Testing Lipschitz Functions on Hypergrid Domains*

Pranjal Awasthi[†]
Carnegie Mellon University

Madhav Jha[‡]
Pennsylvania State University

Marco Molinaro[§]
Carnegie Mellon University

Sofya Raskhodnikova[‡]
Pennsylvania State University

Abstract

A function $f(x_1, ..., x_d)$, where each input is an integer from 1 to n and output is a real number, is Lipschitz if changing one of the inputs by 1 changes the output by at most 1. In other words, Lipschitz functions are not very sensitive to small changes in the input.

Our main result is an efficient tester for the Lipschitz property of functions $f : [n]^d \to \delta\mathbb{Z}$, where $\delta \in (0, 1]$ and $\delta\mathbb{Z}$ is the set of integer multiples of δ. A property tester is given an oracle access to a function f and a proximity parameter ϵ, and it has to distinguish, with high probability, functions that have the property from functions that differ on at least an ϵ fraction of values from every function with the property. The Lipschitz property was first studied by Jha and Raskhodnikova (FOCS'11) who motivated it by applications to data privacy and program verification. They presented efficient testers for the Lipschitz property of functions on the domains $\{0, 1\}^d$ and $[n]$. Our tester for functions on the more general domain $[n]^d$ runs in time $O(d^{1.5} n \log n)$ for constant ϵ and δ.

The main tool in the analysis of our tester is a smoothing procedure that makes a function Lipschitz by modifying it at a few points. Its analysis is already nontrivial for the 1-dimensional version, which we call Bubble Smooth, in analogy to Bubble Sort. In one step, Bubble Smooth modifies two values that violate the Lipschitz property, namely, differ by more than 1, by transferring δ units from the larger to the smaller. We define a *transfer graph* to keep track of the transfers, and use it to show that the ℓ_1 distance between f and BubbleSmooth(f) is at most twice the ℓ_1 distance from f to the nearest Lipschitz function. Bubble Smooth has several other important properties that allow us to obtain a *dimension reduction*, i.e., a reduction from testing functions on multidimensional domains to testing functions on one-dimensional domains. Our dimension reduction incurs only a small multiplicative overhead in the running time and thus avoids the exponential dependence on the dimension.

1 Introduction

Property testing aims to understand how much information is needed to decide (approximately) whether an object has a property. A *property tester* [RS96, GGR98] is given oracle access to an object and a proximity parameter $\epsilon \in (0, 1)$. If the object has the desired property then the tester *accepts* it with probability[1]

*A preliminary version of this paper appeared in RANDOM'12 [AJMR12].

†P.A. is supported by NSF grant CCF-1116892.

‡M.J. and S.R. are supported by NSF CAREER grant CCF-0845701 and NSF grant CCF-0729171.

§M.M. is supported by NSF grant CMMI-1024554.

[1] In the definition of a property tester, we set the success probability to 2/3. By standard arguments, it can be amplified to $1 - \gamma$ for any $\gamma \in (0, 1/3)$ with $\Theta(\log 1/\gamma)$ overhead in the complexity of the tester.

at least $2/3$; if the object is ϵ-far from having the desired property then the tester *rejects* it with probability at least $2/3$. Specifically, for properties of functions, ϵ-far means that a given function differs on at least an ϵ fraction of the domain points from any function with the property[2]. Properties of many different types of objects have been studied, including graphs, metrics spaces, images and functions (see, e.g., [Ron09, CS10, Gol11, RS11] for recent surveys.) Specifically, for non-Boolean functions, sublinear-time testers of linearity [BLR93], being a low-degree polynomial [RS96, KR06, JPRZ09], monotonicity [EKK+00, GGL+00, DGL+99, BRW05, FLN+02, HK08, ACCL07, BGJ+12], submodularity [SV14] and the Lipschitz property [JR13] have been proposed.

We present efficient testers for the Lipschitz property of functions[3] $f : [n]^d \to \delta\mathbb{Z}$, where $\delta \in (0, 1]$ and $\delta\mathbb{Z}$ is the set of integer multiples of δ. A function f is *c-Lipschitz* (with respect to the ℓ_1 metric on the domain) if $|f(x) - f(y)| \leq c \cdot |x - y|_1$ for all x, y in the domain. Points in the domain $[n]^d$ can be thought of as vertices of a d-dimensional *hypergrid*, where every pair of points at ℓ_1 distance 1 is connected by an edge. Each edge (x, y) imposes a constraint $|f(x) - f(y)| \leq c$ and a function f is c-Lipschitz iff every edge constraint is satisfied. We say a function is *Lipschitz* if it is 1-Lipschitz. (Note that rescaling by a factor of $1/c$ converts a c-Lipschitz function into a Lipschitz function.)

Testing of the Lipschitz property was first studied by Jha and Raskhodnikova [JR13] who motivated it by applications to data privacy and program verification. They presented testers for the Lipschitz property of functions on the domains $\{0, 1\}^d$ (the *hypercube*) and $[n]$ (the *line*) that run in time $O(d^2/(\delta\epsilon))$ and $O(\log n/\epsilon)$, respectively. Even though the applications in [JR13] are most convincing for functions on general hypergrid domains (in one of their applications, for instance, a point in $[n]^d$ represents a histogram of a private database), no nontrivial tester for functions on such general domains was known prior to this work.

1.1 Our Results

We present two efficient testers of the Lipschitz property of functions of the form $f : [n]^d \to \delta\mathbb{Z}$ with running time polynomial in d, n and $(\delta\epsilon)^{-1}$. Our testers are faster for functions whose image has small diameter.

Definition 1.1 (Image diameter)**.** *Given a function* $f : [n]^d \to \mathbb{R}$*, its* image diameter *is*

$$\mathrm{ImgD}(f) = \max_{x \in [n]^d} f(x) - \min_{y \in [n]^d} f(y).$$

Observe that a Lipschitz function on $[n]^d$ must have image diameter at most nd. However, image diameter can be arbitrarily large for a non-Lipschitz function.

Our testers are *nonadaptive*, that is, their queries do not depend on answers to previous queries. The first tester has *1-sided error*, that is, it always accepts Lipschitz functions. The second tester is faster (when $\sqrt{d} \gg \log(1/\epsilon)$ and $\mathrm{ImgD}(f)$ is large), but has 2-sided error, that is, it can err with probability at most 1/3 on both positive and negative instances.

[2] The distance measure on the space of functions used here and in previous work on property testing is Hamming distance. After the conference version of this article was published, Berman et al. [BRY14] proposed to use ℓ_p distances in property testing of real-valued functions. We note that even though testing with respect to ℓ_p distances is very useful, testing with respect to Hamming distance is still important. In particular, the applications of Lipschitz testers to privacy developed in [JR13, DJRT13] require the use of Hamming distance.

[3] The set $\{1, \ldots, n\}$ is denoted by $[n]$.

Theorem 1.1 (Lipschitz testers)**.** *For*[4] *all $\delta \in (0,1], \epsilon \in (0,1)$, the Lipschitz property of functions $f : [n]^d \to \delta\mathbb{Z}$ can be tested nonadaptively with the following time complexity:*

(1) in $O\left(\frac{d}{\delta\epsilon} \cdot \min\{\mathrm{ImgD}(f), nd\} \cdot \log \min\{\mathrm{ImgD}(f), n\}\right)$ time with 1-sided error.

(2) in $O\left(\frac{d}{\delta\epsilon} \cdot \min\left\{\mathrm{ImgD}(f), n\sqrt{d\log(1/\epsilon)}\right\} \cdot \log \min\{\mathrm{ImgD}(f), n\}\right)$ time with 2-sided error.

If the image diameter, δ and ϵ are constant, then both testers run in $O(d)$ time. This is tight already for the range $\{0, 1, 2\}$, even for the special case of the hypercube domain [JR13].

1.2 Our Techniques

For clarity of presentation, we first state and prove all our theorems for $\delta = 1$, i.e., for integer-valued functions. In Section 5, by discretizing (as was done in [JR13]), we extend our results to the range $\delta\mathbb{Z}$.

The main challenge in designing a tester for functions on the hypergrid domains is avoiding an exponential dependence on the dimension d. We achieve this via a *dimension reduction*, i.e., a reduction from testing functions on the hypergrid $[n]^d$ to testing functions on the line $[n]$, that incurs only an $O(d\cdot\min\{\mathrm{ImgD}, nd\})$ multiplicative overhead in the running time. In order to do this, we relate the distance to the Lipschitz property of a function f on the hypergrid to the average distance to the Lipschitz property of restrictions of f to 1-dimensional (axis-parallel) lines. For $i \in [d]$, let $e^i \in [n]^d$ be 1 on the ith coordinate and 0 on the remaining coordinates. Then for every dimension $i \in [d]$ and $\alpha \in [n]^d$ with $\alpha_i = 0$, the *line g of f along dimension i with position α* is the restriction of f defined by $g(x_i) = f(\alpha + x_i \cdot e^i)$, where x_i ranges over $[n]$. We denote the set of lines of f along dimension i by L^i_f and the set of all lines, i.e., $\bigcup_{i\in[d]} L^i_f$, by L_f. We denote the relative distance of a function h to the Lipschitz property, i.e., the fraction of input points where the function needs to be changed in order to become Lipschitz, by $\epsilon^{Lip}(h)$. The technical core of our dimension reduction is the following theorem that demonstrates that if a function on the hypergrid is far from the Lipschitz property then a random line from L_f is, in expectation, also far from it.

Theorem 1.2 (Dimension reduction)**.** *For all functions $f : [n]^d \to \mathbb{Z}$, the following holds:*

$$\mathbb{E}_{g\leftarrow L_f}\left[\epsilon^{Lip}(g)\right] \geq \frac{\epsilon^{Lip}(f)}{2\cdot d\cdot \mathrm{ImgD}(f)}.$$

To obtain this result, we introduce a smoothing procedure that "repairs" a function (i.e., makes it Lipschitz) one dimension at a time, while modifying it at a few points. Such procedures have been previously designed for restoring monotonicity of Boolean functions [GGL+00, DGL+99] and for restoring the Lipschitz property of functions on the hypercube domain [JR13]. The key challenge is to find a smoothing procedure that satisfies the following three requirements:

1. It makes all lines along dimension i (i.e., in L^i_f) Lipschitz.

2. It changes only a small number of function values.

3. It does not make lines in other dimensions less Lipschitz, according to some measure.

There are known smoothing operators (e.g., graph Laplacian) that make a function more Lipschitz [Oll09], but to the best of our knowledge there are no appropriate bounds on the number of function values that are changed.

[4] If $\delta > 1$ then f is Lipschitz iff it is 0-Lipschitz (that is, constant). Testing if a function is constant takes $O(1/\epsilon)$ time.

Smoothing Procedure for 1-dimensional Functions. Our first technical contribution is a local smoothing procedure for functions $f : [n] \to \mathbb{Z}$, which we call **BubbleSmooth**, in analogy to Bubble Sort. In one *basic step*, **BubbleSmooth** modifies two consecutive values (i.e., $f(i)$ and $f(i+1)$ for some $i \in [n-1]$) that violate the Lipschitz property, namely, differ by more than 1. It decreases the larger and increases the smaller by 1, i.e., it transfers a unit from the larger to the smaller. See Algorithm 1 for the description of the order in which basic steps are applied. **BubbleSmooth** is a natural generalization of the *averaging operator* in [JR13], used to repair an edge of the hypercube, that can also be viewed as several applications of the basic step to the edge.

