Testing k-Submodularity

Themistoklis Haris
Department of Computer Science, Boston University, MA, USA

Diptaksho Palit
Department of Computer Science, Boston University, MA, USA

Abstract

We initiate the study of property testing for k–submodular functions, a higher-dimensional analogue of submodular functions defined on partial partitions of a ground set. While k–submodularity retains the diminishing-returns flavor of ordinary submodularity, it also introduces a pairwise monotonicity constraint comparing competing assignments of the same element. This additional local structure makes the testing problem qualitatively different from the classical case.

Our results show a sharp contrast between distance regimes. In the ℓ_p regime for $p \geq 1$, we prove that every bounded k–submodular function is close to a junta on the hypergrid. Combined with an implicit-learning tester for hypergrid domains, this yields a constant-query tester for k–submodularity. In the Hamming distance regime, k–submodularity admits two qualitatively different local witnesses – violated *squares* for diminishing marginal gains, and violated *triangles* for pairwise-monotonicity failures – and the latter has no counterpart at $k = 1$. We prove density theorems for both witness types via repair on filters and ideals of partial partitions, yielding non-adaptive, one-sided sub-exponential-query testers for the two component properties of k–submodularity. We then exhibit a configuration in which the two repair directions are forced into opposition on a shared vertex, identifying a structural barrier to combining these into a tester for the full property.

Finally, for bounded-range functions, we give an adaptive tester for monotone k–submodularity via a pseudo-DNF representation and learning on the hypergrid. Several of the structural and learning tools developed here may be useful for testing other properties over product domains.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms

Keywords and phrases property testing, sublinear algorithms, submodular functions

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2026.52

Category RANDOM

Related Version *Full Version*: `https://arxiv.org/pdf/2606.30433` [9]

Funding *Diptaksho Palit*: Supported by the U.S. National Science Foundation under Grant No. 2022446.

Acknowledgements T.H. worked on k-submodularity testing as a final project for the 2024 Sublinear Algorithms course taught by Ronitt Rubinfeld. T.H. and D.P. started working on this project for a course on Sublinear Algorithms taught by Sofya Raskhodnikova [14]. We thank Fabian Spaeh for inspiring us to investigate this question through his work on k-submodular optimization.

1 Introduction

Submodularity is one of the canonical examples of a global property with a simple local interpretation. A set function is submodular when it satisfies the principle of *diminishing marginal returns*: the incremental value of adding an element can only decrease as the set to which it is added grows. Formally, a set function $f : 2^{[n]} \to \mathbb{R}$ is said to be *submodular* if, for all subsets $X, Y \subseteq [n]$, it satisfies the inequality:

$$f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y).$$

This local-to-global nature makes submodularity a natural target for property testing [8], where one is given oracle access to an unknown function and must distinguish functions satisfying a property from those that are far from satisfying it. Yet even for ordinary submodularity, the query complexity of testing remains poorly understood. Seshadhri and Vondrák [16] gave a tester using roughly $(1/\varepsilon)^{\sqrt{n}}$ queries and proved a linear lower bound; the latter was later improved to a nearly quadratic lower bound by Hatami and Vondrák [10]. The resulting exponential gap has remained open, and has motivated the study of more structured regimes, such as bounded-range submodular functions [15] and testing under ℓ_p distance rather than Hamming distance [1].

In this work we ask what happens to this picture for a higher-dimensional analogue of submodularity. For a fixed integer $k \geq 1$, a point in $[k+1]^n$ can be viewed as a *partial partition* of the ground set $[n]$: each element is either assigned to one of k parts, or left unassigned. A function on these partial partitions is *k–submodular* if marginal gains diminish no matter which part an element is assigned to. The case $k = 1$ is ordinary submodularity, while already for $k = 2$ the domain has multiple incompatible directions in which a single element may be added. This function class has been studied extensively in optimization [12, 4], with applications including tensor placement [12] and online ad allocation [5].

