Directed Isoperimetric Theorems for Boolean Functions on the Hypergrid and an $\widetilde{O}(n\sqrt{d})$ Monotonicity Tester

Hadley Black
hablack@ucla.edu
University of California, Los Angeles
USA

Deeparnab Chakrabarty
deeparnab@dartmouth.edu
Dartmouth College
USA

C. Seshadhri
deeparnab@dartmouth.edu
University of California, Santa Cruz
USA

ABSTRACT

The problem of testing monotonicity for Boolean functions on the hypergrid, $f : [n]^d \to \{0, 1\}$ is a classic topic in property testing. When $n = 2$, the domain is the hypercube. For the hypercube case, a breakthrough result of Khot-Minzer-Safra (FOCS 2015) gave a non-adaptive, one-sided tester making $\widetilde{O}(\varepsilon^{-2}\sqrt{d})$ queries. Up to polylog d and ε factors, this bound matches the $\widetilde{\Omega}(\sqrt{d})$-query non-adaptive lower bound (Chen-De-Servedio-Tan (STOC 2015), Chen-Waingarten-Xie (STOC 2017)). For any $n > 2$, the optimal non-adaptive complexity was unknown. A previous result of the authors achieves a $\widetilde{O}(d^{5/6})$-query upper bound (SODA 2020), quite far from the $\sqrt{d}$ bound for the hypercube.

In this paper, we resolve the non-adaptive complexity of monotonicity testing for all constant n, up to $\text{poly}(\varepsilon^{-1} \log d)$ factors. Specifically, we give a non-adaptive, one-sided monotonicity tester making $\widetilde{O}(\varepsilon^{-2}n\sqrt{d})$ queries. From a technical standpoint, we prove new directed isoperimetric theorems over the hypergrid $[n]^d$. These results generalize the celebrated directed Talagrand inequalities that were only known for the hypercube.

CCS CONCEPTS

• **Theory of computation → Streaming, sublinear and near linear time algorithms**.

KEYWORDS

Property Testing, Monotonicity Testing, Boolean Functions, Isoperimetry Theorems

ACM Reference Format:
Hadley Black, Deeparnab Chakrabarty, and C. Seshadhri. 2023. Directed Isoperimetric Theorems for Boolean Functions on the Hypergrid and an $\widetilde{O}(n\sqrt{d})$ Monotonicity Tester. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC '23), June 20–23, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3564246.3585167

1 INTRODUCTION

Monotonicity testing, especially over hypergrid domains, is one of the most well studied problems in property testing. We use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. The set $[n]^d$ is the d-dimensional hypergrid where $\mathbf{x} \in [n]^d$ is a d-dimensional vector with $\mathbf{x}_i \in [n]$. The hypergrid is equipped with the natural partial order $\mathbf{x} \preceq \mathbf{y}$ iff $\mathbf{x}_i \leq \mathbf{y}_i$ for all $i \in [d]$. Note that when $n = 2$, the hypergrid $[n]^d$ is isomorphic to the hypercube $\{0, 1\}^d$.

Let $f : [n]^d \to \{0, 1\}$ be a Boolean function defined on the hypergrid. The function f is monotone if $f(\mathbf{x}) \leq f(\mathbf{y})$ whenever $\mathbf{x} \preceq \mathbf{y}$. The Hamming distance between two Boolean functions f and g, denoted as $\Delta(f, g)$, is the fraction of points where they differ. The *distance to monotonicity* of a function $f : [n]^d \to \{0, 1\}$ is defined as $\varepsilon_f := \min_{g \text{ monotone}} \Delta(f, g)$. The Boolean monotonicity testing problem on the hypergrid takes parameter ε and oracle access to $f : [n]^d \to \{0, 1\}$. The objective is to design a randomized algorithm, called the tester, that accepts a monotone function with probability $\geq 2/3$ and rejects a function f with $\varepsilon_f \geq \varepsilon$ with probability $\geq 2/3$. A tester is one-sided if it accepts a monotone function with probability 1. A tester is non-adaptive if all its queries are made in one round before seeing any responses.

There has been a rich history of results on monotonicity testing over hypergrids, with a significant focus on hypercubes [3, 4, 8–10, 16–18, 21–24, 29, 32, 33]. We discuss the history more in Section 1.5, but for now, we give the state of the art. For hypercubes, after a long line of work, the breakthrough result [33] of Khot, Minzer, and Safra gave an $\widetilde{O}_\varepsilon(\sqrt{d})$-query non-adaptive, one-sided tester. This result is tight due to a nearly matching $\widetilde{\Omega}(\sqrt{d})$-query lower bound for non-adaptive testers due to Chen, Waingarten, and Xie [23]. For general hypergrids, the best upper bound is the $\widetilde{O}_\varepsilon(d^{5/6})$-query tester of the authors [8, 9].

This $\widetilde{\Omega}(\sqrt{d})$ vs $\widetilde{O}(d^{5/6})$ gap for non-adaptive testers is a tantalizing and important open question in property testing. Even for the domain $[3]^d$, the optimal non-adaptive monotonicity testing bound is unknown. One of the main questions driving our work is:

Are there $\widetilde{O}_\varepsilon(\sqrt{d})$-query monotonicity testers for domains beyond the hypercube?

Directed isoperimetric theorems. The initial seminal work on monotonicity testing, by Goldreich, Goldwasser, Lehman, Ron, and Samorodnitsky [29] and Dodis, Goldreich, Lehman, Ron, Raskhodnikova and Samorodnitsky [24] prove the existence of $\widetilde{O}_\varepsilon(d)$-query testers. For almost a decade, it was not clear whether $o(d)$-query testers were possible. In [18], the last two authors gave the first such tester via an exciting connection with *robust directed isoperimetric theorems*. Indeed, all $o(d)$-query testers are achieved through such theorems.

Think of a Boolean function f as the indicator for a subset of the domain. The variance of f, $\text{var}(f)$, is a measure of the volume of the indicated subset. An isoperimetric theorem for Boolean functions relates the variance of f to the "boundary" of the function which corresponds to the sensitive edges and/or their endpoints.

STOC '23, June 20–23, 2023, Orlando, FL, USA

ACM ISBN 978-1-4503-9913-5/23/06.
https://doi.org/10.1145/3564246.3585167

The deep insight of these theorems comes from sophisticated ways of measuring boundary size, involving both the vertex and edge boundary. A *directed* isoperimetric theorem is an analog where we only measure "up-boundary" formed by monotonicity violations. Rather surprisingly, in the directed case, one can replace the variance as a measure of volume by the distance to monotonicity.

In Table 1, we list some classic isoperimetric results and their directed analogues for the hypercube. For a point $\mathbf{x}$, $I_f(\mathbf{x})$ is the number of sensitive edges incident to $\mathbf{x}$. We use I_f to denote $\mathbf{E}_\mathbf{x}[I_f(\mathbf{x})]$, the total influence of f, which the number of sensitive edges in f divided by the domain size 2^d. The quantity Γ_f is the vertex boundary size divided by 2^d. The directed analogues of these, $I^-_f, \Gamma^-_f, I^-_f(\mathbf{x})$, only consider sensitive edges that violate monotonicity.

Observe the remarkable parallel between the standard isoperimetric results and their directed versions. The Talagrand inequality is the strongest statement, and implies all other bounds. The directed versions imply the undirected versions, using standard inequalities regarding monotone functions. The [33] $\widetilde{O}_\varepsilon(\sqrt{d})$-query tester is based on the directed Talagrand inequality.

