Lower Bounds for Testing Properties of Functions over Hypergrid Domains

Eric Blais
MIT
Cambridge, MA
Email: eblais@csail.mit.edu

Sofya Raskhodnikova
Pennsylvania State University and Boston University
State College, PA and Boston, MA
Email: sofya@cse.psu.edu

Grigory Yaroslavtsev
Brown University, ICERM
Providence, RI
Email: grigory@grigory.us

***Abstract*—We show how the communication complexity method introduced in (Blais, Brody, Matulef 2012) can be used to prove lower bounds on the number of queries required to test properties of functions with non-hypercube domains. We use this method to prove strong, and in many cases optimal, lower bounds on the query complexity of testing fundamental properties of functions $f\colon \{1,\dots,n\}^d \to \mathbb{R}$ over hypergrid domains: monotonicity, the Lipschitz property, separate convexity, convexity and monotonicity of higher-order derivatives. There is a long line of work on upper bounds and lower bounds for many of these properties that uses a diverse set of combinatorial techniques. Our method provides a unified treatment of lower bounds for all these properties based on Fourier analysis.**

A key ingredient in our new lower bounds is a set of Walsh functions, a canonical Fourier basis for the set of functions on the line $\{1,\dots,n\}$. The orthogonality of the Walsh functions lets us use a product construction to extend our method from properties of functions over the line to properties of functions over hypergrids. Our product construction applies to properties over hypergrids that can be expressed in terms of axis-parallel directional derivatives, such as monotonicity, the Lipschitz property and separate convexity. We illustrate the robustness of our method by making it work for convexity, which is the property of the Hessian matrix of second derivatives being positive semidefinite and thus cannot be described by axis-parallel directional derivatives alone. Such robustness contrasts with the state of the art in the upper bounds for testing properties over hypergrids: methods that work for other properties are not applicable for testing convexity, for which no nontrivial upper bounds are known for $d \geq 2$.

***Keywords*-Property testing; monotonicity; functions on hypergrids; communication complexity; Walsh functions**

I. Introduction

Property testing examines the following general question: given a property $\mathcal{P}$ of functions mapping one set D to another set R, how many queries does a randomized algorithm with oracle access to some unknown function $f : D \to R$ need to distinguish functions with the property $\mathcal{P}$ from those that are "far" from having this property? (See Section II for formal definitions.) Over the last two decades, many powerful tools have been developed for designing efficient algorithms for testing various properties (see, e.g., [19], [32], [33] for recent surveys). In contrast, few tools are known for establishing the limitations of these algorithms.

One such tool is the *communication complexity method* recently introduced by Blais, Brody, and Matulef [6]. This method yields new lower bounds on the query complexity of property testing problems from known lower bounds in communication complexity. It has been remarkably successful in establishing strong lower bounds on the query complexity for testing many properties of functions mapping the hypercube $\{0,1\}^d$ to some (finite or infinite) set R. The best previously known lower bounds for testing monotonicity [6], [8], k-linearity [6], low Fourier degree [6], [24], the Lipschitz property [25], and function linear isomorphism [23] have all been established using this method.

Yet, despite the success in establishing lower bounds for properties of functions on the hypercube, so far the communication complexity method has not yielded property testing lower bounds in any other setting. The state of affairs is not due to any inherent limitation of the method itself. Rather, it is due to the specialized nature of the constructions developed so far in applications of the method. Roughly speaking, most existing constructions rely on the fact that they can treat the d dimensions of the Boolean hypercube $\{0,1\}^d$ "independently" to obtain the desired lower bounds. In particular, many of these constructions use the parity functions, an orthonormal basis for functions on the hypercube, as a basic building block. To obtain lower bounds for properties of functions over other domains, new construction techniques and new building blocks are required.

We give the first applications of the communication complexity method to the setting of testing properties of functions over non-hypercube domains. Specifically, we focus our attention on functions over the line $[n] := \{1,2,\dots,n\}$ and the hypergrid $[n]^d$. An extensive research effort has been devoted to the study of testing fundamental properties of functions over these domains, with particular emphasis on testing monotonicity [15], [16], [17], [1], [5], [12], the Lipschitz property [25], [3], [12], and convexity [29], [31], [30]. (Subsequent to the publication of the preprint of this article [7], several more works on testing functions over hypergrid domain appeared [13], [9], [4], [10].) Yet, prior to this work, large gaps remained between the best upper and lower bounds on the query complexity of these property testing problems. We establish strong, and in many cases optimal, lower bounds for testing all of these properties. See Table I for a summary of our lower bounds.

The basic building block used in our constructions is the set of *Walsh functions*, which form a canonical Fourier basis for the set of functions over the line and the hypergrid. The choice of an orthonormal Fourier basis is crucial because it allows us to express the rich families of functions used in our reductions concisely, i.e., using a small number of bits, which is necessary for the application of the communication complexity framework. Moreover, it often allows us to lift our constructions from the line to the high-dimensional hypergrids using a generic product rule without losing optimality of the results (see the first part of Table I). Finally, the expressive power of the Fourier basis allows us to obtain lower bounds for properties for which no good upper bounds are known (specifically, convexity, separate convexity and monotonicity of high-order derivatives).

We also streamline the formulation of the communication complexity method, which results in simpler proofs. After the publication of a preprint of this article [7], Goldreich [20] generalized the streamlined formulation of the communication complexity method and gave a thorough comparison with the original formulation.

A. Our results

We give lower bounds for several properties of functions on the hypergrid. For each of these properties, we first construct a lower bound for one-dimensional functions. Many properties we consider can be expressed as conditions of the axis-parallel derivatives of the function. For these properties, the orthogonality of Walsh functions enables us to extend the lower bounds to the hypergrid setting with a natural product construction.

1) Monotonicity: The function $f : [n]^d \to R$ is *monotone* if $f(x) \leq f(y)$ for every pair of inputs $x = (x_1, \ldots, x_n), y = (y_1, \ldots, y_n) \in [n]^d$ that satisfy $x_1 \leq y_1, \ldots, x_n \leq y_n$. Monotonicity testing is a classic problem in property testing that has been studied extensively for functions on the line [16], [17], on the hypercube [21], [15], [18], [6], [12], [11], on general partially ordered domains [18], and on hypergrid domains [15], [1], [12]. The best upper bound for testing monotonicity on the hypergrid is due to Chakrabarty and Seshadhri [12], who recently showed that $O(d \log n)$ queries suffice to test whether $f : [n]^d \to R$ is monotone, for any range $R \subseteq \mathbb{R}$.

Prior to this work, however, there were *no* general lower bounds for the problem of testing monotonicity of functions on the hypergrid. We give the first lower bound for this problem. Furthermore, the bound that we obtain is optimal for nonadaptive tests,[1] since it matches the upper bound of Chakrabarty and Seshadhri [12].

Theorem I.1. *Fix $\epsilon \in (0, \frac{1}{8}]$ and $m, r \in \mathbb{N}$. Let $n = 2^m$. Any nonadaptive ϵ-test for monotonicity of functions $f : [n]^d \to [nd]$ makes $\Omega(d \log n)$ queries.*

The special case of the theorem with $d = 1$ also gives a new lower bound for the classic problem of testing monotonicity of functions on the line. Theorem III.6 gives a more nuanced lower bound for this special case, claimed in Table I. Ergun et al. [16] showed that $\Theta(\log n)$ queries are necessary and sufficient for testing monotonicity of $f : [n] \to \mathbb{R}$ nonadaptively with one-sided error, and Fischer [17] showed that the lower bound also holds for adaptive testers with two-sided error. But Fischer's proof relies on Ramsey theory arguments that only hold when the range of f is extremely large (i.e., at least exponential in n). Theorem III.6 gives the first lower bound for two-sided error monotononicity testers of functions with smaller ranges.

2) Convexity: The function $f : [n]^d \to R$ is *convex* if for all $x, y \in [n]^d$ and all $\rho \in [0, 1]$ such that $\rho x + (1 - \rho)y \in [n]^d$, the function f satisfies $f(\rho x + (1 - \rho)y) \leq \rho f(x) + (1 - \rho)f(y)$. Parnas, Ron, and Rubinfeld [29] showed that we can test if $f : [n] \to \mathbb{R}$ is convex with $O(\log n)$ queries. They also stated the open problem of testing convexity of functions on the hypergrid. Our next lower bound represents the first progress on this ten-year-old problem.

Theorem I.2. *Fix $\epsilon \in (0, \frac{1}{16}]$ and $m, r \in \mathbb{N}$. Let $n = 2^m$. Any nonadaptive ϵ-test for convexity of functions $f : [n]^d \to \mathbb{R}$ makes $\Omega(d \log n)$ queries.*

Notably, the special case of the theorem where $d = 1$ gives the first lower bound for testing convexity on the line. This lower bound is optimal because it matches the query complexity of the nonadaptive tester in [29].

Convexity, unlike the other properties we consider in this paper, cannot be expressed in terms of conditions on axis-parallel derivatives—it is a property of the Hessian matrix of all partial derivatives of a function being positive semidefinite. As a result, our lower bound construction for convexity on the hypergrid is more technically involved.

In contrast, a closely related property, *separate convexity*, can be expressed in terms of conditions on axis-parallel derivatives. The function $f : [n]^d \to R$ is *separately convex* if for every $i \in [d]$ and $x \in [n]^d$, the function $g : [n] \to R$ defined by $g(y) = f(x_1, \ldots, x_{i-1}, y, x_{i+1}, \ldots, x_d)$ is convex. Separate convexity is a strictly weaker condition than convexity (namely, all convex functions are also separately convex, but the converse statement is false—consider, for example, $f(x, y) = xy$). Separate convexity has been studied in many settings, including convex analysis [35], probability theory [2], and computational geometry [26], [27]. We give the first lower bound for the query complexity of testing separate convexity.