One challenge in analyzing **BubbleSmooth** is that when it is applied to all lines in one dimension, it may increase the average distance to the Lipschitz property for the lines in the remaining dimensions. Our second key technical insight is to use the ℓ_1 distance to the Lipschitz property to measure the performance of our procedure on the line and its effect on other dimensions. The ℓ_1 distance between functions f and f' on the same domain, denoted by $|f - f'|_1$, is the sum of $|f(x) - f'(x)|$ over all values x in the domain. The ℓ_1 distance of a function f to Lipschitz, denoted $\ell_1^{Lip}(f)$, is $\min_{f'} |f - f'|_1$, where the minimum is taken over all Lipschitz functions with the same domain as f. Observe that the Hamming distance and the ℓ_1 distance from a function to a property can differ by at most $\mathrm{ImgD}(f)$. Later, we leverage the fact that Lipschitz functions have a relatively small image diameter to relate the ℓ_1 distance to the Hamming distance.

We prove that **BubbleSmooth** returns a Lipschitz function and that it makes at most twice as many changes in terms of ℓ_1 distance as necessary to make a function Lipschitz.

Theorem 1.3. *Consider a function $f : [n] \to \mathbb{Z}$ and let f' be the function returned by **BubbleSmooth**(f). Then (1) function f' is Lipschitz and (2) $|f - f'|_1 \leq 2 \cdot \ell_1^{Lip}(f)$.*

The proof of the second part of this theorem requires several technical insights. One of the challenges is that **BubbleSmooth** changes many function values, but then undoes most changes during subsequent steps. We define a transfer graph to keep track of the transfers that move a unit of function value during each basic step. Its vertex set is $[n]$ and an edge (x, y) represents that a unit was transferred from $f(x)$ to $f(y)$. Since two transfers (x, y) and (y, z) are equivalent to a transfer (x, z), we can merge the corresponding edges in the transfer graph, proceeding with such merges until no vertex in it has both incoming and outgoing edges. As a result, we get a transfer graph where the number of edges, $|E|$, is half the ℓ_1 distance from the original to the final function.

To prove that $|E| \leq \ell_1^{Lip}(f)$, we show that the transfer graph has a matching with the violation score at least $|E|$. The *violation score* of an edge (or a pair) (x, y) is the quantity by which $|f(x) - f(y)|$ exceeds the distance between x and y. (Recall that $|f(x) - f(y)| \leq |x - y|$ for all Lipschitz functions f on domain $[n]$.) The violation score of a matching is the sum of the violation scores over all edges in the matching. We observe (in Lemma 2.4) that $\ell_1^{Lip}(f)$ is bounded below by a violation score of any matching. The crucial step in obtaining a matching with a large violation score is pinpointing a provable, but strong enough property of the transfer graph that guarantees such a matching. Specifically, we show that the violation score of each edge in the graph is at least the number of edges adjacent to its endpoints at its (suitably defined) *moment of creation* (Lemma 2.2). For example, this statement is not true if we consider adjacent edges in the final transfer graph. The construction of a matching with a large violation score in the transfer graph is one of the key technical contributions of this paper. It is the focus of Section 2.

Dimension Reduction with respect to ℓ_1. Our smoothing procedure for functions on the hypergrids applies **BubbleSmooth** to repair all lines in dimensions $1, 2, ..., d$, one dimension at a time. We show that for all $i, j \in [d]$ applying **BubbleSmooth** in dimension i does not increase the expected $\ell_1^{Lip}(f)$ for a random line g in dimension j. The key feature of our smoothing procedure that makes the analysis tractable is that it can be broken down into steps, each consisting of one application of the basic step of **BubbleSmooth** to the same positions $(k, k+1)$ on all lines in a specific dimension. This allows us to show that one such step does not make other dimensions worse in terms of the ℓ_1 distance to the Lipschitz property. The cleanest statement of the resulting dimension reduction is with respect to the ℓ_1 distance.

Theorem 1.4. *For all functions $f : [n]^d \to \mathbb{Z}$, we have: $\sum_{g \in L_f} \ell_1^{Lip}(g) \geq \frac{\ell_1^{Lip}(f)}{2}$.*

Our Testers and Effective Image Diameter. The main component of our tester (Algorithm 5) repeats the following procedure: *Pick a line uniformly at random and run one step of the line tester.* (We use the line tester from [JR13]. One step of this tester samples a random pair (x, x') from a collection of special pairs and rejects if f violates the Lipschitz condition for the pair (x, x'). The special pairs are edges of a sparse *2-transitive-closure-spanner* (2-TC-spanner) of the line graph on n vertices. Specifically, there are $O(n \log n)$ special pairs.) Our dimension reduction (Theorem 1.2) is crucial in analyzing this component. However, the bound in Theorem 1.2 depends on the image diameter of the function f. In the case of a non-Lipschitz function, it can be arbitrarily large, but for a Lipschitz function on $[n]^d$ it is at most the diameter of the space, namely nd (notice this factor in part (1) of Theorem 1.1). In fact, for our application we can also use the *observable diameter* of the space [Gro99]: since the hypergrid exhibits Gaussian-type concentration of measure, one obtains that a Lipschitz function maps the vast majority of points to an interval of size $O(n\sqrt{d})$ (notice this factor in part (2) of Theorem 1.1). Our testers use a preliminary step to rule out functions with large image diameter (resulting in 1-sided error) or with large observable diameter (resulting in 2-sided error).

1.3 Comparison to Previous and Subsequent Work

Results. Jha and Raskhodnikova [JR13] gave a 1-sided error nonadaptive testers for the Lipschitz property of functions of the form $f : \{0,1\}^d \to \delta\mathbb{Z}$ and $f : [n] \to \mathbb{R}$ that run in time $O\left(\frac{d}{\delta\epsilon} \cdot \min\{\mathrm{ImgD}(f), d\}\right)$ and $O\left(\frac{\log n}{\epsilon}\right)$, respectively. They also showed that $\Omega(d)$ queries are necessary for testing the Lipschitz property on the domain $\{0,1\}^d$, even when the range is $\{0,1,2\}$. No nontrivial tester of the Lipschitz property of functions on the domain $[n]^d$ was known prior to this work.

Our first tester from Theorem 1.1 naturally generalizes the testers of [JR13] to functions on the domain $[n]^d$. As in [JR13], our tester has at most quadratic dependence on the dimension d. Our second tester from Theorem 1.1 gives an improvement in the running time over the hypercube tester in [JR13] at the expense of allowing 2-sided error. In this specific case, Theorem 1.1 gives a tester with running time $\tilde{O}(d^{1.5}/(\delta\epsilon))$.

Subsequently to our work, Chakrabarty and Seshadhri [CS13] gave an ingenious tester for the Lipschitz property of functions $[n]^d \to \mathbb{R}$ that runs in $O(d \log n/\epsilon)$ time. Later, Chakrabarty et al. [CDJS15] presented a dimension reduction for the Lipschitz property with respect to Hamming distance. Our dimension reduction with respect to ℓ_1 was used by Berman et al. [BRY14] to obtain testers for the Lipschitz property that work with respect to ℓ_p distances.

Techniques. Relating the distance to the property of a given function with the distance to the property of random restrictions has been successfully used to obtain testers for many properties. Notably, for functions on multi-dimensional domains, it has been done for testing low degree, monotonicity and, for the special case of the hypercube, the Lipschitz property. Two ideas that appeared repeatedly in proofs of this type of statements are self-correction (e.g., in low-degree testing) and repair (e.g., in monotonicity and Lipschitz testing). Specifically, in [GGL+00, DGL+99, JR13] the function is repaired one dimension at a time. We note that many new ideas are required to generalize the repair procedure in [JR13] to functions on the hypergrid domains. Their repair procedure takes an integer average of values for each edge in a given dimension. So, our main challenge was designing and analyzing a natural generalization of this procedure. The procedure in [GGL+00, DGL+99], for repairing monotonicity of Boolean functions on the hypergrid domains sorts the 0-1 values on each line in a given dimension. There are at least three obstacles that make the design and analysis of our repair procedure significantly harder: (1) Our function values are not limited to 0s and 1s. (2) There is no natural unique Lipschitz function to which we should reconstruct (in the case of monotonicity, sorting gives such a function). (3) Unlike in the case of sorting, the Hamming distance does not work as a measure of progress for our operator.

The repair procedure in [DGL+99] for restoring monotonicity of functions on general ranges applies induction on the size of the range, using Boolean range as the base case. Observe that in the case of the Lipschitz property, functions with Boolean ranges are always Lipschitz, so there is nothing to test. In addition, in this case, not only the size of the range, but also the distances between points in the range play a role. Even though for monotonicity, repairing a function with a range of size greater than 2 in one dimension at a time does not work, this is exactly what we do here.

1.4 Organization

In Section 2, we present and analyze **BubbleSmooth**, our procedure for smoothing 1-dimensional functions, and prove Theorem 1.3. In Section 3, we use **BubbleSmooth** to construct a smoothing procedure for multi-dimensional functions that leads to the dimension reduction of Theorems 1.2 and 1.4. Our Lipschitz testers for functions on hypergrids claimed in Theorem 1.1 are presented in Section 4.

2 BubbleSmooth and its Analysis

In this section, we describe **BubbleSmooth** and prove Theorem 1.3 which asserts that **BubbleSmooth**(f) outputs a Lipschitz function that does not differ too much from f in the ℓ_1 distance. In Section 2.1, we present **BubbleSmooth** (Algorithm 1) and show that it outputs a Lipschitz function. Sections 2.2 and 2.3 are devoted to proving part (2) of Theorem 1.3. At the high level, the proof follows the ideas explained in Section 1.2 (right after Theorem 1.3). In Section 2.2, we define our transfer graph (Definition 2.3) and prove its key property (Lemma 2.2). In Section 2.3, we show that the existence of a matching with a large violation score implies that f is far from Lipschitz in the ℓ_1 distance (Lemma 2.4) and complete the proof of part (2) of Theorem 1.3 by constructing such a matching in the transfer graph.

2.1 Description of BubbleSmooth and Proof of Part (1) of Theorem 1.3

We begin this section by recalling two basic definitions from [JR13].

Definition 2.1 (Violation score)**.** *Let f be a function and x, y be points in its domain. The pair (x, y) is* violated *by f if $|f(x) - f(y)| > |x - y|_1$. The* violation score *of (x, y), denoted by* $\mathrm{vs}_f(x, y)$*, is $|f(x) - f(y)| - |x - y|_1$ if it is violated and 0 otherwise.*

Definition 2.2 (Basic operator)**.** *Given $f : [n]^d \to \mathbb{Z}$ and $x, y \in [n]^d$, where $|x - y|_1 = 1$ and vertex names x and y are chosen so that $f(x) \le f(y)$, the basic operator $\mathbb{B}_{x,y}$ works as follows: If the pair (x, y) is not violated by f then $\mathbb{B}_{x,y}[f]$ is identical to f. Otherwise, $\mathbb{B}_{x,y}[f](x) = f(x) + 1$ and $\mathbb{B}_{x,y}[f](y) = f(y) - 1$.*

In this section, we view a function $f : [n] \to \mathbb{Z}$ as an integer-valued sequence $f(1), f(2), \ldots, f(n)$. We denote the subsequence $f(i), f(i+1), \ldots, f(j)$ by $f[i..j]$. Naturally, a sequence $f[i..j]$ is *Lipschitz* if $|f(k) - f(k+1)| \le 1$ for all $i \le k \le j-1$. Algorithm 1 presents a formal description of **BubbleSmooth**.