From a testing perspective, k–submodularity is not merely submodularity on a larger domain. A useful characterization, due to Ward and Živný [17], decomposes k–submodularity into two conditions. The first is *diminishing marginal gains*, which generalizes the familiar submodular inequality inside each direction of the hypergrid. The second is *pairwise monotonicity*, a constraint comparing the marginal gains of assigning the same element to two different parts. One way to understand this decomposition is to separate two kinds of local interactions. When two partial partitions make compatible choices, k–submodularity behaves like ordinary diminishing returns along the corresponding direction. When they make incompatible choices for the same element, the join operation discards that element from both parts; the resulting inequality says that assigning the element to two different parts cannot have two strongly negative marginal gains at the same time. This second condition has no analogue when $k = 1$, and it is the source of much of the new combinatorial structure in the testing problem.

We initiate the study of testing k–submodularity. Our results show a sharp contrast between two distance regimes. In the ℓ_p regime, analytic structure survives the passage from the Boolean hypercube to the hypergrid: we prove a junta-approximation theorem for k–submodular functions and combine it with an implicit-learning tester on hypergrids. In the Hamming distance regime, the local structure fragments: we obtain testers for the two local constraints by developing square and triangle witness structures, and show that their repair procedures expose a genuine obstruction to the standard local-repair paradigm for full k–submodularity. Along the way, we develop testing and learning tools for hypergrid domains that may be useful beyond this problem.

1.1 Our Results

We organize our contributions around three questions. First, does the analytic structure that makes submodularity testable in ℓ_p distance survive on the richer domain of partial partitions? Second, in the Hamming distance regime, can the local-violation framework for submodularity be adapted to the two distinct constraints that characterize k–submodularity? Third, can bounded range assumptions recover efficient testability for a meaningful subclass of k–submodular functions?

ℓ_p testing from hypergrid junta structure

Our first result is a constant-query tester for k–submodularity in ℓ_p distance. Here and throughout this section, constant means independent of the ambient dimension n.

▶ **Theorem 1.** *For all $p \geq 1$, there exists an adaptive, one-sided error ℓ_p-tester for k-submodularity making $q = \left(\frac{p}{\varepsilon}\right)^2 \cdot 2^{\left(\frac{p}{\varepsilon}\right)^2 \log\left(\frac{p}{\varepsilon}\right)}$ queries.*

The main structural input is not a black-box reduction to the submodular case. We prove that every bounded k–submodular function is close to a function depending on only a small number of coordinates, even though each coordinate can move in k competing directions. This requires choosing influential coordinates in a way that respects all parts of a partial partition, and uses the fact that k–submodular functions satisfy a self-bounding variance inequality.

▶ **Theorem 2.** *Let $f : [k+1]^J \to [0,1]$ be a k–submodular function and $\varepsilon > 0$ be a small enough constant. Then there exists a k–submodular function $h : [k+1]^J \to [0,1]$ that depends only on $J' \subseteq J$ dimensions, where*

$$|J'| = O\left(\frac{1}{\varepsilon^2} \cdot \log \frac{1}{\varepsilon}\right),$$

and the function h satisfies

$$\|f - h\|_2 \leq \varepsilon.$$

We then develop the testing component needed to turn this structure into an algorithm. The implicit-learning framework of Blais and Bommireddi [1] relies on identifying influential coordinates and checking consistency with a small junta. We formulate and analyze this framework on hypergrid domains using Fourier analysis over generalized product spaces, and simplify parts of the original argument by separating the learning and final testing samples.

▶ **Theorem 3.** *Let $p \geq 1$. Suppose $\mathcal{P} \subseteq \{f : [k+1]^n \to [0,1]\}$ is a property of functions such that for any $f \in \mathcal{P}$ there exists a κ-junta $h : [k+1]^n \to [0,1]$ satisfying $||f - h||_2 \leq \varepsilon$. Then, there exists a tester that, given query access to an unknown function f, will make $\varepsilon^{-9} \cdot 2^{O(\kappa \log \kappa)}$ queries to f and with probability at least $2/3$ accept if $f \in \mathcal{P}$ and reject if f is ε-far from $\mathcal{P}$ in the ℓ_2 norm.*

Local testing in Hamming distance

Our second set of results addresses the Hamming-distance regime, where the Seshadhri–Vondrák local-repair paradigm for ordinary submodularity does not extend directly. Pairwise monotonicity is vacuous when $k = 1$, but on the hypergrid its violations form a new witness shape – *triangles* – incomparable with the *squares* that witness diminishing-marginal-gain failures. Repair likewise shifts from Boolean subcubes to filters and ideals of partial partitions, and the density arguments must be re-proved in this setting. We obtain non-adaptive, one-sided testers for the two component properties of k–submodularity:

▶ **Theorem 4.** *Let $f : [k+1]^n \to \mathbb{R}$ be given via oracle access. There exist non-adaptive, one-sided error testers for the diminishing marginal gains and pairwise monotonicity properties with the following guarantees:*

- *If f satisfies the property, the tester accepts with probability 1.*
- *If f is ε-far from the property (i.e., at least $\varepsilon \cdot [k+1]^n$ values must be modified to satisfy it), the tester rejects with probability at least $2/3$.*

The tester for diminishing marginal gains makes $q = 4n^2 \cdot (1/\varepsilon)^{\Theta(\sqrt{n \log n})}$ *queries; the tester for pairwise monotonicity makes* $q = 3kn \cdot (1/\varepsilon)^{\Theta(\sqrt{n \log n})}$ *queries.*

Notably, only the pairwise-monotonicity tester has query complexity depending on k, despite k governing the structure of the domain.

Combined, these results expose a structural difference between k–submodularity and ordinary submodularity that the analytic (ℓ_p) regime does not see. In Section 4.3 we exhibit an explicit configuration (Figure 1) on a high item-weight filter in which repairing a triangle violation forces value changes in the *opposite* direction from those needed to repair an overlapping square violation, on a shared vertex. The two repair directions are not merely incompatible at one point – they are jointly inconsistent across an entire region. We record this as Observation 32: it rules out the natural component-wise extension of local repair to full k–submodularity, and isolates a concrete barrier that any sub-exponential local-witness tester must overcome by enlarging the witness vocabulary beyond squares and triangles.

Bounded range and monotone k–submodularity

Finally, we show that efficient Hamming-distance testing can be recovered for monotone k–submodular functions with bounded range. The result is again obtained through learning, but the relevant representation is no longer a junta. Instead, we utilize pseudo-DNF representations for functions on the hypergrid and prove that bounded-range monotone k–submodular functions have bounded-width pseudo-DNFs.

▶ **Theorem 5.** *Let* $\varepsilon, \delta \in (0, 1)$ *be constants. There exists a tester* $\mathcal{T}$ *which, given query access to a function* $f : [k+1]^n \to \{0, 1, \cdots, r\}$ *satisfies the following.*

1. *If* f *is monotone* k*–submodular, then* $\mathcal{T}$ **accepts** *with probability* $\geq 1 - \delta$.
2. *If* f *is* ε*-far from monotone* k*–submodularity, then* $\mathcal{T}$ **rejects** *with probability* $\geq 1 - \delta$.
3. $\mathcal{T}$ *makes at most* $\mathsf{poly}(n, 1/\varepsilon, 1/\delta, (kr)^{O(rk^2 \log(r/\varepsilon))})$ *queries.*

The learning result for bounded-width pseudo-DNFs uses a hypergrid version of the switching-lemma/Fourier-sparsity route and yields a tester via the Kushilevitz-Mansour algorithm. We expect this learner to be useful beyond k–submodularity.

1.2 Prior Work

Testing whether a given function is submodular was first examined by Parnas, Ron and Rubinfeld [13], who studied the limited setting of testing for a class of matrices called *Monge matrices.* In its full generality, the problem was considered by Seshadhri and Vondrák [16]. They developed the local-witness and repair framework for Hamming-distance testing on the Boolean hypercube, proving an upper bound of $(1/\varepsilon)^{O(\sqrt{n \log n})}$ and a linear lower bound. The latter was later improved to $\Omega(n^2/\log n)$ by Hatami and Vondrák [10]. Blais and Bommireddi [1] subsequently studied the problem in the ℓ_p norm, and using an approximation result from [6], designed an implicit-learning tester with constant query complexity in the input length. Raskhodnikova and Yaroslavtsev [15] studied bounded-range submodular functions through a learning-based approach and proved a polynomial-time upper bound in this regime. Chakrabarty and Huang [3] studied the problem of testing for coverage functions, which are a special type of submodular function.

Our work is informed most directly by the three submodularity-testing frameworks of Seshadhri and Vondrák [16], Blais and Bommireddi [1], and Raskhodnikova and Yaroslavtsev [15]. These works provide the conceptual starting points for the three regimes we study.