[1] A property tester is nonadaptive if its choice of queries does not depend on the answers to the previous queries. See Definition II.2.

Table I
QUERY COMPLEXITY BOUNDS FOR TESTING PROPERTIES OF THE FUNCTION $f : [n]^d \to \mathbb{Z}$ (TOP) AND OF THE FUNCTION $f : [n] \to [r]$ (BOTTOM). ALL THE BOUNDS ARE FOR NONADAPTIVE TESTS WITH TWO-SIDED ERROR UNLESS MARKED OTHERWISE.

Functions on the hypergrid

	Our lower bounds	Previous lower bounds		Upper bounds	
Monotonicity	$\Omega(d \log n)$	$\Omega(d)$ (adaptive, $n = 2$)	[6]	$O(d \log n)$	[12]
Convexity	$\Omega(d \log n)$	—		—	
Separate convexity	$\Omega(d \log n)$	—		—	
Lipschitz	$\Omega(d \log n)$	$\Omega(d)$ (adaptive, $n = 2$)	[25]	$O(d \log n)$	[12]

Functions on the line

	Our lower bounds	Previous lower bounds		Upper bounds	
Monotonicity	$\Omega(\min\{\log r, \log n\})$	$\Omega(\min\{\log r, \log n\})$ (1.-s. err.) $\Omega(\log n)$ (adaptive, $r \gg n$)	[16] [17]	$O(\log n)$	[16]
Convexity	$\Omega(\log n)$ ($r = \Omega(n^2)$)	—		$O(\log n)$	[29]
Lipschitz	$\Omega(\min\{\log r, \log n\})$	$\Omega(\min\{\log r, \log n\})$ (1-s. err.)	[25]	$O(\log n)$	[25]
Monotone ℓ-th derivative	$\Omega(\log n)$ ($r = \Omega(n^{\ell+1})$)	—		—	

Theorem I.3. *Fix $\epsilon \in (0, \frac{1}{16}]$ and $m, r \in \mathbb{N}$. Let $n = 2^m$. Any nonadaptive ϵ-test for separate convexity of functions $f : [n]^d \to [r]$, where $r = \Omega(dn^2)$, makes $\Omega(d \log n)$ queries.*

3) Lipschitz property: The function $f : [n]^d \to R$ is *Lipschitz* if $|f(x_1, \ldots, x_n) - f(y_1, \ldots, y_n)| \leq \sum_{i=1}^n |x_i - y_i|$ for every $(x_1, \ldots, x_n), (y_1, \ldots, y_n) \in [n]^d$. Lipschitz functions play a fundamental role in many areas of mathematics and computer science. Of particular interest to our present study, the problem of testing whether a function $f : [n]^d \to \mathbb{R}$ is Lipschitz was recently found to have important applications to data privacy and program checking [25], [14]. These applications motivated a flurry of research on the topic [25], [3], [12], [14], [14]. A highlight of this line of work is is Chakrabarty and Seshadhri's nonadaptive tester which needs $O(d \log n)$ queries to test whether $f : [n]^d \to \mathbb{R}$ is Lipschitz [12]. We establish the first lower bound on the query complexity of this problem. Our bound is optimal because it matches the upper bound in [12].

Theorem I.4. *Fix $\epsilon \in (0, \frac{1}{8}]$ and $m, r \in \mathbb{N}$. Let $n = 2^m$. Any nonadaptive ϵ-test for the Lipschitz property of functions $f : [n]^d \to [r]$, where $r = \Omega(dn)$, makes $\Omega(d \log n)$ queries.*

The special case of Theorem I.4 when $d = 1$ is also new. Jha and Raskhodnikova [25] showed that a nonadaptive one-sided error algorithm requires $\Omega(\min\{\log n, \log r\})$ queries to test if $f : [n]^d \to [r]$ is Lipschitz. Theorem III.12, a more nuanced version of Theorem I.4 for $d = 1$, shows that the same lower bound also holds for testers with two-sided error.

4) Generalizations: Our techniques are extendable to other properties as well. In the full version of this article, we illustrate this on two classes of properties of functions on the line: (α, β)-Lipschitz properties and the properties of nonnegativity of higher order discrete derivatives.

For any parameters $-\infty \leq \alpha \leq \beta \leq \infty$, a function $f : [n] \to \mathbb{R}$ is (α, β)*-Lipschitz* if $\alpha \leq f(x+1) - f(x) \leq \beta$ for every $x \in [n-1]$. The class of (α, β)-Lipschitz properties, introduced by Chakrabarty and Seshadhri [12], includes monotonicity and the Lipschitz property as special cases. Our lower bound constructions for these two properties can be generalized to to all (α, β)-Lipschitz properties.

As we discuss in Section III-B, convexity of a function $f : [n] \to \mathbb{R}$ is equivalent to the nonnegativity of its discrete derivative f' defined by $f'(x) = f(x+1) - f(x)$. In the full version of the article, we extend the lower bound construction for testing convexity to give a unified lower bound for testing the nonnegativity of any higher discrete derivative of a given function. This is in stark contrast to the situation with the upper bounds, where significantly different algorithms and analyses are used to test monotonicity [16] (nonnegativity of the first derivative) and convexity [29] (nonnegativity of the second derivative), and *no* algorithm is known for testing nonnegativity of higher derivatives.

B. Discussion and open problems

All lower bounds presented in this paper are for nonadaptive tests. Interestingly, *all* the best known upper bounds on the query complexity of testing monotonicity, convexity, or the Lipschitz property (for functions over any domain) are achievable with nonadaptive tests, with one exception: the new adaptive bound for testing Boolean functions on constant-dimensional hypergrids from [4].

Subsequent to the publication of a preprint of this article [7], Chakrabarty and Seshadhri [13] and later Dixit et al. [9] gave lower bounds of $\Omega(d \log n)$ queries for testing (adaptively or not) whether the function $f : [n]^d \to \mathbb{R}$ is monotone and, respectively, Lipschitz. These results follow from an extension of the Ramsey theory argument of Fischer [17]. Like Fischer's lower bound, their method only applies to functions with very large ranges. These results leave two open problems that we find particularly intriguing. Can the adaptive lower bounds also be established for functions with small ranges? Can they be obtained via the communication complexity method?

Organization: The basic definitions and facts for property testing and communication complexity are introduced in Section II. In Section III, we prove our lower bounds for functions on the line. The more general lower bounds for functions with hypergrid domains are presented in Section IV.

II. Preliminaries

A. Property testing

This section is devoted to basic property testing definitions. For a more thorough introduction to the area, we recommend [32], [33].

Definition II.1 (Distance)**.** The *distance* between two functions $f, g : D \to R$ is the fraction of points x in D for which $f(x) \neq g(x)$. The distance between f and a property $\mathcal{P}$ of functions mapping D to R is the minimal distance between f and any $g \in \mathcal{P}$. We say f is ϵ*-far* from $\mathcal{P}$ if its distance to $\mathcal{P}$ is at least ϵ.

Definition II.2 (Property tester [34], [22])**.** Fix $\epsilon \in (0, 1)$. An ϵ-tester for a property $\mathcal{P}$ is a randomized algorithm which, given oracle access to a function f, accepts with probability at least 2/3 if $f \in \mathcal{P}$, and rejects with probability at least 2/3 if f is ϵ-far from $\mathcal{P}$.

A tester has *one-sided error* if it always accepts functions in $\mathcal{P}$ and has *two-sided error* otherwise. It is *nonadaptive* if the queries to f do not depend on the answers to the previous queries; otherwise, it is *adaptive*.

B. Communication complexity

In a (two-player) *communication game* C, Alice receives some input a, Bob receives some input b, and they must compute the value of some function $f_C(a, b)$ on their joint input. A *protocol* defines how Alice and Bob communicate. The maximum number of bits exchanged by Alice and Bob during the execution of a protocol over the possible inputs a and b is the *complexity* of the protocol. A randomized protocol is valid for f_C if for every input, the protocol computes f_C correctly with probability at least $2/3$. The *communication complexity* of f_C is the minimum complexity of any protocol that is valid for f_C.

A number of different communication models have been extensively studied. We focus on the *one-way shared randomness* model. In this model, communication is allowed only from Alice to Bob. Alice and Bob share access to a common source of randomness that can be used to determine the protocol. The communication complexity of f_C in the one-way shared randomness model is denoted $R^{A\to B}(f_C)$.

A fundamental function f_C studied in the one-way shared randomness model is $\textsc{AugmentedIndex}_t$, where $t \geq 1$ is a parameter specifying the instance size. Alice's input to this function is a set $A \subseteq [t]$ while Bob's input is an index $i \in [t]$ and the set $B = A \cap [i-1]$. The output of $\textsc{AugmentedIndex}_t$ is 1 if $i \in A$ and 0 otherwise. No randomized one-way communication protocol for this function does significantly better than the naïve protocol where Alice communicates her whole set to Bob.

Theorem II.3 ([28])**.** *The one-way communication complexity of* $\textsc{AugmentedIndex}_t$ *in the shared randomness model is* $R^{A\to B}(\textsc{AugmentedIndex}_t) = \Theta(t)$.