The story for hypergrids is much more complicated. From an isoperimetric perspective, a common approach is to consider the *augmented hypergrid*, wherein we add edges between pairs in the same line. The dimension reduction technique in [24] used to prove the $\widetilde{O}_\varepsilon(d)$ testers can be thought of as establishing a directed Poincaré inequality . In previous work [8], the authors proved a directed Margulis inequality, which led to the $\widetilde{O}_\varepsilon(d^{5/6})$ query tester. Another motivating question for our work is:

Can the directed Talagrand inequality be generalized beyond the hypercube?

1.1 Main Results

We answer both questions mentioned above in the affirmative. To state our results more formally, we begin with some notation. For any $i \in [d]$, we use $\mathbf{e}_i$ to denote the d-dimensional vector which has 1 on the ith coordinate and zero everywhere else. For a dimension i, a pair $(\mathbf{x}, \mathbf{y})$ is called *i-aligned* if $\mathbf{x}$ and $\mathbf{y}$ only differ on their i-coordinate. An *i-line* is a 1D line of n points obtained by fixing all but the ith coordinate.

We define a notion of directed influence of Boolean functions on hypergrids, which generalizes the notion for Boolean functions on hypercubes. In plain English, for a point $\mathbf{x}$ we count the number of *dimensions* in which $\mathbf{x}$ takes part in a violation. We call this the *thresholded negative influence* of $\mathbf{x}$. Note that $\mathbf{x}$ could participate in multiple violations along the same dimension. Throughout this paper, we will be only talking about negative influences of functions on the hypergrid, and thus will often refer to the above as just thresholded influence, and for brevity's sake we also don't use the superscript "−" in the notation below to denote the negative aspect.

Definition 1.1 (Thresholded Influence). *Fix $f : [n]^d \to \{0, 1\}$ and a dimension $i \in [d]$. Fix a point $\mathbf{x} \in [n]^d$. The thresholded influence of $\mathbf{x}$ along coordinate i is denoted $\Phi_f(\mathbf{x}; i)$, and has value 1 if there exists an i-aligned violation $(\mathbf{x}, \mathbf{y})$. The thresholded influence of $\mathbf{x}$ is $\Phi_f(\mathbf{x}) = \sum_{i=1}^d \Phi_f(\mathbf{x}; i)$.*

Note that the thresholded influence coincides with the hypercube directed influence when $n = 2$. Also note that for any $\mathbf{x}$, $\Phi_f(\mathbf{x}) \in \{0, 1, \ldots, d\}$ and is independent of n. We prove the following theorem, a directed Talagrand theorem for hypergrids, which generalizes the [33] result.

Theorem 1.2. *Let $f : [n]^d \to \{0, 1\}$ be ε-far from monotone.*

$$\mathbf{E}_{\mathbf{x}\in[n]^d}\left[\sqrt{\Phi_f(\mathbf{x})}\right] = \Omega\left(\frac{\varepsilon}{\log n}\right)$$

Robust isoperimetric theorems and monotonicity testing. For the application to monotonicity testing, as [33] showed, a significant strengthening of Theorem 1.2 is required. The weakness of Theorem 1.2, as stated, is that the same violation/influence is "double-counted" at both its endpoints. The LHS can significantly vary depending on whether we choose to only "count" influences at zero-valued or one-valued points, and this is true even on the hypercube. As a simple illustration, consider the function f that is 1 at the all zeros point and 0 everywhere else. Suppose we only count influences at one-valued points. Then the only vertex with any $I^-_f(\mathbf{x})$ is the all 0's point, and this value is d. Therefore, the Talagrand objective is $\frac{\sqrt{d}}{2^d}$. On the other hand, if we count influences at zero-valued points, then $I^-_f(\mathbf{x}) = 1$ for the d points $\mathbf{e}_1$ to $\mathbf{e}_d$, and 0 everywhere else. The Talagrand objective counted from zero-valued points is now much larger: $\frac{d}{2^d}$. Therefore, depending on how we count, one can potentially reduce the Talagrand objective, $\mathbf{E}_\mathbf{x}[\sqrt{I^-_f(\mathbf{x})}]$.

[33] define a general way of deciding which endpoint "pays" for a violated edge. Consider a *coloring*[1] $\chi : E \to \{0, 1\}$ of every edge $(\mathbf{x}, \mathbf{y}) \in E$ of the hypercube to either 0 or 1. Now, given a violated edge $(\mathbf{x}, \mathbf{y})$, we use this coloring to decide whose influence this edge contributes towards. More precisely, given this coloring χ, the *colored* directed influence $I^-_{f,\chi}(\mathbf{x})$ of $\mathbf{x}$ is defined as the number of violated edges $(\mathbf{x}, \mathbf{y})$ incident on $\mathbf{x}$ which have the same color as $f(\mathbf{x})$. Given a coloring, the *colorful* Talagrand objective equals the expected root colored directed influence. What [33] prove is that no matter what coloring χ one chooses, the Talagrand objective is still large, and in particular $\mathbf{E}_\mathbf{x}\left[\sqrt{I^-_{f,\chi}(\mathbf{x})}\right] = \Omega(\frac{\varepsilon_f}{\log d})$.

We define the robust/colorful generalizations of the thresholded negative influence on hypergrids. Consider the *fully augmented hypergrid*, where we put the edge $(\mathbf{x}, \mathbf{y})$ if $\mathbf{x}$ and $\mathbf{y}$ differ on only one coordinate. Let E be the set of edges in the fully augmented hypergrid.

Definition 1.3 (Colorful Thresholded Influence). *Fix $f : [n]^d \to \{0, 1\}$ and $\chi : E \to \{0, 1\}$. Fix a dimension $i \in [d]$ and a point $\mathbf{x} \in [n]^d$. The colorful thresholded negative influence of $\mathbf{x}$ along coordinate i is denoted $\Phi_{f,\chi}(\mathbf{x}; i)$, and has value 1 if there exists an i-aligned violation $(\mathbf{x}, \mathbf{y})$ such that $\chi(\mathbf{x}, \mathbf{y}) = f(\mathbf{x})$, and has value 0 otherwise. The colorful thresholded negative influence of $\mathbf{x}$ is $\Phi_{f,\chi}(\mathbf{x}) = \sum_{i=1}^d \Phi_{f,\chi}(\mathbf{x}; i)$.*

The main result of our paper is a robust directed Talagrand isoperimetry theorem for Boolean functions on the hypergrid. It is a strict generalization of the KMS Talagrand theorem for hypercubes.

Theorem 1.4. *Let $f : [n]^d \to \{0, 1\}$ be ε-far from monotone, and let $\chi : E \to \{0, 1\}$ be an arbitrary coloring of the edges of the*

[1] [33] considered the colorings to be red/blue, but we find the 0, 1-coloring more natural.

Table 1: Boolean hypercube isoperimetry results and their directed analogues. Pallavoor, Raskhodnikova, and Waingarten [37] removed the $\log d$-dependence in the directed Talagrand inequality.

Undirected Isoperimetry	Directed Isoperimetry
$I_f \geq \Omega(\mathrm{var}(f))$ (*Poincaré inequality, Folklore*)	$I_f^- \geq \Omega(\varepsilon_f)$ (*Goldreich et al.[29]*)
$I_f \cdot \Gamma_f \geq \Omega(\mathrm{var}(f)^2)$ (*Margulis [35]*)	$I_f^- \cdot \Gamma_f^- \geq \Omega(\varepsilon_f^2)$ (*Chakrabarty, Seshadhri [18]*)
$\mathbf{E}_{\mathbf{x}}\left[\sqrt{I_f(\mathbf{x})}\right] \geq \Omega(\mathrm{var}(f))$ (*Talagrand [41]*)	$\mathbf{E}_{\mathbf{x}}\left[\sqrt{I_f^-(\mathbf{x})}\right] = \Omega(\frac{\varepsilon_f}{\log d})$ (*Khot, Minzer, Safra [33]*)

augmented hypergrid.

$$\mathbf{E}_{\mathbf{x}\in[n]^d}\left[\sqrt{\Phi_{f,\chi}(\mathbf{x})}\right] = \Omega\left(\frac{\varepsilon}{\log n}\right)$$

As a consequence of this theorem, we can (up to log factors) resolve the question of non-adaptive monotonicity testing on hypergrids with constant n. We note that the best bound for any $n > 2$ was $\widetilde{O}(d^{5/6})$. Even for the simplest non-hypercube case of $n = 3$, it was open whether the optimal non-adaptive complexity of monotonicity testing is $\sqrt{d}$.