The passage to k–submodularity introduces new difficulties in each case: local witnesses split into squares and triangles in the Hamming regime, influential coordinates must account for competing assignments in the ℓ_p regime, and bounded-range learning requires representations over the hypergrid rather than the Boolean cube.

To our knowledge, there is no prior work on testing k-submodularity, though the property has been studied extensively from the optimization perspective [12, 4]. We also use the structural characterization of Ward and Živný [17], which identifies k–submodularity with the conjunction of diminishing marginal gains and pairwise monotonicity.

1.3 Technical Overview

The main technical theme is that the hypergrid introduces several inequivalent notions of locality. In the ℓ_p regime, locality is analytic and is captured by influential coordinates. In the Hamming regime, locality is combinatorial and is captured by small violated configurations. For bounded-range functions, locality appears through compact symbolic representations. In each part of the paper we develop the corresponding notion of locality for partial partitions.

The analytic regime: finding the right influential coordinates

The ℓ_p tester has two ingredients: a junta-approximation theorem for k–submodular functions, and a tester for properties that are close to juntas on the hypergrid. The obstacle in the first ingredient is that a coordinate has k possible nonzero directions. Several natural ways of declaring a coordinate influential in one part at a time almost work, but fail to interact correctly with the boosting lemma for down-monotone families (Lemma 12).

Our solution is to define the relevant influence threshold by looking across all partitions of the remaining coordinates. This lets the selection rule remain compatible with the combinatorial boosting argument while still respecting the multi-directional nature of the domain. The second key input is a variance bound: we prove that k–submodular functions are 2-self bounding (Lemma 14). Together, these ingredients yield a junta depending on only $O(\varepsilon^{-2}\log(1/\varepsilon))$ coordinates.

Once the junta structure is established, the testing step is conceptually separate. We analyze implicit learning directly on $[k+1]^n$ using Fourier analysis over product domains. The tester identifies a candidate set of influential coordinates and then checks whether the observed behavior is consistent with a junta from the target property. Using an independent sample in the final testing stage also simplifies the decorrelation issues that arise in the original Boolean-domain analysis of [1].

The Hamming regime: local witnesses and repair

In Hamming distance, a tester needs many local witnesses whenever the function is far from the property. For ordinary submodularity, the relevant witness is a violated square in the Boolean hypercube, and the analysis of Seshadhri and Vondrák [16] shows that few violated squares imply a global repair.

For k–submodularity, a single witness shape is no longer sufficient. Using the characterization of Ward and Živný [17], we separate the property into diminishing marginal gains and pairwise monotonicity. Diminishing marginal gains is witnessed by generalized violated squares, while pairwise monotonicity is witnessed by violated triangles. We prove density statements for both witnesses by designing repair procedures on large inclusion-closed regions of the hypergrid: ideals $\{Y \preceq X\}$ and filters $\{Y \succeq X\}$. This gives sub-exponential testers for the two component properties.

The same repair viewpoint also yields a structural observation about the full Hamming-distance problem (Observation 32, illustrated in Figure 1). On high item-weight filters, the repair direction needed for a violated square is forced into conflict with the repair direction needed for an overlapping violated triangle on a shared vertex. This is not a limitation of our proof – it is a structural property of the problem: any tester built on the squares-and-triangles vocabulary, no matter how the repair is organized, must encounter a configuration of this kind. A sub-exponential tester for full k–submodularity in Hamming distance therefore requires an enriched witness vocabulary, not merely a more clever combination of square and triangle repairs.

The bounded-range regime: representation before testing

For bounded-range monotone k–submodular functions, we take a representation-theoretic route. We first show that such functions admit bounded-width pseudo-DNF representations over the hypergrid. This representation captures the monotone threshold-like structure imposed jointly by bounded range and diminishing marginal gains.

We then prove a hypergrid switching lemma for pseudo-DNFs, using low-depth k-ary decision trees in place of binary trees. Combined with Fourier analysis on the hypergrid, this implies sparse Fourier approximation for bounded-width pseudo-DNFs. The resulting learner, together with the Kushilevitz-Mansour algorithm [11], yields the bounded-range tester.