Algorithm 1: BubbleSmooth (Input: an integer sequence $f[1 \ldots n]$)

for $i = n-1$ *to* 1 **do**
  `// Start phase` **Fix**(i).
  while $|f(i) - f(i+1)| > 1$ **do** `//` $(i, i+1)$ `is violated by` f
    LinePass(i).
return f

Algorithm 2: LinePass (Input: integer i)

for $j = i$ *to* $n-1$ **do**
  $f \leftarrow \mathbb{B}_{j,j+1}[f]$.
  `// Apply basic operator (see Definition 2.2.)`

See Figure 1 for an illustration of **BubbleSmooth**. We start analyzing the behavior of **BubbleSmooth** by proving part (1) of Theorem 1.3, which states that **BubbleSmooth** returns a Lipschitz function.

Proof of part (1) of Theorem 1.3. Consider an integer sequence $f[1..n]$ and let $f'[1..n]$ be the sequence returned by **BubbleSmooth**(f). We prove that f' is Lipschitz by induction on the phase of **BubbleSmooth**. Initially, $f(n)$ is vacuously Lipschitz. We fix $i \in [n]$, assume $f[i+1..n]$ is Lipschitz at the beginning of phase **Fix**(i) and show this phase terminates and that $f[i..n]$ is Lipschitz at the end of the phase.

Consider an execution of **LinePass**(i). Assume $f[i+1..n]$ is Lipschitz in the beginning of this execution.

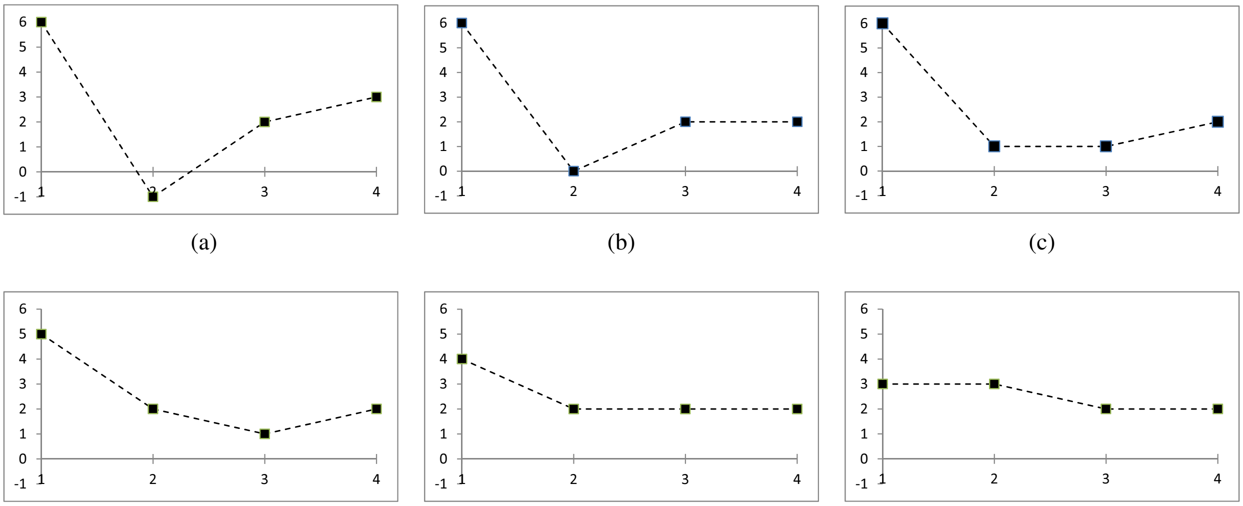
Observation 2.1. *Let j be the index, such that at the beginning of the execution, $f[i..j]$ is the longest strictly monotone sequence starting from $f(i)$. Then **LinePass**(i) modifies two elements: $f(i)$ and $f(j)$.*

If $f(i) > f(j)$ then $f(i)$ is decreased by 1 and $f(j)$ is increased by 1, i.e., 1 unit is *transferred* from i to j. Similarly, if $f(i) < f(j)$ then 1 unit is transferred from j to i. It is easy to see that after this transfer is performed, $f[i+1..n]$ is still Lipschitz. Moreover, each iteration of **LinePass**(i) reduces the violation score of the pair $(i, i+1)$ by at least 1. Thus, phase **Fix**(i) terminates with $f[i..n]$ being Lipschitz. □

2.2 Transfer Graph

In the proof of part (1) of Theorem 1.3, we established that for all $i \in [n]$, each iteration of **LinePass**(i) transfers one unit to or from i. We record the transfers in the *transfer graph* $T = ([n], E)$, defined next. A transfer from x to y is recorded as a directed edge (x, y). The edges of the transfer graph are ordered (indexed), according to when they were added to the graph. The edge (i, j) (resp., (j, i)) corresponding to the most recent transfer is combined with a previously added edge (j, k) (resp., (k, j)) if such an edge exists. This is done because transfers from x to y and from y to z are equivalent to a transfer from x to z. If a new edge (x, y) is merged with an existing edge (y, z), the combined edge retains the index of the edge (y, z).

Definition 2.3 (Transfer graph)**.** *The* transfer graph *is a directed graph $T = ([n], E)$, where the edge set $E = (e_1, \ldots, e_t)$ is ordered and edges are not necessarily distinct. The graph is defined by the following procedure. Initially, $E = \emptyset$ and $t = 0$. Each new run of **LinePass** during the execution of **BubbleSmooth**, transfers a unit from i to j (or resp., from j to i) for some i and j. If j has no outgoing (resp., incoming) edge in T, then increment t by 1 and add the edge $e_t = (i, j)$ (resp., $e_t = (j, i)$) to E. Otherwise, let e_s be an outgoing edge (j, k) (resp., an incoming edge (k, j)) with the largest index s. Replace (j, k) with (i, k), i.e., $e_s \leftarrow (i, k)$. (Replace (k, j) with (k, i), i.e., $e_s \leftarrow (k, i)$.) The final transfer graph is denoted by T^*.*



(a) (b) (c)

(d) (e) (f)

Figure 1: An illustration of **BubbleSmooth**. (a) The initial function. Phase **Fix**(3) does not alter the function. (b)-(c) Phase **Fix**(2): Functions obtained after one and two executions of **LinePass**(2), respectively. (d)-(f) Phase **Fix**(1): Functions obtained after each execution of **LinePass**(1).

As mentioned previously, the order of creation of edges is important to formalize the desired property of the transfer graph, so we need to consider the subgraphs that consist of the first s edges $e_1, \ldots, e_s$ of E.

Definition 2.4 (Degrees)**.** *Consider a transfer graph T at some time during the execution of **BubbleSmooth**. For all $s \in \{0, \ldots, t\}$, its subgraph graph T_s is defined as $([n], (e_1, \ldots, e_s))$, where $(e_1, \ldots, e_t)$ is the ordered edge set of T. (When $s = 0$, the edge set of T_s is empty.) The degree of a vertex $x \in [n]$ of T_s is denoted by $deg_s(x)$; when T_s is a subgraph of the final transfer graph, it is denoted by $deg^*_s(x)$.*

Observe that at no point in time can a vertex in T simultaneously have an incoming and an outgoing edge because such edges would get merged into one edge.

Lemma 2.2 (Key property of transfer graph)**.** Let f be an input function given to **BubbleSmooth**. Then for each edge $e_s = (x, y)$ of the final transfer graph T^*, the following holds:

$$\mathrm{vs}_f(x, y) \geq deg^*_s(x) + deg^*_s(y) - 1.$$

To prove this lemma, we consider each phase of **BubbleSmooth** separately and formulate a slightly stronger invariant that holds at every point during that phase.

Definition 2.5. *For all $i \in [n-1]$, let Δ_i be the degree of i in the transfer graph at the end of phase **Fix**(i).*

The following stronger invariant of the transfer graph directly implies Lemma 2.2.

Claim 2.3 (Invariant for phase **Fix**(i))**.** Let f be an input function given to **BubbleSmooth**. At every point during the execution of **BubbleSmooth**(f), for each edge $e_s = (x, y)$ of the transfer graph T,

$$f(x) - f(y) \geq deg_s(x) + deg_s(y) - 1 + |x - y|.$$

Moreover, for phase **Fix**(i) for each $i \in [n-1]$, after each execution of **LinePass**(i), for each edge e_s incident on vertex i, the following (stronger) condition holds:

if the edge $e_s = (i,j)$, i.e., it is outgoing from i, then $f(i) - f(j) \geq \Delta_i + deg_s(j) - 1 + |i-j|$;
if the edge $e_s = (j,i)$, i.e., it is incoming into i, then $f(j) - f(i) \geq \Delta_i + deg_s(j) - 1 + |i-j|$.

Observe that all transfers involving i during phase **Fix**(i) are in the same direction: if in the beginning of the phase we have $f(i) > f(i+1)$, then all transfers are from i; if we have $f(i) < f(i+1)$ instead, then all transfers are to i. In particular, whenever an edge incident to i is added, it is not modified subsequently during phase **Fix**(i). So for all s, $deg_s(i)$ never exceeds Δ_i during phase **Fix**(i) and the condition in Claim 2.3 is indeed stronger than that in Lemma 2.2.

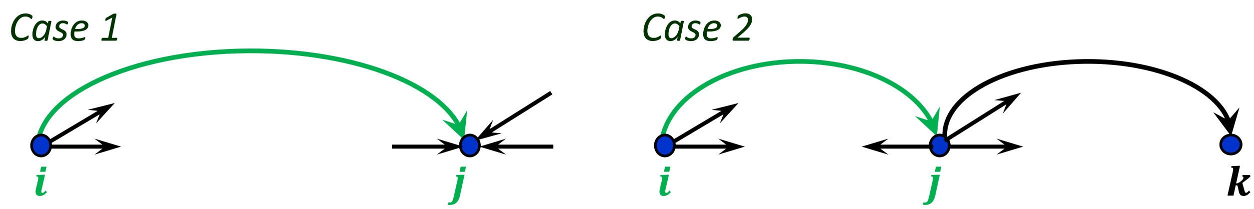


Figure 2: Two cases in the proof of Claim 2.3.

Proof of Claim 2.3. Initially the transfer graph has no edges, so the invariant in Claim 2.3 holds. Observe that for all $i \in [n-2]$, if the invariant holds at the end of phase **Fix**$(i+1)$, it also holds in the beginning of the following phase **Fix**(i). This is because the condition on each edge not incident to i in phase **Fix**$(i+1)$ is the same or stronger than in phase **Fix**(i) (notice that in the beginning of phase **Fix**(i) there are no edges incident to i). It remains to prove that if the invariant holds before an iteration of **LinePass**, it also holds after the iteration.