C. Communication complexity method

A *combining operator* ψ takes as input a and b, the inputs of Alice and Bob for a given communication game C, and returns a function $\psi[a, b]$. It is a *one-way one-bit combining operator* if for every a and b, and every element x in the domain of $\psi[a, b]$, Bob can compute the value of $\psi[a, b](x)$ with only one bit of communication from Alice. A combining operator is also called a *reduction operator* if it satisfies the conditions we require to complete a reduction from C to a property testing problem:

Definition II.4 (Reduction operator)**.** A one-bit one-way combining operator ψ is a *reduction operator* for the communication game C, the property $\mathcal{P}$, and the parameter $\epsilon_0 \in (0, 1)$ if for all possible inputs a and b of Alice and Bob, respectively,

1) if $f_C(a, b) = 0$, then $\psi[a, b] \in \mathcal{P}$, and
2) if $f_C(a, b) = 1$, then $\psi[a, b]$ is ϵ_0-far from $\mathcal{P}$.

The following lemma is the main tool in our lower bound constructions. The proof of this lemma is implicit in [6]. For completeness, we include it below.

Lemma II.5 (Reduction lemma)**.** *If there exists a reduction operator for the communication game C, the property $\mathcal{P}$ and the parameter $\epsilon_0 \in (0, 1)$, then for all $\epsilon \in (0, \epsilon_0]$, every nonadaptive ϵ-tester for $\mathcal{P}$ makes $R^{A\to B}(C)$ queries.*

Proof: Let ψ be a reduction operator for C, $\mathcal{P}$, and ϵ_0. Consider a nonadaptive ϵ-tester T for $\mathcal{P}$ that makes at most q queries for some $\epsilon \in (0, \epsilon_0]$. Let Alice and Bob use their shared randomness to both simulate the tester T and identify the inputs $x^{(1)}, \ldots, x^{(q)}$ queried by T. The tester T is nonadaptive, so they can both identify the queried inputs without observing the value of $\psi[a, b]$ on any of these inputs. Since ψ is a one-way one-bit combining operator, Alice only needs to send q bits of information to enable Bob to compute $\psi[a, b](x^{(1)}), \ldots, \psi[a, b](x^{(q)})$. Bob completes the execution of T then outputs 0 if T accepts or 1 if T rejects. The correctness of this protocol is guaranteed by conditions 1 and 2 of Definition II.4. ■

The definition of the reduction operator and the reduction lemma can be generalized to handle two-way bounded-bit combining operators. Goldreich [20] introduces this generalized formulation and provides a thorough comparison with

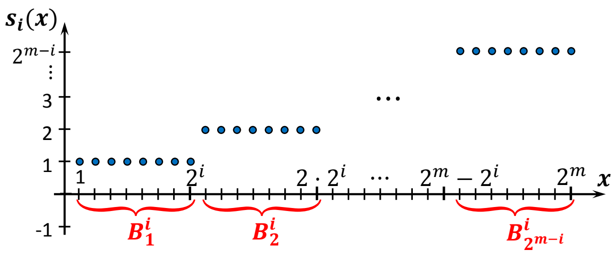

Figure 1. Blocks B_k^i and step functions s_i: an illustration of Definitions III.1 and III.2.

the original formulation of the communication complexity method. All our reductions use one-way one-bit combining operators, and in fact they are all obtained from the AUGMENTED INDEX communication game. We write $\psi[A, i, B]$ (instead of $\psi[A, (i, B)]$) to denote the functions obtained by the reduction operator ψ for this game. The following corollary follows directly from the reduction lemma (Lemma II.5) and Theorem II.3.

Corollary II.6 (Reduction corrolary)**.** *If there exists a reduction operator for* AUGMENTEDINDEX$_t$*, the property $\mathcal{P}$ and the parameter $\epsilon_0 \in (0, 1)$, then for all $\epsilon \in (0, \epsilon_0]$, every nonadaptive ϵ-tester for $\mathcal{P}$ makes $\Omega(t)$ queries.*

III. LOWER BOUNDS ON THE LINE

In this section, we consider properties of functions mapping the domain $[2^m] = \{1, \ldots, 2^m\}$ (where $m \in \mathbb{N}$) to a range $R \subseteq \mathbb{R}$. Two classes of functions play a central role in the study of these properties: step functions and Walsh functions. The functions in both of these classes are constant on *blocks* of inputs in $[2^m]$, which we define next.

Definition III.1 (Blocks)**.** Let $i \in \{0, \ldots, m\}$. For $k \in [2^{m-i}]$, the kth block of length 2^i is the set of integers $\{2^i(k-1)+1, \ldots, 2^i k\}$. We denote this block B_k^i.

Definition III.2 (Step functions)**.** For $i \in \{0, \ldots, m\}$, the *step function* of block length 2^i is the function $s_i : [2^m] \to [2^{m-i}]$ defined by $s_i(x) = k$, such that $x \in B_k^i$. (Equivalently, $s_i(x) = \lfloor \frac{x-1}{2^i} \rfloor + 1$.)

The definitions of blocks and step functions are illustrated in Figure 1. Note that blocks of length 2^i partition $[2^m]$ and that the step functions of block length 2^i are constant on each block B_k^i.

The Walsh functions can be defined in terms of blocks. Specifically, the Walsh function indexed by i is equal to 1 on the first half of each block B_k^i and to -1 on the second half. In other words, the value of the ith Walsh function on input x is determined by the ith bit of the binary representation of $x-1$. We denote this value by $\mathsf{bit_i}(x-1)$, where the bits are numbered starting from the least significant.

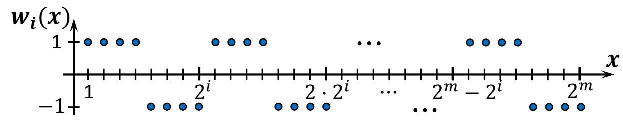

Figure 2. Walsh functions w_i: an illustration of Definition III.3.

Definition III.3 (Walsh functions)**.** For $i \in [m]$, the function $w_i : [2^m] \to \{-1, 1\}$ is defined by $w_i(x) = (-1)^{\mathsf{bit_i}(x-1)}$. For any $S \subseteq [m]$, the *Walsh function* $w_S : [2^m] \to \{-1, 1\}$ corresponding to S is $w_S(x) = \prod_{i \in S} w_i(x)$. (If $S = \emptyset$ then $w_S(x) = 1$ for all x.) Lastly, we define $w_{m+1}(x) = 1$.

The Walsh functions are illustrated in Figures 2 and 3. We use two basic properties of Walsh functions in this section.

Proposition III.4. *For every $S \subseteq [m]$, the Walsh function w_S satisfies $\sum_{x \in [2^m]} w_S(x) \geq 0$.*

For two functions $f, g : [n] \to \mathbb{R}$, we write $f \cdot g$ to denote the pointwise product of the two functions: for every $x \in [n]$, $f \cdot g(x) = f(x)g(x)$.

Proposition III.5. *For every $A, B \subseteq [m]$, the Walsh function $w_{A \triangle B} : [2^m] \to \{-1, 1\}$ corresponding to the symmetric difference between A and B satisfies $w_{A \triangle B} = w_A \cdot w_B$.*

A. Monotonicity

In this section, we establish the following lower bound for testing monotonicity of functions on the line.

Theorem III.6. *Fix $\epsilon \in (0, \frac{1}{4}]$ and $m, r \in \mathbb{N}$. Let $n = 2^m$. Any nonadaptive ϵ-tester for monotonicity of functions $f : [n] \to [r]$ makes $\Omega(\min(\log n, \log r))$ queries.*

A central component of the proof of Theorem III.6 is the following observation regarding combinations of step functions and Walsh functions.

Lemma III.7. *Fix $i \in [m]$ and $S \subseteq \{i, \ldots, m\}$. Define $h = 2s_i + w_S$ and $h_- = 2s_i - w_S$.*

1) *If $i \notin S$, then h and h_- are monotone;*
2) *If $i \in S$, then h is $\frac{1}{4}$-far from monotone.*

Proof: When $i \notin S$, then $S \subseteq \{i+1, \ldots, m\}$ and the functions s_i, w_S and $-w_S$ are constant on each block B_k^i (for $k \in [2^{m-i}]$). This means that the value of the functions w_S and $-w_S$ can decrease (from 1 to -1) only between adjacent blocks (i.e., the inequality $w_S(x) > w_S(x+1)$ can only hold when $x \in B_k^i$ and $x+1 \in B_{k+1}^i$ for some $k \in [2^{m-i} - 1]$). But the step function s_i increases by 1 between adjacent blocks, so h and h_- are monotone.

When $i \in S$, then the Walsh function w_S changes value in the middle of each block B_k^i. If this change is from 1 to -1, then w_S is 1/2-far from monotone on this block, and so is h because the step function s_i is constant on each

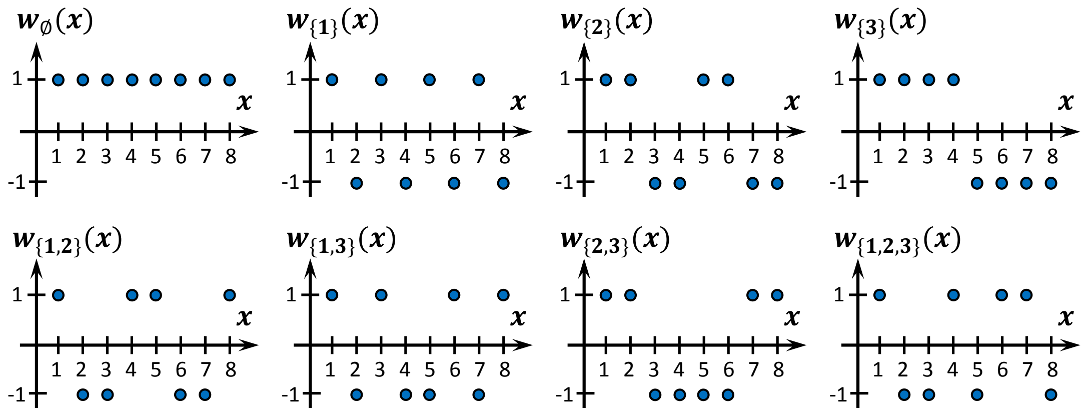

Figure 3. Walsh functions w_S for $m = 3$ and all subsets S of $[3]$: an illustration of Definition III.3.