1.4 Discussion and Open Problems

Our results leave several natural questions open. The most immediate is whether full k–submodularity can be tested in Hamming distance with sub-exponential query complexity. Our component testers for diminishing marginal gains and pairwise monotonicity show that each constraint has many local witnesses when violated, but the repair procedures for the two constraints can be forced to conflict. Resolving this – either by exhibiting a richer witness family with a compatible repair procedure, or by proving a sub-exponential lower bound that confirms the barrier is fundamental – is, in our view, the right next step in this regime.

A second direction is to understand the bounded-range setting without monotonicity. Our tester in this regime relies on a pseudo-DNF representation for monotone k–submodular functions, and it is unclear whether a comparable representation exists for non-monotone functions.

Finally, it would be interesting to prove lower bounds for testing k–submodularity. The known lower bounds for submodularity do not transfer directly, since natural embeddings can destroy pairwise monotonicity. Thus even formulating the right hard instances appears to require using the genuinely multi-part structure of the domain.

1.5 Organization

The paper is organized as follows. In Section 2, we formally define the problem and related structures on the hypergrid. In Section 3, we prove the junta approximability theorem for k–submodular functions. In Section 4, we develop Hamming-distance testers for *diminishing marginal gains* and *pairwise monotonicity*, and analyze the obstruction to combining their repairs into a local-repair tester for full k–submodularity. We refer to the full version of the paper [9] for the analysis of the hypergrid implicit-learning theorem and construction of our ℓ_0 tester for monotone k–submodularity in the bounded range regime.

2 Preliminaries

▶ **Definition 6** (Partial partition). *A* partial partition *of* $[n]$ *into* k *subsets is defined* $X = (X_1, ..., X_k)$*, where each* $X_i \subseteq [n]$ *and* $X_i \cap X_j = \emptyset$ *for all* $i \neq j \in [k]$*. We let* $X_{k+1} = [n] \setminus \cup_{i=1}^{k} X_i$*, and this allows us to associate* partial partitions *of* $[n]$ *into* k *subsets with the hypergrid* $[k+1]^n$ *(each element of* $[n]$ *is associated to a bucket in* $[k+1]$*, yielding a vector in* $[k+1]^n$*). If* $e \notin X_{k+1}$*, we say that* $e \in X$*. We impose a* partial order *on* partial partitions *by defining* $X \preceq Y$ *if and only if* $X_i \subseteq Y_i$ *for all* $i \in [k]$*. We let* $|X|$ *denote the* item-weight *of a partial partition, which is the total number of items picked. Numerically, this is equal to* $\sum_{i=1}^{k} |X_i|$*.*

▶ **Definition 7** (Marginal gains). *Define the* marginal gains *of a function* f *on a partial partition* X *with respect to an index* $i \in [k]$ *and an element* $e \in X_{k+1}$ *as*

$$\Delta_f(X, i, e) := f(X_1, ..., X_i \cup \{e\}, ..., X_k) - f(X).$$

Letting $X_e^i := (X_1, ..., X_i \cup \{e\}, ..., X_k)$*, we can rewrite this as* $\Delta_f(X, i, e) = f(X_e^i) - f(X)$*.*

▶ **Definition 8** (k–submodularity). *A function* $f : [k+1]^n \to \mathbb{R}$ *is* k–submodular *if, for all partial partitions* X, Y*, the following holds.*

$$f(X) + f(Y) \geq f(X \sqcup Y) + f(X \sqcap Y),$$

where $X \sqcap Y := (X_1 \cap Y_1, \cdots, X_k \cap Y_k)$ *and*

$$X \sqcup Y = \left((X_1 \cup Y_1) \setminus \bigcup_{i \neq 1} (X_i \cup Y_i), \cdots, (X_k \cup Y_k) \setminus \bigcup_{i \neq k} (X_i \cup Y_i) \right).$$

The following theorem of Ward and Živný [17] gives us an equivalent definition of k–submodularity, one that is described by two more intuitive properties. The intuition is that every local comparison between two partial partitions can be decomposed into compatible and conflicting moves. Compatible moves are governed by the usual diminishing-returns principle: adding an unassigned element to part i becomes less valuable as the current partial partition grows. Conflicting moves arise only when the same element could be assigned to two different parts. For example, comparing X_e^i and X_e^j gives $X_e^i \sqcap X_e^j = X$ and $X_e^i \sqcup X_e^j = X$, so the k–submodular inequality becomes

$$\Delta_f(X, i, e) + \Delta_f(X, j, e) \geq 0.$$

Thus pairwise monotonicity is the local condition that prevents two competing assignments of the same element from both being too costly.

