$-monotonicity properties, proven in [21]. In other words, there is a single distribution on $O(d)$ queries that defines a nonadaptive property tester for $\mathbf{b}$-monotonicity, regardless of $\mathbf{b}$. Since unateness is the union of all $\mathbf{b}$-monotonicity properties, our construction of hard inputs must be able to fool such algorithms. In particular, if $\mathbf{b}$ is fixed in advance, the algorithm from [21] will work, so it should be hard to learn $\mathbf{b}$ for functions in the lower bound construction. Once a tester finds a non-constant edge in each dimension, the problem reduces to testing $\mathbf{b}$-monotonicity for a vector $\mathbf{b}$ determined by the directions (increasing or decreasing) of the non-constant edges. That is, intuitively, most edges in our construction must be constant. This is one of the main technical challenges. The previous lower bound constructions for monotonicity testing [16, 23] crucially used the fact that all edges in the hard functions were non-constant.

We briefly describe how we overcome the problems mentioned above. By Yao's minimax principle, it suffices to construct two distributions, $\mathcal{D}^+$ and $\mathcal{D}^-$, on unate and on far-from-unate functions, respectively, that a deterministic nonadaptive tester cannot distinguish. First, for some parameter m, we partition the hypercube into m subcubes based on the first $\log_2 m$ most significant coordinates. Both distributions, $\mathcal{D}^+$ and $\mathcal{D}^-$, sample a uniform k from $[K]$, where $K = \Theta(\log d)$, and a set $R \subseteq [d]$ of cardinality 2^k. Furthermore, each subcube $j \in [m]$ selects an "action dimension" $r_j \in R$ uniformly at random. For both distributions, in any particular subcube j, the function value is completely determined by the coordinates *not in* R, and the random coordinate $r_j \in R$. Note that all the i-edges for $i \in (R \setminus \{r_j\})$ are constant. Within the subcube, the function is a linear function with exponentially increasing coefficients. In the distribution $\mathcal{D}^+$, any two subcubes j, j' with the same action dimension orient the edges in that dimension the same way (both increasing or both decreasing), whereas in the distribution $\mathcal{D}^-$ each subcube decides on the orientation independently. The former correlation maintains unateness while the latter independence creates distance to unateness. We prove that to distinguish the distributions, any comparison-based nonadaptive tester must find two distinct subcubes with the same action dimension r_j and, furthermore, make a specific query (in both) that reveals the coefficient of r_j. We show that, with $o(d \log d)$ queries, the probability of this event is negligible.

2 Upper bounds

In this section, we prove parts 1-2 of Theorem 1.1, starting from the hypercube domain.

Recall the definition of i-edges and i-lines from Section 1.1.1 and what it means for an edge to be increasing, decreasing, and constant.

The starting point for our algorithms is the dimension reduction theorem from [21]. It bounds the distance of $f : [n]^d \to \mathbb{R}$ to monotonicity in terms of the average distances of restrictions of f to one-dimensional functions.

Theorem 2.1 (Dimension Reduction, Theorem 1.8 in [21]). *Fix a bit vector* $\mathbf{b} \in \{0,1\}^d$ *and a function* $f : [n]^d \to \mathbb{R}$ *which is* ε*-far from* $\mathbf{b}$*-monotonicity. For all* $i \in [d]$*, let* μ_i *be the average distance of* $f_{|\ell}$ *to* $\mathbf{b}_i$*-monotonicity over all* i*-lines* ℓ*. Then*

$$\sum_{i=1}^{d} \mu_i \geq \frac{\varepsilon}{4}.$$

For the special case of the hypercube domains, i-lines become i-edges, and the average distance μ_i to $\mathbf{b}_i$-monotonicity is the fraction of i-edges on which the function is not $\mathbf{b}_i$-monotone.

2.1 The nonadaptive tester over the hypercube

We now describe Algorithm 1, the nonadaptive tester for unateness over the hypercube domains.

Algorithm 1: The Nonadaptive Unateness Tester over the Hypercube

```
input : a distance parameter $\varepsilon \in (0, 1/2)$; query access to a function $f : \{0,1\}^d \to \mathbb{R}$.
1 for $r = 1$ to $\lceil \log(16d/\varepsilon) \rceil$ do
2     repeat $s_r = \left\lceil \frac{16d \ln 4}{\varepsilon \cdot 2^r} \right\rceil$ times
3         Sample a dimension $i \in [d]$ uniformly at random.
4         Sample $3 \cdot 2^r$ $i$-edges uniformly and independently at random and reject if there exist an
          increasing edge and a decreasing edge among the sampled edges.
5 accept
```

It is evident that Algorithm 1 is a nonadaptive, one-sided error tester. Furthermore, its query complexity is $O((d/\varepsilon)\log(d/\varepsilon))$. It suffices to prove the following.

Lemma 2.2. *If f is ε-far from unate, Algorithm 1 rejects with probability at least* $2/3$.

Proof. Recall that α_i is the fraction of i-edges that are decreasing, β_i is the fraction of i-edges that are increasing and $\mu_i = \min(\alpha_i, \beta_i)$.

Define the d-dimensional bit vector $\mathbf{b}$ as follows: for each $i \in [d]$, let $\mathbf{b}_i = 0$ if $\alpha_i < \beta_i$ and $\mathbf{b}_i = 1$ otherwise. Then the average distance of f to $\mathbf{b}_i$-monotonicity over a random i-edge is precisely μ_i. Since f is ε-far from being unate, f is also ε-far from being $\mathbf{b}$-monotone. By Theorem 2.1, $\sum_{i\in[d]} \mu_i \geq \varepsilon/4$. Hence, $\mathbf{E}_{i\in[d]}[\mu_i] \geq \varepsilon/(4d)$. We now apply the work investment strategy due to Berman et al. [11] to get an upper bound on the probability that Algorithm 1 fails to reject.

Theorem 2.3 ([11])**.** *For a random variable $X \in [0,1]$ with $\mathbf{E}[X] \geq \mu$, let $p_r = \Pr[X \geq 2^{-r}]$ and $\delta \in (0,1)$ be the desired error probability. Let $s_r = \frac{4\ln 1/\delta}{\mu\cdot 2^r}$. Then*

$$\prod_{r=1}^{\lceil \log(4/\mu) \rceil} (1-p_r)^{s_r} \leq \delta.$$

Consider running Algorithm 1 on a function f that is ε-far from unate. Let $X = \mu_i$ where i is sampled uniformly at random from $[d]$. Then $\mathbf{E}[X] \geq \varepsilon/(4d)$. Applying the work investment strategy (Theorem 2.3) on X with $\mu = \varepsilon/(4d)$, we get that the probability that, in some iteration, Step 3 samples a dimension i such that $\mu_i \geq 2^{-r}$ is at least $1-\delta$. We set $\delta = 1/4$. Conditioned on sampling such a dimension, the probability that Step 4 fails to obtain an increasing edge and a decreasing edge among its $3\cdot 2^r$ samples is at most $2\left(1-2^{-r}\right)^{3\cdot 2^r} \leq 2\mathrm{e}^{-3} < 1/9$, as the fraction of both increasing and decreasing edges in the dimension is at least 2^{-r}. Hence, the probability that Algorithm 1 rejects f is at least $(3/4)\cdot(8/9) = 2/3$, which completes the proof of Lemma 2.2. □

2.2 The adaptive tester over the hypercube

We now describe Algorithm 2, an adaptive tester for unateness over the hypercube domains with good expected query complexity. The final tester is obtained by repeating this tester and accepting if the number of queries exceeds a specified bound.

Claim 2.4. *The expected number of queries made by Algorithm 2 is $O(d/\varepsilon)$.*

Proof. Consider one iteration of the **repeat**-loop in Step 1. We prove that the expected number of queries in this iteration is $4d$. The total number of queries in Step 3 is $2d$, as 2 points per dimension are queried. Let E_i be the event that edge e_i is non-constant and T_i be the random variable for the number of i-edges sampled in Step 5. Then

$$\mathbf{E}[T_i] = \frac{1}{\alpha_i + \beta_i} = \frac{1}{\Pr[E_i]}\,.$$

Therefore, the expected number of all edges sampled in Step 5 is

$$\sum_{i=1}^{d} \Pr[E_i]\cdot\mathbf{E}[T_i] = \sum_{i=1}^{d} \Pr[E_i]\cdot\frac{1}{\Pr[E_i]} = d\,.$$

Algorithm 2: The Adaptive Unateness Tester over the Hypercube

```
  input : a distance parameter $\varepsilon \in (0, 1/2)$; query access to a function $f : \{0,1\}^d \to \mathbb{R}$.
1 repeat $\lceil 8/\varepsilon \rceil$ times
2     for $i = 1$ to $d$ do
3         Sample an $i$-edge $e_i$ uniformly at random.
4         if $e_i$ is non-constant (i. e., increasing or decreasing) then
5             Sample $i$-edges uniformly at random until we obtain a non-constant edge $e'_i$.
6             reject if one of the edges $e_i$ and $e'_i$ is increasing and the other is decreasing.
7 accept
```

Hence, the expected number of queries in Step 5 is $2d$, and the total expected number of queries in one iteration is $4d$. Since there are $\lceil 8/\varepsilon \rceil$ iterations in Step 1, the expected number of queries in Algorithm 2 is $O(d/\varepsilon)$. □

Claim 2.5. *If f is ε-far from unate, Algorithm 2 accepts with probability at most $1/6$.*

Proof. First, we bound the probability that a violation of unateness is detected in some dimension $i \in [d]$ in one iteration of the **repeat**-loop in Step 1. Consider the probability of finding a decreasing i-edge in Step 3 and of finding an increasing i-edge in Step 5. The former is exactly α_i, and the latter is $\beta_i/(\alpha_i + \beta_i)$. Similarly, the probability of finding an increasing i-edge in Step 3 and of finding a decreasing i-edge in Step 5 is β_i and $\alpha_i/(\alpha_i + \beta_i)$, respectively. Therefore, the probability we detect a violation from dimension i is

$$2 \cdot \frac{\alpha_i \beta_i}{\alpha_i + \beta_i} \geq \min(\alpha_i, \beta_i) = \mu_i.$$

The probability that we fail to detect a violation in any of the d dimensions in one iteration is at most

$$\prod_{i=1}^{d} (1 - \mu_i) \leq \exp\Big(-\sum_{i=1}^{d} \mu_i\Big) \leq \mathrm{e}^{-\varepsilon/4},$$

where the last inequality follows from Theorem 2.1 (Dimension Reduction). Thus, the probability that Algorithm 2 fails to reject in all iterations of Step 1 is at most $(\exp(-\varepsilon/4))^{(8/\varepsilon)} = \mathrm{e}^{-2} < 1/6$. □

Proof of Theorem 1.1, Part 2 (for the special case of the hypercube domain). We run Algorithm 2, aborting and accepting if the number of queries exceeds the expectation by a factor of 6. By Markov's inequality, the probability of aborting is at most $1/6$. By Claim 2.5, if f is ε-far from unate, Algorithm 2 accepts with probability at most $1/6$. The theorem follows by a union bound. □

2.3 Extension to hypergrids

We start by establishing terminology for lines and pairs. Consider a function $f : [n]^d \to \mathbb{R}$. Recall the definition of i-lines from Section 1.1.1. A pair of points that differ only in coordinate i is called an i-pair. An i-pair $\{x, y\}$ with $x_i < y_i$ is called *increasing* if $f(x) < f(y)$, *decreasing* if $f(x) > f(y)$, and *constant* if

$f(x) = f(y)$. Recall that we call a function $h : [n] \to \mathbb{R}$ *monotone* if $h(x) \leq h(y)$ for all $x < y$ and *antitone* if $h(x) \geq h(y)$ for all $x < y$.

Recall that a proximity-oblivious tester (POT) for a property $\mathcal{P}$ rejects every input with probability proportional to the distance from the input to $\mathcal{P}$. As discussed in Section 1.1.1, there are several 1-sided error, nonadaptive POTs for monotonicity on the line that we can use to extend Algorithms 1 and 2 to work on hypergrids. One of them is the *tree tester*, designed by Ergun et al. [34], that simply picks a point $x \in [n]$ uniformly at random, queries all points visited in a binary search for x, and rejects iff x forms a decreasing pair with one of the queried points. By symmetry, when this algorithm is modified to reject iff it finds an increasing pair, it tests antitonicity. The tree tester (and any other 1-sided error POT for sortedness) can be easily modified to return whether it found any increasing/decreasing edges by including $\uparrow$ / $\downarrow$ in its output.

Lemma 2.6 ([34, 12, 22, 21])**.** *There exists a nonadaptive algorithm $\mathcal{A}_{dir}$ that gets query access to a function $h : [n] \to \mathbb{R}$, makes $O(\log n)$ queries, and returns a set* $\mathsf{dir} \subseteq \{\uparrow, \downarrow\}$.

- *If h is monotone, then $\downarrow$ is not in* dir. *The probability that* dir *contains $\downarrow$ is at least the distance from h to monotonicity.*
- *Similarly, if h is antitone, then $\uparrow$ is not in* dir. *The probability that* dir *contains $\uparrow$ is at least the distance from h to antitonicity.*

We now describe Algorithm 3, an adaptive tester for unateness over the hypergrid domains with good expected query complexity. As in the case of the hypercube domains, the final tester is obtained by repeating this tester and accepting if the number of queries exceeds a specified bound.