B_k^i. Note that this change is from 1 to -1 for all blocks on which $w_{S\setminus\{i\}}$ evaluates to 1. By Proposition III.4, this is the case for at least half of the blocks. Thus, h is $\frac{1}{4}$-far from monotone. ■

Proof of Theorem III.6: To prove the lower bound of $\Omega(\log n)$ queries (for $n < r$), we use the reduction corollary (Corollary II.6) with the parameter t in the corollary set to m. To get the bound of $\Omega(\log r)$ queries (for $r \le n$), we use the same proof with t set to $\lfloor \log_2(r-1) \rfloor$ and with the additional restriction that the sets given to Alice and Bob reside in $\{m-t+1, \ldots, m\}$ instead of $[m]$.

Let ψ be the combining operator that receives Alice's set A, Bob's index i and set B as input and returns the function $h : [2^m] \to \mathbb{Z}$ defined by

$$h(x) = 2s_i(x) + w_{A\triangle B}(x). \tag{1}$$

Note that $A\triangle B = A \cap \{i, \ldots, m\}$ and that the range of h is $[2 \cdot 2^{t-1} + 1] = [2^t + 1]$. That is, the range is $[n+1]$ when $t = m$ and is $[r]$ when $t = \lfloor \log_2(r-1) \rfloor$.

By Proposition III.5, $w_{A\triangle B} = w_A \cdot w_B$. Bob knows B, so to determine $h(x)$ he only needs Alice to communicate a single bit—namely, the value of $w_A(x)$. Thus, ψ is a one-bit one-way combining operator. Furthermore, by Lemma III.7 the function h is monotone when $i \notin A$ and it is $\frac{1}{4}$-far from monotone when $i \in A$, so ψ is a reduction operator for monotonicity of functions of the form $f : [2^m] \to [t+1]$ and $\epsilon_0 = 1/4$. Then, by Corollary II.6, for any $\epsilon < \frac{1}{4}$, every nonadaptive ϵ-tester for monotonicity requires $\Omega(t) = \Omega(\min(\log n, \log r))$ queries. ■

B. Convexity

The main result of this section is the following lower bound on the query complexity for testing the convexity of functions on the line.

Theorem III.8. *Fix $\epsilon \in (0, \frac{1}{8}]$ and $n = 2^m$ for some $m \ge 1$. Any nonadaptive ϵ-test for convexity of functions $f : [n] \to [r]$, where $r = \Omega(n^2)$, makes $\Omega(\log n)$ queries.*

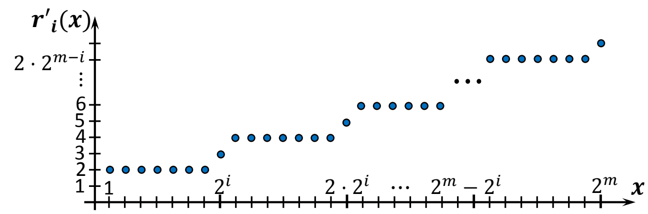

Figure 4. Double-step functions r'_i: an illustration of Definition III.10.

Recall that the function $f : [n] \to \mathbb{R}$ is convex if for all $x, y \in [n]$ and all $\rho \in [0,1]$ such that $\rho x + (1-\rho)y$ is also an integer in $[n]$, the function f satisfies $f(\rho x + (1-\rho)y) \le \rho f(x) + (1-\rho)f(y)$. Equivalently, we can define convexity in terms of the discrete derivative of functions on the line.

Definition III.9 (Discrete derivative, convexity)**.** The *discrete derivative* of $f \colon [n] \to \mathbb{R}$ is the function $f' \colon [n-1] \to \mathbb{R}$ defined by $f'(x) = f(x+1) - f(x)$. The function $f : [n] \to \mathbb{R}$ is *convex* (resp., *concave*) if its derivative f' is a monotone nondecreasing (resp., nonincreasing) function.

The proof of Theorem III.8 uses two variants of the step functions: rising-step-size functions and double-step functions.

Definition III.10 (Rising-step-size and double-step functions)**.** Fix $i \in [m]$. The *rising-step-size function* $r_i : [n] \to [n^2]$ is defined by $r_i(x) = s_i(x) + 2\sum_{y=1}^{x-1} s_i(y)$. Its discrete derivative, $r'_i(x) = s_i(x+1) + s_i(x)$, is called a *double-step function*. Equivalently, for every $k \in [2^{m-i}]$ the function $r'_i(x)$ is equal to $2k$ on all but the last element x of the block B_k^i and to $2k+1$ on the last element of B_k^i.

Lemma III.11. *Fix $i \in [m]$ and $S \subseteq \{i, \ldots, m\}$. Define $h = r_i + \frac{1}{2}(w_S + 1)$ and $h_- = r_i - \frac{1}{2}(w_S + 1)$.*

1) *If $i \notin S$, then h and h_- are both convex.*
2) *If $i \in S$, then h is $\frac{1}{8}$-far from convex.*

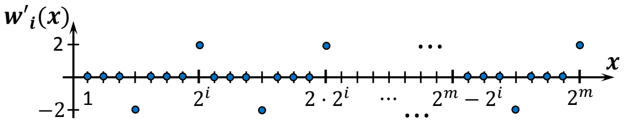

Figure 5. Derivative of singleton Walsh functions w_i'. Illustration for the proof of Lemma III.11

Proof: First, consider the case where $i \notin S$. The discrete derivative of h is $h'(x) = r_i'(x) + \frac{1}{2}w_S'(x)$. It is sufficient to prove that h' is nondecreasing. Since $S \subseteq \{i+1, \ldots, m\}$, the function w_S is constant on each block B_k^i (for $k \in [2^{m-i}]$). That is, for all but the last element x of a block B_k^i, the discrete derivative $w'(x) = 0$ and, consequently, $h'(x) = r_i'(x) = 2k$. Now consider $h'(x)$, where x is the last element of a block B_k^i. Recall that $r_i'(x) = 2k+1$. Since Walsh functions are ± 1-valued, the value $\frac{1}{2}w_S'(x)$ is in $\{-1, 0, 1\}$ (see Fig. 5 for an illustration of a derivative of a singleton Walsh function w_i'). Thus, $h'(x) \in [2k, 2k+2]$, i.e., $h'(x-1) \le h'(x) \le h'(x+1)$. Therefore, h' is a nondecreasing function. The same argument shows that when $i \notin S$, the function h_- is also convex.

Now consider the case where $i \in S$. We start the analysis of this case by showing that for at least half of the blocks B_k^i, the derivative $w_S'(x) = -2$ on the 2^{i-1}th element of B_k^i (i.e., on the input $x = 2^i(k-1) + 2^{i-1}$.) Note that $w_S = w_i \cdot w_{S \setminus \{i\}}$. By Proposition III.4, $w_{S\setminus\{i\}}(x) = 1$ for at least half of the inputs $x \in [2^m]$. Since $S \cap [i-1] = \emptyset$, the function $w_{S\setminus\{i\}}$ is constant within the blocks B_k^i. Thus, for at least half of these blocks it is a constant 1. For each block B_k^i, the function w_i is 1 on the first half of the block and -1 on the second half. Combining these observations, for half of the blocks B_k^i, the derivative of w_S on the middle point $x = 2^i(k-1) + 2^{i-1}$ of the block satisfies $w_S'(x) = w_S(x+1) - w_S(x) = w_{S\setminus\{i\}}(x+1) \cdot w_i(x+1) - w_{S\setminus\{i\}}(x) \cdot w_i(x) = -2$.

Let B_k^i be a block where $w_S'(x) = -2$ on the 2^{i-1}th element x of B_k^i. Note that $w_S'(x) = 0$ on all other inputs in the block apart from the last one because w_S is constant on all blocks B_j^{i-1}. Consider any three points $x, y, z \in B_k^i$ such that $x \le (k-1)2^i + 2^{i-1} < y < z$, namely, x is in the first half of the block B_k^i while y and z are in the second half. Then $h'(y) = h'(y+1) = \cdots = h'(z-1) = 2k$ so $(h(z) - h(y))/(z-y) = 2k$. However, $h'((k-1)2^i + 2^{i-1}) = 2k-2$ so $(h(y) - h(x))/(y-x) < 2k$, which violates convexity. To fix convexity on all such triples, we must change the value of h on all the points $(k-1)2^i + 1, \ldots, (k-1)2^i + 2^{i-1}$ in the first half of the block B_k^i, or on all but one point in the second half of B_k^i. Thus, we need to change at least $1/4$ of the points in B_k^i. Since this is the case for at least half of all blocks, h is $1/8$-far from convex. ■

Proof of Theorem III.8: We use the reduction corollary (Corollary II.6) with the parameter t in the corollary set to m. Given Alice's set $A \subseteq [m]$ and Bob's index $i \in [m]$ and the prefix set $B = A \cap [i-1]$, the combining operator $\psi[A, i, B]$ returns the function

$$h(x) = r_i(x) + \tfrac{1}{2}(w_{A \triangle B}(x) + 1).$$

Note that $A \triangle B = A \cap \{i, \ldots, m\}$. Since $w_{A\triangle B} = w_A \cdot w_B$, the operator ψ is a one-bit one-way combining operator. Furthermore, by Lemma III.11, if $i \notin A$ then h is convex and if $i \in A$ then h is $1/8$-far from convex. So ψ is a reduction operator for convexity with parameter $\epsilon_0 = \frac{1}{8}$ and the theorem follows from Corollary II.6. ■

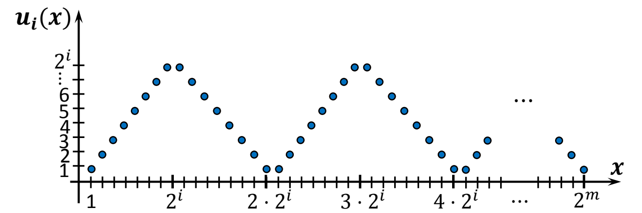

Figure 6. Up-down staircase functions u_i: an illustration of Definition III.13.