▶ **Theorem 9** ([17]). *A function* $f : [k+1]^n \to \mathbb{R}$ *is* k–submodular *if and only if the following two conditions hold.*

- ***Diminishing marginal gains.*** *For all partial partitions* X, Y *such that* $X \preceq Y$*,* $e \notin Y$*, and* $i \in [k]$*, the following holds.*

  $$\Delta_f(X, i, e) \geq \Delta_f(Y, i, e).$$

- ***Pairwise monotonicity.*** *For all partial partitions* X*,* $e \notin X$*, and* $i \neq j \in [k]$*, the following holds.*

  $$\Delta_f(X, i, e) + \Delta_f(X, j, e) \geq 0.$$

3 Junta approximation of k–submodular functions

In the following two sections, we prove the structural and algorithmic ingredients behind our ℓ_p tester for k–submodularity. The structural ingredient is a junta approximation theorem for k–submodular functions on the hypergrid. The main difficulty is choosing influential coordinates in a way that accounts for all k directions of a coordinate while still giving a dimension-independent approximation.

This approximation theorem, combined with an implicit learning tester for properties on hypergrids with a good junta approximation, yields the desired tester.

We prove the following approximation theorem.

▶ **Theorem 10** (Reminder of Theorem 2). *Let $f : [k+1]^J \to [0,1]$ be a k–submodular function and $\varepsilon > 0$ be a small enough constant. Then there exists a k–submodular function $h : [k+1]^J \to [0,1]$ that depends only on $J' \subseteq J$ dimensions, where*

$$|J'| = O\left(\frac{1}{\varepsilon^2} \cdot \log \frac{1}{\varepsilon}\right),$$

and the functions h satisfies

$$\|f - h\|_2 \leq \varepsilon.$$

Before we get to the proof of Theorem 2, we first set up some necessary machinery.

3.1 Setup

There are three main parts to the setup.

Boosting lemma. We cite the following boosting lemma due Goemans and Vondrák [7].

▶ **Definition 11** (Down-monotone family). *Let X be a finite set and $\mathcal{F} \subseteq 2^X$ be a collection of subsets of X. The family $\mathcal{F}$ is called down-monotone if the following condition holds.*

$$A \in \mathcal{F} \wedge B \subseteq A \implies B \in \mathcal{F}.$$

▶ **Lemma 12** ([7] Boosting lemma). *Let X be a finite set and $\mathcal{F} \subseteq 2^X$ be a down-monotone family. For $p \in (0,1)$, define*

$$\sigma_p := \Pr[X(p) \in \mathcal{F}],$$

where $X(p)$ is a random subset of X such that each element is picked independently with probability p. Then

$$\sigma_p = (1-p)^{\phi(p)},$$

where $\phi(p)$ is a non-decreasing function for $p \in (0,1)$.

Self–bounding functions and concentration. A very useful tool in bounding the variance of submodular functions comes from the literature of concentration for *self–bounding* functions.

▶ **Definition 13** (Self–bounding function). *A function $f : \mathcal{X}^n \to [0,1]$ is **α-self–bounding** if there exist functions $f_i : \mathcal{X}^{n-1} \to [0,1]$ such that for all $x_1, \cdots, x_n \in \mathcal{X}$ and all $i \in [n]$, the following conditions hold.*

1. $0 \leq f(x_1, \cdots, x_n) - f_i(x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n) \leq 1$*; and*
2. $\sum_{i=1}^{n} (f(x_1, ..., x_n) - f_i(x_1, x_2, ..., x_{i-1}, x_{i+1}, ..., x_n)) \leq \alpha f(x_1, ..., x_n)$.

▶ **Lemma 14** ([2] Bounded variance lemma). *If a function $f : \mathcal{X}^n \to [0,1]$ is α-self–bounding then*

$$\mathrm{Var}[f(X)] \leq \alpha \cdot \mathbb{E}[f(X)].$$

Bounded variance of k–submodular functions. Submodular functions that are Lipschitz are self–bounding, which allows us to bound their variance. We show that a similar property holds for k–submodular functions.