Consider a phase **Fix**(i) for some $i \in [n-1]$ and an execution of **LinePass**(i) that transfers a unit from i to j for some $j \in \{i+1, \ldots, n\}$. (The case when a unit is transferred in the other direction is symmetric.) Let f^- be the function and T^- be the transfer graph right *before* the considered execution of **LinePass**. Define deg_s^- as in Definition 2.4, with respect to the transfer graph T^-. Define T^+ and deg_s^+ analogously for the moment right *after* the considered execution of **LinePass**. Let t be the number of edges in T^-.

Since the current transfer occurred, the sequence $f^-[i+1,j]$ is monotone decreasing, giving $f^-(i+1) - f^-(j) \geq |i+1-j|$. The number of transfers from i that occurred before the current transfer is $deg_t^-(i)$. The number of the remaining transfers, including the current one, in phase **Fix**(i) is thus $\Delta_i - deg_t^-(i)$. Since each such transfer from i can happen only if the pair $(i, i+1)$ is violated, and moreover it lowers the violation score of the pair $(i, i+1)$ by at least 1, it follows that $\mathrm{vs}_{f^-}(i, i+1) \geq \Delta_i - deg_t^-(i)$ or, equivalently, $f^-(i) - f^-(i+1) \geq \Delta_i - deg_t^-(i) + 1$. Therefore,

$$f^-(i) - f^-(j) = [f^-(i) - f^-(i+1)] + [f^-(i+1) - f^-(j)] \geq \Delta_i - deg_t^-(i) + |i-j|. \tag{1}$$

The effect the current transfer has on the transfer graph T^- depends on whether T^- contains an outgoing edge from j. We consider the two corresponding cases separately.

Case 1: transfer graph T^- contains no outgoing edge from j. Then T^+ is obtained from T^- by adding the edge $e_{t+1} = (i,j)$. (See Figure 2.)

Recall that all transfers involving i made during phase **Fix**(i) are in the same direction, either all from i or all to i. So, by the assumption that the current transfer is from i, vertex i can have only outgoing edges in the transfer graph during phase **Fix**(i). That is,

$$f(i) = f^{-}(i) + deg_t^{-}(i). \tag{2}$$

By assumption of Case 1, vertex j can have only incoming edges. That is,

$$f(j) = f^{-}(j) - deg_t^{-}(j). \tag{3}$$

Applying first (2), (3), and then (1), we get:

$$\begin{aligned} f(i) - f(j) &= f^{-}(i) - f^{-}(j) + deg_t^{-}(i) + deg_t^{-}(j) \\ &\geq \Delta_i - deg_t^{-}(i) + |i-j| + deg_t^{-}(i) + deg_t^{-}(j) \\ &= \Delta_i + deg_{t+1}^{+}(j) - 1 + |i-j|. \end{aligned}$$

The last equality holds because the edge $e_{t+1} = (i, j)$ is added to T after the current transfer, so $deg_{t+1}^{+}(j) = deg_t^{-}(j)$+1. We proved that the invariant in Claim 2.3 holds for the new edge.

Since all other edges and their ordering are unchanged, $deg_s^{+}(x) = deg_s^{-}(x)$ for all $x \in [n]$ and $s \leq t$. Thus, the invariant of Claim 2.3 holds for all edges of T^{+}.

Case 2: transfer graph T^{-} contains an outgoing edge from j. Let $e_r = (j, k)$ be such an edge with the largest index r. Then T^{+} is obtained from T^{-} by replacing the edge (j, k) with the edge (i, k), with this new edge receiving the index r. (See Figure 2.) Notice that k could be larger or smaller[5] than j.

Recall that each vertex in T^{-} can only have one type of incident edges: either incoming or outgoing. In this case, both i and j can only have outgoing edges. Since T^{-} has t edges,

$$f(i) = f^{-}(i) + deg_t^{-}(i) \text{ and } f(j) = f^{-}(j) + deg_t^{-}(j). \tag{4}$$

By assumption, the invariant in Claim 2.3 holds for the transfer graph T^{-}. Since the edge (j, k) has index r in T^{-}, i.e., $(j, k) = e_r$,

$$f(j) - f(k) \geq deg_r^{-}(j) + deg_r^{-}(k) - 1 + |j-k|.$$

Finally, $deg_t^{-}(j) = deg_r^{-}(j)$ because of our choice of r. Putting the equations for this case together, then applying (1) and finally using the triangle inequality, we get:

$$\begin{aligned} f(i) - f(k) &= [f(i) - f(j)] + [f(j) - f(k)] \\ &\geq [f^{-}(i) + deg_t^{-}(i) - f^{-}(j) - deg_t^{-}(j)] + [deg_r^{-}(j) + deg_r^{-}(k) - 1 + |j-k|] \\ &\geq \Delta_i - deg_t^{-}(i) + |i-j| + deg_t^{-}(i) + deg_r^{-}(k) - 1 + |j-k| \\ &\geq \Delta_i + deg_r^{-}(k) - 1 + |i-k| = \Delta_i + deg_t^{+}(k) - 1 + |i-k|. \end{aligned}$$

Thus, the invariant in Claim 2.3 holds for the newly formed edge.

When the transfer graph is modified to reflect the current transfer, only edges incident to i,j or k are affected by the changes. For all other nodes x and all $s < t$, the corresponding degrees remain the same:

[5] For example, the case when k is smaller than j occurs when running **BubbleSmooth** over the sequence $f(1) = 5$, $f(2) = 0$, and $f(3) = 2$, during phase **Fix**(1).

$deg_s^+(x) = deg_s^-(x)$. For the node k and all $s < t$, the degrees are unchanged: $deg_s^+(k) = deg_s^-(k)$. For the node j and all $s \leq t$, the degrees decrease or remain the same: $deg_s^+(j) \leq deg_s^-(j)$. Therefore, for all edges in T^+ not affected by the current merge and not incident to i, the invariant in Claim 2.3 still holds. Finally, the stronger invariant for edges incident to i also holds for T^+ because it depends on a fixed parameter Δ_i (instead of $deg_s^+(i)$). □

2.3 Matchings of Violated Pairs

Part (2) of Theorem 1.3 states that the ℓ_1 distance between f and **BubbleSmooth**(f) is at most $2 \cdot \ell_1^{Lip}(f)$. By definition of the transfer graph $T = ([n], E)$, the distance $|f - \textbf{BubbleSmooth}(f)|_1 = 2|E|$. Lemma 2.4 shows that $\ell_1^{Lip}(f)$ is bounded below by the violation score of any matching. We complete the proof of Theorem 1.3 by showing that T has a matching with violation score $|E|$.

Definition 2.6 (Violation score of a set of pairs)**.** *Let M be a set of pairs violated by f. The violation score of the set M, denoted $\mathrm{vs}_f(M)$, is the sum of violation scores of all pairs in M.*

Lemma 2.4. Let M be a matching of pairs (x, y) where x and y are in the (discrete) domain of a function f. Then $\ell_1^{Lip}(f) \geq \mathrm{vs}_f(M)$.

Proof. Let f^* be a closest Lipschitz function to f (on the same domain as f) with respect to the ℓ_1 distance, i.e., $|f - f^*|_1 = \ell_1(f, Lip)$. Consider a pair $(x, y) \in M$. Since $|f(x) - f(y)| = |x - y|_1 + \mathrm{vs}_f(x, y)$ and $|f^*(x) - f^*(y)| \leq |x - y|_1$, it follows by the triangle inequality that $|f(x) - f^*(x)| + |f(y) - f^*(y)| \geq \mathrm{vs}_f(x, y)$. Since M is a matching, we can add over all of its pairs to obtain

$$\begin{aligned} \ell_1(f, Lip) = |f - f^*|_1 &\geq \sum_{(x,y)\in M} (|f(x) - f^*(x)| + |f(y) - f^*(y)|) \\ &\geq \sum_{(x,y)\in M} \mathrm{vs}_f(x, y) = \mathrm{vs}_f(M), \end{aligned}$$

which concludes the proof. □

Now using Lemma 2.2 we exhibit a matching in the final transfer graph which has large violation score, concluding the proof of Theorem 1.3.

Proof of part (2) of Theorem 1.3. Let $T^* = ([n], E)$ be the final transfer graph corresponding to the execution of **BubbleSmooth** on f and let $E = \{e_1, \ldots, e_t\}$. By definition of the transfer graph, $|f - f'|_1 = \sum_{i\in[n]} deg_t^*(i) = 2|E|$, where f' is the output of **BubbleSmooth**(f). By Lemma 2.4, it is enough to show that there is a matching M of pairs violated by f with the violation score $\mathrm{vs}_f(M) \geq |E|$.

We claim that T^* contains such a matching. It can be constructed greedily by setting $T = T^*$ and repeating the following step until no edges remain in T: let $e_s = (x, y)$ be the edge in T with the largest index $s \in [t]$; add e_s to M and then remove e_s and all other edges adjacent to its endpoints from T. In each step, at most $deg_s^*(x) + deg_s^*(y) - 1$ edges are removed from T, where degrees deg_s^* are defined with respect to the original graph T^*. This is because the degree of x (respectively, y) in T just before e_s is removed from T is at most $deg_s^*(x)$ (respectively, $deg_s^*(y)$), since e_s is the edge with the highest index present in T. By Lemma 2.2, $\mathrm{vs}_f(x, y) \geq deg_s^*(x) + deg_s^*(y) - 1$. So, at each step of the greedy procedure, the violation score of the pair (x, y) added to M is at least the number of edges removed from T. Therefore, $\mathrm{vs}_f(M) \geq |E|$. □

3 Dimension Reduction: Proof of Theorems 1.2 and 1.4

In this section, we prove Theorems 1.2 and 1.4 that connect the distance of a function to being Lipschitz to the distance of its lines to being Lipschitz. Effectively, these results reduce the task of testing a multidimensional function to the task of testing its lines. Our main contribution in this section is a smoothing procedure that makes a function Lipschitz by repairing one dimension at a time while modifying only a few function values. In Section 3.1, we present the *dimension operator* that repairs all lines in a specified dimension by applying **BubbleSmooth** to each of them. The important properties of the dimension operator are summarized in Lemma 3.1 which is the key ingredient in the proofs of Theorems 1.2 and 1.4. Section 3.2 proves auxiliary claims used in the proof of Lemma 3.1. Section 3.3 completes the proofs of Theorems 1.2 and 1.4.

3.1 Dimension operator and its properties

Recall from the discussion in Section 1.2 that we denote the set of lines of f along dimension i by L_f^i and the set of all lines of f by $L_f = L_f^i$. Notice that there are $d \cdot n^{d-1}$ lines.

Definition 3.1 (Dimension operator A_i)**.** *Given $f : [n]^d \to \mathbb{Z}$ and dimension $i \in [d]$, the* dimension operator *A_i applies **BubbleSmooth** to every function $g \in L_f^i$ and returns the resulting function, denoted $A_i[f]$.*

Next lemma summarizes the properties of the dimension operator.

Lemma 3.1 (Properties of the dimension operator A_i)**.** For all $i \in [d]$, the dimension operator A_i satisfies the following properties for every function $f : [n]^d \to \mathbb{Z}$.