C. The Lipschitz property

Theorem III.12. *Fix $\epsilon \in (0, \frac{1}{4}]$ and $m, r \in \mathbb{N}$. Let $n = 2^m$. Any nonadaptive ϵ-test for the Lipschitz property of functions $f : [n] \to [r]$ makes $\Omega(\min(\log n, \log r))$ queries.*

The proof of Theorem III.12 uses yet another variant on the step functions: up-down staircase functions.

Definition III.13 (Up-down staircase functions)**.** For all $i \in \{0, 1, \ldots, m\}$, let the *up-down staircase function* of block-length 2^i be the function $u_i : [2^m] \to [2^i]$, such that $u_i(1) = 1$ and the discrete derivative of u_i is

$$u_i'(x) = \begin{cases} 0 & \text{if } x \text{ is divisible by } 2^i; \\ w_{i+1}(x) & \text{otherwise.} \end{cases}$$

Equivalently, the function u_i takes the values $1, \ldots, 2^i$ on consecutive inputs from the block B_j^i if j is odd, and the values $2^i, \ldots, 1$ if j is even. (See Figure 6.)

Lemma III.14. *Fix $i \in [m]$ and $S \subseteq \{i, \ldots, m\}$. Define $h(x) = u_i(x) - \frac{1}{2}(w_S(x) + 1)$ and $h_-(x) = u_i(x) - \frac{1}{2}(-w_S(x) + 1)$.*

1) *If $i \notin S$, then h and h_- are both Lipschitz.*
2) *If $i \in S$, then h is $\frac{1}{4}$-far from Lipschitz.*

Proof: If $i \notin S$, i.e., $S \subseteq \{i+1, \ldots, m\}$, then the function w_S is constant on each block B_k^i (for $k \in [2^{m-i}]$). Let $w(x) = -\frac{1}{2}(w_S(x) + 1)$. Since Walsh functions are ± 1-valued, the discrete derivative $w'(x)$ is in $\{-1, 0, 1\}$ for all x, and $w'(x) = 0$ for all x *not* divisible by 2^i. By definition of the up-down staircase functions, $u_i'(x) \in \{-1, 0, 1\}$ for all x, and $u_i'(x) = 0$ for all x divisible by 2^i. Thus, $h' = u_i' + w'$ takes values only in $\{-1, 0, 1\}$, implying that h is Lipschitz. The proof that h_- is Lipschitz is analogous.

When $i \in S$, i.e., i is the smallest element in S, the rescaled Walsh function $w(x) = -\frac{1}{2}(w_S(x) + 1)$ changes value in the middle of each block B_k^i. This change is either from -1 to 0 or vice versa. In the former case, the discrete derivative w' is 1 on the 2^{i-1}th element of the block, in the latter, it is -1. In both cases, it is 0 on all other elements of the block besides the last one. Next we show that if the former case occurs on a block with odd i (similarly, if the latter case occurs on a block with even i), then h is $1/2$-far from Lipschitz on this block.

Consider the case when i is odd and w' is 1 on the 2^{i-1}th element of a block B_k^i. Since i is odd, u_i' takes value 1 on all but the last element of B_k^i. Then $h' = u_i' + w'$ is 2 on the 2^{i-1}th element of B_k^i, and 1 on all other elements of the block besides the last one. We pair up all elements of B_k^i as follows: each element x in the first half of the block is paired up with the element $x+2^{i-1}$. The function h is not Lipschitz on each such pair: $h(x+2^{i-1})-h(x) = \sum_{y=x}^{x+2^{i-1}-1} h'(y) = 2^{i-1} + 1$. Thus, h is $1/2$-far from Lipschitz on each such block. The other case (when i is even and w' is -1 on the 2^{i-1}th element of a block B_k^i) is analogous—the only difference is that h' takes negative values.

We can rephrase what we just proved as follows: the function h is $1/2$-far from Lipschitz on all blocks B_k^i with $k \in [2^{m-i}]$, where $w_{S\setminus\{i\}}(x) = w_{i+1}(x)$ for all $x \in B_k^i$. Equivalently, $w_{S\setminus\{i\}}(x) \cdot w_{i+1}(x) = w_{(S\setminus\{i\})\triangle\{i+1\}}(x) = 1$ for all $x \in B_k^i$. By Proposition III.4 and the fact that it is constant on each block B_k^i, the function $w_{(S\setminus\{i\})\triangle\{i+1\}}$ is the constant 1 function on at least half of the blocks. Thus, h is $1/2$-far from Lipschitz on at least half of the blocks B_k^i. That is, overall h is $1/4$-far from Lipschitz. ∎

Proof of Theorem III.12: The structure of the proof is very similar to that of the previous two lower bounds in this section. As in the monotonicity testing lower bound, when $n < r$ we will invoke Corollary II.6 with parameter t set to m, and when $r \le n$ we use the same proof with t set to $\lfloor \log_2(r-1) \rfloor$ and add the restriction that Alice and Bob's sets reside in $\{m-t+1, \ldots, m\}$ instead of in $[m]$.

Define a combining operator ψ that receives Alice's set A, and Bob's index i and set B as input then returns the function $h : [2^m] \to \mathbb{Z}$ defined by

$$h(x) = u_i(x) - \frac{1}{2}(w_{A\triangle B}(x) + 1),$$

where $A\triangle B = A \cap \{i, \ldots, m\}$. The additional restriction on the sets A and B that we introduced when $r \le n$ guarantee that in this case the range of the function is $\{0, 1, \ldots, 2^t\} \subseteq \{0, 1, \ldots, r-1\}$. Since $w_{A\triangle B} = w_A \cdot w_B$, the operator ψ is a one-bit one-way combining operator. And by Lemma III.14, when $i \notin A$ then h is Lipschitz and when $i \in A$ then h is $1/4$-far from Lipschitz. Therefore, we can apply Corollary II.6 to obtain the desired lower bound. ∎

IV. Lower bounds on the hypergrid

In this section, we generalize the lower bounds for testing functions on the line to the hypergrid setting. Specifically, we consider properties mapping the domain $[2^m]^d$ to some range $R \subseteq \mathbb{R}$. All of the lower bounds in this section are obtained via reductions from the AugmentedIndex$_{md}$ problem. In order to obtain these reductions, we associate each subset of $[md]$ with a d-dimensional vector of subsets of $[m]$ and each index in $[md]$ with a d-dimensional vector of indices in $\{0, 1, \ldots, m\}$.

Definition IV.1 (Vector representation)**.** Fix $m, d \in \mathbb{N}$. The *d-dimensional representation* of the set $S \subseteq [md]$ is the vector $\mathbf{S} = (\mathbf{S}_1, \ldots, \mathbf{S}_d)$ defined by $\mathbf{S}_j = \{\ell \in [m] : (j-1)m+\ell \in S\}$ for each $j \in [d]$. The d-dimensional representation of the index $i \in [md]$ is the vector $\boldsymbol{i} = (\boldsymbol{i}_1, \ldots, \boldsymbol{i}_d)$ defined by $\boldsymbol{i}_j = \max\{0, \min\{m, i-(j-1)m\}\}$ for each $j \in [d]$.

Equivalently, the d-dimensional representation of the index $i \in [md]$ is the vector $\boldsymbol{i} = (m, \ldots, m, \boldsymbol{i}_{j^*}, 0, \ldots, 0)$, where $j^* = \lceil i/m \rceil$ and $\boldsymbol{i}_{j^*} = i - (j^*-1)m$. We call j^* the *active coordinate* of the vector $\boldsymbol{i}$. Observe that $i \in S$ iff $\boldsymbol{i}_{j^*} \in \mathbf{S}_{j^*}$.

The notions of step functions and Walsh functions extend very naturally to the d-dimensional setting.

Definition IV.2 (Multidimensional step functions)**.** The *step function* indexed by the d-dimensional vector $\boldsymbol{i} \in [m]^d$ is the function $s_{\boldsymbol{i}} : [2^m]^d \to [d2^m]$ defined by

$$s_{\boldsymbol{i}}(x_1, \ldots, x_d) = \sum_{j=1}^{d} s_{\boldsymbol{i}_j}(x_j).$$

Definition IV.3 (Multidimensional Walsh functions)**.** The *Walsh function* indexed by the d-dimensional vector $\mathbf{S}$ of subsets of $[m]$ is the function $w_{\mathbf{S}} : [2^m]^d \to \{-1, 1\}$ defined by

$$w_{\mathbf{S}}(x_1, \ldots, x_d) = \prod_{j=1}^{d} w_{\mathbf{S}_j}(x_i).$$

The multidimensional Walsh functions satisfy the same basic properties that we used in our lower bound constructions for properties of functions on the line (c.f. Propositions III.4 and III.5).