Theorem 1.5. *Consider Boolean functions over the hypergrid, $f : [n]^d \to \{0,1\}$. There is a one-sided, non-adaptive tester for monotonicity that makes $O(\varepsilon^{-2}n\sqrt{d}\log^5(nd))$ queries.*

The Importance of Being Robust. We briefly explain why the robust Talagrand version is central to the monotonicity testing application. All testers that have a $o(d)$-query complexity are versions of a *path tester*, which can be thought of as querying endpoints of a directed random walk in the hypercube. Consider a function f as the indicator for a set $\mathbf{1}_f$, where the violating edges form the "up-boundary" between $\mathbf{1}_f$ and its complement. To analyze the random walk, we would like to lower bound the probability that a random walk starts in $\mathbf{1}_f$, crosses over the boundary, and stays in $\overline{\mathbf{1}_f}$, that is, the set of 0's. To analyze this, one needs some structural properties in the graph induced by the boundary edges, which [33] express via their notion of a "good subgraph". In particular, one needs that there be a large number of edges, but also that they are regularly spread out among the vertices. It doesn't seem that the "uncolored" Talagrand versions (like Theorem 1.2) are strong enough to prove this regularity, but the robust version can "weed out" high-degree vertices via a definition of a suitable coloring function χ. In short, the robust version of the Talagrand-style isoperimetric theorem is much more expressive. Indeed, these style of robust results have found other applications in distribution testing [15] as well.

The Dependence on n. Given Theorem 1.5, it is natural to ask whether the dependence on n is necessary. Previous *domain reduction* theorems have shown that one can reduce n to $\mathrm{poly}(d)$ in a black box manner [9, 32]. The monotonicity tester based on the directed Margulis inequality for hypergrids has a logarithmic dependence on n [8]. Combining with domain reduction, we get a $\widetilde{O}(\mathrm{poly}(\varepsilon^{-1})d^{5/6})$-query tester. It is an outstanding open problem to remove the dependence on n from Theorem 1.5.

Independent Results by [13]. Independent of our work, Braverman, Kindler, Khot, and Minzer recently give an $\widetilde{O}(\varepsilon^{-2}\mathrm{poly}(n)\sqrt{d})$-query one-sided, non-adaptive Boolean monotonicity tester for hypergrids [13]. Their approach is different, and goes via monotone embeddings of hypergrids into hypercubes. This embedding (locally) constructs a hypercube function that has the same distance to monotonicity as the original hypergrid function. The hypercube function has dimension $\mathrm{poly}(n)d$, on which the KMS tester/result can be applied. Their final tester has query complexity $\widetilde{O}(\varepsilon^{-2}n^3\sqrt{d})$. They also use monotone embeddings to prove a directed isoperimetric inequality for hypergrid functions. They prove a version of Theorem 1.2, but the RHS has an n^3 loss.

1.2 Challenges

We explain the challenges faced in proving Theorem 1.4 and Theorem 1.5. The KMS proof of the directed Talagrand inequality for the hypercube is a tour-de-force [33], and there are many parts of their proof that do not generalize for $n > 2$. We begin by giving an overview of the KMS proof for the hypercube case.

For the time being, let us focus on the uncolored case. For convenience, let $T(f) = \mathbf{E}_{\mathbf{x}}[\sqrt{I_f^-(\mathbf{x})}]$ denote the hypercube directed Talagrand objective for a $f : \{0,1\}^d \to \{0,1\}$. To lower bound $T(f)$, [33] transform the function f to a function g using a sequence of what they call *split* operators. The ith split operator applied to f replaces the ith coordinate/dimension by two new coordinates $(i,+)$ and $(i,-)$. One way to think of the split operator is that takes the $((0,\mathbf{x}_{-i}),(1,\mathbf{x}_{-i}))$ edge and converts it into a square. (Here, $\mathbf{x}_{-i}$ denotes the collection of coordinates in $\mathbf{x}$ skipping $\mathbf{x}_i$.) The "bottom" and "top" corners of the square store the original values of the edge, while the "diagonal" corners store the min and max values (of the edge). The definition of this remarkably ingenious operator ensures that the split function is monotone in $(i,+)$ and anti-monotone in $(i,-)$. The final function $g : \{0,1\}^{2d} \to \{0,1\}$ obtained by splitting on all coordinates has the property that it is either monotone or anti-monotone on all coordinates. That is, g is unate (or pure, as [33] call them), and for such functions the directed Talagrand inequality can be proved via a short reduction to the undirected case.

The utility of the split operator comes from the main technical contribution of [33] (Section 3.4), where it is shown that splitting cannot increase the directed Talagrand objective. This is a "roll-your-sleeve-and-calculate" argument that follows a case-by-case analysis. So, we can lower bound $T(f) \geq T(g)$. Since g is unate, one can prove $T(g) = \Omega(\varepsilon_g)$ (the distance of g to monotonicity). But how does one handle ε_g, or g more generally? This is done by relating splitting to the classic *switch operator* in monotonicity testing, introduced in [29]. The switch operator for the ith coordinate can be thought of as modifying the edges along the i-dimension: for any i-edge violation $(\mathbf{x}, \mathbf{y})$, this operator switches the values, thereby fixing the violation. The switching operator has the remarkable property of never increasing monotonicity violations in other dimensions; hence, switching in all dimensions leads to a monotone function. [33] observe that the function g basically "embeds" disjoint variations of f, wherein each variation is obtained by performing a distinct sequence of switches on f. The function g contains all possible such variations of f, stored cleverly so that g is unate. One can then use properties of the switch operators to relate ε_g to ε_f. (The truth is more complicated; we will come back to this point later.)

Challenge #1, Splitting on Hypergrids? The biggest challenge in trying to generalize the [33] argument is to generalize the split operator. One natural starting point would be to consider the *sort* operator, defined in [24], which generalizes the switch operator: the sort operator in the ith coordinate sorts the function along all i-lines. But it is not at all clear how to split the ith coordinate into a set of coordinates that contains the information about the sort operator thereby leading to a pure/unate function. In short, sorting is a much more complicated operation than switching, and it is not clear how to succinctly encode this information using a single operator.

We address this challenge by a reorientation of the KMS proof. Instead of looking at operators on dimensions to understand effects of switching/sorting, we do this via what we call "tracker functions" which are n^d different Boolean functions tracking the changes in f. We discuss this more in Section 1.3.

Challenge #2, the Case Analysis for Decreasing Talagrand Objective. As mentioned earlier, the central calculation of KMS is in showing that splitting does not increase the directed Talagrand objective. This is related (not quite, but close enough) to showing that the switch operator does not increase the Talagrand objective. A statement like this is proven in KMS by case analysis; there are 4 cases, for the possible values a Boolean function takes on an edge. One immediately sees that such an approach cannot scale for general n, since the number of possible Boolean functions on a line is 2^n. Even with our new idea of tracking functions, we cannot escape this complexity of arguing how the Talagrand-style objective decreases upon a sorting operation, and a case-by-case analysis depending on the values of the function is infeasible.