Algorithm 3: The Adaptive Unateness Tester over the Hypergrid

```
input : a distance parameter $\varepsilon \in (0, 1/2)$; query access to a function $f : [n]^d \to \mathbb{R}$.
1 repeat $\lceil 8/\varepsilon \rceil$ times
2     for $i = 1$ to $d$ do
3         Sample an $i$-line $\ell_i$ uniformly at random.
4         Let $\mathsf{dir}_i$ be the output of $\mathcal{A}_{\mathrm{dir}}$ (from Lemma 2.6) on $f_{|\ell_i}$.
5         if $\mathsf{dir}_i \neq \emptyset$ then
6             repeat
7                 Sample an $i$-line $\ell'_i$ uniformly at random and let $\mathsf{dir}'_i$ be the output of $\mathcal{A}_{\mathrm{dir}}$ on $f_{|\ell'_i}$.
              until $\mathsf{dir}'_i \neq \emptyset$
8             If $\mathsf{dir}_i \cup \mathsf{dir}'_i = \{\uparrow, \downarrow\}$, reject.
9 accept
```

Proof of Theorem 1.1, Part 2. First, we explain how Lemma 2.6 and Theorem 2.1 are used in the analysis of the adaptive tester. For a dimension $i \in [d]$, let α_i and β_i denote the average distance of $f_{|\ell}$ to monotonicity and antitonicity, respectively, over all i-lines ℓ. Then $\mu_i := \min(\alpha_i, \beta_i)$ is the average fraction of points per i-line that needs to change to make f unate. Define the $\mathbf{b}$-vector with $\mathbf{b}_i = 0$ if

$\alpha_i < \beta_i$, and $\mathbf{b}_i = 1$ otherwise. By Theorem 2.1, if f is ε-far from unate, and thus ε-far from $\mathbf{b}$-monotone, then $\sum_{i=1}^{d} \mu_i \geq \varepsilon/4$. By Lemma 2.6, the probability that the output of $\mathcal{A}_{\mathrm{dir}}$ on a uniformly random i-line $f_{|\ell_i}$ contains $\downarrow$ is at least α_i , and the probability that it contains $\uparrow$ is at least β_i . The rest of the analysis of Algorithm 3 is similar to that in the hypercube case.

In the proof of Claim 2.4, the expected number of edges sampled in one iteration of Algorithm 2 is $2d$. Similarly, the expected number of lines sampled in one iteration of Algorithm 3 is $2d$. Since $\mathcal{A}_{\mathrm{dir}}$ makes $O(\log n)$ queries, the overall expected number of queries is $O((d \log n)/\varepsilon)$.

The proof of Claim 2.5 carries over almost word for word. Fix a dimension i. The probability that $\downarrow \in \mathsf{dir}_i$ in Step 4 is at least α_i. The probability that $\uparrow \in \mathsf{dir}'_i$ in Step 7 is at least $\beta_i/(\alpha_i + \beta_i)$. The rest of the calculation is identical to that of the proof of Claim 2.5.

Finally, we run Algorithm 3 and abort and accept if the number of queries exceeds the expectation by a factor of 6. As in the proof of Theorem 1.1, Part 2, the resulting algorithm always accepts unate functions and accepts functions that are ε-far from unate with probability at most $1/3$, completing the proof. □

Next we show how to modify any POT for sortedness in a black-box manner to obtain a POT for unateness on the line with the same guarantees.

Algorithm 4: The POT for Unateness over the Line

```
input : query access to a function $h : [n] \to \mathbb{R}$.
1 Run algorithm $\mathcal{A}_{\mathrm{dir}}$ on $h$ and let dir be its output.
2 Pick $x \in [n]$ uniformly at random and query $h$ on points $1, x$, and $n$.
3 If any pair in $\{1,x,n\}$ is increasing, update dir $\leftarrow$ dir $\cup \{\uparrow\}$.
4 If any pair in $\{1,x,n\}$ is decreasing, update dir $\leftarrow$ dir $\cup \{\downarrow\}$.
5 Reject if dir $= \{\uparrow,\downarrow\}$; otherwise, accept.
```

Lemma 2.7. *If $h : [n] \to \mathbb{R}$ is ε-far from unate, then Algorithm 4 rejects with probability at least ε.*

Proof. First, consider the case when $h(1) = h(n)$. Since h is ε-far from unate, it is also ε-far from being a constant function equal to $h(1)$ on all points in $[n]$. Therefore, with probability at least ε, point x chosen in Step 2 of the algorithm satisfies $h(x) \neq h(1)$. If this holds, then one of the pairs $\{1,x\}$ and $\{x,n\}$ is increasing and the other is decreasing. So, Algorithm 4 indeed rejects with probability at least ε.

Next, consider the case when $h(1) < h(n)$. Then $\{1,n\}$ is an increasing pair. Since h is ε-far from monotone, by Lemma 2.6, the output of $\mathcal{A}_{\mathrm{dir}}$ contains $\downarrow$ with probability at least ε. So, again Algorithm 4 rejects with probability at least ε.

Finally, the case when $h(1) > h(n)$ is symmetric to the case when $h(1) < h(n)$. □

Our nonadaptive hypergrid tester is stated in Algorithm 5. A crucial part of its analysis is Lemma 2.8 that demonstrates that every function on the line that is far from monotone is also far from unate or a large fraction of pairs are decreasing with respect to that function.

Lemma 2.8. *Consider a function $h : [n] \to \mathbb{R}$ which is ε-far from monotone. At least one of the following holds:*

Algorithm 5: The Nonadaptive Unateness Tester over the Hypergrid

```
input : distance parameter $\varepsilon \in (0,1/2)$; query access to a function $f : [n]^d \to \mathbb{R}$.
1  repeat $\lceil \frac{24 \ln 3}{\varepsilon} \rceil$ times
2      for $i = 1$ to $d$ do
3          Sample an $i$-line $\ell$ uniformly at random.
4          Reject if Algorithm 4 rejects on input $f_{|\ell}$.
5  for $r = 1$ to $\lceil \log(96d/\varepsilon) \rceil$ do
6      repeat $s_r = \lceil \frac{96d \ln 4}{\varepsilon \cdot 2^r} \rceil$ times
7          Sample a dimension $i \in [d]$ uniformly at random.
8          Sample $3 \cdot 2^r$ $i$-pairs uniformly and independently at random.
9          If we find an increasing and a decreasing pair among the sampled pairs, reject.
10 accept
```

1. *The function h is $\frac{\varepsilon}{2}$-far from unate.*

2. $\Pr_{u,v \in [n]} \left[\textit{pair } \{u,v\} \textit{ is decreasing} \right] \geq \frac{\varepsilon}{4}$.

Proof. Suppose Item 1 does not hold, that is, there exists a set $G \subseteq [n]$ of size greater than $n(1-\varepsilon/2)$ on which $h_{|G}$ is unate and therefore antitone. (Unate means monotone or antitone, but $h_{|G}$ cannot be monotone because h is ε-far from monotone.) Observe that if $u, v \in G$ and $h(u) \neq h(v)$ then $\{u,v\}$ is a decreasing pair because $h_{|G}$ is antitone. For each point $u \in G$, let $D_u \subseteq G$ be the set of points on which h differs from $h(u)$. Since h is ε-far from monotone and, consequently, ε-far from constant, $|D_u| \geq \varepsilon/2$ for all $u \in G$. Thus,

$$\Pr_{u,v \in [n]}[\text{pair } \{u,v\} \text{ is decreasing}] \geq \Pr[u \in G \text{ and } v \in D_u] \geq \left(1 - \frac{\varepsilon}{2}\right) \cdot \frac{\varepsilon}{2} \geq \frac{\varepsilon}{4}.$$

The last inequality follows because $\varepsilon \leq 1$. We proved that if Item 1 does not hold then Item 2 must hold. □

Theorem 2.9. *If $f : [n]^d \to \mathbb{R}$ is ε-far from unate, then Algorithm 5 rejects with probability at least* $2/3$.

Proof. Suppose $f : [n]^d \to \mathbb{R}$ is ε-far from unate. We will show that when Algorithm 5 runs on f, one of the two loops (in Steps 1 and 5) rejects with probability at least 2/3.

For every line ℓ, we define the following quantities.

- α_ℓ: the distance of $f_{|\ell}$ to monotonicity.
- β_ℓ: the distance of $f_{|\ell}$ to antitonicity.
- $\mu_\ell = \min(\alpha_\ell, \beta_\ell)$: the distance of $f_{|\ell}$ to unateness.
- α'_ℓ: the probability that a uniformly random pair in ℓ is decreasing.

- β'_ℓ: the probability that a uniformly random pair in ℓ is increasing.

By Lemma 2.8 and symmetry, for every line ℓ,

$$\mu_\ell + 2\alpha'_\ell \geq \frac{\alpha_\ell}{2} \text{ and } \mu_\ell + 2\beta'_\ell \geq \frac{\beta_\ell}{2}. \tag{2.1}$$

Let L_i be the set of i-lines. By Theorem 2.1,

$$\frac{1}{n^{d-1}} \sum_{i=1}^{d} \min\left(\sum_{\ell \in L_i} \alpha_\ell, \sum_{\ell \in L_i} \beta_\ell \right) \geq \frac{\varepsilon}{4}. \tag{2.2}$$

Define $u = \sum_{i=1}^{d} u_i$, where

$$u_i = \frac{1}{n^{d-1}} \sum_{\ell \in L_i} \mu_\ell$$

is the average distance to unateness in dimension i. (In general, u_i is different from the quantity μ_i used in the analysis of the adaptive tester. Recall that μ_i was used to denote the minimum of the average distance to monotonicity and the average distance to antitonicity in dimension i. We are using different notation to avoid confusion.) Let α'_i to be the average of α'_ℓ for all lines $\ell \in L_i$. Define β'_i analogously, and let $\mu'_i = \min(\alpha'_i, \beta'_i)$. Let random variable X be μ'_i where i is sampled uniformly at random from $[d]$. Then

$$\begin{aligned} u + 2d \cdot \mathbf{E}[X] &= \frac{1}{n^{d-1}} \sum_{i=1}^{d} \left[\sum_{\ell \in L_i} \mu_\ell + 2 \cdot \min\left(\sum_{\ell \in L_i} \alpha'_\ell, \sum_{\ell \in L_i} \beta'_\ell \right) \right] \\ &= \frac{1}{n^{d-1}} \sum_{i=1}^{d} \min\left(\sum_{\ell \in L_i} (\mu_\ell + 2\alpha'_\ell), \sum_{\ell \in L_i} (\mu_\ell + 2\beta'_\ell) \right) \\ &\geq \frac{1}{n^{d-1}} \sum_{i=1}^{d} \min\left(\sum_{\ell \in L_i} \frac{\alpha_\ell}{2}, \sum_{\ell \in L_i} \frac{\beta_\ell}{2} \right) \geq \frac{\varepsilon}{8}. \end{aligned}$$

The last two inequalities follow from Equations (2.1) and (2.2), respectively. We conclude that at least one of u and $\mathbf{E}[X]$ has to be large, specifically, $u \geq \varepsilon/24$ or $\mathbf{E}[X] \geq \varepsilon/(24d)$.

Case 1: $u \geq \varepsilon/24$. Consider the first loop of Algorithm 5 (in Step 1). By Lemma 2.7, the probability that a uniformly random i-line is rejected by Algorithm 4 is at least u_i. The probability that one iteration of the loop in Step 1 fails to reject is

$$\prod_{i=1}^{d} (1 - u_i) \leq \prod_{i=1}^{d} \exp(-u_i) = \exp(-\sum_{i=1}^{d} u_i) = \exp(-u) \leq \exp\left(-\frac{\varepsilon}{24}\right).$$

Thus, all iterations of the loop fail to reject with probability at most $(\exp(-\varepsilon/24))^{(24 \ln 3)/\varepsilon} = 1/3$.

Case 2: $\mathbf{E}[X] \geq \varepsilon/(24d)$. Applying the work investment strategy (Theorem 2.3) on X with $\mu = \varepsilon/(24d)$ and using a calculation analogous to that in the proof of Lemma 2.2, the probability that Step 9 rejects in some iteration is at least $2/3$. □

3 The lower bound for nonadaptive testers over the hypercube

In this section, we prove Theorem 1.2, which gives a lower bound for nonadaptive unateness testers for functions over the hypercube.

Fischer [35] showed that in order to prove lower bounds for a general class of properties on the line domain, it is sufficient to consider a special class of testers called *comparison-based testers*. The properties he looked at are called *order-based properties* (see Definition 3.2), and they include monotonicity and unateness. A tester is *comparison-based* if it bases its decisions only on the *order* of the function values at the points it queried, and not on the values themselves. Chakrabarty and Seshadhri [23] extended Fischer's proof to monotonicity on any partially-ordered domain for the case when all function values are distinct. As we show in Section 3.1 below, Chakrabarty and Seshadhri's proof goes through for all order-based properties on partially-ordered domains. Moreover, the assumption of distinctness for function values can be removed. We include this proof for completeness, filling in the details needed to generalize the original proof.

Our main technical contribution is the construction of a distribution of functions $f : \{0,1\}^d \to \mathbb{N}$ on which every nonadaptive comparison-based tester must query $\Omega(d \log d)$ points to determine whether the sampled function is unate or far from unate. We describe this construction in Section 3.2 and show its correctness in Sections 3.3-3.4.

3.1 Reduction to comparison-based testers

In this section, we prove that if there exists a tester for an order-based property $\mathcal{P}$ of functions over a partially-ordered domain, then there exists a comparison-based tester for $\mathcal{P}$ with the same query complexity. This is stated in Theorem 3.3. Before stating the theorem, we introduce some definitions.

Definition 3.1. A (t,ε,δ)-tester for a property is a 2-sided error tester with distance parameter ε that makes at most t queries and errs with probability at most δ.

Definition 3.2 (Order-based property)**.** For an arbitrary partial order D and an arbitrary total order R, a property $\mathcal{P}$ of functions $f : D \to R$ is *order-based* if, for all strictly increasing maps $\phi : R \to R$ and all functions f, we have $\mathsf{dist}(f,\mathcal{P}) = \mathsf{dist}(\phi \circ f,\mathcal{P})$.

In particular, unateness is an order-based property. The following theorem is an extension of Theorem 5 in [35] and Theorem 2.1 in [23]. Specifically, Theorem 2.1 in [23] was proved with the assumption that the function values are distinct. We generalize the theorem by removing this assumption.

Theorem 3.3 (Generalization of [35, 23])**.** *Let $\mathcal{P}$ be an order-based property of functions $f : D \to \mathbb{N}$ over a finite domain D. Suppose there exists a (t,ε,δ)-tester for $\mathcal{P}$. Then there exists a comparison-based $(t,\varepsilon,2\delta)$-tester for $\mathcal{P}$.*

The rest of this section is devoted to proving Theorem 3.3. Our proof closely follows the proof of Theorem 2.1 in [23]. The proof has two parts. The first part describes a reduction from a tester to a *discretized tester*, and the second part describes a reduction from a discretized tester to a comparison-based tester.