1. (Repairs dimension i.) Every $g \in L^i_{A_i[f]}$ is Lipschitz.
2. (Does not modify the function too much.) $|f - A_i[f]|_1 \le 2 \cdot \sum_{g \in L_f^i} \ell_1^{Lip}(g)$.
3. (Does not worsen other dimensions.) For all $j \neq i$ in $[d]$, it does not increase the expected ℓ_1 distance of a random line in dimension j to the Lipschitz property, i.e., $\mathbb{E}_{g \leftarrow L^j_{A_i[f]}}[\ell_1^{Lip}(g)] \le \mathbb{E}_{g \leftarrow L_f^j}[\ell_1^{Lip}(g)]$.

Proof. **Item 1.** Item 1 follows from part (1) of Theorem 1.3.

Item 2. Since the dimension operator A_i operates by applying **BubbleSmooth** to all (disjoint) lines in L_f^i, we get $|f - A_i[f]|_1 = \sum_{g \in L_f^i} |g - \textbf{BubbleSmooth}[g]|_1$. The latter is at most $\sum_{g \in L_f^i} 2 \cdot \ell_1^{Lip}(g)$ by Part (2) of Theorem 1.3, thus proving the item.

Item 3. Fix i and j. First, we give a standard argument [GGL+00, DGL+99, JR13] that it is enough to prove this statement for $n \times n$ grids. Namely, every $\alpha \in [n]^d$ with $\alpha_i = \alpha_j = 0$ defines a restriction of a function f to an $n \times n$ grid by $h(x_i, x_j) = f(\alpha + x_i \cdot e^i + x_j \cdot e^j)$, where x_i and x_j range over $[n]$. (Recall that $e^i \in [n]^d$ is 1 on the ith coordinate and 0 on the remaining coordinates.) If the item holds for all 2-dimensional grids, we can average over all such grids defined by different α to obtain the statement for the d-dimensional function f. Now fix an arbitrary restriction $h : [n]^2 \to \mathbb{Z}$ as discussed and think of h as an $n \times n$ matrix with rows (resp., columns) corresponding to lines in dimension i (resp., in dimension j).

The key feature of our dimension operator A_i is that it can be broken down into steps, each consisting of one application of the basic step of **BubbleSmooth** to the same positions $(k, k+1)$ on all lines in dimension i. In other words, the basic step is applied to all rows of columns k and $k+1$. To see this, observe that we can replace the **while** loop condition on Line 2 of Algorithm 2 with "repeat t times", where t should be large enough to guarantee that the line segment under consideration is Lipschitz after t iterations of **LinePass**.

(E.g., $t = n \cdot \mathrm{ImgD}(f)$ repetitions suffices.) If this version of **BubbleSmooth** is run synchronously and in parallel on all lines in dimension i, the basic step will be applied to the same positions $(k, k+1)$ on all lines.

Since in each parallel update step only two adjacent columns of h are affected, it is sufficient to prove the item for two adjacent columns of h. Accordingly, consider two adjacent columns C_1 and C_2 of h. Let M_1 and M_2 be Lipschitz columns that are closest in the ℓ_1 distance to C_1 and C_2, respectively. Thus, $\ell_1^{Lip}(C_1) = |C_1 - M_1|_1$ and $\ell_1^{Lip}(C_2) = |C_2 - M_2|_1$. Let C_1' and C_2' be the columns of the matrix resulting from applying the basic operator to the rows of the matrix (C_1, C_2). Similarly, define M_1' and M_2' to be the columns of the matrix resulting from applying the basic operator to the rows of (M_1, M_2). (See also Figure 4.) By Corollary 3.3, M_1' and M_2' are Lipschitz. Therefore, $\ell_1^{Lip}(C_1') \leq |C_1' - M_1'|_1$ and $\ell_1^{Lip}(C_2') \leq |C_2' - M_2'|_1$. Finally, using the inequality $|C_1' - M_1'|_1 + |C_2' - M_2'|_1 \leq |C_1 - M_1|_1 + |C_2 - M_2|_1$ proved in Corollary 3.4 below, the proof of Item 3 is completed as follows:

$$\begin{aligned} \ell_1^{Lip}(C_1) + \ell_1^{Lip}(C_2) &= |C_1 - M_1|_1 + |C_2 - M_2|_1 \\ &\geq |C_1' - M_1'|_1 + |C_2' - M_2'|_1 \geq \ell_1^{Lip}(C_1') + \ell_1^{Lip}(C_2'). \end{aligned}$$

□

3.2 Basic operator on a square

To analyze the behavior of the dimension operator, we need to understand the effect of the basic operator $\mathbb{B}_{k,k+1}$ on a multidimensional function. We remark that our definition of the basic operator coincides with that of [JR13] (see Definition 3.5 of [JR13]) for integer-valued functions defined on domain $\{0,1\}^2$. We recall and extend properties of the basic operator from [JR13] when applied to functions defined on $\{0,1\}^2$ in Claim 3.2 below.

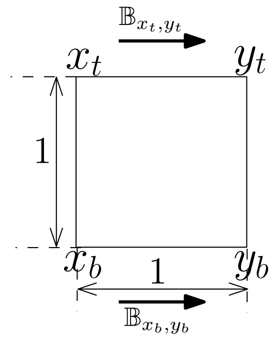
Claim 3.2 (Properties of basic operator on a square)**.** Consider a function $f : \{x_t, x_b, y_t, y_b\} \to \mathbb{Z}$ where x_b denotes 00, $x_t = 10$, $y_b = 01$ and $y_t = 11$. (See Figure 3.) Let f' be the function obtained by applying basic operators $\mathbb{B}_{x_t,y_t}$ and $\mathbb{B}_{x_b,y_b}$ along the horizontal edges. Then the following holds for the vertical edges.

1. [JR13, proof of Lemma 3.9, paragraph 3] The sum of violation scores of vertical edges does not increase:
$$\mathrm{vs}_{f'}(x_t, x_b) + \mathrm{vs}_{f'}(y_t, y_b) \leq \mathrm{vs}_f(x_t, x_b) + \mathrm{vs}_f(y_t, y_b).$$
2. The absolute difference of values along vertical edges does not increase:
$$|f'(x_t) - f'(x_b)| + |f'(y_t) - f'(y_b)| \leq |f(x_t) - f(x_b)| + |f(y_t) - f(y_b)|.$$

Proof of Claim 3.2, Item 2. If neither horizontal edge is violated then $f' = f$, and we are done. So assume without loss of generality $\{x_t, y_t\}$ is violated and $f(y_t) \geq f(x_t) + 2$. Then $\mathbb{B}_{x_t,y_t}$ increases $f(x_t)$ by 1 and decreases $f(y_t)$ by 1 leading to inequality (i): $f'(y_t) \geq f(x_t) + 1$. If, for both vertical edges, the absolute difference of the values does not increase, we are also done. Assume without loss of generality that the absolute difference of the values of the left vertical edge increases strictly, namely, $|f'(x_b) - f'(x_t)| = |f(x_b) - f(x_t)| + \Delta$ for $\Delta \in \{1, 2\}$. This implies that $\mathbb{B}_{x_b,y_b}$ does not increase the value at x_b and the following holds: (ii) $f(x_b) \leq f(x_t)$. The former also implies that: (iii) $f'(y_b) \leq f'(x_b) + 1$. The definition of Δ further gives that: (iv) $f'(x_b) = f(x_b) - (\Delta - 1)$ and (v) $f'(y_b) = f(y_b) + (\Delta - 1)$. Using inequalities (i), (ii), (iv) and (iii), we get

$$f'(y_t) \geq f(x_t) + 1 \geq f(x_b) + 1 = f'(x_b) + \Delta \geq f'(y_b) + (\Delta - 1),$$

Figure 3: The sum of absolute difference of values of the endpoints of vertical edges does not increase as a result of application of the basic operator along horizontal edges.



and hence $f'(y_t) - f'(y_b) \geq 0$. Using the last inequality, equation (v) and the fact that $f'(y_t) = f(y_t) - 1$, we get

$$|f'(y_t) - f'(y_b)| = f'(y_t) - f'(y_b) = f'(y_t) - f(y_b) - (\Delta - 1) = f(y_t) - f(y_b) - \Delta \leq |f(y_t) - f(y_b)| - \Delta.$$

Thus, the absolute difference of the values for $\{x_t, x_b\}$ increases by Δ, whereas that quantity for $\{y_t, y_b\}$ decreases by Δ. This completes the proof of the claim. $\square$

Claim 3.2 implies Corollaries 3.3 and 3.4 which were used in the proof of Lemma 3.1.

Corollary 3.3. *Let $M \in \mathbb{Z}^{n\times 2}$ be a matrix consisting of two Lipschitz columns. If the basic operator is applied to the rows of this matrix then the resulting matrix M' still has Lipschitz columns.*

Proof. Applying part 1 of Claim 3.2 to each 2×2 grid formed by taking 2 adjacent rows of M (respectively, M'), we get the desired statement. $\square$

The second corollary is about one-dimensional functions $C_1, C_2, M_1, M_2, C'_1, C'_2, M'_1$ and M'_2 used in proof of Lemma 3.1.

Corollary 3.4. $|C_1 - M_1|_1 + |C_2 - M_2|_1 \geq |C'_1 - M'_1|_1 + |C'_2 - M'_2|_1$.

Proof. Expanding each term in the statement, we get that it is equivalent to

$$\sum_{z\in[n]} \left[(|C_1(z) - M_1(z)| + |C_2(z) - M_2(z)|) - \left(|C'_1(z) - M'_1(z)| + |C'_2(z) - M'_2(z)| \right) \right] \geq 0. \quad (5)$$

We show that for each term in the sum, the inequality holds separately. Accordingly, fix $z \in [n]$ and let $f : \{0,1\}^2 \to \mathbb{Z}$ be the function defined as follows: $f(x_b) = C_1(z)$, $f(y_b) = M_1(z)$, $f(x_t) = C_2(z)$ and $f(y_t) = M_2(z)$. Similarly, let $f' : \{0,1\}^2 \to \mathbb{Z}$ be defined as follows: $f'(x_b) = C'_1(z)$, $f'(y_b) = M'_1(z)$, $f'(x_t) = C'_2(z)$ and $f'(y_t) = M'_2(z)$. Then the inequality for the term in (5) relative to z follows from the second part of Claim 3.2. Hence the corollary follows. $\square$

3.3 Proof of Theorems 1.2 and 1.4

To prove Theorems 1.2 and 1.4, we use the following observation that relates $\ell_1^{Lip}(f)$ to $\epsilon^{Lip}(f)$.

Observation 3.5. *For all $f : [n]^d \to \delta\mathbb{Z}$, the following holds:*

$$\delta\epsilon^{Lip}(f) \cdot n^d \leq \ell_1^{Lip}(f) \leq \epsilon^{Lip}(f) \cdot n^d \cdot \mathrm{ImgD}(f).$$

The first inequality follows directly from definitions, while the second follows from the fact that one can make a function f Lipschitz by changing $\epsilon^{Lip}(f)$ fraction of values, each by at most $\mathrm{ImgD}(f)$.