We address this challenge by a connection to the theory of majorization. We show that the sort operator is (roughly) a majorizing operator on the vector of influences. The concavity of the square root function implies that sorting along lines cannot increase the Talagrand objective. More details are given in the next section.

Challenge #3, the Colorings. Even if we circumvented the above issues, the robust colored Talagrand objective brings a new set of issues. Roughly speaking, colorings decide which points "pay" for violations of the Talagrand objective, the switching/sorting operator move points around by changing values, and the high-level argument to prove $T(f)$ drops is showing that these violations "pay" for the moves. In the hypercube, a switch either changes the values on all the points of the edge or none of the points, and this binary nature makes the handling of colors in the KMS proof fairly easy, merely introducing a few extra cases in their argument. Sorting, on the other hand, can change an arbitrary set of points, and in particular, even in the case of $n = 3$, a point participating in a violation may not change value in a sort.

To address this challenge, as we apply the sort operators to obtain a handle on our function, we also need to *recolor* the edges such that we obtain the drop in the T-objective. Once again, the theory of majorization is the guide. This part of the proof is perhaps the most technical portion of our paper.

Other Challenges: The Telescoping Argument and Tester Analysis. The issues detailed here are not really conceptual challenges, but they do require some work to handle the richer hypergrid domain.

Recall that the KMS analysis proves the chain of inequalities, $T(f) \geq T(g) = \Omega(\varepsilon_g)$. Unfortunately, it can happen that $\varepsilon_g \ll \varepsilon_f$. In this case, KMS observe that one could redo the entire argument on random restrictions of f to half the coordinates. If the corresponding ε_g is still too small, then one restricts on one-fourth of the coordinates, so on and so forth. One can prove that somewhere along these $\log d$ restrictions, one must have $\varepsilon_g = \Omega(\varepsilon_f)$. Pallavoor, Raskhodnikova, and Waingarten [37] improve this analysis to remove a $\log d$ loss from the final bound. We face the same problems in our analysis, and have to adapt the analysis to our setting.

Finally, the tester analysis of KMS for the hypercube can be ported to the hypergrid path tester, with some suitable adaptations of their argument. It is convenient to think of the *fully augmented hypergrid*, where all pairs that lie along a line are connected by an edge. We can essentially view the hypergrid tester as sampling a random hypercube from the fully augmented hypergrid, and then performing a directed random walk on this hypercube. We can then piggyback on various tools from KMS for the hypercube tester, to bound the rejection probability of the path tester for hypergrids.

1.3 Main Ideas

We sketch some key ideas needed to prove Theorem 1.4 and address the challenges detailed earlier. We begin with a key conceptual contribution of this paper. Given a function $f : [n]^d \to \{0, 1\}$, we define a collection of Boolean functions on the hypercube called *tracker functions*. We will lower bound the directed Talagrand objective on the hypergrid by the undirected Talagrand objective on these tracker functions. Indeed, the inspiration of these tracker functions arose out of understanding the analysis in [33], in particular, the intermediate "g" function in their Section 4. As an homage, we also denote our tracker functions with the same Roman letter, even though it is different from their function.

1.3.1 *Tracker Functions $g_{\mathbf{x}}$ for all $\mathbf{x} \in [n]^d$.*

Let us begin with the sort operator discussed earlier. Without loss of generality, fix the ordering of coordinates in $[d]$ to be $(1, 2, \ldots, d)$. The operator $\mathtt{sort}_i$ for $i \in [d]$ sorts the function on every i-line. Given a subset $S \subseteq [d]$ of coordinates, the function $(S \circ f)$ is obtained by sorting f on the coordinates in S in that order.

Sorting along any dimension cannot increase the number of violations along any other dimension, and therefore upon sorting on all dimensions, the result is a monotone function [24]. Suppose f is ε-far from monotone. Clearly, the total number of points changed by sorting along all dimensions must be at least εn^d. While this is not obvious here, it will be useful to to *track* how the function value changes when we sort along a certain subset S of coordinates. The intuitive idea is: if the function value changes for most such partial sortings, then perhaps the function is far from being monotone. To this end, for every point $\mathbf{x} \in [n]^d$, we define a Boolean function $g_{\mathbf{x}} : 2^{[d]} \to \{0,1\}$ that tracks how the function value f changes as we apply the sort operator a subset S of the coordinates. It is best to think of the domain of $g_{\mathbf{x}}$ as subsets $S \subseteq [d]$.

Definition 1.6 (Tracker Functions $g_{\mathbf{x}}$). *Fix an $\mathbf{x} \in [n]^d$. The tracker function $g_{\mathbf{x}} : \{0,1\}^d \to \{0,1\}$ is defined as*

$$\forall S \subseteq [d],\ g_{\mathbf{x}}(S) := (S \circ f)(\mathbf{x})$$

We provide an illustration of this definition in Figure 1.
Note that when f is a monotone function, all the functions $g_{\mathbf{x}}$ are constants. Sorting does not change any values, so $g_{\mathbf{x}}(S)$ is always $f(\mathbf{x})$. On the other hand, if f is not monotone along dimension i, then there are points such that $g_{\mathbf{x}}(\{i\}) \neq f(\mathbf{x})$. Indeed, one would expect the typical variance of these $g_{\mathbf{x}}$ functions to be related to the distance to monotonicity of f (technically not true, but we come to this point later).

The tracker functions help us lower bound the (colorful) Talagrand objective for thresholded influence, in particular, the LHS in Theorem 1.4. Recall that the Talagrand objective is the expected square root of the colorful thresholded influences on the hypergrid function f. We lower bound this quantity by the expected Talagrand objective on the *undirected* (colorful, however) influence of the various $g_{\mathbf{x}}$ functions. Note that $g_{\mathbf{x}}$ functions are defined on hypercubes. So we reduce the robust directed Talagrand inequality on hypergrids to robust undirected Talagrand inequalities on hypercubes. This is the main technical contribution of our paper. Let us define the (colored) influences of these $g_{\mathbf{x}}$ functions.

Definition 1.7 (Influence of the Tracking Functions). *Fix a $\mathbf{x} \in [n]^d$ and consider the tracking function $g_{\mathbf{x}} : \{0,1\}^d \to \{0,1\}$. Fix a coordinate $j \in [d]$. The influence of $g_{\mathbf{x}}$ at a subset S along the jth coordinate is defined as*

$$I^{=j}_{g_{\mathbf{x}}}(S) = 1 \text{ iff } g_{\mathbf{x}}(S) \neq g_{\mathbf{x}}(S \oplus j) \text{ that is } (S \circ f)(\mathbf{x}) \neq (S \oplus j\ \circ f)(\mathbf{x})$$

In plain English, the influence of the jth coordinate at a subset S is 1 if the function value (the hypergrid function) changes when we include the dimension j to be sorted. Once again, note that the same sensitive edge $(S, S \oplus j)$ is contributing towards both $I^{=j}_{g_{\mathbf{x}}}(S)$ and $I^{=j}_{g_{\mathbf{x}}}(S \oplus j)$. We define a robust, colored version of these influences.