Let $\mathcal{P}$ be a property of functions $f : D \to R$ for an arbitrary partial order D and an arbitrary total order $R \subseteq \mathbb{N}$. Let $\mathcal{T}$ be a (t,ε,δ)-tester for $\mathcal{P}$. First, we define a family of probability functions that completely characterizes $\mathcal{T}$. Fix some $s \in [t]$. Consider the point in time in an execution of the tester $\mathcal{T}$ on some input function f, where exactly s queries have been made. Suppose these queries are $x_1, x_2, \dots, x_s \in D$ and the corresponding answers are $a_1 = f(x_1), a_2 = f(x_2), \dots, a_s = f(x_s)$. Let the *query vector* X be $(x_1,\dots,x_s)$ and the *answer vector* A be $(a_1,\dots,a_s)$. The next action of the algorithm is either choosing the $(s+1)^{\text{st}}$ query from D or outputting *accept* or *reject*. For each action $y \in D \cup \{\texttt{accept},\texttt{reject}\}$, let $p_X^y(A)$ denote the probability that $\mathcal{T}$ chooses action y after making queries X and receiving answers A. Since $p_X^y(A)$ is a probability distribution,

$$\forall s < t, \forall X \in D^s, \forall A \in R^s, \quad \sum_{y \in D \cup \{\texttt{accept},\texttt{reject}\}} p_X^y(A) = 1.$$

Furthermore, the tester cannot make more than t queries, and so the action $(t+1)$ must be either `accept` or `reject`. Formally,

$$\forall X \in D^t, \forall A \in R^t, \quad \sum_{y \in \{\texttt{accept},\texttt{reject}\}} p_X^y(A) = 1.$$

If a tester decides to accept or reject before making t queries, i. e., $p_X^{y'}(A) = 1$ for some $X = (x_1,\dots,x_s), A = (a_1,\dots,a_s)$, where $s < t$ and $y' \in \{\texttt{accept, reject}\}$, then we fill in the values for p, so that the action of the tester is defined to be the same y' for all values until $t+1$. Specifically, we set $p_{X'}^{y'}(A') = 1$ for all $X' \in D^{s'}, A' \in R^{s'}$ where $s < s' \le t$ and the first s queries (in X') and their corresponding answers (in A') are $x_1,\dots,x_s$ and $a_1,\dots,a_s$, respectively.

Definition 3.4 (Discretized tester)**.** A tester $\mathcal{T}$ is *discretized* if all $p_X^y(A)$-values associated with $\mathcal{T}$ come from the range $\{i/K : i \in \{0,1,\dots,K\}\}$ for some positive integer K.

Chakrabarty and Seshadhri [23] proved that if there exists a (t,ε,δ)-monotonicity tester $\mathcal{T}$ for functions $f : D \to \mathbb{N}$, then there exists a discretized $(t,\varepsilon,2\delta)$-monotonicity tester $\mathcal{T}'$ for the same class of functions. Both the statement and the proof in [23] hold not only for testers of monotonicity, but for testers of all properties of functions $f : D \to R$.

Lemma 3.5 (Implicit in [23, Lemma 2.2])**.** *Suppose there exists a (t,ε,δ)-tester $\mathcal{T}$ for a property $\mathcal{P}$ of functions $f : D \to R$. Then there exists a $(t,\varepsilon,2\delta)$-discretized tester $\mathcal{T}'$ for $\mathcal{P}$.*

This completes the first part of the proof.

Next, we show how to transform a discretized tester into a comparison-based tester. Intuitively, a tester is comparison-based if each action of the tester depends only on the ordering of the answers to the previous queries, not on the values themselves. We define a family of probability functions q in order to characterize comparison-based testers. The q-functions are defined in terms of p-functions, but, in their definition, we decouple the set of values that were received as answers from their positions in the answer vector. Specifically, the set of values that were received as answers (i. e., information irrelevant for a comparison-based tester) is given as the argument to the q-functions. All the remaining information is given as subscripts and superscripts. Let V represent the set $\{a_1,\dots,a_s\}$ of answer values (without duplicates). Let r be the number of (distinct) values in V. Note that $r \le s$. Suppose V is

$\{v_1, v_2, \ldots, v_r\}$, where $v_1, \ldots, v_r \in R$ and $v_1 < v_2 < \ldots < v_r$. Let ρ be the map from positions of values in the answer vector to their corresponding indices in V, that is, $\rho : [s] \to [r]$. Observe that ρ is surjective. The q-functions are defined as follows:

$$q^y_{X,\rho}(V) = p^y_X((v_{\rho(1)}, v_{\rho(2)}, \ldots, v_{\rho(s)})).$$

For any set S, let $S^{(r)}$ denote the set of all subsets of S of size r.

Definition 3.6 (Comparison-based tester)**.** A tester $\mathcal{T}$ for an order-based property $\mathcal{P}$ is *comparison-based* for functions $f : D \to R$, if for all r, s satisfying $r \le s \le t$, and all $X \in D^s$, $y \in D \cup \{\texttt{accept}, \texttt{reject}\}$ and surjections $\rho : [s] \to [r]$, the function $q^y_{X,\rho}$ is constant on $R^{(r)}$. That is, for all $V, V' \in R^{(r)}$, we have $q^y_{X,\rho}(V) = q^y_{X,\rho}(V')$.

To complete the proof of Theorem 3.3, we show that if there exists a discretized tester $\mathcal{T}$ for an order-based property $\mathcal{P}$ over the functions $f : D \to \mathbb{N}$, then there exists an infinite set $R \subseteq \mathbb{N}$ such that, for functions $f : D \to R$, the tester $\mathcal{T}$ is comparison-based. Specifically, the q-functions that describe the tester do not depend on V, the specific set of answer values, as long as $V \subset R$. At the end of the proof, we construct a new comparison-based tester that modifies the input function so that its range is R (without changing the distance of the function to the property $\mathcal{P}$) and runs $\mathcal{T}$ on the modified function. The existence of the infinite set R is proved using Ramsey theory arguments.

We introduce some Ramsey theory terminology. Consider a positive integer C, where $[C]$ represents a set of colors. For any positive integer i, a *finite coloring* of $\mathbb{N}^{(i)}$ is a function $\texttt{col}_i : \mathbb{N}^{(i)} \to [C]$. That is, this function assigns one of C colors to each subset of $\mathbb{N}$ with i elements. An infinite set $R \subseteq \mathbb{N}$ is *monochromatic* with respect to $\texttt{col}_i$ if for all i-element subsets $V, V' \in R^{(i)}$, the color $\texttt{col}_i(V) = \texttt{col}_i(V')$. In other words, each i-element subset of R is colored with the same color by the coloring function $\texttt{col}_i$. A *k-wise finite coloring of* $\mathbb{N}$ is a collection of k-colorings $\texttt{col}_1, \texttt{col}_2, \ldots, \texttt{col}_k$. Note that each coloring function $\texttt{col}_1, \ldots, \texttt{col}_k$ is defined over subsets of a different size, and together they assign a color to each subset with at most k elements. An infinite subset $R \subseteq \mathbb{N}$ is *k-wise monochromatic* with respect to $\texttt{col}_1, \ldots, \texttt{col}_k$ if R is monochromatic with respect to all $\texttt{col}_i$ for $i \in [k]$. That is, all subsets of R of the same size get assigned the same color by the coloring functions.

We use the following variant of Ramsey's theorem which was also used in [35, 23].

Theorem 3.7 (Theorem 2.3 in [23])**.** *For any k-wise finite coloring of $\mathbb{N}$, there exists an infinite k-wise monochromatic subset $R \subseteq \mathbb{N}$.*

Proof of Theorem 3.3. Suppose there exists a (t, ε, δ)-tester for property $\mathcal{P}$ of functions $f : D \to \mathbb{N}$. By Lemma 3.5, there exists a $(t, \varepsilon, 2\delta)$-discretized tester $\mathcal{T}$ for $\mathcal{P}$. Consider the family of q-functions that characterizes $\mathcal{T}$.

The main idea of the proof is to view the behavior of the tester $\mathcal{T}$ on each possible subset of answer values V (with at most t elements) as the color of V. More precisely, the color of V is the corresponding vector of all q-functions evaluated at V. If two sets V and V' are mapped to the same color, it means, intuitively, that the tester $\mathcal{T}$ is ignoring whether the specific answer values are V or V'. We want to show that there is a large subset R of values, such that the tester $\mathcal{T}$ ignores the answer values, as long as they come from R. Since $\mathcal{T}$ is discretized, the set of colors is finite, and we are able to apply Theorem 3.7 to get an infinite subset R of $\mathbb{N}$ which is t-wise monochromatic. That means that $\mathcal{T}$ ignores V, as long as $V \subset R$. That is, $\mathcal{T}$ is already comparison-based on R. At the end, we use the fact that $\mathcal{P}$ is an order-based property to get a comparison-based tester for the whole range.

We define a t-wise finite coloring of $\mathbb{N}$. For each $r \in [t]$ and $V \in \mathbb{N}^{(r)}$, the color $\texttt{col}_r(V)$ is defined as a vector of probability values $q^y_{X,\rho}(V)$. The vector is indexed by (y, X, ρ) for each $y \in D \cup \{\texttt{accept}, \texttt{reject}\}, X \in D^s$ for an integer s satisfying $r \leq s \leq t$, and a surjection $\rho : [s] \to [r]$. The value at the index (y, X, ρ) in $\texttt{col}_r(V)$ is equal to $q^y_{X,\rho}(V)$. Note that, there are finitely many possible values for y and X, and surjections ρ. So, the dimension of the vector $\texttt{col}_r(V)$ is finite. Furthermore, since the tester is discretized, the number of different values that the q-functions take is also finite. Hence, the range of $\texttt{col}_r$ is finite. Now, we have a t-wise finite coloring $\texttt{col}_1, \ldots, \texttt{col}_t$ of $\mathbb{N}$. By Theorem 3.7, there exists an infinite t-wise monochromatic set $R \subseteq \mathbb{N}$. Thus, for each $r \in [t]$ and $V, V' \in R^{(r)}$, we have $\texttt{col}_r(V) = \texttt{col}_r(V')$, implying that $q^y_{X,\rho}(V) = q^y_{X,\rho}(V')$ for all y, X, and ρ. Thus, $\mathcal{T}$ is comparison-based for functions $f : D \to R$.

Finally, we construct a comparison-based tester $\mathcal{T}'$ for the whole range $\mathbb{N}$. Consider a strictly monotone increasing map $\phi : \mathbb{N} \to R$. Given any function $f : D \to \mathbb{N}$, consider $\phi \circ f : D \to R$. Define an algorithm $\mathcal{T}'$, which on input f, runs $\mathcal{T}$ on $\phi \circ f$. Since $\mathcal{P}$ is order-based, $\text{dist}(f, \mathcal{P}) = \text{dist}(\phi \circ f, \mathcal{P})$. Hence, $\mathcal{T}'$ is a $(t, \varepsilon, 2\delta)$-tester for $\mathcal{P}$. Moreover, since the tester $\mathcal{T}'$ just runs $\mathcal{T}$ on $\phi \circ f : D \to R$, and $\mathcal{T}$ is comparison-based for $\phi \circ f$, the tester $\mathcal{T}'$ is also comparison-based. □

3.2 The hard distributions

Our main lower bound theorem is stated next. Together with Theorem 3.3, it implies Theorem 1.2.

Theorem 3.8 (Main). *Any nonadaptive comparison-based unateness tester of functions $f : \{0,1\}^d \to \mathbb{N}$ that works with distance parameter $\varepsilon = 1/8$ and errs with probability at most 1/6 must make $\Omega(d \log d)$ queries.*

The proof of Theorem 3.8 is presented in Sections 3.2-3.4 and forms the core technical content of this work. We will use Yao's minimax principle [61], specifically, the version stated in [57, Claim 5]. It asserts that to prove a lower bound for a randomized tester, it is sufficient to give two distributions, one on positive and one on negative instances, that every deterministic tester fails to distinguish with high probability. Next, we define two input distributions.

It may be useful for the reader to recall the sketch of the main ideas given in Section 1.1.1. Without loss of generality,[2] let d be an even power of 2 and $d' := d + \log_2 d$. We will focus on functions $h : \{0,1\}^{d'} \to \mathbb{N}$, and prove the lower bound of $\Omega(d \log d)$ for this class of functions, as $\Omega(d \log d) = \Omega(d' \log d')$.

We partition $\{0,1\}^{d'}$ into d subcubes based on the $\log_2 d$ most significant bits. Specifically, for $i \in [d]$, the i^{th} subcube is defined as

$$C_i := \{x \in \{0,1\}^{d'} \mid \mathsf{val}(x_{d'} x_{d'-1} \cdots x_{d+1}) = i - 1\},$$

where $\mathsf{val}(z) := \sum_{i=1}^{p} z_i 2^{i-1}$ denotes the integer equivalent of the binary string $z_p z_{p-1} \ldots z_1$.

[2]If the number of dimensions is d'' where there is no positive integer d such that $d'' = d + \log_2 d$, then consider the largest integer d' such that $d' < d''$ and $d' = d + \log_2 d$. We can construct any function over $\{0,1\}^{d'}$ and extend it to $\{0,1\}^{d''}$ by adding $d'' - d'$ dummy variables. Furthermore, $d' = \Theta(d'')$, and hence, $\Omega(d'' \log d'') = \Omega(d' \log d')$.

Let $m := d$. For ease of notation, we denote the set of subcube indices by $[m]$ and the set of dimensions in a subcube by $[d]$. We use $i, j \in [m]$ to index subcubes, and $a, b \in [d]$ to index dimensions. We now define a collection of random variables, where each variable may depend on the previous ones:

- The logarithm of the number of *action dimensions*, k: a number picked uniformly at random from $[(1/2)\log_2 d]$.
- The set of all action dimensions, R: a uniformly random subset of $[d]$ of size 2^k.
- The action dimension for each subcube, r_i: for each $i \in [m]$, r_i is picked from R uniformly and independently at random.
- The direction for each dimension (in $\mathcal{D}^+$ distribution), α_b: for each $b \in [d]$, α_b is picked from $\{-1,+1\}$ uniformly and independently at random. (Technically, α_b will only be used for each $b \in R$. We define it for all $b \in [d]$ so that it is independent of R.)
- The direction of potential violations for each subcube (in $\mathcal{D}^-$ distribution), β_i: for each $i \in [m]$, β_i is picked from $\{-1,+1\}$ uniformly and independently at random.