Proof of Theorems 1.2 and 1.4. Let A_i be the dimension repair operator of Definition 3.1. For $i \in [d]$, define f_i inductively by letting $f_i = A_i[f_{i-1}]$ with the *base case* being $f_0 = f$. Items 1 and 3 of Lemma 3.1 give that f_d is Lipschitz. Specifically, Item 1 implies that the application of the dimension operator A_i makes f_{i-1} Lipschitz along the ith dimension while Item 3 ensures that each such application does not introduce violations in the already repaired dimensions. Using properties of A_i from Lemma 3.1, the following holds for all $i \in [d]$.

$$|f_{i-1} - f_i|_1 = |f_{i-1} - A_i[f_{i-1}]|_1 \leq 2 \cdot \sum_{g \in L^i_{f_{i-1}}} \ell_1^{Lip}(g) \leq 2 \cdot \sum_{g \in L^i_f} \ell_1^{Lip}(g).$$

Specifically, the two inequalities above follow from Items 2 and 3 of Lemma 3.1, respectively. By the triangle inequality, $\ell_1^{Lip}(f) \leq \sum_{i=1}^d |f_{i-1} - f_i|_1$. This fact together with the above bound on $|f_{i-1} - f_i|_1$, leads to the following chain of (in)equalities and proves Theorem 1.4.

$$\ell_1^{Lip}(f) \leq \sum_{i=1}^d |f_{i-1} - f_i|_1 \leq \sum_{i=1}^d 2 \cdot \sum_{g \in L^i_f} \ell_1^{Lip}(g) = 2 \cdot \sum_{g \in L_f} \ell_1^{Lip}(g).$$

For proving Theorem 1.2, we apply Observation 3.5 to both sides of the inequality of Theorem 1.4 leading to the first inequality below.

$$n^d \epsilon^{Lip}(f) \leq 2 \cdot \sum_{g \in L_f} \epsilon^{Lip}(g) \cdot \mathrm{ImgD}(g) \cdot n \leq 2 \cdot \mathrm{ImgD}(f) \cdot n \sum_{g \in L_f} \epsilon^{Lip}(g) = 2 \cdot d \cdot \mathrm{ImgD}(f) \cdot n^d \cdot \mathbb{E}_{g \in L_f}[\epsilon^{Lip}(g)].$$

The last inequality follows from the fact that $\mathrm{ImgD}(f)$ is a trivial upper bound on $\mathrm{ImgD}(g)$ for every $g \in L_f$. Finally, expressing the sum as an expectation (as done in the last equality), we get Theorem 1.2. $\square$

4 Algorithms for Testing the Lipschitz Property on Hypergrids

In this section, we present our testers for the Lipschitz property of functions $f : [n]^d \to \mathbb{Z}$. Theorem 1.2 relates the distance of a function f from the Lipschitz property to the (expected) distance of its lines to this property. The resulting bound, however, depends on the image diameter of f. The image diameter is small (at most nd) for Lipschitz functions, but it is not bounded for general functions. Next we give a high-level description of our testers:

1. Estimate the image diameter of f and **reject** if it is too large.

2. Repeatedly sample a line g of f at random, run one step of a Lipschitz tester for the line on g and **reject** if a violated pair is discovered; otherwise, **accept**.

Step 1 ensures that a small sample of lines is enough to succeed with constant probability. The two testers we present differ only in one parameter which quantifies what "too large" means in Step 1.

4.1 Estimating the Effective Image Diameter

As mentioned before, a Lipschitz function on $[n]^d$ has image diameter at most nd, which can serve as a threshold for rejection in Step 1 of the informal procedure above. However, if we are willing to tolerate two-sided error, it is sufficient to use a smaller threshold, equal the *effective* diameter of the function. For a given $\epsilon \in (0,1)$, define $\mathrm{ImgD}_\epsilon(f)$ as the smallest value α such that f is ϵ-close to having image diameter α:

$$\mathrm{ImgD}_\epsilon(f) = \min_{U \subseteq [n]^d : |U| \geq (1-\epsilon)n^d} \{\max_{x \in U} f(x) - \min_{x \in U} f(x)\}.$$

Although the image diameter of a Lipschitz function f can indeed achieve value nd, the effective $\mathrm{ImgD}_\epsilon(f)$ is upper bounded by the potentially smaller quantity $O(n\sqrt{d \ln(1/\epsilon)})$. The next lemma makes this precise. It follows directly from McDiarmid's inequality (stated in Appendix A).

Lemma 4.1 (Effective image diameter of Lipschitz functions)**.** For all $\epsilon \in (0,1)$, each Lipschitz function $f : [n]^d \to \mathbb{R}$ is $(\epsilon/21)$-close to having image diameter at most $n\sqrt{d \ln(42/\epsilon)}$.

Our testers use estimates of the image diameter or effective diameter to reject functions. For this, we use the procedure SAMPLE-DIAMETER (Algorithm 3) defined in [JR13].

Algorithm 3: SAMPLE-DIAMETER

input : function $f : [n]^d \to \mathbb{R}$, $\epsilon \in (0,1)$

1 Let $s = \lceil 5/\epsilon \rceil$.
2 Select samples $z_1, z_2, \ldots, z_s$ from $[n]^d$ uniformly and independently at random.
3 Return $r = \max_{i=1}^s f(z_i) - \min_{i=1}^s f(z_i)$.

The next lemma provides guarantees for the estimate returned by SAMPLE-DIAMETER.

Lemma 4.2. The value r returned by SAMPLE-DIAMETER satisfied the following: (i) $\mathrm{ImgD}_\epsilon(f) \leq r$ with probability at least $3/4$; (ii) $r \leq \mathrm{ImgD}(f)$ (always) and (iii) $r \leq \mathrm{ImgD}_{\epsilon/21}(f)$ with probability at least $2/3$. Moreover, the algorithm runs in time $O(1/\epsilon)$.

Proof. It is clear that $r \leq \mathrm{ImgD}(f)$ and the first part of the lemma is proved in Claim 3.10 of [JR13]. It remains to prove that $r \leq \mathrm{ImgD}_{\epsilon/21}(f)$ with probability at least $2/3$.

Recall the definition of $\mathrm{ImgD}_{\epsilon/21}(f)$. Let $S \subseteq [n]^d$ be a set of size at most $\epsilon n^d/21$ such that $|f(x) - f(y)| \leq \mathrm{ImgD}_{\epsilon/21}(f)$ for every $x, y \in [n]^d \setminus S$. We have $r \leq \mathrm{ImgD}_{\epsilon/21}(f)$ whenever all the samples z_i lie outside S. By the union bound, the probability of this event is at least $1 - \left(\frac{5}{\epsilon} + 1\right)\left(\frac{\epsilon}{21}\right) \geq 2/3$, as required. $\square$

4.2 Tester for Hypergrid Domains

Our tester for functions on hypergrids uses a tester (Algorithm 4) for functions on lines from [JR13].

Algorithm 4: LINE-TESTER

input : function $g : [n] \to \mathbb{R}$, $r \in \mathbb{R}$

1. Let $H = ([n], E)$ be a 2-TC-spanner of the line graph, and let $E' \subseteq E$ be the set of edges (x, y) such that $|x - y| < r$.
2. Sample an edge (x, y) uniformly at random from E'
3. If $|g(x) - g(y)| > |x - y|$, output g **not Lipschitz**, else output g **is Lipschitz**.

Lemma 4.3 (Claim 3.13, [JR13])**.** Consider a function $g : [n] \to \mathbb{R}$ and $r \geq \mathrm{ImgD}(g)$. Then LINE-TESTER on input g and r is a 1-sided tester that always accepts if g is Lipschitz and otherwise rejects with probability at least $\frac{\epsilon^{Lip}(g)}{10 \log \min\{r,n\}}$. Moreover, it makes only a constant number of queries to g.

To analyze our testers, we also need to estimate the probability that a random line $g \leftarrow L_f$ is rejected by LINE-TESTER(g, r) with $r \geq \mathrm{ImgD}_{\epsilon/2}(f)$. Such an upper bound r on $\mathrm{ImgD}_{\epsilon/2}(f)$ will be obtained via Lemma 4.2. Since r may be much smaller than $\mathrm{ImgD}(f)$, Lemma 4.3 does not apply directly. Nevertheless, the next lemma shows how to circumvent this difficulty.

Lemma 4.4. Let $f : [n]^d \to \mathbb{Z}$ be ϵ-far from Lipschitz. Consider a real number $r \geq \mathrm{ImgD}_{\epsilon/2}(f)$. For a random line $g \leftarrow L_f$, the probability that LINE-TESTER(g, r) rejects is at least $\frac{\epsilon}{40 d \cdot r \cdot \log \min\{r,n\}}$.

Proof. Define the function f' by truncating f as follows. Consider integers $a < b$ such $|a-b| \leq \mathrm{ImgD}_{\epsilon/2}(f)$ and at most $\epsilon n^d/2$ points $x \in [n]^d$ have $f(x) \notin [a, b]$. Define

$$f'(x) = \begin{cases} a & \text{if } f(x) < a; \\ f(x) & \text{if } f(x) \in [a, b]; \\ b & \text{if } f(x) > b. \end{cases}$$

Clearly, $\mathrm{ImgD}(f') \leq \mathrm{ImgD}_{\epsilon/2}(f) \leq r$ and, since f and f' differ on at most $\epsilon/2$ points, f' is $\epsilon/2$-far from Lipschitz.

We first analyze how LINE-TESTER behaves on f'. Let $g' \leftarrow L_{f'}$ be a random line of f'. Since r is an upper bound on $\mathrm{ImgD}(f')$, and hence also an upper bound on $\mathrm{ImgD}(g')$, we can use Lemma 4.3 to obtain that LINE-TESTER(g', r) rejects with probability at least $\mathbb{E}_{g' \leftarrow L_{f'}}[\epsilon^{Lip}(g')/10 \log \min\{r, n\}]$. By the dimension reduction (Theorem 1.2), the probability of rejection is at least

$$\frac{\epsilon^{Lip}(f')}{20d \cdot \mathrm{ImgD}(f') \log \min\{r, n\}} \geq \frac{\epsilon}{40d \cdot r \cdot \log \min\{r, n\}}.$$

The following claim directly implies the desired result.

Claim 4.5. *Consider a line $g \in L_f$ of f and let g' be the corresponding line of f'. The probability that* LINE-TESTER(g, r) *rejects is at least as large as the probability that* LINE-TESTER(g', r) *rejects.*

Proof. Notice that every pair $x, y \in [n]$ that is violated by g', must be also violated by g. To see this, assume without loss of generality that $g'(x) < g'(y) - |x-y|$. We obtain, in particular, that $g'(x) < \max_z f'(z) \leq b$. Notice that by construction of f', this also implies that $g(x) \leq g'(x)$, since whenever we decrease a value in the construction of f' from f, this new value becomes equal to b. Similarly, $g'(y) > a$, and hence $g(y) \geq g'(y)$. These bounds give that $g(x) < g(y) - |x - y|$ and hence (x, y) is violated with respect to g. Note that every pair violated by g' is also violated by g. Since LINE-TESTER only checks pairs for violations, the claim follows. ☐

This concludes the proof of the lemma. ☐

Algorithm 5 presents our tester for the Lipschitz property on hypergrid domains. One of its inputs is a threshold R for rejection in Step 1. The testers in Theorem 1.1 are obtained by setting R appropriately.

Algorithm 5: Tester for the Lipschitz property for functions on hypergrids.

input : function $f : [n]^d \to \mathbb{Z}$, $\epsilon \in (0, 1)$, and value $R \in \mathbb{R}$

1 Let $r \leftarrow$ SAMPLE-DIAMETER$(f, \epsilon/2)$. If $r > R$, **reject**.

2 **for** $i = 1$ **to** $\ell = \frac{120d \cdot r \log \min\{r,n\}}{\epsilon}$ **do**

3   Select a line g uniformly from L_f and **reject** if LINE-TESTER(g, r) does.

4 **Accept**.