Definition 1.8 (Colorful Influence of the Tracking Functions). *Fix a $\mathbf{x} \in [n]^d$ and consider the tracking function $g_{\mathbf{x}} : \{0,1\}^d \to \{0,1\}$. Fix any arbitrary coloring $\xi_{\mathbf{x}} : E(2^{[d]}) \to \{0,1\}$ of the Boolean* hypercube. *Fix a coordinate $j \in [d]$. The influence of $g_{\mathbf{x}}$ at a subset S along the jth coordinate is defined as*

$$I^{=j}_{g_{\mathbf{x}},\xi_{\mathbf{x}}}(S) = 1 \text{ iff } g_{\mathbf{x}}(S) \neq g_{\mathbf{x}}(S \oplus j) \textbf{ and } g_{\mathbf{x}}(S) = \xi_{\mathbf{x}}(S, S \oplus j)$$

The colorful total influence at the point S in $g_{\mathbf{x}}$ is defined as

$$I_{g_{\mathbf{x}},\xi_{\mathbf{x}}}(S) := \sum_{j=1}^{d} I^{=j}_{g_{\mathbf{x}},\xi_{\mathbf{x}}}(S)$$

As before, for a sensitive edge $(S, S \oplus j)$ of $g_{\mathbf{x}}$, we count it towards the influence of the endpoint whose value equals the color $\xi_{\mathbf{x}}(S, S \oplus j)$. The main technical contribution of this paper is proving that for any function $f : [n]^d \to \{0,1\}$ and any arbitrary coloring $\chi : E \to \{0,1\}$ of the hypergrid edges, for every $\mathbf{x} \in [n]^d$ there exists a coloring $\xi_{\mathbf{x}} : E(2^{[d]}) \to \{0,1\}$ of the Boolean *hypercube* edges, such that

$$T_{\Phi_\chi}(f) := \mathbf{E}_{\mathbf{x} \in [n]^d}\left[\sqrt{\Phi_{f,\chi}(\mathbf{x})}\right] \gtrapprox \mathbf{E}_{\mathbf{x} \in [n]^d}\mathbf{E}_{S \subseteq [d]}\left[\sqrt{I_{g_{\mathbf{x}},\xi_{\mathbf{x}}}(S)}\right] \tag{H1}$$

We explain the $\approx$ in the above inequality in the next subsection.

Why is a statement like (H1) useful? Because the RHS terms are Talagrand objectives on colored influences on the usual undirected hypercube. Therefore, we can apply undirected Talagrand bounds known from [33] to get an upper bound on the variance.

Corollary 1.9 (Corollary of Theorem 1.8 in [33]). *Fix $f : [n]^d \to \{0,1\}$. Fix an $\mathbf{x} \in [n]^d$ and consider the tracking function $g_{\mathbf{x}} : \{0,1\}^d \to \{0,1\}$. Consider any* arbitrary *coloring $\xi_{\mathbf{x}} : E(2^{[d]}) \to \{0,1\}$ of the Boolean* hypercube. *Then, for every $\mathbf{x} \in [n]^d$, we have*

$$\mathbf{E}_{S \subseteq [d]}\left[\sqrt{I_{g_{\mathbf{x}},\xi_{\mathbf{x}}}(S)}\right] = \Omega(\mathrm{var}(g_{\mathbf{x}}))$$

The final piece of the puzzle connects $\mathrm{var}(g_{\mathbf{x}})$'s with the distance to monotonicity. Ideally, we would have liked to have a statement such as the following true.

$$\mathbf{E}_{\mathbf{x} \in [n]^d}\left[\mathrm{var}(g_{\mathbf{x}})\right] \approx \Omega(\varepsilon_f) \tag{H2}$$

We now see that (H1), Corollary 1.9, and (H2) together implies Theorem 1.4 (indeed without the $\log n$).

1.3.2 High Level Description of our Approaches.

Addressing the $\approx$ in (H1) via Semisorting. As stated, we do not know if (H1) is true. However, we establish (H1) for *semisorted* functions $f : [n]^d \to \{0,1\}$. A function f is semisorted if on any line ℓ, the restriction of the function on the first half is sorted and the restriction on the second half is sorted. This may seem like a simple subclass of functions, but note that all functions on the Boolean hypercube ($n = 2$) are vacuously semisorted. Thus, proving Theorem 1.4 on semi-sorted functions is already a generalization of the [33] result.

We reduce Theorem 1.4 on general functions to the same bound for semisorted functions. Consider semisorting f, which means we sort f on each half of every line. Suppose the Talagrand objective did not increase *and* the distance to monotonicity did not decrease. Then Theorem 1.4 on the semisorted version of f implies Theorem 1.4 on f. What we can prove is that: given the semisorted function, one can find a *recoloring* of the hypergrid edges such that the Talagrand objective doesn't increase. We comment on our techniques to prove such a statement in a later paragraph.

Although semisorting can't increase the Talagrand objective, it can clearly reduce the distance to monotonicity. However, a relatively simple inductive argument proves Theorem 1.4 with a $\log n$

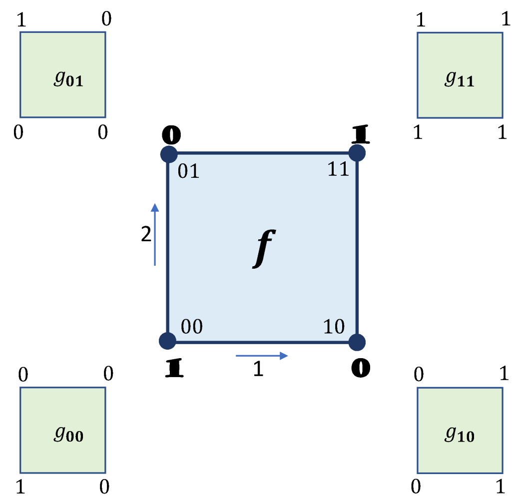

Figure 1: The blue function $f : [n]^d \to \{0,1\}$ is defined in the middle using bold, gothic characters. We have $d = 2$ and $n = 2$. For each of the 4 points of this square, we have four different $g_{\mathbf{x}} : \{0,1\}^2 \to \{0,1\}$ and they are described in the four green squares. For any $S \subseteq \{1,2\}$, if we focus on the corresponding corners of the four squares, then we get the function $(S \circ f)$. For instance, if $S = \{2\}$, then if we focus on the top left corners, then starting from g_{00} and moving clockwise we get $(0,1,1,0)$. These will precisely the function f (read clockwise from 00) after we sort along dimension 2.

loss. Any function can be turned into a completely sorted (aka monotone) function by performing "$\log n$ semisorting steps" at varying scales. In each scale, we consider many disjoint small hypergrids, and convert a semisorted function defined over a small hypergrid to another semisorted function over a hypergrid of double the size (the next scale). In one of these scales, we will find a semisorted function that has $\Omega(\varepsilon/\log n)$ distance from its sorted version. One can average Theorem 1.4 over all the small hypergrids at this scale to bound the Talagrand objective of the whole function by $\Omega(\varepsilon/\log n)$. This is the step where we incur the $\log n$-factor loss.

The real work happens in proving (H1) for semisorted functions.

Approach to Proving (H1) *for Semisorted Functions.* Recall, we have a fixed adversarial coloring $\chi : E \to \{0,1\}$. The proof follows a "hybrid argument" where we define a potential that is modified over $d+1$ rounds. At the beginning of round 0 it takes the value $\mathbf{E}_{\mathbf{x}\in[n]^d}[\sqrt{\Phi_{f,\chi}(\mathbf{x})}]$ which is the LHS of (H1). At the end of round d it takes the value $\mathbf{E}_{\mathbf{x}\in[n]^d}\mathbf{E}_{S\subseteq[d]}[\sqrt{I_{g_{\mathbf{x}},\xi_{\mathbf{x}}}(S)}]$ which is the RHS of (H1). The proof follows by showing that the potential decreases in each round.