Proposition IV.4. *For every $S \subseteq [md]$ with d-dimensional representation $\mathbf{S}$, the Walsh function $w_{\mathbf{S}}$ satisfies $\sum_{x \in [2^m]^d} w_{\mathbf{S}}(x) \ge 0$.*

Proof: It is sufficient to prove that if the random variables $X_1, \ldots, X_d$ are i.i.d. and uniform over $[2^m]$ then $\Pr[w_{\mathbf{S}}(X_1, \ldots, X_d) = 1] \ge 1/2$. If $\mathbf{S}_j = \emptyset$ then $w_{\mathbf{S}_j}(X_j) = 1$. For all $j \in [d]$ such that $\mathbf{S}_j \neq \emptyset$, the random variables $w_{\mathbf{S}_j}(X_j) \in \{-1,1\}$ are i.i.d. and uniformly distributed over $\{-1,1\}$. Thus, $\Pr[w_{\mathbf{S}}(X_1,\ldots,X_d)=1] = \Pr[\prod_{j\in[d]} w_{\mathbf{S}_j}(X_j)=1] \geq 1/2$. ■

Corollary IV.5. *Let $\mathbf{S}$ be the d-dimensional representation of $S \subseteq [md]$. The product $\prod_{k\in[d]\setminus\{j\}} w_{\mathbf{S}_k}(x_k)$, where $x_k \in [2^m]$ for all $k \in [d]\setminus\{j\}$, evaluates to 1 for at least half of the settings of variables x_k.*

Proof: Let $\mathbf{S}'$ be the $(d-1)$-dimensional vector $(\mathbf{S}_1,\ldots,\mathbf{S}_{j-1},\mathbf{S}_{j+1},\ldots,\mathbf{S}_d)$. Then $\prod_{k\in[d]\setminus\{j\}} w_{\mathbf{S}_k}(x_k) = w_{\mathbf{S}'}(x_1,\ldots,x_{j-1},x_{j+1},\ldots,x_d)$. By Proposition IV.4, this expression is 1 for at least half of the settings of x_k. ■

Proposition IV.6. *Fix $A,B \subseteq [md]$ and $S = A\triangle B$. Let $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{S}$ be the d-dimensional vector representations of the sets A, B, and S, respectively. Then $w_{\mathbf{S}} : [2^m]^d \to \{-1,1\}$ satisfies $w_{\mathbf{S}}(x) = w_{\mathbf{A}}(x)\cdot w_{\mathbf{B}}(x)$ for all $x \in [2^m]^d$.*

A. Monotonicity

The lower bound for testing monotonicity over the hypergrid domain is conceptually similar to the monotonicity lower bound for the line domain. For the hypergrid domain, however, we start with the AUGMENTEDINDEX$_{md}$ problem and use the d-dimensional representation of Alice and Bob's inputs A,B, and i to define a combining operator ψ that returns a function h that (a) is monotone in every dimension when $i \notin A$, and (b) is far from monotone in one dimension j^* when $i \in A$. The details follow.

Proof of Theorem I.1: We use Corollary II.6 with parameter $t = md$. Let $A \subseteq [md]$ be Alice's input and $i \in [md]$ and $B = A \cap [i-1]$ be Bob's input.

The combining operator ψ is defined as follows. It receives A, i, B as input. Then it computes $S = A\triangle B = A \cap \{i,\ldots,md\}$ and the d-dimensional vectors $\boldsymbol{i}$ and $\mathbf{S}$ corresponding to i and S, respectively. It returns the function $h : [n]^d \to \{d-1,\ldots,dn+1\}$ defined by

$$h(x) = 2s_{\boldsymbol{i}}(x) + w_{\mathbf{S}}(x).$$

By Proposition IV.6, $w_{\mathbf{S}} = w_{\mathbf{A}}\cdot w_{\mathbf{B}}$, where $\mathbf{A}$ and $\mathbf{B}$ are the d-dimensional representations of A and B, respectively. Bob knows i and B and can compute their vector representations. To determine $h(x)$, he only needs Alice to communicate the bit $w_{\mathbf{A}}(x)$. Thus, ψ is a one-bit one-way combining operator. Lemma IV.7, below, concludes the proof that ψ is a reduction operator for monotonicity and $\epsilon_0 = 1/8$, implying the theorem. ■

Lemma IV.7. *Fix $i \in [md]$ and $S \subseteq \{i,\ldots,dm\}$, and let $\boldsymbol{i}$ and $\mathbf{S}$, respectively, be their d-dimensional vector representations. If $i \notin S$, then h is monotone. Otherwise, h is $\frac{1}{8}$-far from monotone.*

Proof: Let $j^* = \lceil i/m \rceil$. We will show that all line restrictions of h to dimensions other than j^* are monotone. If $i \notin S$, we will show that all line restrictions of h to dimension j^* are also monotone, so h itself is monotone. Conversely, if $i \in S$, we will show that at least half of the line restrictions of h to dimension j^* are $1/4$-far from monotone, so h itself is $1/8$-far from monotone.

Consider the restriction of $h = 2s_{\boldsymbol{i}} + w_{\mathbf{S}}$ to a line in dimension $j \in [d]$, i.e., a function $\bar{h} : [2^m] \to \mathbb{N}$ defined by $\bar{h}(x_j) = h(\bar{x}_1,\ldots,\bar{x}_{j-1},x_j,\bar{x}_{j+1},\ldots,\bar{x}_d)$, where the values $\bar{x}_k \in [2^m]$ are fixed for all $k \in [d]\setminus\{j\}$. Then

$$\begin{aligned}\bar{h}(x_j) &= 2\sum_{k\neq j} s_{\boldsymbol{i}_k}(\bar{x}_k) + 2s_{\boldsymbol{i}_j}(x_j) + w_{\mathbf{S}_j}(x_j)\cdot\prod_{k\neq j} w_{\mathbf{S}_k}(\bar{x}_k)\\ &= 2s_{\boldsymbol{i}_j}(x_j) \pm w_{\mathbf{S}_j}(x_j) + c, \qquad (2)\end{aligned}$$

where $\pm$ means "either $+$ or $-$" and c is a constant independent of x_j.

If $j < j^*$ then $\mathbf{S}_j = \emptyset$, $\boldsymbol{i}_j = m$ and $\bar{h} = 2s_m \pm w_\emptyset + c = 2 \pm 1 + c$. And if $j > j^*$ then $\boldsymbol{i}_j = 0$, so $\bar{h}(x_j) = 2x_j \pm w_{\mathbf{S}_j}(x_j) + c$. In both cases, the function $\bar{h}$ is monotone.

Finally, if $j = j^*$ then $\boldsymbol{i}_j = i-(j-1)m$. In this case, $i \in S$ iff $\boldsymbol{i}_j \in \mathbf{S}_j$. If $\boldsymbol{i}_j \notin \mathbf{S}_j$ then, by (2) and Lemma III.7, $\bar{h}(x_j)$ is monotone. Since all line restrictions of $h(x)$ are monotone, the overall function $h(x)$ is monotone. Now suppose $\boldsymbol{i}_j \in \mathbf{S}_j$. Consider the product $\prod_{k\neq j} w_{\mathbf{S}_k}(\bar{x}_k)$ that determines whether the expression $\pm$ in (2) is actually a plus or a minus. By Corollary IV.5, this product evaluates to 1 for at least half of the line restrictions $\bar{h}$ of h in dimension j. For those restrictions, $\bar{h}(x_j) = 2s_{\boldsymbol{i}_j}(x_j) + w_{\mathbf{S}_j}(x_j) + c$ and, since $\boldsymbol{i}_j \in \mathbf{S}_j$, Lemma III.7 implies that $\bar{h}$ is $\frac{1}{4}$-far from monotone. Thus, at least half of the line restrictions of h in dimension j are $1/4$-far from monotone. Since the domains of line restrictions of h in dimension j partition the domain of h, it implies that the overall function $h(x)$ is $\frac{1}{8}$-far from monotone. ■

B. Convexity

The lower bound for testing separate convexity on the hypergrid domain is obtained with an argument similar to the one in Section IV-A: we define a combining operator ψ for the AUGMENTEDINDEX$_{md}$ problem that returns a function h that is (a) convex in every dimension when $i \notin A$, and (b) far from convex in one dimension when $i \in A$.

This approach does not suffice for the convexity lower bound, however, since the convexity of the restriction of a function h in every dimension does not imply that h itself is convex; to ensure that h is convex, we need to construct a reduction such that when $i \notin A$, the projection of h is convex on *every* line, not just the axis-parallel ones.

The proofs of the lower bounds for testing separate convexity and for testing convexity share some common elements, so we present them together.

Proof of Theorems I.2 and I.3: We apply Corollary II.6 with parameter $t = md$. Let $A \subseteq [md]$ be the set received by Alice and let $i \in [md]$ and $B = A\cap[i-1]$ be Bob's input. Let $j^* = \lceil i/m \rceil$. Let $\mathbf{A},\mathbf{B}$ and $\boldsymbol{i}$ be the d-dimensional vectors corresponding to A, B and i respectively. The combining operator ψ receives $\mathbf{A}$ and $\boldsymbol{i}$ as input and returns the function $h : [n]^d \to \mathbb{R}$ defined by

$$h(x) = \alpha\left(\frac{1}{2}\left(w_{\mathbf{S}}(x)+1\right) + r_{\boldsymbol{i}_{j^*}}(x_{j^*})\right) + \sum_{j=j^*+1}^{d} {x_j}^2,$$

where $\mathbf{S}$ is the d-dimensional vector corresponding to $S = A \triangle B = A \cap \{i, \ldots, md\}$ and $r_{\boldsymbol{i}_{j^*}}$ is a rising-step-size function (see Definition III.10). The parameter α is set to 1 for separate convexity. In this case, the range of h is $[r]$ for $r = O(dn^2)$ because for every $k \in [m]$ the range of r_k is $O(n^2)$. For convexity, $\alpha \in (0,1)$ is selected later, to satisfy Lemma IV.8 below. For any $x \in [n]^d$, Bob only needs the single bit $w_{\mathbf{A}}(x)$ from Alice to compute $h(x)$, so ψ is a one-bit one-way combining operator.