We use $\mathbf{T}$ to refer to the entire collection $(k, R, \{r_i \mid i \in [m]\}, \{\alpha_b \mid b \in [d]\}, \{\beta_i \mid i \in [m]\})$ of random variables. We denote the tuple $(k, R, \{r_i \mid i \in [m]\})$ by $\mathbf{S}$, also referred to as the *shared randomness* common to the distributions $\mathcal{D}^+$ and $\mathcal{D}^-$. Given $\mathbf{T}$, the distributions $\mathcal{D}^+$ and $\mathcal{D}^-$ generate the functions $f_{\mathbf{T}}$ and $g_{\mathbf{T}}$, respectively, where

$$f_{\mathbf{T}}(x) := \sum_{b \in [d'] \setminus R} x_b 3^b + \alpha_{r_i} x_{r_i} 3^{r_i},$$
$$g_{\mathbf{T}}(x) := \sum_{b \in [d'] \setminus R} x_b 3^b + \beta_i x_{r_i} 3^{r_i}$$

and i is the subcube containing x, i. e., $i = \mathsf{val}(x_{d'} x_{d'-1} \cdots x_{d+1}) + 1$.

3.3 The sign function and the distance to unateness for hard distributions

We need to analyze when functions in our hard distributions are increasing or decreasing in a specified dimension. More generally, since we are looking at comparison-based testers, we need to understand how $h(x)$ and $h(y)$ compare for any points x, y in the hypercube $\{0,1\}^{d'}$ for any function h in the support of the hard distributions. To help us with that, we define the sign function $\mathsf{sgn}_h(x,y)$ and analyze its behavior on our hard distribution.

Recall that $\mathsf{val}(z) := \sum_{i=1}^{p} z_i 2^{i-1}$ denotes the integer equivalent of the binary string $z = z_p z_{p-1} \ldots z_1$. We say that $x <_{\mathsf{val}} y$ if $\mathsf{val}(x) < \mathsf{val}(y)$. Note that $<_{\mathsf{val}}$ forms a total ordering on $\{0,1\}^{d'}$.

Definition 3.9. Given $x, y \in \{0,1\}^{d'}$ such that $x <_{\mathsf{val}} y$, and a function $h : \{0,1\}^{d'} \to \mathbb{N}$, define

$$\mathsf{sgn}_h(x,y) = \begin{cases} 1 & \text{if } h(x) < h(y), \\ 0 & \text{if } h(x) = h(y), \\ -1 & \text{if } h(x) > h(y). \end{cases}$$

Next, we show that for every function in the support of our hard distributions and two points $x <_{\mathsf{val}} y$ in different subcubes, $\mathsf{sgn}_h(x,y) = 1$.

For a distribution $\mathcal{D}$, let $\mathsf{supp}(\mathcal{D})$ denote the support of $\mathcal{D}$.

Claim 3.10. *For all $h \in \mathsf{supp}(\mathcal{D}^+) \cup \mathsf{supp}(\mathcal{D}^-), x \in C_i$ and $y \in C_j$ such that $i, j \in [m]$ and $i < j$, we have*

$$\mathsf{sgn}_h(x,y) = 1.$$

Proof. Let b be the most significant coordinate on which x and y differ, i. e., $x_b \neq y_b$. Since $x \in C_i, y \in C_j$ and $i < j$, we know that $x_b = 0, y_b = 1$ and $b > d$. For all $a > d$, the coefficient of x_a in the definition of h is 3^a, irrespective of whether h is $f_{\mathbf{T}}$ or $g_{\mathbf{T}}$. Then,

$$h(x) \leq \sum_{a \in [d']} 3^a x_a \leq \sum_{a > b} 3^a x_a + \sum_{a < b} 3^a;$$
$$h(y) \geq \sum_{a \geq b} 3^a y_a - \sum_{a < b} 3^a.$$

Thus, $h(y) - h(x) = 3^b - 2\sum_{a<b} 3^a > 0$. □

Now we analyze the sign function on points x, y in the same subcube. Its value is determined by the most significant coordinate on which x and y differ that has a nonzero coefficient in the definitions of hard functions $f_{\mathbf{T}}$ and $g_{\mathbf{T}}$. We define this coordinate next.

Definition 3.11. For any setting of the shared randomness $\mathbf{S}$, index $i \in [m]$, and points $x, y \in C_i$, define $t^i_{\mathbf{S}}(x,y)$ to be $\perp$ if x and y are the same on all coordinates in $([d] \setminus R) \cup \{r_i\}$; otherwise, define $t^i_{\mathbf{S}}(x,y)$ to be the *most significant coordinate* in $([d] \setminus R) \cup \{r_i\}$ on which x and y differ.

Note that $\mathbf{S}$ determines R and $\{r_i\}$. For any $\mathbf{T}$ that extends $\mathbf{S}$, the restrictions of both $f_{\mathbf{T}}$ and $g_{\mathbf{T}}$ to C_i is constant with respect to the coordinates in $R \setminus \{r_i\}$. Thus, $t^i_{\mathbf{S}}(x,y)$ is $\perp$ if x and y differ only on coordinates that do not influence the value of the function in C_i; otherwise, it is the first coordinate on which x and y differ that is influential in C_i.

Claim 3.12. *Fix some shared randomness $\mathbf{S}$, index $i \in [m]$, and points $x, y \in C_i$ where $x <_{\mathsf{val}} y$. Let $a = t^i_{\mathbf{S}}(x,y)$. For any $\mathbf{T}$ that extends $\mathbf{S}$,*

- *if $a = \perp$, then $\mathsf{sgn}_{f_{\mathbf{T}}}(x,y) = \mathsf{sgn}_{g_{\mathbf{T}}}(x,y) = 0$;*
- *if $a \in ([d] \setminus R)$, then $\mathsf{sgn}_{f_{\mathbf{T}}}(x,y) = \mathsf{sgn}_{g_{\mathbf{T}}}(x,y) = 1$;*
- *if $a = r_i$, $\mathsf{sgn}_{f_{\mathbf{T}}}(x,y) = \alpha_a$ and $\mathsf{sgn}_{g_{\mathbf{T}}}(x,y) = \beta_i$.*

Proof. Recall that

$$f_{\mathbf{T}}(x) = \sum_{b \in [d'] \setminus R} x_b 3^b + \alpha_{r_i} x_{r_i} 3^{r_i};$$
$$g_{\mathbf{T}}(x) = \sum_{b \in [d'] \setminus R} x_b 3^b + \beta_i x_{r_i} 3^{r_i}.$$

First, consider the case $a = \perp$. Then $x_b = y_b$ for all $b \in ([d'] \setminus R) \cup \{r_i\}$. By the definition of $f_{\mathbf{T}}$ and $g_{\mathbf{T}}$, we get $f_{\mathbf{T}}(x) = f_{\mathbf{T}}(y)$ and $g_{\mathbf{T}}(x) = g_{\mathbf{T}}(y)$. Hence, $\mathsf{sgn}_{f_{\mathbf{T}}}(x,y) = \mathsf{sgn}_{g_{\mathbf{T}}}(x,y) = 0$.

Next, consider the case $a \neq r_i$. Then $a \notin R$ by the definition of $t^i_{\mathbf{S}}(x,y)$. Also, $x_b = y_b$ for all $b > a$ such that $b \notin R$. Since $x <_{\mathsf{val}} y$, we have $x_a = 0$ and $y_a = 1$. Thus, $f_{\mathbf{T}}(y) - f_{\mathbf{T}}(x) \geq 3^a - \sum_{b<a} 3^b > 0$ implying that $\mathsf{sgn}_{f_{\mathbf{T}}}(x,y) = 1$. The same argument holds for $g_{\mathbf{T}}$.

Finally, consider the case $a = r_i$. Thus, $f_{\mathbf{T}}(y) - f_{\mathbf{T}}(x) = \alpha_a 3^a + \sum_{b<a, b\notin R}(y_b - x_b)3^b$. Note that $-3^a < \sum_{b<a,b\notin R}(y_b - x_b)3^b < 3^a$. Hence, $f_{\mathbf{T}}(y) - f_{\mathbf{T}}(x)$ is positive when $\alpha_a = 1$ and negative when $\alpha_a = -1$. This implies that $\mathsf{sgn}_{f_{\mathbf{T}}}(x,y) = \alpha_a$. By an analogous argument, $\mathsf{sgn}_{g_{\mathbf{T}}}(x,y) = \beta_i$. □

We complete Section 3.3 by using Claims 3.10 and 3.12 to analyze the distance to unateness for functions distributed according to our hard distributions.

Claim 3.13. *Every function $f_{\mathbf{T}} \in \mathsf{supp}(\mathcal{D}^+)$ is unate.*

Proof. Fix some $f_{\mathbf{T}} \in \mathsf{supp}(\mathcal{D}^+)$. We need to prove that along each dimension $b \in [d']$, the function $f_{\mathbf{T}}$ is either nonincreasing or nondecreasing. Recall that a *b-edge* is a pair (x,y) of points in $\{0,1\}^{d'}$, where $x_b = 0, y_b = 1$, and $x_a = y_a$ for all $a \in [d'] \setminus \{b\}$. Consider a b-edge $\{x,y\}$. If $b \in [d'] \setminus R$, then, by Claims 3.10 and 3.12, $\mathsf{sgn}_{f_{\mathbf{T}}}(x,y) = 1$, and hence $f_{\mathbf{T}}$ is increasing in dimension b.

If $b \in R$, then, by Claim 3.12, $\mathsf{sgn}_{f_{\mathbf{T}}}(x,y)$ is 0 or α_b. That is, if $\alpha_b = 1$, then $f_{\mathbf{T}}$ is nondecreasing, and if $\alpha_b = -1$, then $f_{\mathbf{T}}$ is nonincreasing. □

We write $f \sim \mathcal{D}$ to denote that f is sampled from distribution $\mathcal{D}$.

Claim 3.14. *For sufficiently large d, a function $g_{\mathbf{T}} \sim \mathcal{D}^-$ is $1/8$-far from unate with probability at least $19/20$.*

Proof. For each $r \in R$, let $A_r = \{i \in [m] \mid r_i = r\}$ be the set of subcube indices with $r_i = r$. Then

$$\sum_{r\in R} |A_r| = m = d. \tag{3.1}$$

Hence, $\mathbf{E}[|A_r|] = d/|R| \geq \sqrt{d}$, since $|R| \leq \sqrt{d}$. By Chernoff and union bounds, for all $r \in R$, we have $|A_r| \geq \sqrt{d}/2$ with probability at least $1 - d\exp(-\sqrt{d}/8)$.

Condition on the event that $|A_r| \geq \sqrt{d}/2$ for all $r \in R$. Fix an $r \in R$. For each $i \in A_r$, there is a random choice of β_i. Partition A_r into A_r^+ and A_r^-, depending on whether β_i is $+1$ or -1. Again, by Chernoff and union bounds, for all $r \in R$, we have $\min(|A_r^+|, |A_r^-|) \geq |A_r|/4$ with probability at least $1 - d\exp(-\sqrt{d}/32)$.

Thus, for sufficiently large d and any choice of R,

$$\Pr\left[\forall r \in R, \min(|A_r^+|, |A_r^-|) \geq \frac{|A_r|}{4}\right] \geq 1 - d\left(\exp\left(-\frac{\sqrt{d}}{8}\right) + \exp\left(-\frac{\sqrt{d}}{32}\right)\right) \geq \frac{19}{20}.$$

Note that for all $i \in [m]$, the size $|C_i| = 2^d$. By Claim 3.12, for any r-edge (x,y) that lies within subcube C_i where $i \in A_r$, the sign $\mathsf{sgn}_{g_{\mathbf{T}}}(x,y) = \beta_i$. Hence, in $g_{\mathbf{T}}$, for all $i \in A_r^+$, all r-edges in subcube

C_i are increasing, whereas, for all $j \in A_r^-$, all r-edges in C_j are decreasing. In any unate function, all these r-edges are either nonincreasing or nondecreasing. Thus $g_{\mathbf{T}}$ differs from a unate function on at least

$$\frac{2^d}{2} \cdot \min(|A_r^+|, |A_r^-|) \geq \frac{2^d |A_r|}{8}$$

points. Overall, we need to change at least $(2^d/8)\sum_{r \in R}|A_r|$ values. Since the A_r's partition the set of subcube indices, $(2^d/8)\sum_{r \in R}|A_r| = 2^{d'}/8$. □

3.4 Bad events

Fix a set of queries Q made by a deterministic, nonadaptive, comparison-based tester. We first identify certain *bad* values of $\mathbf{S}$, on which Q could potentially distinguish between $f_{\mathbf{T}}$ and $g_{\mathbf{T}}$ for any $\mathbf{T}$ that extends $\mathbf{S}$. In this section, we prove that, for a given Q, the probability of a bad $\mathbf{S}$ is small. In Section 3.5, we show that the tester with queries Q cannot distinguish between $f_{\mathbf{T}}$ and $g_{\mathbf{T}}$ for any $\mathbf{T}$ that extends good $\mathbf{S}$.

Recall that when a comparison-based tester queries points x and y, it only sees $\mathsf{sgn}_h(x,y)$. In order for $\mathsf{sgn}_h(x,y)$ to have a possibility of being different for $h \sim \mathcal{D}^+$ and $h \sim \mathcal{D}^-$, by Claims 3.10 and 3.12, queries x and y have to be in the same subcube C_i for some $i \in [m]$ and, moreover, $t^i_{\mathbf{S}}(x,y)$, the most significant coordinate in $([d] \setminus R) \cup \{r_i\}$ on which x and y differ has to be the action dimension r_i for the subcube C_i. Subcubes are fixed, whereas action dimensions R and r_i are chosen randomly. Intuitively, the value of $\mathsf{sgn}_h(x,y)$ is likely to be determined by the top several coordinates in $[d]$ on which x and y differ. (And specifically, considering 5 top coordinates is sufficient to get high enough probability.) Next, we formalize this intuition in the definition of dimensions captured by (x,y) and, more generally, by a set of query points.