Let us describe the potential. Let us first write this without any reference to the colorings (so no χ's and $\xi_{\mathbf{x}}$'s), and then subsequently address the colorings. At stage i, fix a subset $S \subseteq [i]$. Define

$$R_i(S) := \mathbf{E}_{\mathbf{x}\in[n]^d}\left[\sqrt{\sum_{j=1}^{i} I_{g_{\mathbf{x}}}^{=j}(S) + \sum_{j=i+1}^{d} \Phi_{S\circ f}(\mathbf{x};j)}\right] \qquad \text{(Hybrid)}$$

We remind the reader that $S\circ f$ is the function f after the dimensions corresponding to $i \in S$ have been sorted. Thus, $R_i(S)$ is a "hybrid" Talagrand objective, with two different kinds of influences being summed. Consider point $\mathbf{x} \in [n]^d$. On the first i coordinates, we sum the undirected influence (along these coordinates) of S on the function $g_{\mathbf{x}}$. On the coordinates $i+1$ to d, we sum to directed influence along these coordinates in the function $S\circ f$. The potential is $\Lambda_i := \mathbf{E}_{S\subseteq[i]}[R_i(S)]$.

To make some sense of this, consider the extreme cases of $i = 0$ and $i = d$. When $i = 0$, we only have the second $\Phi_{S\circ f}$ term. Furthermore, S is empty since $S \subseteq [i]$. So Λ_0 is precisely the original directed Talagrand objective, the LHS of (H1). When $i = d$, we only have the $I_{g_{\mathbf{x}}}^{=j}$ terms. Taking expectation over $S \subseteq [d]$ to get Λ_d, we deduce that Λ_d is the RHS of (H1).

We will prove $\Lambda_{i-1} \geq \Lambda_i$ for all $1 \leq i \leq d$. To choose a uar set in $[i]$, we can choose a uar subset of $[i-1]$ and then add i with 1/2 probability. Hence, $\Lambda_i = (\mathbf{E}_{S\subseteq[i-1]}[R_i(S) + R_i(S+i)])/2$, while $\Lambda_{i-1} = \mathbf{E}_{S\subseteq[i-1]}[R_{i-1}(S)]$. So, if we prove that $R_{i-1}(S)$ is at least both $R_i(S)$ and $R_i(S+i)$, then $\Lambda_{i-1} \geq \Lambda_i$. The bulk of the technical work in this paper is involved in proving these two inequalities, so let us spend a little time explaining what proving this entails.

Let's take the inequality $R_{i-1}(S) \geq R_i(S)$. Refer again to (Hybrid). When we go from $R_{i-1}(S)$ to $R_i(S)$, under the square root, the term $\Phi_{S\circ f}(\mathbf{x};i)$ is replaced by $I_{g_{\mathbf{x}}}^{=i}(S)$. To remind the reader, the former term is the indicator of whether $\mathbf{x}$ participates in a i-violation after the coordinates in $S \subseteq [i-1]$ have been sorted. The latter term is whether $g_{\mathbf{x}}(S+i)$ equals $g_{\mathbf{x}}(S)$, that is, whether the (hypergrid) function value at $\mathbf{x}$ changes between sorting on coordinates in S and $S+i$. Just by parsing the definitions, one can observe that $\Phi_{S\circ f}(\mathbf{x};i) \geq I_{g_{\mathbf{x}}}^{=i}(S)$; if a point is modified on sorting in the i-coordinate, then it must be participating in some i-violation (note that vice-versa may not be true and thus we have an inequality and not an equality). The quantity under the square-root *point-wise* dominates (ie, for every $\mathbf{x}$) when we move from $R_{i-1}(S)$ to $R_i(S)$. Thus, $R_{i-1}(S) \geq R_i(S)$.

The other inequality $R_{i-1}(S) \geq R_i(S+i)$, however, is much trickier to establish. In $R_i(S+i)$, the second summation under the square-root, the Φ terms, are actually on a *different* function. The $\Phi_{S\circ f}(\mathbf{x};j)$ terms in $R_{i-1}(S)$ are the thresholded influences of the function after sorting on coordinates in S. But in $R_i(S+i)$, these terms are $\Phi_{(S+i)\circ f}(\mathbf{x};j)$, the thresholded influences of $\mathbf{x}$ for the function after sorting on $S+i$. Although, it is true that sorting on more coordinates cannot increase the total number of violations along any dimension, this fact is *not* true point-wise. So, a point-wise argument as in the previous inequality is not possible.

The argument for this inequality proceeds *line-by-line*. One fixes an i-line ℓ and considers the vector of "hybrid function" values on this line. We then consider this vector when moving from $R_{i-1}(S)$ to $R_i(S+i)$, and we need to show that the *sum of square roots* can get only smaller. This is where one of our key insights comes in: the theory of majorization can be used to assert these bounds. Roughly speaking, a vector $\mathbf{a}$ (weakly) majorizes a vector $\mathbf{b}$ if the sum of the k-largest coordinates of $\mathbf{a}$ dominates the sum of the k-largest coordinates of $\mathbf{b}$, for every k. A less balanced vector majorizes a more balanced vector. If the ℓ_1-norms of these vectors are the same, then the sum of square roots of the entries of $\mathbf{a}$ is at most the sum of square roots of that of $\mathbf{b}$. This follows from concavity of the square-root function.

Our overarching mantra throughout this paper is this: whenever we perform an operation and the hybrid-influence-vector induced by a line changes, the new vector majorizes the old vector. Specifically, these vectors are generated by look at the terms of $R_{i-1}(S)$ and $R_i(S+i)$ restricted to i-lines.

To prove this vector-after-operation majorizes vector-before-operation, we need some structural assumptions on the function. Otherwise, it's not hard to construct examples where this just fails. The structure we need is precisely the *semisortedness* of f. When a function is semisorted, the majorization argument goes through. At a high level, when f is semisorted, the vector of influences (along a line) satisfy various monotonicity properties. In particular, when we (fully) sort on some coordinate i, we can show the points losing violations had low violations to begin with. That is, the vector of violations becomes less balanced, and the majorization follows.

The above discussion disregarded the colors. With colors, the situation is noticeably more difficult. Although the function f is assumed to be semisorted, the coloring $\chi : E \longrightarrow \{0,1\}$ is adversarial. So even though the vector of influences may have monotonicity properties, the colored influences may not have this structure. So a point with high influence could have much lower colored influence. Note that the sort operator is insensitive to the coloring. So the majorization argument discussed above might not hold when looking at colored influences.

With colors, (Hybrid) is replaced by the "colored version" of the same. To carry out the majorization argument, we need to construct a family of colorings $\xi_{\mathbf{x}}$ on the n^d different hypercubes. We also need 2^d many different auxiliary colorings χ_S of the hypergrid, constructed after every sort operation. The argument is highly technical. But all colorings are chosen to follow our mantra: vector after operation should majorize vector before operation. The same principle is also used to prove the previous claim that semisorting an interval can only decrease the Talagrand objective, after a recoloring.

Addressing the ≈ in (H2) via Random Sorts. To finally complete the argument, we need (H2) that relates the average variance of the $g_{\mathbf{x}}$ functions to the distance to monotonicity of f. As discussed earlier, (H2) is false, even for the case of hypercubes. Nevertheless, one can use (H1) and Corollary 1.9 to prove a lower bound on $T_{\Phi_\chi}(f)$ with respect to ε_f. This is the telescoping argument of KMS, refined in [37]. We describe the main ideas below. The first observation is that $\mathbf{E}_{\mathbf{x}\in[n]^d}[\mathrm{var}(g_{\mathbf{x}})]$ is roughly $\mathbf{E}_S[\Delta(S\circ f, \overline{S}\circ f]$ where S is a uniform random subset of coordinates. The distance to monotonicity ε_f is approximated by $\Delta\left(f, S\circ\overline{S}\circ f\right)$ which, by the triangle inequality, is at most $\Delta(f, S\circ f) + \Delta(S\circ f, \overline{S}\circ f)$. Thus, we get a relation between ε_f, the expected $\mathrm{var}(g_{\mathbf{x}})$, and the distance between f and a "random sort" of f. Therefore, if (H2) is not true, then a random sort of f must be still far from being monotone, and then one can repeat the whole argument on just this random sort itself. In one of these $\log d$ "repetitions", the (H2) must be true since in the end we get a monotone function (which can't be far from being monotone). And this suffices to establish Theorem 1.4. We re-assert that the main ideas are already present in [33, 37]. However, we require a more general presentation to make things work for hypergrids.