To show that ψ is a reduction operator for convexity (resp., separate convexity) we need to show that if $i \notin S$ (or equivalently $\boldsymbol{i}_{j^*} \notin \mathbf{S}_{j^*}$) then h is convex (resp., separately convex) and otherwise h is $\frac{1}{16}$-far from convex (resp., separately convex). We do so with the help of the following lemma. To apply Lemma IV.8 in the case of convexity recall that the distance of a function f to convex is at least the distance of f to separately convex. ∎

Lemma IV.8. *Fix $i \in [md]$ and $S \subseteq \{i, \ldots, dm\}$, and let $\boldsymbol{i}$ and $\mathbf{S}$, respectively, be their d-dimensional vector representations. $j^* = \lceil i/m \rceil$. If $\boldsymbol{i}_{j^*} \notin \mathbf{S}_{j^*}$ then (1) for $\alpha = 1$ the function h is separately convex; (2) there exists $\alpha > 0$ such that the function h is convex. Otherwise (if $\boldsymbol{i}_{j^*} \in \mathbf{S}_{j^*}$), the function h is $\frac{1}{16}$-far from separately convex for all $\alpha > 0$.*

Proof: To prove part (1), it suffices to show that every restriction of h to any dimension $j \in [d]$ is a convex function.

Every one-dimensional restriction $\bar{h}$ of h in dimension j^* can be expressed as $\bar{h}(x_{j^*}) = \alpha(r_{\boldsymbol{i}_{j^*}}(x_i) \pm \frac{1}{2} w_{\mathbf{S}_{j^*}}(x_{j^*})) + c$, where c is some constant independent of x_{j^*}. Since $\boldsymbol{i}_{j^*} \notin \mathbf{S}_{j^*}$, this function is convex by Lemma III.11. For all $j < j^*$, every one-dimensional restriction $\bar{h}$ of h to dimension j is a constant function. For all $j > j^*$, the restrictions of h to dimension j can be expressed as $\bar{h}(x_j) = \pm\frac{1}{2}\alpha w_{\mathbf{S}_j}(x_j) + {x_j}^2 + c$. The derivative of the first term $w_{\mathbf{S}_j}$ satisfies that $|\frac{1}{2}\alpha w'_{\mathbf{S}_j}(x_j)| \le \alpha$ and the derivative of the second term is $2x_j$, so for $\alpha \le 1$ the derivative $\bar{h}'$ is a nondecreasing function and $\bar{h}$ is convex. Hence, the function h is separately convex for all $\alpha \le 1$. This completes the proof of part (1).

To prove part (2), we show how to pick a parameter $\alpha \in (0,1)$ such that the function h is convex. By definition, to prove that h is convex we need to show that $h(z) \le \gamma h(x) + (1-\gamma)h(y)$ for every pair of points $(x,y) \in [n]^d \times [n]^d$ and every $\gamma \in (0,1)$ for which $z = \gamma x + (1-\gamma)y \in [n]^d$.

The function h is independent of the first $j^* - 1$ coordinates, so $h(x) = h(y_1, \ldots, y_{j^*-1}, x_{j^*}, \ldots, x_d)$ and $h(z) = h(y_1, \ldots, y_{j^*-1}, z_{j^*}, \ldots, z_d)$.

First, consider the case when $x_j = y_j$ for all $j > j^*$, so we have $x = (x_1, \ldots, x_{j^*}, y_{j^*+1}, \ldots, y_d)$. By Lemma IV.8 (Part 1), all the restrictions $\bar{h}$ of h to dimension j^* are convex, so in this case $h(z) \le \gamma h(x) + (1-\gamma)h(y)$.

Otherwise, fix an index $j > j^*$ such that $x_j \ne y_j$.

Proposition IV.9. *Define $\phi_{j^*}(x) = \sum_{t=j^*+1}^{d} {x_t}^2$. For all $n, d \ge 1$ there exists a value $\delta^*(n,d) > 0$ such that*

$$\phi_{j^*}(\gamma x + (1-\gamma)y) \le \gamma\phi_{j^*}(x) + (1-\gamma)\phi_{j^*}(y) - \delta^*(n,d)$$

for all pairs (x,y), where $x_j \ne y_j$ for some $j > j^*$, and all $\gamma \in (0,1)$, where $\gamma x + (1-\gamma)y \in [n]^d$.

Proof: Let j be an index such that $x_j \ne y_j$ and $j > j^*$. Then

$$\begin{aligned}
&\phi_{j^*}(\gamma x + (1-\gamma)y) - \gamma\phi_{j^*}(x) - (1-\gamma)\phi_{j^*}(y) \\
&\quad = \sum_{t=j^*+1}^{d}\left((\gamma x_t + (1-\gamma)y_t)^2 - \gamma {x_t}^2 - (1-\gamma){y_t}^2\right) \\
&\quad \le \left((\gamma x_j + (1-\gamma)y_j)^2 - \gamma {x_j}^2 - (1-\gamma){y_j}^2\right) < 0.
\end{aligned}$$

The first inequality uses convexity of x^2. The second inequality uses its strict convexity and the fact that $x_j \ne y_j$. Let

$$\begin{aligned}
&\delta(x,y,j,\gamma,n,d) \\
&\quad = -\left((\gamma x_j + (1-\gamma)y_j)^2 - \gamma {x_j}^2 - (1-\gamma){y_j}^2\right) > 0.
\end{aligned}$$

Note that j and γ can take at most d and n^d different values respectively for any fixed pair (x,y). Thus there are at most dn^{3d} different valid tuples (x,y,j,γ). The claim follows by letting $\delta^*(n,d) = \min_{x,y,j,\gamma} \delta(x,y,j,\gamma,n,d)$. ∎

We set $\alpha = \frac{\delta^*(n,d)}{6(2n^2+1)}$. Using the notation introduced above,

$$\begin{aligned}
h(x) &= \alpha\left(\frac{1}{2}\left(w_{\mathbf{S}}(x)+1\right) + r_{\boldsymbol{i}_{j^*}}(x_{j^*})\right) + \sum_{j>j^*} {x_j}^2 \\
&= \alpha\left(\frac{1}{2}\left(w_{\mathbf{S}}(x)+1\right) + r_{\boldsymbol{i}_{j^*}}(x_{j^*})\right) + \phi_{j^*}(x).
\end{aligned}$$

Since the range of $r_{\boldsymbol{i}_{j^*}}$ is $[2n^2]$,

$$\begin{aligned}
&h(z) - \gamma h(x) - (1-\gamma)h(y) \\
&\quad \le \phi_{j^*}(z) - \gamma\phi_{j^*}(x) - (1-\gamma)\phi_{j^*}(y) + 3\alpha(2n^2+1) \\
&\quad \le -\delta^*(n,d) + 3\alpha(2n^2+1) = -\delta^*(n,d)/2 < 0,
\end{aligned}$$

where the inequalities follow from Proposition IV.9. This concludes the proof of the fact that h is convex (part (2) of Lemma IV.8).

Finally, we consider the case $\boldsymbol{i}_{j^*} \in \mathbf{S}_{j^*}$. By Corollary IV.5, the product $\prod_{k \ne j^*} w_{\mathbf{S}_k}(x_k)$ evaluates to 1 for at least half of the line restrictions $\bar{h}$ of h to dimension j^*. For such restrictions, $\bar{h}(x_{j^*}) = \alpha(\frac{1}{2} w_{\mathbf{S}_{j^*}}(x_{j^*}) + r_{\boldsymbol{i}_{j^*}}(x_{j^*})) + c$, for some constant c. Lemma III.11 implies that $\bar{h}$ is $\frac{1}{8}$-far from convex. The domains of the restrictions $\bar{h}$ of h

in dimension j^* partition the domain of h, so we conclude that the function h is $\frac{1}{16}$-far from separately convex. ■

C. The Lipschitz property

Definition IV.10 (Multidimensional up-down staircase functions)**.** The *up-down staircase function* indexed by the d-dimensional vector $\boldsymbol{i} \in [m]^d$ is the function $u_{\boldsymbol{i}} : [2^m]^d \to [d2^m]$ defined by $u_{\boldsymbol{i}}(x_1, \ldots, x_d) = \sum_{j=1}^{d} u_{\boldsymbol{i}_j}(x_j)$.

Proof of Theorem I.4: The starting point of the reduction is the same as in the proof of the lower bound for monotonicity in Section IV-A. We use the same notation for the parameters of the reduction from AUGMENTEDINDEX$_{md}$, Alice's and Bob's inputs, the set $S = A \triangle B = A \cap \{i, \ldots, md\}$ and the vector representation of these objects. The combining operator ψ returns the function

$$h(x) = u_{\boldsymbol{i}}(x) - \tfrac{1}{2}(w_{\mathbf{S}}(x) + 1).$$

As in the proof of Theorem I.1, ψ is a one-bit one-way combining operator. The next lemma completes the proof of the theorem. ■

Lemma IV.11. *Fix $i \in [md]$ and $S \subseteq \{i, \ldots, dm\}$, and let $\boldsymbol{i}$ and $\mathbf{S}$ be their respective d-dimensional vector representations. If $i \notin S$, then h is Lipschitz. Otherwise, h is $\frac{1}{8}$-far from Lipschitz.*

Proof: Consider a restriction of h to a line in dimension $j \in [d]$, that is, a univariate function $\bar{h}(x_j) = h(\bar{x}_1, \ldots, \bar{x}_{j-1}, x_j, \bar{x}_{j+1}, \ldots, \bar{x}_d)$, where the values $\bar{x}_k \in [2^m]$ are fixed for all $k \in [d] \setminus \{j\}$. Then

$$\begin{aligned}\bar{h}(x_j) =& \sum_{k \neq j} u_{\boldsymbol{i}_k}(\bar{x}_k) + u_{\boldsymbol{i}_j}(x_j) \\ & - \frac{1}{2}\Big(w_{\mathbf{S}_j}(x_j) \cdot \prod_{k \neq j} w_{\mathbf{S}_k}(\bar{x}_k) + 1\Big) \\ =& u_{\boldsymbol{i}_j}(x_j) - \frac{1}{2}(\pm w_{\mathbf{S}_j}(x_j) + 1) + c, \end{aligned} \tag{3}$$

where $\pm$ means "either $+$ or $-$" and c is a constant independent of x_j.