Definition 3.15 (Captured coordinates)**.** For points $x,y \in \{0,1\}^{d'}$, define $\mathtt{cap}(x,y)$ to be

1. the set of all coordinates on which x and y differ if the size of the set is at most 5;
2. the set of 5 most significant coordinates on which x and y differ, otherwise.

We say that the pair (x,y) *captures* the set of coordinates $\mathtt{cap}(x,y)$. For a set of points $P \subseteq \{0,1\}^{d'}$, define $\mathtt{cap}(P) := \bigcup_{x,y \in P} \mathtt{cap}(x,y)$ to be the set of all coordinates captured by the set P.

For each $i \in [m]$, let $Q_i := Q \cap C_i$ denote the set of the queried points that lie in the subcube C_i. We define two bad events for $\mathbf{S}$.

Definition 3.16 (Bad events)**.** $\mathbf{S}$ is *bad* if at least one of the following bad events holds:

- Abort Event $\mathcal{A}$: There exist $x,y \in Q$ such that $|\mathtt{cap}(x,y)| = 5$ and $\mathtt{cap}(x,y) \subseteq R$.
- Collision Event $\mathcal{C}$: There exist distinct $i,j \in [m]$ with $r_i = r_j$, such that $r_i \in \mathtt{cap}(Q_i)$ and $r_j \in \mathtt{cap}(Q_j)$.

If $\mathcal{A}$ does not occur, then for every pair (x,y), the sign $\mathsf{sgn}_h(x,y)$ is determined by $\mathtt{cap}(x,y)$ for any $h \in \mathsf{supp}(\mathcal{D}^+) \cup \mathsf{supp}(\mathcal{D}^-)$. And, if $\mathcal{C}$ does not occur, conditioned on $\mathcal{A}$ not occurring, then the tester cannot distinguish $\mathcal{D}^+$ and $\mathcal{D}^-$. The heart of the analysis lies in proving that the bad events happen rarely.

Lemma 3.17. *Let $\delta_0 = 1/20000$. If d is sufficiently large and $|Q| \le \delta_0 d \log d$, then* $\Pr[\mathcal{A}] < 0.01$.

Proof. Recall that k is the log of the number of action dimensions. Fix any choice of k (in $\mathbf{S}$). For each pair of points $x, y \in Q$ such that $|\mathtt{cap}(x,y)| = 5$,

$$\Pr[\mathtt{cap}(x,y) \subseteq R] = \frac{2^k}{d} \cdot \frac{2^k - 1}{d-1} \cdots \frac{2^k - 4}{d-4} < \left(\frac{2^k}{d-4}\right)^5.$$

Since $d - 4 \ge d/2$ for all $d \ge 8$ and $k \le (\log d)/2$, the probability is at most $32 d^{-5/2}$. By the union bound,

$$\Pr[\mathcal{A}] < |Q \times Q| \cdot 32 d^{-5/2} < \delta_0^2 d^2 \log^2 d \cdot 32 d^{-5/2} = 32 \delta_0^2 d^{-1/2} \log^2 d < 0.01$$

for all sufficiently large d. □

To analyze the probability of the collision event $\mathcal{C}$, we prove the following combinatorial lemma that will be used to bound the number of coordinates Q captures.

Lemma 3.18. *Let $c, d \in \mathbb{N}$ and V be a set of d-dimensional vectors over an arbitrary alphabet. For all $x, y \in V$,*

1. *if x and y differ on more than c coordinates, then let $\mathtt{cap}_c(x,y)$ denote the set of the first c coordinates on which x and y differ;*
2. *otherwise, let $\mathtt{cap}_c(x,y)$ denote the set of all coordinates on which x and y differ.*

Define $\mathtt{cap}_c(V) := \bigcup_{x,y \in V} \mathtt{cap}_c(x,y)$. Then $|\mathtt{cap}_c(V)| \le c(|V| - 1)$.

Proof. We construct c different edge-colored graphs $G_1, \ldots, G_c$ over the vertex set V with colors from $[d]$. For every coordinate $i \in \mathtt{cap}_c(V)$, add one edge of color i to exactly one of the graphs G_t. Since $i \in \mathtt{cap}_c(V)$, there exists at least one pair of vectors x, y such that $i \in \mathtt{cap}_c(x,y)$. Thinking of each $\mathtt{cap}_c(x,y)$ as an ordered set, find one pair of vectors (x,y) where i appears "earliest" in $\mathtt{cap}_c(x,y)$. Let the position of i in this $\mathtt{cap}_c(x,y)$ be denoted by t. Add the edge (x,y) to G_t and color it i. Note that each edge (x,y) is added to G_t at most once, and hence each graph G_t is simple.

We claim that each G_t is acyclic. Suppose not. Let C be a cycle in G_t. Let (x,y) be the edge in C with the smallest color i. Clearly, $x_i \ne y_i$, since $i \in \mathtt{cap}_c(x,y)$. There must exist another edge (u,v) in C such that $u_i \ne v_i$. Furthermore, the color of (u,v) is $j > i$. Thus, j is the t^{th} entry in $\mathtt{cap}_c(u,v)$. Note that $i \in \mathtt{cap}_c(u,v)$, since $u_i \ne v_i$ and $j > i$. It follows that i appears earlier in $\mathtt{cap}_c(u,v)$ than in $\mathtt{cap}_c(x,y)$. So the edge colored i should not be in G_t, a contradiction. □

Lemma 3.18 implies that a small Q can capture only a few coordinates. Next we bound the probability of the collision event.

Lemma 3.19. *If $|Q| \le \delta_0 d \log d$, then* $\Pr[\mathcal{C}] < 0.04$.

Proof. By Lemma 3.18,

$$\sum_{i \in [m]} |\mathtt{cap}(Q_i)| \le 5 \sum_{i \in [m]} |Q_i| = 5|Q| \le 5 \delta_0 d \log d. \tag{3.2}$$

For each $r \in [d]$, define $A_r := \{j \in [m] \mid r \in \texttt{cap}(Q_j)\}$ to be the set of indices of the subcubes in which coordinate r is captured. Let $a_r := |A_r|$. For each $\ell \in [\log d]$, define $n_\ell := |\{r \in [d] \mid a_r \in (2^{\ell-1}, 2^\ell]\}|$ to be the number of coordinates that are captured by more than $2^{\ell-1}$, but at most 2^ℓ subcubes. Observe that

$$\sum_{\ell \in [\log d]} n_\ell 2^\ell \leq 2 \sum_{r \in [d]} a_r = 2 \sum_{i \in [m]} |\texttt{cap}(Q_i)| \leq 10\delta_0 d \log d, \tag{3.3}$$

where the first inequality holds because if coordinate r is included in the count n_ℓ then $2^{\ell-1} < a_r$, and the last inequality follows from Equation (3.2).

Fix the log of the action dimensions, k, to be some $\kappa \in [(1/2)\log d]$. For each $r \in [d]$, we say the event $\mathcal{C}_r$ occurs if $r \in R$ and there exist distinct $i, j \in [m]$ such that $r_i = r_j = r$, and $r \in \texttt{cap}(Q_i) \cap \texttt{cap}(Q_j)$. By the union bound, $\Pr[\mathcal{C} \mid k = \kappa] \leq \sum_{r \in [d]} \Pr[\mathcal{C}_r \mid k = \kappa]$.

Now, we compute $\Pr[\mathcal{C}_r | k = \kappa]$. Conditioning on $k = \kappa$, the size of the set of the action dimensions, $|R|$, is 2^κ. Only sets Q_j with $j \in A_r$ capture coordinate r. Event $\mathcal{C}_r$ occurs if at least two of these sets have $r_i = r_j = r$. Hence,

$$\Pr[\mathcal{C}_r \mid k = \kappa] = \Pr[r \in R] \cdot \Pr[\exists \text{ distinct } i, j \in A_r \text{ such that } r_i = r_j = r \mid r \in R] \tag{3.4}$$

$$= \frac{2^\kappa}{d} \cdot \sum_{c=2}^{a_r} \Pr\left[\text{There are exactly } c \text{ subcube indices } i \in A_r \text{ with } r_i = r \mid r \in R\right]$$

$$= \frac{2^\kappa}{d} \cdot \sum_{c=2}^{a_r} \binom{a_r}{c} \left(\frac{1}{2^\kappa}\right)^c \left(1 - \frac{1}{2^\kappa}\right)^{a_r - c}, \tag{3.5}$$

where we used $\Pr[r \in R] = 2^\kappa/d$ to get the second equality, and $\Pr[r_i = r \mid r \in R] = 2^{-\kappa}$ for each $i \in [m]$ to get the third equality.

If $a_r > 2^\kappa/4$, by Equation (3.4),

$$\Pr[\mathcal{C}_r \mid k = \kappa] \leq \Pr[r \in R] = 2^\kappa/d. \tag{3.6}$$

If $a_r \leq 2^\kappa/4$, by Equation (3.5),

$$\begin{aligned} \Pr[\mathcal{C}_r \mid k = \kappa] &= \frac{2^\kappa}{d} \cdot \sum_{c=2}^{a_r} \binom{a_r}{c} \left(\frac{1}{2^\kappa}\right)^c \left(1 - \frac{1}{2^\kappa}\right)^{a_r - c} \\ &< \frac{2^\kappa}{d} \left(1 - \frac{1}{2^\kappa}\right)^{a_r} \sum_{c=2}^{a_r} \left(a_r \cdot \frac{1}{2^\kappa} \cdot \left(1 - \frac{1}{2^\kappa}\right)^{-1}\right)^c \\ &\leq \frac{2^\kappa}{d} \sum_{c=2}^{a_r} \left(\frac{a_r}{2^{\kappa-1}}\right)^c \\ &< \frac{2^\kappa}{d} \cdot 2 \left(\frac{a_r}{2^{\kappa-1}}\right)^2 = \frac{8a_r^2}{2^\kappa d}, \end{aligned} \tag{3.7}$$

where we used

$$\binom{a_r}{c} < a_r^c, \quad (1 - 2^{-\kappa})^{-1} \leq 2 \quad \text{and} \quad \frac{a_r}{2^{\kappa-1}} \leq \frac{1}{2}$$

to get the first, second and third inequalities, respectively. Using the union bound over all $r \in [d]$ and grouping according to n_ℓ, we get

$$\begin{aligned}
\Pr[\mathcal{C} \mid k=\kappa] \le \sum_{r=1}^{d} \Pr[\mathcal{C}_r \mid k=\kappa] &= \sum_{r\in[d]:a_r>2^\kappa/4} \Pr[\mathcal{C}_r \mid k=\kappa] + \sum_{r\in[d]:a_r\le 2^\kappa/4} \Pr[\mathcal{C}_r \mid k=\kappa] \\
&\le \sum_{r\in[d]:a_r>2^{\kappa-2}} \frac{2^\kappa}{d} + \sum_{r\in[d]:a_r\le 2^{\kappa-2}} \frac{8a_r^2}{2^\kappa d} \\
&\le \frac{2^\kappa}{d} \sum_{\ell=\max(\kappa-1,1)}^{\log d} n_\ell + \frac{8}{d}\sum_{\ell=1}^{\kappa-2} n_\ell 2^{2\ell-\kappa}, \qquad (3.8)
\end{aligned}$$

where the second last inequality follows from Equations (3.6) and (3.7), and the last inequality holds because if coordinate r is included in the count n_ℓ, then $2^{\ell-1} < a_r \le 2^\ell$.

Averaging over all the values of k, we get

$$\begin{aligned}
\Pr[\mathcal{C}] &= \frac{2}{\log d} \sum_{\kappa=1}^{\frac{\log d}{2}} \Pr[\mathcal{C}|k=\kappa] \\
&\le \frac{16}{d\log d} \sum_{\kappa=1}^{\frac{\log d}{2}} \left(2^\kappa \sum_{\ell=\max(\kappa-1,1)}^{\log d} n_\ell + \sum_{\ell=1}^{\kappa-2} n_\ell 2^{2\ell-\kappa} \right) \\
&\le \frac{16}{d\log d} \left(\sum_{\ell=1}^{\log d} n_\ell \sum_{\kappa=1}^{\ell+1} 2^\kappa + \sum_{\ell=1}^{\frac{\log d}{2}-2} n_\ell \sum_{\kappa=\ell+2}^{\frac{\log d}{2}} 2^{2\ell-\kappa} \right), \qquad (3.9)
\end{aligned}$$

where the first inequality follows from Equation (3.8) and the last inequality is obtained by switching the order of summations and rearranging the terms. Now,

$$\sum_{\kappa=1}^{\ell+1} 2^\kappa < 2^{\ell+2} \text{ and } \sum_{\kappa=\ell+2}^{\frac{\log d}{2}} 2^{2\ell-\kappa} < 2^\ell .$$

Using these inequalities to bound the sum in Equation (3.9) and then applying Equation (3.3), we get

$$\Pr[\mathcal{C}] < \frac{80}{d\log d} \sum_{\ell=1}^{\log d} n_\ell 2^\ell \le \frac{80 \cdot 10\delta_0 d \log d}{d \log d} = 0.04 . \qquad \square$$

3.5 Indistinguishability of hard distributions

Proof of Theorem 3.8. Fix a deterministic, nonadaptive, comparison-based tester making a set of queries Q whose size $|Q| \le \delta_0 d \log d$. (Recall that $\delta_0 = 1/20000$.) The tester decides whether the input function h is unate based on the $\binom{|Q|}{2}$ comparisons between the function values at points in Q. We define a labeled undirected graph G_h^Q to be the clique on the node set Q, where each edge $\{x,y\}$ such that $x <_{\mathsf{val}} y$ is labeled by $\mathsf{sgn}_h(x,y)$. The graph G_h^Q defines the tester's "view" of the function h after it has made its queries Q.