1.4 Full Version

The implementation and discussion of the above ideas take some space to describe and won't fit in the space provided for the extended abstract. We refer the reader to the full version of this paper [7] for the detailed explanation of the ideas above.

1.5 Related Work

Monotonicity testing has seen much activity since its introduction around 25 years ago [1–6, 8–12, 14, 16–18, 21–34, 38–40].

We have already covered much of the previous work on Boolean monotonicity testing over the hypercube, but give a short recap. For convenience of presentation, in some results, we subsume ε-dependencies using the notation O_ε. The problem was introduced by Goldreich et al. [29] and Raskhodnikova [38], who described an $O(d/\varepsilon)$-query tester. Chakrabarty and Seshadhri [18] achieved the first sublinear in dimension query complexity of $\widetilde{O}_\varepsilon(d^{7/8})$ using directed isoperimetric inequalities. Chen, Servedio, and Tan [22] improved the analysis to $\widetilde{O}_\varepsilon(d^{5/6})$ queries. Fischer et al. [28] had first shown an $\Omega(\sqrt{d})$-query lower bound for non-adaptive, one-sided testers, by a short and neat construction. The non-adaptive, two-sided $\widetilde{\Omega}(\sqrt{d})$ lower bound is much harder to attain, and was done by Chen, Waingarten, and Xie [23], improving on the $\Omega(d^{1/2-c})$ bound from [21], which itself improved on the $\widetilde{\Omega}(d^{1/5})$ bound of [22]. [33] gave an $\widetilde{O}_\varepsilon(\sqrt{d})$-query tester, via the robust directed Talagrand inequality.

While this resolves the non-adaptive testing complexity (up to $\mathrm{poly}(\varepsilon^{-1}\log d)$ factors) for the hypercube, the adaptive complexity is still open. The first polynomial lower bound of $\widetilde{\Omega}(d^{1/4})$ for adaptive testers was given by Belovs and Blais [3] and has since been improved to $\widetilde{\Omega}(d^{1/3})$ by Chen, Waingarten, and Xie [23]. Chakrabarty and Seshadhri [20] gave an adaptive $\widetilde{O}_\varepsilon(I_f)$-query tester, thereby showing that adaptivity can help in monotonicity testing. The $d^{1/3}$ vs $\sqrt{d}$ query complexity gap is an outstanding open question in property testing.

There has been work on approximating the distance to monotonicity in $\mathrm{poly}(d, \varepsilon_f)$-queries. Fattal and Ron [26] gave the first non-trivial result of an $O(d)$-approximation, and Pallavoor, Raskhodnikova, and Waingarten [37] gave a non-adaptive $O(\sqrt{d})$-approximation (all running in $\mathrm{poly}(d, \varepsilon_f)$ time). They also show that non-adaptive $\mathrm{poly}(d)$-time algorithms cannot beat this approximation factor.

The above discussion is only for Boolean valued functions on the hypercube. For arbitrary ranges, the original results on monotonicity testing gave an $O(d^2/\varepsilon)$-query tester [24, 29]. Chakrabarty and Seshadhri [17] proved that $O(d/\varepsilon)$-queries suffices for monotonicity testing, matching the lower bound of $\Omega(d)$ of Blais, Brody, and Matulef [11]. The latter bound holds even when the range size is $\sqrt{d}$. A recent result of Black, Kalemaj, and Raskhodnikova showed a smooth trade-off between the $\sqrt{d}$ bound for the Boolean range and the d bound for arbitrary ranges ([10]). Consider functions $f : \{0,1\}^d \to [r]$. They gave a tester with query complexity $\widetilde{O}_\varepsilon(r\sqrt{d})$, achieved by extending the directed Talagrand inequality to arbitrary range functions. Their techniques are quite black-box and carry over to other posets. We note that their techniques can also be ported to our setting, so we can get an $\widetilde{O}_\varepsilon(rn\sqrt{d})$-query monotonicity tester for functions $f : [n]^d \to [r]$.

We now discuss monotonicity testing on the hypergrid. We discuss more about the ε-dependencies, since there have been interesting relevant discoveries. As mentioned above, [24] gives a non-adaptive, one-sided $O((d/\varepsilon)\log^2(d/\varepsilon))$-query tester. This was improved to $O((d/\varepsilon)\log(d/\varepsilon))$ by Berman, Raskhodnikova, and Yaroslavtsev [4]. This paper also showed an interesting adaptivity gap for 2D functions $f : [n]^2 \to \{0,1\}$: there exists an $O(1/\varepsilon)$-query adaptive tester (in fact, for any constant dimension d), and they show an $\Omega(\log(1/\varepsilon)/\varepsilon)$ lower bound for non-adaptive testers. Previous work [8] by the authors gave an $\widetilde{O}_\varepsilon(d^{5/6}\log n)$-query tester, by proving a directed Margulis inequality on augmented hypergrids. Another work [9] of the authors, and subsequently a work [32] by Harms and Yoshida, designed domain reduction methods for monotonicity testing, showing how n can be reduced to $\mathrm{poly}(\varepsilon^{-1}, d)$ by subsampling the hypergrid.

For hypergrid functions with arbitrary ranges, the optimal complexity is known to be $\Theta(d\log n)$ [17, 19]. When the range is $[r]$ and $d = 1$, one can get $O(\log r)$-query testers [36].

ACKNOWLEDGMENTS

Hadley Black was supported by NSF award AF:Small 2007682, NSF Award: Collaborative Research Encore 2217033. Deeparnab Chakrabarty was supported by NSF-CAREER award 2041920. C. Seshadhri was supported by NSF DMS-2023495, CCF-1740850, 1839317, 1813165, 1908384, 1909790, and ARO Award W911NF1910294.

REFERENCES

[1] Nir Ailon and Bernard Chazelle. 2006. Information Theory in Property Testing and Monotonicity Testing in Higher Dimension. *Information and Computation* 204, 11 (2006), 1704–1717.

[2] Nir Ailon, Bernard Chazelle, Seshadhri Comandur, and Ding Liu. 2007. Estimating the distance to a monotone function. *Random Structures Algorithms* 31, 3 (2007), 371–383. Prelim. version in Proc., RANDOM 2004.

[3] Aleksandrs Belovs and Eric Blais. 2021. A Polynomial Lower Bound for Testing Monotonicity. *SIAM Journal on Computing (SICOMP)* 50, 3 (2021), 406–433. Prelim. version in Proc., STOC 2016.

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

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

[6] Arnab Bhattacharyya, Elena Grigorescu, Madhav Jha, Kyoming Jung, Sofya Raskhodnikova, and David Woodruff. 2012. Lower Bounds for Local Monotonicity Reconstruction from Transitive-Closure Spanners. *SIAM Journal on Discrete Mathematics (SIDMA)* 26, 2 (2012), 618–646. Prelim. version in Proc., RANDOM 2010.