Proof of Theorem 1.1. We show that Algorithm 5 when run with $R = nd$ (respectively, $R = n\sqrt{d \ln(84/\epsilon)}$) satisfies Theorem 1.1.

First, we focus on the correctness of the testers. Suppose that the input function f is Lipschitz. Since Lipschitz functions do not have any violated pairs, Algorithm 5 may only reject f in Step 1. When $R = nd$, this happens with probability 0, since Lemma 4.2 guarantees that $r \leq \mathrm{ImgD}(f) \leq nd$. Then Algorithm 5 with $R = nd$ is a 1-sided error tester. Now consider the case when $R = n\sqrt{d \ln(84/\epsilon)}$. By the second part of Lemma 4.2 and Lemma 4.1, with probability at least $2/3$ we have $r \leq R$ (notice that SAMPLE-DIAMETER is evoked with parameter $\epsilon/2$). Thus, Algorithm 5 with $R = n\sqrt{d \ln(84/\epsilon)}$ accepts the Lipschitz function f with probability at least $2/3$.

Now consider the case when f is ϵ-far from Lipschitz. We show that with probability at least $2/3$, f is rejected in some iteration of Step 3. This part of the analysis is independent of the setting of R. Let E be the event that r is a good estimate, namely that $r \geq \mathrm{ImgD}_{\epsilon/2}(f)$; from part 1 of Lemma 4.2, E holds with probability at least $3/4$. Then f is rejected with probability at least $\Pr(\text{Step 3 rejects and } E \text{ holds}) \geq \Pr(\text{Step 3 rejects} \mid E) \cdot (3/4)$. Conditioned on E (or more precisely conditioning on a realization of r satisfying E), Lemma 4.4 gives that the probability p of rejection on a single execution of Step 3 rejects is at least $p = \frac{\epsilon}{40d \cdot r \cdot \log \min\{r,n\}}$. Therefore, using the standard approximation $(1 - x) \leq e^{-x}$ valid for all x, we obtain that $\Pr(\text{Step 3 rejects} \mid E)$ is at least $1 - (1 - p)^{\ell} \geq 8/9$; it then follows that the probability of rejection by the procedure is at least $(8/9) \cdot (3/4) = 2/3$.

Finally, we analyze the running time of the testers. Observe that since all operations performed by the algorithms (computing the maximum and simple comparisons) take time at most linear in the number of queries, the time complexity is the same as query complexity (in the model where each required random number is generated in one step). It remains to analyze the query complexity of Algorithm 5 for both settings of R. First, SAMPLE-DIAMETER in Step 1 makes $O(1/\epsilon)$ queries. Each iteration of Step 3 makes only 2 queries. Finally, by construction of the estimator, $r \leq \mathrm{ImgD}(f)$ and whenever the **for** loop is executed we also have $r \leq R$. This gives that the total number of iteration of the **for** loop is at most $O(1) \cdot d \min\{\mathrm{ImgD}(f), R\} \log \min\{\mathrm{ImgD}(f), R, n\}/\epsilon$, so the total number of queries made by Algorithm 5 is also of this order. Our choices of R give the desired query complexity, concluding the proof of Theorem 1.1. $\square$

5 Testers for functions with range $\delta\mathbb{Z}$

In this section, we discuss modifications in the proofs required to obtain the desired testers for functions $f : [n]^d \to \delta\mathbb{Z}$ with $\delta \in (0, 1]$.

5.1 Modifications to Section 2

First, the main product of Section 2, namely, Theorem 1.3, holds as stated for the more general functions $f : [n] \to \delta\mathbb{Z}$ with $\delta \leq 1$. To prove it, we start by changing the definition of the basic operator to modify the values of the function by $\pm\delta$.

Definition 5.1 (Basic operator)**.** *Given $f : [n]^d \to \delta\mathbb{Z}$ and $x, y \in [n]^d$ where $|x - y|_1 = 1$ and vertex names x and y are chosen so that $f(x) \leq f(y)$, the basic operator $\mathbb{B}_{x,y}$ works as follows: If the pair (x, y) is not violated by f then $\mathbb{B}_{x,y}[f]$ is identical to f. Otherwise, $\mathbb{B}_{x,y}[f](x) = f(x) + \delta$ and $\mathbb{B}_{x,y}[f](y) = f(y) - \delta$.*

We continue with procedures **LinePass** and **BubbleSmooth**. Now **LinePass** uses the new basic operator defined above, but no other changes are required in these procedures. However, notice that whenever **LinePass** (i) is applied to a function f, there is a transfer of δ units between nodes i and j (recall that when $\delta = 1$, one unit is transferred). Moreover, node j that participates in this transfer is as described in Observation 2.1: it has the largest index such that the sequence $f(i), f(i+1), \ldots, f(j)$: (i) is monotone and (ii) has consecutive terms that differ by exactly 1.

The definition of the transfer graph is unchanged, but its key property (Lemma 2.2) is scaled by a factor of δ.

Lemma 5.1 (Key property of transfer graph)**.** Let $f : [n] \to \delta\mathbb{Z}$ be an input function given to **BubbleSmooth**. Then for each edge $e_s = (x, y)$ of the final transfer graph T^*, the following holds:

$$\mathrm{vs}_f(x, y) \geq \delta(deg_s^*(x) + deg_s^*(y) - 1).$$

Naturally, Claim 2.3 used to prove Lemma 2.2 is also scaled accordingly. The proof of Lemma 5.1 follows directly from the claim below.

Claim 5.2 (Invariant for phase **Fix**(i))**.** Let $f : [n] \to \delta\mathbb{Z}$ be an input function given to **BubbleSmooth**. At every point during the execution of **BubbleSmooth**(f), for each edge $e_s = (x, y)$ of the transfer graph T,

$$f(x) - f(y) \geq \delta(deg_s(x) + deg_s(y) - 1) + |x - y|.$$

Moreover, for each phase **Fix**(i) for some $i \in [n - 1]$, after each execution of **LinePass**(i), for each edge e_s incident on vertex i, the following (stronger) condition holds:

if the edge $e_s = (i, j)$, i.e., it is outgoing from i, then $f(i) - f(j) \geq \delta(\Delta_i + deg_s(j) - 1) + |i - j|$;

if the edge $e_s = (j, i)$, i.e., it is incoming into i, then $f(j) - f(i) \geq \delta(\Delta_i + deg_s(j) - 1) + |i - j|$.

The proof of Claim 2.3 can be used almost directly to prove Claim 5.2, the only modifications required being the following. First, (1) becomes

$$f^-(i) - f^-(j) \geq \delta(\Delta_i - deg_t^-(i)) + |i - j|$$

(because each transfer lowers the violation score of the pair $(i, i+1)$ by at least δ). Since each transfer moves δ units of mass instead of 1, equations (2) and (3) become, respectively, $f(i) = f^-(i) + \delta \cdot deg_t^-(i)$ and $f(j) = f^-(j) - \delta \cdot deg_t^-(j)$. For the same reason, equation (4) becomes $f(i) = f^-(i) + \delta \cdot deg_t^-(i)$ and $f(j) = f^-(j) + deg_t^-(j)$. The rest of the proof can be used exactly as in the case $\delta = 1$ to obtain Claim 5.2.

With Lemma 5.1 at hand, we can prove Theorem 1.3 for functions $f : [n] \to \delta\mathbb{Z}$ just as before. Indeed, the proof of part 1 of the theorem requires no changes. For part 2, we note that $|f - \textbf{BubbleSmooth}(f)|_1 = 2\delta|E|$, where E is the set of edges in the final transfer graph. Moreover, the same greedy procedure as before yields a matching in the final transfer graph with violation score at least $\delta|E|$; part 2 of Theorem 1.3 then follows from Lemma 2.4.

5.2 Modifications to Section 3

The proof of the dimension reduction uses Claim 3.2. Below we present the corresponding lemma for the modified definition of the basic operator and its proof. This lemma is sufficient to prove Item 3 of Lemma 3.1 for the modified definition of basic operator; the proofs of Items 1 and 2 are unchanged.

5.2.1 Modifications to the basic operator on a square

Claim 5.3 (Properties of modified basic operator on a square)**.** Consider a function $f : \{x_t, x_b, y_t, y_b\} \to \delta\mathbb{Z}$, where vertices are labels of the four vertices of the square $\{0,1\}^2$. (See Figure 3.) Let f' be the function obtained by applying modified basic operators $\mathbb{B}_{x_t,y_t}$ and $\mathbb{B}_{x_b,y_b}$ along the horizontal edges. Then the following holds for the vertical edges.

1. [JR13]: $\mathrm{vs}_{f'}(x_t, x_b) + \mathrm{vs}_{f'}(y_t, y_b) \leq \mathrm{vs}_f(x_t, x_b) + \mathrm{vs}_f(y_t, y_b)$.
2. $|f'(x_t) - f'(x_b)| + |f'(y_t) - f'(y_b)| \leq |f(x_t) - f(x_b)| + |f(y_t) - f(y_b)|$.

Proof. If neither horizontal edge is violated then $f' = f$, and we are done. So assume without loss of generality $\{x_t, y_t\}$ is violated such that $f(y_t) \geq f(x_t) + 1 + \delta$. Then $\mathbb{B}_{x_t,y_t}$ increases $f(x_t)$ by δ and decreases $f(y_t)$ by δ leading to inequality (i): $f'(y_t) \geq f(x_t) + 1$.

If, for both vertical edges, the absolute difference on the values does not increase, we are also done. Assume without loss of generality that the absolute difference of the values of the left vertical edge increases strictly, namely $|f'(x_b) - f'(x_t)| = |f(x_b) - f(x_t)| + \Delta$ for $\Delta \in \{\delta, 2\delta\}$. This implies that $\mathbb{B}_{x_b,y_b}$ did not increase the value at x_b and also that the following holds: (ii) $f(x_b) \leq f(x_t)$. The former also implies that: (iii) $f'(y_b) \leq f'(x_b) + 1$. The definition of Δ further gives that: (iv) $f'(x_b) = f(x_b) - (\Delta - \delta)$ and (v) $f'(y_b) = f(y_b) + (\Delta - \delta)$.

Using inequalities (i), (ii), (iv) and (iii), we get

$$f'(y_t) \geq f(x_t) + 1 \geq f(x_b) + 1 = f'(x_b) + (\Delta - \delta) + 1 \geq f'(y_b) + (\Delta - \delta).$$

Hence $f'(y_t) \geq f'(y_b)$. Further using equation (v), we get

$$|f'(y_t) - f'(y_b)| = f'(y_t) - f'(y_b) = f(y_t) - \delta - (f(y_b) + (\Delta - \delta)) \leq |f(y_t) - f(y_b)| - \Delta.$$

Thus, the absolute difference of the values for the edge $\{x_b, x_t\}$ increases by Δ, whereas this quantity for the edge $\{y_b, y_t\}$ decreases by at least Δ. This completes the proof of the claim. $\square$

5.2.2 Remaining modifications

The proof of Theorem 1.4 for functions $f : [n] \to \delta\mathbb{Z}$ follows exactly as before. The general dimension reduction looses a factor of $1/\delta$ in the bound, as compared to Theorem 1.2. It follows directly from Theorem 1.4 and Observation 3.5.