For any distribution $\mathcal{D}$ over functions $f : \{0,1\}^{d'} \to \mathbb{N}$, let $\mathcal{D}$-view denote the distribution of the labeled graphs G_h^Q when $h \sim \mathcal{D}$. For two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$, let $\mathcal{D}_1 \approx_\alpha \mathcal{D}_2$ denote that the statistical distance between $\mathcal{D}_1$ and $\mathcal{D}_2$ is at most α. By the version of Yao's principle stated in [57, Claim 5], it is sufficient to give two distributions, $\mathcal{D}_1$ and $\mathcal{D}_2$, on positive and negative instances, respectively, such that the statistical distance between $\mathcal{D}_1$-view and $\mathcal{D}_2$-view is less than $1/6$ for every deterministic tester that makes at most $\delta_0 d \log d$ queries. Specifically, we show that for such testers, $\mathcal{D}_1$-view $\approx_{3/19}$ $\mathcal{D}_2$-view.

We first show that, conditioned on the bad events $\mathcal{A}$ and $\mathcal{C}$ not occurring, the view of the tester is the same for both $\mathcal{D}^+$ and $\mathcal{D}^-$. For a distribution $\mathcal{D}$ and an event E, let $\mathcal{D}|_E$ denote the distribution $\mathcal{D}$ conditioned on the event E.

Lemma 3.20. *For every deterministic, nonadaptive, comparison-based tester,*

$$\mathcal{D}^+\text{-}view|_{\overline{\mathcal{A}\cup\mathcal{C}}} = \mathcal{D}^-\text{-}view|_{\overline{\mathcal{A}\cup\mathcal{C}}}.$$

Proof. Fix some $\mathbf{S} = S$ for Q such that the bad events $\mathcal{A}$ and $\mathcal{C}$ do not occur. Let E_S denote the event that the collection of random variables $\mathbf{T}$ used to define the support of $\mathcal{D}^+$ and $\mathcal{D}^-$ is extended from S. We show that for any labeled clique G over the vertex set Q,

$$\Pr_{f\sim\mathcal{D}^+|_{E_S}}[G_f^Q = G] = \Pr_{g\sim\mathcal{D}^-|_{E_S}}[G_g^Q = G].$$

Let $F = \mathsf{supp}(\mathcal{D}^+|_{E_S}) \cup \mathsf{supp}(\mathcal{D}^-|_{E_S})$. Fix some $G \in \{G_h^Q \mid h \in F\}$. Consider an edge $\{x,y\} \in G$ with $x <_{\mathsf{val}} y$ and $x,y \in \{0,1\}^{d'}$. If x and y lie in different subcubes, then, by Claim 3.10, its label $\mathsf{sgn}_h(x,y) = 1$ for all $h \in F$. Similarly, if x and y lie in the same subcube C_i for some $i \in [m]$, but with $t_{\mathbf{S}}^i(x,y) \neq r_i$, then by Claim 3.12, $\mathsf{sgn}_h(x,y)$ is the same for all $h \in F$. Hence the label of the edge $\{x,y\}$ can potentially distinguish between $\mathcal{D}^+$ and $\mathcal{D}^-$ only if x and y lie in the same subcube C_i for some $i \in [m]$ and $t_{\mathbf{S}}^i(x,y) = r_i$. We call such edges *interesting*. In the rest of this proof, we focus on interesting edges, since the labels of non-interesting edges cannot be used to distinguish between $\mathcal{D}^+$ and $\mathcal{D}^-$.

Fix $i \in [m]$. Recall that $Q_i = Q \cap C_i$ is the set of queried points that lie in the subcube C_i. Let $G(Q_i)$ denote the subgraph of G induced on vertex set Q_i. By Claim 3.12, all interesting edges in $G(Q_i)$ have the same label. Denote this label by ℓ_i.

Let $I = \{i \in [m] \mid G(Q_i) \text{ contains an interesting edge}\}$ denote the set of indices of all subcubes with at least one interesting edge.

Now, we focus on $g \sim \mathcal{D}^-|_{E_S}$. For any edge $\{x,y\} \in G$ with $x <_{\mathsf{val}} y$, let $\ell(x,y)$ denote its label. The probability

$$\begin{aligned}
\Pr_{g\sim\mathcal{D}^-|_{E_S}}[G_g^Q = G] &= \Pr\Big[\bigwedge_{i\in I} \bigwedge_{\substack{\{x,y\}\in G(Q_i),\\ \{x,y\} \text{ is interesting}}} \ell(x,y) = \mathsf{sgn}_g(x,y)\Big] \\
&= \Pr\Big[\bigwedge_{i\in I} \bigwedge_{\substack{\{x,y\}\in G(Q_i),\\ \{x,y\} \text{ is interesting}}} \ell(x,y) = \beta_i\Big] && \text{(by Claim 3.12)} \\
&= \Pr\Big[\bigwedge_{i\in I}(\ell_i = \beta_i)\Big] \\
&= \prod_{i\in I} \Pr[\ell_i = \beta_i] = 2^{-|I|}, && (3.10)
\end{aligned}$$

since each β_i is chosen uniformly and independently at random from $\{+1,-1\}$.

Similarly, for $f \sim \mathcal{D}^+|_{E_S}$, the probability

$$\Pr_{f\sim\mathcal{D}^+|_{E_S}}[G_f^Q = G] = \Pr\Big[\bigwedge_{i\in I}(\ell_i = \alpha_{r_i})\Big]. \tag{3.11}$$

Note that unless there exist distinct $i, j \in I$ such that $r_i = r_j$, the individual events in the probability expression Equation (3.11) are mutually independent. We show that if the bad events $\mathcal{A}$ and $\mathcal{C}$ do not occur then all subcubes C_i with interesting edges have distinct action dimensions r_i.

Claim 3.21. *If $\overline{\mathcal{A}\cup\mathcal{C}}$ then $r_i \neq r_j$ for all distinct $i, j \in I$.*

Proof. Assume for the sake of contradiction that there exist distinct $i, j \in I$ with $r_i = r_j$. The fact that $\overline{\mathcal{C}}$ holds ensures that if $r_i \in \mathtt{cap}(Q_i)$ and $r_j \in \mathtt{cap}(Q_j)$ for distinct $i, j \in [m]$, then $r_i \neq r_j$. Hence, $r_i \notin \mathtt{cap}(Q_i)$ or $r_j \notin \mathtt{cap}(Q_j)$. Suppose without loss of generality $r_i \notin \mathtt{cap}(Q_i)$. Since $i \in I$, there must exist an interesting edge $\{x, y\} \in G(Q_i)$ with $x <_{\mathsf{val}} y$. We know $r_i \notin \mathtt{cap}(Q_i)$, so $r_i \notin \mathtt{cap}(x, y)$. By Definition 3.15, if x and y differ on at most five coordinates, then $r_i \in \mathtt{cap}(x, y)$. Hence, x and y differ on at least six coordinates, and r_i is not among the five most significant coordinates. Since the abort event $\mathcal{A}$ does not occur, $\mathtt{cap}(x, y) \nsubseteq R$. Hence, there exists a coordinate $a \in \mathtt{cap}(x, y)$ that is not in R and is more significant than r_i. Then, $t^i_{\mathbf{S}}(x, y) \geq a$, contradicting the fact that $t^i_{\mathbf{S}}(x, y) = r_i$. Hence, $r_i \neq r_j$ for all distinct $i, j \in I$. □

By Equation (3.11) and Claim 3.21,

$$\Pr_{f\sim\mathcal{D}^+|_{E_S}}[G_f^Q = G] = \prod_{i\in I}\Pr[\ell_i = \alpha_{r_i}] = 2^{-|I|}, \tag{3.12}$$

since α_b is chosen uniformly and independently at random from $\{-1,+1\}$ for each $b \in [d]$. By Equations (3.10) and (3.12),

$$\Pr_{f\sim\mathcal{D}^+|_{E_S}}[G_f^Q = G] = \Pr_{g\sim\mathcal{D}^-|_{E_S}}[G_g^Q = G],$$

completing the proof of Lemma 3.20. □

We now wrap up the proof of Theorem 3.8. Let FAR denote the event that a function $h \sim \mathcal{D}^-$ is $1/8$-far from unate. Define $\widehat{\mathcal{D}}^-$ to be $\mathcal{D}^-|_{\mathsf{FAR}}$. Note that every function $h \sim \widehat{\mathcal{D}}^-$ is $1/8$-far from unate. By Claim 3.13, every function $h \sim \mathcal{D}^+$ is unate. Hence, to complete the proof of this theorem, it suffices to show that $\mathcal{D}^+$-view $\approx_{3/19}$ $\widehat{\mathcal{D}}^-$-view.

By Claim 3.14, $\Pr[\mathsf{FAR}] \geq 19/20$. We use the following claim from [57] to show that $\mathcal{D}^-$ and $\widehat{\mathcal{D}}^-$ are close.

Claim 3.22 (Claim 4 in [57])**.** *Let E be an event that happens with probability at least $\alpha = 1 - 1/a$ under the distribution $\mathcal{D}$. Then, $\mathcal{D}|_E \approx_{\alpha'} \mathcal{D}$ where $\alpha' = 1/(a-1)$.*

By Claim 3.22, $\mathcal{D}^- \approx_{1/19} \widehat{\mathcal{D}}^-$. Consequently,

$$\mathcal{D}^-\text{-view} \approx_{1/19} \widehat{\mathcal{D}}^-\text{-view}. \tag{3.13}$$

By Lemmas 3.17 and 3.19, the probability $\Pr[\overline{\mathcal{A}\cup\mathcal{C}}] \geq 19/20$. Using Claim 3.22, we get

$$\mathcal{D}^{+}\text{-view} \approx_{1/19} \mathcal{D}^{+}\text{-view}|_{\overline{\mathcal{A}\cup\mathcal{C}}}, \tag{3.14}$$

$$\mathcal{D}^{-}\text{-view} \approx_{1/19} \mathcal{D}^{-}\text{-view}|_{\overline{\mathcal{A}\cup\mathcal{C}}}. \tag{3.15}$$

Using Equation (3.14), Lemma 3.20, Equation (3.15), and, finally, Equation (3.13), we get

$$\mathcal{D}^{+}\text{-view} \approx_{1/19} \mathcal{D}^{+}\text{-view}|_{\overline{\mathcal{A}\cup\mathcal{C}}} = \mathcal{D}^{-}\text{-view}|_{\overline{\mathcal{A}\cup\mathcal{C}}} \approx_{1/19} \mathcal{D}^{-}\text{-view} \approx_{1/19} \widehat{\mathcal{D}}^{-}\text{-view}.$$

Hence $\mathcal{D}^{+}$-view $\approx_{3/19} \widehat{\mathcal{D}}^{-}$-view, completing the proof of Theorem 3.8. □

4 Other lower bounds

4.1 The lower bound for adaptive testers over the hypergrid

We show that every unateness tester for functions $f : [n]^d \to \mathbb{R}$ requires

$$\Omega\left(\frac{d \log n}{\varepsilon} - \frac{\log 1/\varepsilon}{\varepsilon}\right)$$

queries for $\varepsilon \in (0, 1/4)$ and prove Theorem 1.3.

Proof of Theorem 1.3. By Yao's minimax principle and the reduction to testing with comparison-based testers from [35, 23] (stated for completeness in Theorem 3.3), it is sufficient to give a hard input distribution on which every deterministic comparison-based tester fails with probability at least $2/3$. We use the hard distribution constructed by Chakrabarty and Seshadhri [23], who used it to prove the same lower bound for testing monotonicity. Their distribution is a mixture of two distributions, $\mathcal{D}^+$ and $\mathcal{D}^-$, on positive and negative instances, respectively. The positive instances for their problem are functions that are monotone and, therefore, unate; the negative instances are functions that are ε-far from monotone. We show that their distribution $\mathcal{D}^-$ is supported on functions that are ε-far from unate, i. e., negative instances for our problem. Then the required lower bound for unateness follows from the fact that every deterministic comparison-based tester needs the stated number of queries to distinguish the distributions $\mathcal{D}^+$ and $\mathcal{D}^-$ with high enough probability.

We start by describing the distributions $\mathcal{D}^+$ and $\mathcal{D}^-$ used in [23]. We will define them as distributions on functions over the hypercube domain. Next, we explain how to convert functions over hypercubes to functions over hypergrids.

Without loss of generality, assume n is a power of 2 and let $\ell := \log_2 n$. For all $z \in [n]$, let $\mathsf{bin}(z)$ denote the binary representation of $z-1$ as an ℓ-bit vector $(z_1, \ldots, z_\ell)$, where z_1 is the least significant bit.

We now describe the mapping used to convert functions on hypergrids to functions on hypercubes. Let $\phi : [n]^d \to \{0,1\}^{d\ell}$ be the mapping that takes $y \in [n]^d$ to the concatenation of $\mathsf{bin}(y_1), \ldots, \mathsf{bin}(y_d)$. Any function $f : \{0,1\}^{d\ell} \to \mathbb{R}$ can be easily converted into a function $\widetilde{f} : [n]^d \to \mathbb{R}$, where $\widetilde{f}(y) := f(\phi(y))$.

Let $m := d\ell$. For $x \in \{0,1\}^m$, recall that $\mathsf{val}(x) = \sum_{i=1}^{m} x_i 2^{i-1}$ denotes the integer equivalent of the binary number represented by vector x. For simplicity, assume $1/\varepsilon$ is a power of 2. Partition the set of points $x \in \{0,1\}^m$ according to the most significant $\log(1/2\varepsilon)$ dimensions. That is, for $k \in [1/2\varepsilon]$, let

$$S_k := \{x : \mathsf{val}(x) \in [(k-1)\cdot \varepsilon 2^{m+1}, k \cdot \varepsilon 2^{m+1} - 1]\}.$$

The hypercube is partitioned into $1/2\varepsilon$ sets S_k of equal size, and each S_k forms a subcube of dimension $m' = m - \log(1/\varepsilon) + 1$.

We now describe the distributions $\mathcal{D}^+$ and $\mathcal{D}^-$ for functions on hypercubes. The distribution $\mathcal{D}^+$ consists of a single function $f(x) = 2\mathsf{val}(x)$. The distribution $\mathcal{D}^-$ is the uniform distribution over $m'/2\varepsilon$ functions $g_{j,k}$, where $j \in [m']$ and $k \in [1/2\varepsilon]$, defined as follows:

$$g_{j,k}(x) = \begin{cases} 2\mathsf{val}(x) - 2^j - 1 & \text{if } x_j = 1 \text{ and } x \in S_k; \\ 2\mathsf{val}(x) & \text{otherwise.} \end{cases}$$

To get the distributions $\mathcal{D}^+$ and $\mathcal{D}^-$ for the hypergrid, we convert f to $\widetilde{f}$ and each function $g_{j,k}$ to $\widetilde{g_{j,k}}$, using the transformation defined before.