[7] Hadley Blach, Deeparnab Chakrabarty, and C. Seshadhri. 2022. Directed Isoperimetric Theorems for Boolean Functions on the Hypergrid and an $\widetilde{O}(n\sqrt{d})$ Monotonicity Tester. *Electronic Colloquium on Computational Complexity (ECCC)* 162 (2022). https://eccc.weizmann.ac.il/report/2022/162/

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

[9] Hadley Black, Deeparnab Chakrabarty, and C. Seshadhri. 2020. Domain Reduction: A $o(d)$ Tester for Boolean Functions in d-Dimensions. In *Proceedings, ACM-SIAM Symposium on Discrete Algorithms (SODA)*.

[10] Hadley Black, Iden Kalemaj, and Sofya Raskhodnikova. 2020. Isoperimetric Inequalities for Real-Valued Functions with Applications to Monotonicity Testing. *arXiv* abs/2011.09441 (2020).

[11] Eric Blais, Joshua Brody, and Kevin Matulef. 2012. Property testing lower bounds via communication complexity. *Computational Complexity* 21, 2 (2012), 311–358. Prelim. version in Proc., CCC 2011.

[12] Eric Blais, Sofya Raskhodnikova, and Grigory Yaroslavtsev. 2014. Lower Bounds for Testing Properties of Functions over Hypergrid Domains. In *Proceedings, IEEE Conference on Computational Complexity (CCC)*.

[13] Mark Braverman, Subhash Khot, Guy Kindler, and Dor Minzer. 2023. Improved Monotonicity Testers via Hypercube Embeddings. In *Proceedings, Innovations in Theoretical Computer Science (ITCS)*. 25:1–25:24.

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

[15] Clément L. Canonne, Xi Chen, Gautam Kamath, Amit Levi, and Erik Waingarten. 2021. Random Restrictions of High Dimensional Distributions and Uniformity Testing with Subcube Conditioning. In *Proceedings, ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 321–336.

[16] Deeparnab Chakrabarty, Kashyap Dixit, Madhav Jha, and C Seshadhri. 2017. Property testing on product distributions: Optimal testers for bounded derivative properties. *ACM Trans. on Algorithms (TALG)* 13, 2 (2017), 1–30. Prelim. version in Proc., SODA 2015.

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

[18] Deeparnab Chakrabarty and C. Seshadhri. 2014. An $o(n)$ monotonicity tester for Boolean functions over the hypercube. *SIAM Journal on Computing (SICOMP)* 45, 2 (2014), 461–472. Prelim. version in Proc., STOC 2013.

[19] Deeparnab Chakrabarty and C. Seshadhri. 2014. An Optimal Lower Bound for Monotonicity Testing over Hypergrids. *Theory of Computing* 10 (2014), 453–464. https://doi.org/10.4086/toc.2014.v010a017 Prelim. version in Proc., RANDOM 2013.

[20] Deeparnab Chakrabarty and C. Seshadhri. 2019. Adaptive Boolean Monotonicity Testing in Total Influence Time. In *Proceedings, Innovations in Theoretical Computer Science (ITCS)*. 20:1–20:7.

[21] Xi Chen, Anindya De, Rocco A. Servedio, and Li-Yang Tan. 2015. Boolean Function Monotonicity Testing Requires (Almost) $O(n^{1/2})$ Non-adaptive Queries. In *Proceedings, ACM Symposium on Theory of Computing (STOC)*.

[22] Xi Chen, Rocco A. Servedio, and Li-Yang. Tan. 2014. New Algorithms and Lower Bounds for Monotonicity Testing. In *Proceedings, IEEE Symposium on Foundations of Computer Science (FOCS)*.

[23] Xi Chen, Erik Waingarten, and Jinyu Xie. 2017. Beyond Talagrand: New Lower Bounds for Testing Monotonicity and Unateness. In *Proceedings, ACM Symposium on Theory of Computing (STOC)*.

[24] Yevgeny Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. 1999. Improved Testing Algorithms for Monotonicity. *Proceedings, International Workshop on Randomization and Computation (RANDOM)* (1999).

[25] Funda Ergun, Sampath Kannan, Ravi Kumar, Ronitt Rubinfeld, and Mahesh Viswanathan. 2000. Spot-checkers. *J. Comput. System Sci.* 60, 3 (2000), 717–751. Prelim. version in Proc., STOC 1998.

[26] Shahar Fattal and Dana Ron. 2010. Approximating the Distance to Monotonicity in High Dimensions. *ACM Trans. on Algorithms (TALG)* 6, 3 (2010).
[27] Eldar Fischer. 2004. On the strength of comparisons in property testing. *Information and Computation* 189, 1 (2004), 107–116.
[28] Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, and Ronitt Rubinfeld. 2002. Monotonicity Testing over General Poset Domains. *Proceedings, ACM Symposium on Theory of Computing (STOC)* (2002).
[29] Oded Goldreich, Shafi Goldwasser, Eric Lehman, Dana Ron, and Alex Samordinsky. 2000. Testing Monotonicity. *Combinatorica* 20 (2000), 301–337. Prelim. version in Proc., FOCS 1998.
[30] Shirley Halevy and Eyal Kushilevitz. 2003. Distribution-free Property Testing. *Proceedings, International Workshop on Randomization and Computation (RANDOM)* (2003).
[31] Shirley Halevy and Eyal Kushilevitz. 2008. Testing monotonicity over graph products. *Random Structures Algorithms* 33, 1 (2008), 44–67. Prelim. version in Proc., ICALP 2004.
[32] Nathaniel Harms and Yuichi Yoshida. 2022. Downsampling for Testing and Learning in Product Distributions. In *Proceedings, International Colloquium on Automata, Languages and Programming (ICALP)*, Vol. 229. 71:1–71:19.
[33] Subhash Khot, Dor Minzer, and Muli Safra. 2018. On monotonicity testing and Boolean isoperimetric-type theorems. *SIAM J. Comput.* 47, 6 (2018), 2238–2276. Prelim. version in Proc., FOCS 2015.
[34] Eric Lehman and Dana Ron. 2001. On disjoint chains of subsets. *Journal of Combinatorial Theory, Series A* 94, 2 (2001), 399–404.
[35] Grigory A. Margulis. 1974. Probabilistic Characteristics of Graphs with Large Connectivity. *Problemy Peredachi Informatsii* 10, 2 (1974), 101–108.
[36] Ramesh Krishnan S. Pallavoor, Sofya Raskhodnikova, and Nithin Varma. 2018. Parameterized Property Testing of Functions. *ACM Trans. Comput. Theory* 9, 4 (2018), 17:1–17:19. Prelim. version in Proc., ITCS 2017.
[37] Ramesh Krishnan S. Pallavoor, Sofya Raskhodnikova, and Erik Waingarten. 2022. Approximating the distance to monotonicity of Boolean functions. *Random Structures Algorithms* 60, 2 (2022), 233–260. Prelim. version in Proc., SODA 2020.
[38] Sofya Raskhodnikova. 1999. Monotonicity Testing. *Masters Thesis, MIT* (1999).
[39] Dana Ron, Ronitt Rubinfeld, Muli Safra, Alex Samorodnitsky, and Omri Weinstein. 2012. Approximating the Influence of Monotone Boolean Functions in $O(\sqrt{n})$ Query Complexity. *ACM Trans. Comput. Theory* 4, 4 (2012), 11:1–11:12. Prelim. version in Proc., RANDOM 2011.
[40] Michael E. Saks and C. Seshadhri. 2008. Parallel monotonicity reconstruction. In *Proceedings, ACM-SIAM Symposium on Discrete Algorithms (SODA).*
[41] Michel Talagrand. 1993. Isoperimetry, logarithmic Sobolev inequalities on the discrete cube, and Margulis' graph connectivity theorem. *Geom. Func. Anal.* 3, 3 (1993), 295–314.

Received 2022-11-07; accepted 2023-02-06