Theorem 5.4 (Dimension reduction)**.** *For all functions $f : [n]^d \to \delta\mathbb{Z}$, the following holds:*

$$\mathbb{E}_{g \leftarrow L_f}\left[\epsilon^{Lip}(g)\right] \geq \frac{\epsilon^{Lip}(f)}{2 \cdot d \cdot \delta\,\mathrm{ImgD}(f)}.$$

5.3 Modifications to Section 4

Modifications are required neither in the procedure for estimating the image diameter of a function nor in LINE-TESTER. However, the guarantee from Lemma 4.4 for the latter worsens by a factor of δ.

Lemma 5.5. Consider a function $f : [n]^d \to \delta\mathbb{Z}$ that is ϵ-far from Lipschitz and a real number $r \geq \mathrm{ImgD}_{\epsilon/2}(f)$. For a random line $g \leftarrow L_f$, the probability that LINE-TESTER(g, r) rejects is at least $\frac{\epsilon\delta}{40d \cdot r \cdot \log \min\{r,n\}}$.

Proof. The proof is the same as for Lemma 4.4, except that now we construct the truncation f' of the original function f by taking $a < b$ *in* $\delta\mathbb{Z}$ (as opposed to in $\mathbb{Z}$) such that $|a - b| \leq \mathrm{ImgD}_{\epsilon/2}(f)$ and at most $\epsilon n^d/2$ points $x \in [n]^d$ have $f(x) \notin [a, b]$. Define $f'(x)$ in terms of a, b and $f(x)$, as before. The rest of the proof carries through (now using Theorem 5.4). □

The only modification in our final tester, Algorithm 5, is in the number of iterations of the **for** loop, which is multiplied by $1/\delta$. Correctness and the running time of the modified algorithm for functions $f : [n]^d \to \delta\mathbb{Z}$ are analyzed as before. This concludes the proof of Theorem 1.1 for arbitrary $\delta \in (0, 1]$.

References

[ACCL07] Nir Ailon, Bernard Chazelle, Seshadhri Comandur, and Ding Liu. Estimating the distance to a monotone function. *Random Struct. Algorithms*, 31(3):371–383, 2007.

[AJMR12] Pranjal Awasthi, Madhav Jha, Marco Molinaro, and Sofya Raskhodnikova. Testing Lipschitz functions on hypergrid domains. In Anupam Gupta, Klaus Jansen, José D. P. Rolim, and Rocco A. Servedio, editors, *APPROX-RANDOM*, volume 7408 of *Lecture Notes in Computer Science*, pages 387–398. Springer, 2012.

[BGJ+12] Arnab Bhattacharyya, Elena Grigorescu, Kyomin Jung, Sofya Raskhodnikova, and David P. Woodruff. Transitive-closure spanners. *SIAM J. Comput.*, 41(6):1380–1425, 2012.

[BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *J. Comput. Syst. Sci.*, 47(3):549–595, 1993.

[BRW05] Tugkan Batu, Ronitt Rubinfeld, and Patrick White. Fast approximate PCPs for multidimensional bin-packing problems. *Inf. Comput.*, 196(1):42–56, 2005.

[BRY14] Piotr Berman, Sofya Raskhodnikova, and Grigory Yaroslavtsev. L_p-testing. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 164–173. ACM, 2014.

[CDJS15] Deeparnab Chakrabarty, Kashyap Dixit, Madhav Jha, and C. Seshadhri. Property testing on product distributions: Optimal testers for bounded derivative properties. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015*, 2015.

[CS10] Artur Czumaj and Christian Sohler. Sublinear-time algorithms. In Oded Goldreich, editor, *Property Testing*, volume 6390 of *Lecture Notes in Computer Science*, pages 41–64. Springer, 2010.

[CS13] Deeparnab Chakrabarty and C. Seshadhri. Optimal bounds for monotonicity and Lipschitz testing over hypercubes and hypergrids. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 419–428. ACM, 2013.

[DGL+99] Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. Improved testing algorithms for monotonicity. In *RANDOM*, pages 97–108, 1999.

[DJRT13] Kashyap Dixit, Madhav Jha, Sofya Raskhodnikova, and Abhradeep Thakurta. Testing the Lipschitz property over product distributions with applications to data privacy. In *TCC*, pages 418–436, 2013.

[EKK+00] F. Ergun, S. Kannan, S. R. Kumar, R. Rubinfeld, and M. Viswanathan. Spot-checkers. *J. Comput. Syst. Sci.*, 60(3):717–751, 2000.

[FLN+02] Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. Monotonicity testing over general poset domains. In *STOC*, pages 474–483, 2002.

[GGL+00] Oded Goldreich, Shafi Goldwasser, Eric Lehman, Dana Ron, and Alex Samorodnitsky. Testing monotonicity. *Combinatorica*, 20(3):301–337, 2000.

[GGR98] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998.

[Gol11] Oded Goldreich. Introduction to testing graph properties. In Oded Goldreich, editor, *Studies in Complexity and Cryptography*, volume 6650 of *Lecture Notes in Computer Science*, pages 470–506. Springer, 2011.

[Gro99] Mikhail Gromov. *Metric Structures for Riemannian and non-Riemannian Spaces*. 1999.

[HK08] Shirley Halevy and Eyal Kushilevitz. Testing monotonicity over graph products. *Random Struct. Algorithms*, 33(1):44–67, 2008.

[JPRZ09] Charanjit S. Jutla, Anindya C. Patthak, Atri Rudra, and David Zuckerman. Testing low-degree polynomials over prime fields. *Random Struct. Algorithms*, 35(2):163–193, 2009.

[JR13] Madhav Jha and Sofya Raskhodnikova. Testing and reconstruction of Lipschitz functions with applications to data privacy. *SIAM J. Comput.*, 42(2):700–731, 2013.

[KR06] Tali Kaufman and Dana Ron. Testing polynomials over general fields. *SIAM J. Comput.*, 36(3):779–802, 2006.

[McD89] Colin McDiarmid. On the method of bounded differences. In *Surveys in Combinatorics*. London Math. Soc. Lecture Notes 141, 1989.

[Oll09] Yann Ollivier. Ricci curvature of markov chains on metric spaces. *Journal of Functional Analysis*, 256(3):810–864, 2009.

[Ron09] Dana Ron. Algorithmic and analysis techniques in property testing. *Foundations and Trends in Theoretical Computer Science*, 5(2):73–205, 2009.

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterization of polynomials with applications to program testing. *SIAM J. Comput.*, 25(2):252–271, 1996.

[RS11] Ronitt Rubinfeld and Asaf Shapira. Sublinear time algorithms. *SIAM J. Discrete Math.*, 25(4):1562–1588, 2011.

[SV14] C. Seshadhri and Jan Vondrák. Is submodularity testable? *Algorithmica*, 69(1):1–25, 2014.

A McDiarmid's Inequality

We state the well-known McDiarmid's inequality [McD89] specialized to the domain $[n]^d$.

Theorem A.1 ([McD89]). *For every Lipschitz function $f : [n]^d \to \mathbb{R}$ and uniformly distributed $X \in [n]^d$, the following holds (where the expectation is taken over the uniformly distributed X):* $\Pr[|f(X) - \mathbb{E}[f(X)]| \geq t] \leq 2e^{\frac{-2t^2}{dn^2}}$.

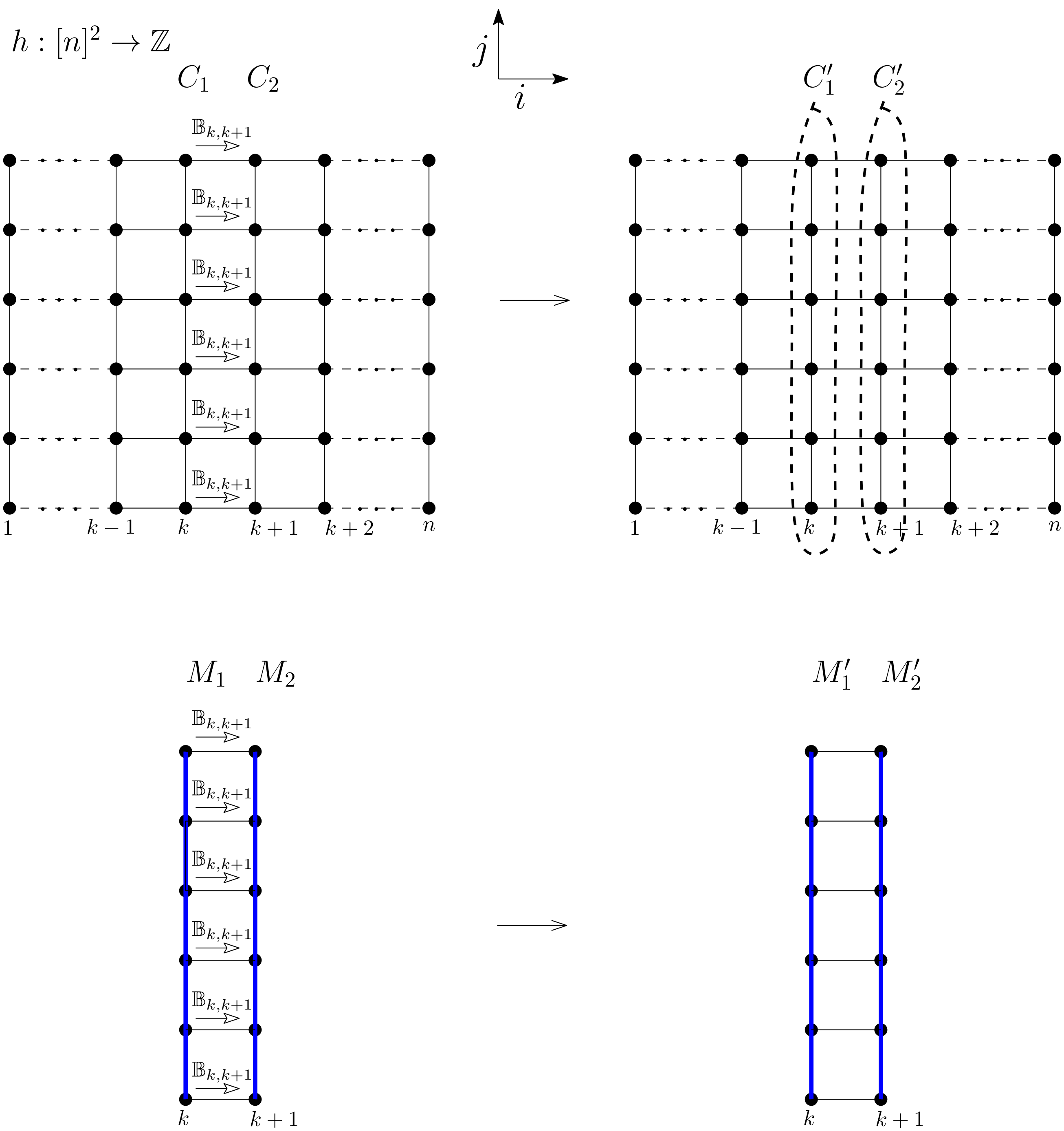


Figure 4: Illustration of proof of Item 3 of Lemma 3.1.