Chakrabarty and Seshadhri [23] proved that f is monotone and each function $\widetilde{g_{j,k}}$ is ε-far from monotone. It remains to show that functions $\widetilde{g_{j,k}}$ are also ε-far from unate.

Claim 4.1. *Each function $\widetilde{g_{j,k}}$ is ε-far from unate.*

Proof. To prove that $\widetilde{g_{j,k}}$ is ε-far from unate, it suffices to show that there exists a dimension i, such that there are at least $\varepsilon 2^{d\ell}$ increasing i-pairs and at least $\varepsilon 2^{d\ell}$ decreasing i-pairs w.r.t. $\widetilde{g_{j,k}}$ and that all of these i-pairs are disjoint. Let $u, v \in [n]^d$ be two points such that $\phi(u)$ and $\phi(v)$ differ only in the jth bit. Clearly, u and v form an i-pair, where $i = \lceil j/\ell \rceil$. Now, if $\phi(u), \phi(v) \in S_k$ and $u \prec v$, then $\widetilde{g_{j,k}}(v) = \widetilde{g_{j,k}}(u) - 1$. So, the i-pair (u, v) is decreasing. The total number of such i-pairs is $2^{d\ell - \log(1/2\varepsilon) - 1} = \varepsilon 2^{d\ell}$. If $\phi(u), \phi(v) \in S_{k'}$ where $k' \neq k$, then the i-pair (u, v) is increasing. Clearly, there are at least $\varepsilon 2^{d\ell}$ such i-pairs. All the i-pairs we mentioned are disjoint. Hence, $\widetilde{g_{j,k}}$ is ε-far from unate. □

This completes the proof of Theorem 1.3. □

4.2 The lower bound for nonadaptive testers over the hypergrid

The lower bound for nonadaptive testers over hypergrids follows from a combination of the lower bound for nonadaptive testers over hypercube and the lower bound for adaptive testers over hypergrids.

Proof of Theorem 1.4. Fix $\varepsilon = 1/8$. The proof consists of two parts. The lower bound for adaptive testers is also a lower bound for nonadaptive testers, and so, the bound of $\Omega(d \log n)$ holds. Next, we extend the $\Omega(d \log d)$ lower bound for hypercubes. Assume n to be even. Define function $\psi : [n] \to \{0,1\}$ as $\psi(a) := \mathbb{1}[a > n/2]$ for $a \in [n]$. For $x = (x_1, x_2, \ldots, x_d) \in [n]^d$, define the mapping $\Psi : [n]^d \to \{0,1\}^d$ as

$$\Psi(x) := (\psi(x_1), \psi(x_2), \ldots, \psi(x_d)).$$

Any function $f : \{0,1\}^d \to \mathbb{R}$ can be extended to $\tilde{f} : [n]^d \to \mathbb{R}$ using the mapping $\tilde{f}(x) = f(\Psi(x))$ for all $x \in [n]^d$. The proof of Theorem 3.8 goes through for hypergrids as well, and so we have an $\Omega(d \log d)$ lower bound. Combining the two lower bounds, we get a bound of $\Omega(d \cdot \max\{\log n, \log d\})$, which is asymptotically equal to $\Omega(d(\log n + \log d))$. □

5 Conclusion and open questions

In this work, we give the first algorithms for testing unateness of real-valued functions over the hypercube as well as the hypergrid domains. We also show that our algorithms are optimal by proving matching lower bounds, thus resolving the query complexity of testing unateness of real-valued functions. Our results demonstrate that, for real-valued functions, adaptivity helps with testing unateness, which is not the case in monotonicity testing.

Subsequent to the initial publication of this work [4], the problem of testing unateness of Boolean functions received significant attention. Concurrent with our work, Chen et al. [29] proved a lower bound of $\Omega\left(d^{2/3}/\log^3 d\right)$ for adaptive unateness testers of Boolean functions over $\{0,1\}^d$. Subsequently, Chen et al. [30] gave an adaptive unateness tester with query complexity $\widetilde{O}\left(d^{3/4}/\varepsilon^2\right)$ for the same class of functions. An exciting recent development is an $\widetilde{O}\left(d^{2/3}/\varepsilon^2\right)$-query algorithm for this problem by Chen and Waingarten [28], which only leaves a polylogarithmic (in d) gap between the upper bound and the lower bound.

Next, we discuss nonadaptive unateness testing of Boolean functions over $\{0,1\}^d$. In a subsequent work, Baleshzar et al. [3] proved a lower bound of $\Omega(d/\log d)$ for one-sided error testers for this problem. Since Boolean functions are a special case of real-valued functions, our nonadaptive algorithm over the hypercube also works for Boolean functions. This algorithm has 1-sided error and its query complexity is $O((d/\varepsilon)\log(d/\varepsilon))$ which is currently the best known upper bound for any (even 2-sided error) nonadaptive algorithm. There is still a polylogarithmic (in d) gap between the upper bound and the lower bound. An interesting open question is to determine if testers with two-sided error have better query complexity than testers with one-sided error in the nonadaptive setting.

Finally, we mention some recent work on approximating the distance to unateness of Boolean functions over the hypercube domain. Levi and Waingarten [48] showed that every algorithm approximating the distance to unateness within a constant factor requires $\widetilde{\Omega}(d)$ queries and strengthened their lower bound to $\widetilde{\Omega}(d^{3/2})$ queries for nonadaptive algorithms. Subsequently, Pallavoor et al. [50] proved that every nonadaptive algorithm approximating the distance to unateness within a $d^{1/2-k}$ factor for $k > 0$ requires 2^{d^k} queries. No nontrivial upper bounds are currently known for this problem.

Acknowledgments. We thank Oded Goldreich for useful discussions and Meiram Murzabulatov for participation in initial discussions on this work. We would like to thank *Theory of Computing* and László Babai personally for the unparalleled copy editing service they provided. We have never seen anything like this from other journals. The comments and suggestions we received helped us greatly improve our paper. We are also pleased with its appearance.

References

[1] NIR AILON AND BERNARD CHAZELLE: Information theory in property testing and monotonicity testing in higher dimension. *Inform. and Comput.*, 204(11):1704–1717, 2006. Preliminary version in STACS'05. [doi:10.1016/j.ic.2006.06.001] 2

[2] PRANJAL AWASTHI, MADHAV JHA, MARCO MOLINARO, AND SOFYA RASKHODNIKOVA: Testing Lipschitz functions on hypergrid domains. *Algorithmica*, 74(3):1055–1081, 2016. Preliminary version in RANDOM'12. [doi:10.1007/s00453-015-9984-y] 2

[3] ROKSANA BALESHZAR, DEEPARNAB CHAKRABARTY, RAMESH KRISHNAN S. PALLAVOOR, SOFYA RASKHODNIKOVA, AND C. SESHADHRI: A lower bound for nonadaptive, one-sided error testing of unateness of Boolean functions over the hypercube. *Electron. Colloq. on Comput. Complexity (ECCC)*, 2017. [ECCC:TR17-111, arXiv:1706.00053] 28

[4] ROKSANA BALESHZAR, DEEPARNAB CHAKRABARTY, RAMESH KRISHNAN S. PALLAVOOR, SOFYA RASKHODNIKOVA, AND C. SESHADHRI: Optimal unateness testers for real-valued functions: Adaptivity helps. In *Proc. 44th Internat. Colloq. on Automata, Languages and Programming (ICALP'17)*, pp. 5:1–5:14. Springer, 2017. [doi:10.4230/LIPIcs.ICALP.2017.5, arXiv:1703.05199] 1, 28

[5] TUĞKAN BATU, RONITT RUBINFELD, AND PATRICK WHITE: Fast approximate PCPs for multidimensional bin-packing problems. *Inform. and Comput.*, 196(1):42–56, 2005. Preliminary version in RANDOM'99. [doi:10.1016/j.ic.2004.10.001] 2

[6] MIHIR BELLARE, DON COPPERSMITH, JOHAN HÅSTAD, MARCOS A. KIWI, AND MADHU SUDAN: Linearity testing in characteristic two. *IEEE Trans. Inform. Theory*, 42(6):1781–1795, 1996. Preliminary version in FOCS'95. [doi:10.1109/18.556674] 2

[7] ALEKSANDRS BELOVS: Adaptive lower bound for testing monotonicity on the line. In *Proc. 22nd Internat. Workshop on Randomization and Computation (RANDOM'18)*, pp. 31:1–31:10. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018. [doi:10.4230/LIPIcs.APPROX-RANDOM.2018.31, arXiv:1801.08709] 2

[8] ALEKSANDRS BELOVS AND ERIC BLAIS: Quantum algorithm for monotonicity testing on the hypercube. *Theory of Computing*, 11(16):403–412, 2015. [doi:10.4086/toc.2015.v011a016, arXiv:1503.02868] 2

[9] ALEKSANDRS BELOVS AND ERIC BLAIS: A polynomial lower bound for testing monotonicity. In *Proc. 48th STOC*, pp. 1021–1032. ACM Press, 2016. [doi:10.1145/2897518.2897567, arXiv:1511.05053] 2

[10] MICHAEL BEN-OR, DON COPPERSMITH, MICHAEL LUBY, AND RONITT RUBINFELD: Non-abelian homomorphism testing, and distributions close to their self-convolutions. *Random Structures Algorithms*, 32(1):49–70, 2008. Preliminary version in RANDOM'04. [doi:10.1002/rsa.20182] 2

[11] PIOTR BERMAN, SOFYA RASKHODNIKOVA, AND GRIGORY YAROSLAVTSEV: L_p-testing. In *Proc. 46th STOC*, pp. 164–173. ACM Press, 2014. [doi:10.1145/2591796.2591887] 2, 4, 7

[12] ARNAB BHATTACHARYYA, ELENA GRIGORESCU, KYOMIN JUNG, SOFYA RASKHODNIKOVA, AND DAVID P. WOODRUFF: Transitive-closure spanners. *SIAM J. Comput.*, 41(6):1380–1425, 2012. Preliminary version in SODA'09. [doi:10.1137/110826655, arXiv:0808.1787] 2, 4, 9

[13] Hadley Black, Deeparnab Chakrabarty, and C. Seshadhri: A $o(d)\cdot$ polylog(n) monotonicity tester for Boolean functions over the hypergrid $[n]^d$. In *Proc. 29th ACM-SIAM Symp. on Discrete Algorithms (SODA'18)*, pp. 2133–2151. ACM Press, 2018. [doi:10.1137/1.9781611975031.139, arXiv:1710.10545] 2

[14] Hadley Black, Deeparnab Chakrabarty, and C. Seshadhri: Domain reduction for monotonicity testing: A $o(d)$ tester for Boolean functions in d-dimensions. In *Proc. 31st ACM-SIAM Symp. on Discrete Algorithms (SODA'20)*, pp. 1975–1994. ACM Press, 2020. [doi:10.1137/1.9781611975994.122, arXiv:1811.01427] 2

[15] Eric Blais and Abhinav Bommireddi: Testing submodularity and other properties of valuation functions. In *Proc. 8th Innovations in Theoret. Computer Science (ITCS'17)*, pp. 33:1–33:17. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017. [doi:10.4230/LIPIcs.ITCS.2017.33, arXiv:1611.07879] 2

[16] Eric Blais, Joshua Brody, and Kevin Matulef: Property testing lower bounds via communication complexity. *Comput. Complexity*, 21(2):311–358, 2012. Preliminary version in CCC'11. [doi:10.1007/s00037-012-0040-x] 2, 3, 5

[17] Eric Blais, Sofya Raskhodnikova, and Grigory Yaroslavtsev: Lower bounds for testing properties of functions over hypergrid domains. In *Proc. 29th IEEE Conf. on Computational Complexity (CCC'14)*, pp. 309–320. IEEE Comp. Soc. Press, 2014. [doi:10.1109/CCC.2014.38] 2

[18] Manuel Blum, Michael Luby, and Ronitt Rubinfeld: Self-testing/correcting with applications to numerical problems. *J. Comput. System Sci.*, 47(3):549–595, 1993. Preliminary version in STOC'90. [doi:10.1016/0022-0000(93)90044-W] 2

[19] 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. Preliminary version in RANDOM'10. [doi:10.1007/s00493-012-2765-1] 2

[20] Deeparnab Chakrabarty: Monotonicity testing. In *Encyclopedia of Algorithms*, pp. 1352–1356. Springer, 2016. [doi:10.1007/978-1-4939-2864-4_699] 2

[21] 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. Preliminary version in SODA'15. [doi:10.1145/3039241, arXiv:1404.0718] 2, 3, 4, 5, 6, 9

[22] Deeparnab Chakrabarty and C. Seshadhri: Optimal bounds for monotonicity and Lipschitz testing over hypercubes and hypergrids. In *Proc. 45th STOC*, pp. 419–428. ACM Press, 2013. [doi:10.1145/2488608.2488661, arXiv:1204.0849] 2, 4, 9

[23] Deeparnab Chakrabarty and C. Seshadhri: An optimal lower bound for monotonicity testing over hypergrids. *Theory of Computing*, 10(17):453–464, 2014. Preliminary version in RANDOM'13. [doi:10.4086/toc.2014.v010a017] 2, 3, 5, 13, 14, 15, 26, 27

[24] DEEPARNAB CHAKRABARTY AND C. SESHADHRI: An $o(n)$ monotonicity tester for Boolean functions over the hypercube. *SIAM J. Comput.*, 45(2):461–472, 2016. Preliminary version in STOC'13. [doi:10.1137/13092770X, arXiv:1302.4536] 2

[25] DEEPARNAB CHAKRABARTY AND C. SESHADHRI: Adaptive Boolean monotonicity testing in total influence time. In *Proc. 10th Innovations in Theoret. Computer Science (ITCS'19)*, pp. 20:1–20:7. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019. [doi:10.4230/LIPIcs.ITCS.2019.20, arXiv:1801.02816] 2