Let $j^* = \lceil i/m \rceil$. If $j < j^*$ then $\mathbf{S}_j = \emptyset$, $\boldsymbol{i}_j = m$ and $\bar{h} = u_{\boldsymbol{i}_j} - \frac{1}{2}(\pm 1 + 1) + c$. Since every up-down staircase function u_i is Lipschitz, and since a Lipschitz function plus a constant function is Lipschitz, the resulting function $\bar{h}$ is Lipschitz. If $j > j^*$ then $\boldsymbol{i}_j = 0$, so $\bar{h}(x_j) = 1 - \frac{1}{2}(\pm w_{\mathbf{S}_j}(x_j) + 1) + c$,, i.e., $\bar{h}$ is again a Lipschitz function because it is the sum of a Lipschitz function and a constant function.

Finally, if $j = j^*$ then $\boldsymbol{i}_j = i - (j-1)m$. In this case, $i \in S$ iff $\boldsymbol{i}_j \in \mathbf{S}_j$. If $\boldsymbol{i}_j \notin \mathbf{S}_j$ then, by (3) and Lemma III.14, $\bar{h}$ is Lipschitz. Since all line restrictions of h are Lipschitz, the overall function h is Lipschitz. Now suppose $\boldsymbol{i}_j \in \mathbf{S}_j$. Consider the product $\prod_{k \neq j} w_{\mathbf{S}_k}(\bar{x}_k)$ that determines whether the expression $\pm$ in (3) is a plus or a minus. By Corollary IV.5, this product evaluates to 1 for at least half of the line restrictions $\bar{h}(x_j)$ of h in dimension j. For those restrictions, $\bar{h}(x_j) = u_{\boldsymbol{i}_j} + \frac{1}{2}(w_{\mathbf{S}_j} + 1)(x_j) + c$ and, since $\boldsymbol{i}_j \in \mathbf{S}_j$, Lemma III.14 implies that $\bar{h}$ is $\frac{1}{4}$-far from Lipschitz. Thus, at least half of the line restrictions of h in dimension j are $1/4$-far from Lipschitz. Since the domains of the line restrictions of h in dimension j partition the domain of h, the overall function h is $\frac{1}{8}$-far from Lipschitz. ■

ACKNOWLEDGMENTS

E.B. is supported by a postdoctoral fellowship from the Simons Foundation. Part of this research was completed while the author was at Carnegie Mellon University. S.R. and G.Y. were supported in part by NSF CAREER award CCF-0845701. In addition, G.Y. was supported by College of Engineering Fellowship at Pennsylvania State University.

We thank Madhav Jha for his participation in the project. He declined to be a co-author, but we would like to acknowledge his contributions to the results presented here. We also thank Joshua Brody and Oded Goldreich for insightful conversations about the communication complexity method.

REFERENCES

[1] N. Ailon and B. Chazelle, "Information theory in property testing and monotonicity testing in higher dimension," *Inf. Comput.*, vol. 204, no. 11, pp. 1704–1717, 2006.

[2] R. J. Aumann and S. Hart, "Bi-convexity and bi-martingales," *Israel Journal of Mathematics*, vol. 54, no. 2, pp. 159–180, 1986.

[3] P. Awasthi, M. Jha, M. Molinaro, and S. Raskhodnikova, "Testing Lipschitz functions on hypergrid domains," in *APPROX-RANDOM*, 2012, pp. 387–398.

[4] P. Berman, S. Raskhodnikova, and G. Yaroslavtsev, "L_p-testing," in *STOC*, 2014.

[5] A. Bhattacharyya, E. Grigorescu, K. Jung, S. Raskhodnikova, and D. P. Woodruff, "Transitive-closure spanners," *SIAM J. Comput.*, vol. 41, no. 6, pp. 1380–1425, 2012.

[6] E. Blais, J. Brody, and K. Matulef, "Property testing lower bounds via communication complexity," *Computational Complexity*, vol. 21, no. 2, pp. 311–358, 2012.

[7] E. Blais, S. Raskhodnikova, and G. Yaroslavtsev, "Lower bounds for testing properties of functions on hypergrid domains," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. TR13-036, 2013.

[8] J. Brody, "A tight lower bound for monotonicity testing over large ranges," *CoRR*, vol. abs/1304.6685, 2013.

[9] D. Chakrabarty, K. Dixit, M. Jha, and C. Seshadhri, "Optimal lower bounds for Lipschitz testing via monotonicity," *Personal communication*, 2013.

[10] ——, "Property testing on product distributions: Optimal testers for bounded derivative propertiess," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. TR14-042, 2014.

[11] D. Chakrabarty and C. Seshadhri, “A $o(n)$ monotonicity tester for Boolean functions over the hypercube,” in *STOC*, 2013, pp. 411–418.

[12] ——, “Optimal bounds for monotonicity and Lipschitz testing over hypercubes and hypergrids,” in *STOC*, 2013, pp. 419–428.

[13] ——, “An optimal lower bound for monotonicity testing over hypergrids,” in *APPROX-RANDOM*, 2013, pp. 425–435.

[14] K. Dixit, M. Jha, S. Raskhodnikova, and A. Thakurta, “Testing the Lipschitz property over product distributions with applications to data privacy,” in *TCC*, 2013, pp. 418–436.

[15] Y. Dodis, O. Goldreich, E. Lehman, S. Raskhodnikova, D. Ron, and A. Samorodnitsky, “Improved testing algorithms for monotonicity.” in *RANDOM*, 1999, pp. 97–108.

[16] F. Ergun, S. Kannan, S. R. Kumar, R. Rubinfeld, and M. Viswanathan, “Spot-checkers,” *J. Comput. Syst. Sci.*, vol. 60, no. 3, pp. 717–751, 2000.

[17] E. Fischer, “On the strength of comparisons in property testing,” *Information and Computation*, vol. 189, no. 1, pp. 107–116, 2004.

[18] E. Fischer, E. Lehman, I. Newman, S. Raskhodnikova, R. Rubinfeld, and A. Samorodnitsky, “Monotonicity testing over general poset domains.” in *STOC*, 2002, pp. 474–483.

[19] O. Goldreich, Ed., *Property Testing: Current Research and Surveys*, ser. Lecture Notes in Computer Science, vol. 6390. Springer, 2010.

[20] O. Goldreich, “On the communication complexity methodology for proving lower bounds on the query complexity of property testing.” in *Electronic Colloquium on Computational Complexity (ECCC)*, vol. TR13-073, 2013.

[21] O. Goldreich, S. Goldwasser, E. Lehman, D. Ron, and A. Samorodnitsky, “Testing monotonicity.” *Combinatorica*, vol. 20, no. 3, pp. 301–337, 2000.

[22] O. Goldreich, S. Goldwasser, and D. Ron, “Property testing and its connection to learning and approximation,” *J. ACM*, vol. 45, no. 4, pp. 653–750, 1998.

[23] E. Grigorescu, K. Wimmer, and N. Xie, “Tight lower bounds for testing linear isomorphism,” in *APPROX-RANDOM*, 2013, pp. 559–574.

[24] P. Hatami, “Lower bounds on testing functions of low Fourier degree,” *CoRR*, vol. abs/1202.3479, 2012.

[25] M. Jha and S. Raskhodnikova, “Testing and reconstruction of Lipschitz functions with applications to data privacy,” *SIAM J. Comput.*, vol. 42, no. 2, pp. 700–731, 2013.

[26] J. Matoušek and P. Plecháč, “On functional separately convex hulls,” *Discrete & Computational Geometry*, vol. 19, no. 1, pp. 105–130, 1998.

[27] J. Matoušek, “On directional convexity,” *Discrete & Computational Geometry*, vol. 25, no. 3, pp. 389–403, 2001.

[28] P. B. Miltersen, N. Nisan, S. Safra, and A. Wigderson, “On data structures and asymmetric communication complexity,” *J. Comput. Syst. Sci.*, vol. 57, no. 1, pp. 37–49, 1998.

[29] M. Parnas, D. Ron, and R. Rubinfeld, “On testing convexity and submodularity,” *SIAM J. Comput.*, vol. 32, no. 5, pp. 1158–1184, 2003.

[30] L. Rademacher and S. Vempala, “Testing geometric convexity,” in *FSTTCS*, 2004, pp. 469–480.

[31] S. Raskhodnikova, “Approximate testing of visual properties,” in *RANDOM-APPROX*, 2003, pp. 370–381.

[32] D. Ron, “Algorithmic and analysis techniques in property testing,” *Foundations and Trends in Theoretical Computer Science*, vol. 5, no. 2, pp. 73–205, 2009.

[33] R. Rubinfeld and A. Shapira, “Sublinear time algorithms,” *SIAM J. Discrete Math.*, vol. 25, no. 4, pp. 1562–1588, 2011.

[34] R. Rubinfeld and M. Sudan, “Robust characterization of polynomials with applications to program testing,” *SIAM J. Comput.*, vol. 25, no. 2, pp. 252–271, 1996.

[35] L. Tartar, “Some remarks on separately convex functions,” in *Microstructure and phase transition*. Springer, 1993, pp. 191–204.