[26] XI CHEN, ANINDYA DE, ROCCO A. SERVEDIO, AND LI-YANG TAN: Boolean function monotonicity testing requires (almost) $n^{1/2}$ non-adaptive queries. In *Proc. 47th STOC*, pp. 519–528. ACM Press, 2015. [doi:10.1145/2746539.2746570, arXiv:1412.5657] 2

[27] XI CHEN, ROCCO A. SERVEDIO, AND LI-YANG TAN: New algorithms and lower bounds for monotonicity testing. In *Proc. 55th FOCS*, pp. 286–295. IEEE Comp. Soc. Press, 2014. [doi:10.1109/FOCS.2014.38, arXiv:1412.5655] 2

[28] XI CHEN AND ERIK WAINGARTEN: Testing unateness nearly optimally. In *Proc. 51st STOC*, pp. 547–558. ACM Press, 2019. [doi:10.1145/3313276.3316351, arXiv:1904.05309] 28

[29] XI CHEN, ERIK WAINGARTEN, AND JINYU XIE: Beyond Talagrand functions: new lower bounds for testing monotonicity and unateness. In *Proc. 49th STOC*, pp. 523–536. ACM Press, 2017. [doi:10.1145/3055399.3055461, arXiv:1702.06997] 2, 28

[30] XI CHEN, ERIK WAINGARTEN, AND JINYU XIE: Boolean unateness testing with $\widetilde{O}(n^{3/4})$ adaptive queries. In *Proc. 58th FOCS*, pp. 868–879. IEEE Comp. Soc. Press, 2017. [doi:10.1109/FOCS.2017.85, arXiv:1708.05786] 28

[31] KASHYAP DIXIT, MADHAV JHA, SOFYA RASKHODNIKOVA, AND ABHRADEEP THAKURTA: Testing the Lipschitz property over product distributions with applications to data privacy. In *Proc. 10th Theory of Cryptography Conf. (TCC'13)*, pp. 418–436. Springer, 2013. [doi:10.1007/978-3-642-36594-2_24, arXiv:1209.4056] 2

[32] KASHYAP DIXIT, SOFYA RASKHODNIKOVA, ABHRADEEP THAKURTA, AND NITHIN M. VARMA: Erasure-resilient property testing. *SIAM J. Comput.*, 47(2):295–329, 2018. Preliminary version in ICALP'16. [doi:10.1137/16M1075661, arXiv:1607.05786] 2

[33] YEVGENIY DODIS, ODED GOLDREICH, ERIC LEHMAN, SOFYA RASKHODNIKOVA, DANA RON, AND ALEX SAMORODNITSKY: Improved testing algorithms for monotonicity. In *Proc. 3rd Internat. Workshop on Randomization and Computation (RANDOM'99)*, pp. 97–108. Springer, 1999. [doi:10.1007/978-3-540-48413-4_10] 2

[34] FUNDA ERGÜN, SAMPATH KANNAN, RAVI KUMAR, RONITT RUBINFELD, AND MAHESH VISWANATHAN: Spot-checkers. *J. Comput. System Sci.*, 60(3):717–751, 2000. Preliminary version in STOC'98. [doi:10.1006/jcss.1999.1692] 2, 4, 9

[35] ELDAR FISCHER: On the strength of comparisons in property testing. *Inform. and Comput.*, 189(1):107–116, 2004. [doi:10.1016/j.ic.2003.09.003] 2, 3, 5, 13, 15, 26

[36] ELDAR FISCHER, ERIC LEHMAN, ILAN NEWMAN, SOFYA RASKHODNIKOVA, RONITT RUBINFELD, AND ALEX SAMORODNITSKY: Monotonicity testing over general poset domains. In *Proc. 34th STOC*, pp. 474–483. ACM Press, 2002. [doi:10.1145/509907.509977] 2

[37] ODED GOLDREICH: *Introduction to Property Testing*. Cambridge Univ. Press, 2017. [doi:10.1017/9781108135252] 4

[38] ODED GOLDREICH, SHAFI GOLDWASSER, ERIC LEHMAN, DANA RON, AND ALEX SAMORODNITSKY: Testing monotonicity. *Combinatorica*, 20(3):301–337, 2000. Preliminary version in FOCS'98. [doi:10.1007/s004930070011] 2

[39] ODED GOLDREICH, SHAFI GOLDWASSER, AND DANA RON: Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998. Preliminary version in FOCS'96. [doi:10.1145/285055.285060] 2

[40] ODED GOLDREICH AND DANA RON: On proximity-oblivious testing. *SIAM J. Comput.*, 40(2):534–566, 2011. Preliminary version in STOC'09. [doi:10.1137/100789646] 4

[41] SHIRLEY HALEVY AND EYAL KUSHILEVITZ: Distribution-free property-testing. *SIAM J. Comput.*, 37(4):1107–1138, 2007. Preliminary version in RANDOM'03. [doi:10.1137/050645804] 2

[42] SHIRLEY HALEVY AND EYAL KUSHILEVITZ: Testing monotonicity over graph products. *Random Structures Algorithms*, 33(1):44–67, 2008. Preliminary version in ICALP'04. [doi:10.1002/rsa.20211] 2

[43] MADHAV JHA AND SOFYA RASKHODNIKOVA: Testing and reconstruction of Lipschitz functions with applications to data privacy. *SIAM J. Comput.*, 42(2):700–731, 2013. Preliminary version in FOCS'11. [doi:10.1137/110840741] 2

[44] TALI KAUFMAN, SIMON LITSYN, AND NING XIE: Breaking the ε-soundness bound of the linearity test over GF(2). *SIAM J. Comput.*, 39(5):1988–2003, 2010. Preliminary version in RANDOM'08. [doi:10.1137/080715548] 2

[45] SUBHASH KHOT, DOR MINZER, AND MULI SAFRA: On monotonicity testing and Boolean isoperimetric-type theorems. *SIAM J. Comput.*, 47(6):2238–2276, 2018. Preliminary version in FOCS'15. [doi:10.1137/16M1065872] 2

[46] SUBHASH KHOT AND IGOR SHINKAR: An $\widetilde{O}(n)$ queries adaptive tester for unateness. In *Proc. 20th Internat. Workshop on Randomization and Computation (RANDOM'16)*, pp. 37:1–37:7. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016. [doi:10.4230/LIPIcs.APPROX-RANDOM.2016.37, arXiv:1608.02451] 2

[47] ERIC LEHMAN AND DANA RON: On disjoint chains of subsets. *J. Combin. Theory Ser. A*, 94(2):399–404, 2001. [doi:10.1006/jcta.2000.3148] 2

[48] AMIT LEVI AND ERIK WAINGARTEN: Lower bounds for tolerant junta and unateness testing via rejection sampling of graphs. In *Proc. 10th Innovations in Theoret. Computer Science (ITCS'19)*, pp. 52:1–52:20. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019. [doi:10.4230/LIPIcs.ITCS.2019.52, arXiv:1805.01074] 28

[49] RAMESH KRISHNAN S. PALLAVOOR, SOFYA RASKHODNIKOVA, AND NITHIN M. VARMA: Parameterized property testing of functions. *ACM Trans. Comput. Theory*, 9(4):17:1–17:19, 2018. Preliminary version in ITCS'17. [doi:10.1145/3155296] 2

[50] RAMESH KRISHNAN S. PALLAVOOR, SOFYA RASKHODNIKOVA, AND ERIK WAINGARTEN: Approximating the distance to monotonicity of Boolean functions. In *Proc. 31st ACM-SIAM Symp. on Discrete Algorithms (SODA'20)*, pp. 1995–2009. ACM Press, 2020. [doi:10.1137/1.9781611975994.123, arXiv:1911.06924] 2, 28

[51] MICHAL PARNAS, DANA RON, AND RONITT RUBINFELD: On testing convexity and submodularity. *SIAM J. Comput.*, 32(5):1158–1184, 2003. Preliminary version in RANDOM'02. [doi:10.1137/S0097539702414026] 2

[52] SOFYA RASKHODNIKOVA: Monotonicity testing. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1999. LINK. 2

[53] SOFYA RASKHODNIKOVA: *Property Testing: Theory and Applications*. Ph. D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2003. LINK. 2

[54] SOFYA RASKHODNIKOVA: Transitive-closure spanners: A survey. In *Property Testing*, volume 6390 of *LNCS*, pp. 167–196. Springer, 2010. [doi:10.1007/978-3-642-16367-8_10] 2, 4

[55] SOFYA RASKHODNIKOVA: Testing if an array is sorted. In *Encyclopedia of Algorithms*, pp. 2219–2222. Springer, 2016. [doi:10.1007/978-1-4939-2864-4_700] 2

[56] SOFYA RASKHODNIKOVA AND RONITT RUBINFELD: Linearity testing/testing Hadamard codes. In *Encyclopedia of Algorithms*, pp. 1107–1110. Springer, 2016. [doi:10.1007/978-1-4939-2864-4_202] 2

[57] SOFYA RASKHODNIKOVA AND ADAM D. SMITH: A note on adaptivity in testing properties of bounded degree graphs. *Electron. Colloq. on Comput. Complexity (ECCC)*, 2006. [ECCC:TR06-089] 16, 24, 25

[58] SOFYA RASKHODNIKOVA AND GRIGORY YAROSLAVTSEV: Learning pseudo-Boolean k-DNF and submodular functions. In *Proc. 24th ACM-SIAM Symp. on Discrete Algorithms (SODA'13)*, pp. 1356–1368. ACM Press, 2013. [doi:10.1137/1.9781611973105.98, arXiv:1208.2294] 2

[59] RONITT RUBINFELD AND MADHU SUDAN: Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996. [doi:10.1137/S0097539793255151] 2

[60] C. SESHADHRI AND JAN VONDRÁK: Is submodularity testable? *Algorithmica*, 69(1):1–25, 2014. Preliminary version in ICS'10. [doi:10.1007/s00453-012-9719-2, arXiv:1008.0831] 2

[61] ANDREW CHI-CHIH YAO: Probabilistic computations: Toward a unified measure of complexity (extended abstract). In *Proc. 18th FOCS*, pp. 222–227. IEEE Comp. Soc. Press, 1977. [doi:10.1109/SFCS.1977.24] 16

AUTHORS

Roksana Baleshzar [about the author]
Software engineer
Google, Inc.
Sunnyvale, CA, USA
roksana.baleshzar@gmail.com

Deeparnab Chakrabarty [about the author]
Assistant professor
Department of Computer Science
Dartmouth College
Hanover, NH, USA
deeparnab@dartmouth.edu
https://www.cs.dartmouth.edu/~deepc/

Ramesh Krishnan S. Pallavoor [about the author]
Graduate student
Department of Computer Science
Boston University
Boston, MA, USA
rameshkp@bu.edu
http://cs-people.bu.edu/rameshkp/

Sofya Raskhodnikova [about the author]
Professor
Department of Computer Science
Boston University
Boston, MA, USA
sofya@bu.edu
http://cs-people.bu.edu/sofya/

C. Seshadhri [about the author]
Associate professor
Department of Computer Science
University of California, Santa Cruz
Santa Cruz, CA, USA
sesh@ucsc.edu
https://users.soe.ucsc.edu/~sesh/

ABOUT THE AUTHORS

ROKSANA BALESHZAR was a graduate student at Penn State while this work was done. She received her M. S. in Computer Science from Penn State under the supervision of Sofya Raskhodnikova, and her Bachelor's degree in Information Technology Engineering from Sharif University of Technology. She currently works as a Software Engineer at Google, Inc.

DEEPARNAB CHAKRABARTY is a faculty member of the Department of Computer Science at Dartmouth. Prior to this, he spent five lovely years at Microsoft Research in Bangalore. Deeparnab received his Ph. D. from the Georgia Institute of Technology under the supervision of Vijay Vazirani. He had postdoctoral stints at the University of Waterloo and the University of Pennsylvania. He gets excited by the application of optimization techniques in the design of algorithms, and is constantly surprised by the breadth of their applicability. He loves teaching undergraduate algorithms at Dartmouth, and hopes that the class sizes remain small.

RAMESH KRISHNAN S. PALLAVOOR recently graduated with a Ph. D. from the Department of Computer Science at Boston University where he was advised by Sofya Raskhodnikova. Previously, he was a Ph. D. student at Penn State, advised by Sofya Raskhodnikova. Before that, he received his B. Tech. in Computer Engineering from the Indian Institute of Information Technology, Design and Manufacturing (IIITD&M), Kancheepuram. His research interests include sublinear algorithms (in particular, property testing), and differential privacy.

He grew up in Chennai, a coastal city of 7 million in the southern part of India. His interest in math was partly spurred by his obsession with cricket statistics. During the initial part of his undergraduate studies, he was not too interested in the theory side of CS, and he thanks his teacher and mentor, Professor N. Sadagopan, for changing his outlook.

SOFYA RASKHODNIKOVA is a professor of Computer Science at Boston University. Previously, she was a Professor of Computer Science and Engineering at Penn State. She received her Ph. D. from MIT. Prior to joining Penn State, she was a postdoctoral fellow at the Hebrew University of Jerusalem and the Weizmann Institute of Science. She has held visiting positions at the Institute for Pure and Applied Mathematics at UCLA, Boston University, Harvard University and at the Simons Institute for the Theory of Computing at Berkeley. Sofya works in the areas of randomized and approximation algorithms. Her main interest is the design and analysis of sublinear-time algorithms for combinatorial problems. She has also made important contributions to data privacy. As far as her hobbies go, recall that she works on privacy.

C. SESHADHRI (Sesh) is a faculty member of the Department of Computer Science and Engineering at the University of California, Santa Cruz. He received his Ph. D. from Princeton University under the supervision of Bernard Chazelle and did a postdoc at IBM Almaden. Prior to joining UCSC, he spent four years as a staff member at Sandia National Laboratories, Livermore. Sesh's research interests are periodically sampled from a distribution that includes sublinear algorithms, social network analysis, and mathematical foundations for massive data analysis.

Sesh got his B. Tech from the Computer Science and Engineering Department at the Indian Institute of Technology, Kanpur. He still remembers his first data structures course with Prof. Manindra Agrawal; in that course, he fell in love with TCS.

Sesh used to have hobbies and was an all around interesting person, but that was before his kids were born.