▶ **Definition 15** (Lipschitz function). *A function $f : [k+1]^n \to [0,1]$ is α-Lipschitz if $|\Delta_f(X, j, i)| \le \alpha$ for all $i \in [n]$ and $j \in [k]$.*

▶ **Lemma 16.** *Let $f : [k+1]^n \to [0,1]$ be 1–Lipschitz and k–submodular. Then:*

$$\mathrm{Var}[f] \le 2 \cdot \mathbb{E}[f].$$

Proof. We prove this by showing that f is 2-self–bounding, and invoking Lemma 14. For $i \in [n]$, define the function f_i as

$$f_i(x_1, \cdots, x_{i-1}, x_{i+1}, \cdots x_n) := \min_{j \in [k+1]} \{f(x_1, \cdots, x_{i-1}, j, x_{i+1}, x_n)\}.$$

Then the first condition

$$0 \le f(x_1, ..., x_n) - f_i(x_1, ..., x_{i-1}, x_{i+1}, ..., x_n) \le 1$$

holds trivially (by choice of minimum value and the range of f being $[0, 1]$). By pairwise monotonicity of f, the following holds.

$$\begin{aligned} &f(x_1, ..., x_n) - f_i(x_1, ..., x_{i-1}, x_{i+1}, ..., x_n) \\ &\qquad \le 2 \cdot (f(x_1, ..., x_n) - f(x_1, ..., x_{i-1}, k+1, x_{i+1}, ..., x_n)) \,. \end{aligned}$$

Here, $x_i = k + 1$ corresponds to item i not being put into any bucket. Finally, the following holds for all $i \in [n]$ due to the marginal gains property.

$$\begin{aligned} &f(x_1, ..., x_n) - f(x_1, ..., x_{i-1}, k+1, x_{i+1}, ..., x_n) \\ &\le f(x_1, \cdots, x_i, k+1, \cdots, k+1) - f(x_1, \cdots, x_{i-1}, k+1, \cdots, k+1). \end{aligned}$$

This yields the desired condition.

$$\begin{aligned} &\sum_{i=1}^{n} (f(x_1, \cdots, x_n) - f_i(x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n)) \\ &\qquad \le \sum_{i=1}^{n} 2 \cdot ((f(x_1, \cdots, x_n) - f(x_1, \cdots, x_{i-1}, k+1, x_{i+1}, \cdots, x_n)) \\ &\qquad \le \sum_{i=1}^{n} 2 \cdot (f(x_1, \cdots, x_i, k+1, \cdots) - f(x_1, \cdots, x_{i-1}, k+1, \cdots)) \\ &\qquad \le 2 \cdot f(x_1, \cdots, x_n). \end{aligned}$$

We have thus shown that f is 2-self–bounding, and by Lemma 16, $\mathrm{Var}[f] \le 2 \cdot \mathbb{E}[f]$. ◀

3.2 Weak junta approximation

We first prove the following "weak" junta approximation for k–submodular functions.

▶ **Theorem 17** (Weak junta approximation). *Let $f : [k+1]^J \to [0,1]$ be a k–submodular function and $\varepsilon \in (0,1)$ be a constant. Then there exists a k–submodular function $h : [k+1]^J \to [0,1]$ that depends only on $J' \subseteq J$ dimensions, where*

$$|J'| \le \frac{32}{\varepsilon^2} \cdot \log \frac{8|J|}{\varepsilon^2},$$

and the function h satisfies

$$\|f - h\|_2 := \left(\mathbb{E}_{x \in [k+1]^J} \left[(f(x) - h(x))^2\right]\right)^{\frac{1}{2}} \le \frac{1}{2} \cdot \varepsilon.$$

We prove Theorem 17 by first describing a procedure to construct the set J' and then showing that there exists a J'-junta satisfying the desired properties.

Let $S \subseteq J$ and $P : J \to [k+1]^n$ be some partition function over J. We denote by S_P the point on the hypergrid obtained by placing each $i \in S$ into bucket $P(i)$. We denote by $S(\delta)_P$ a random set generated by sampling each element in S independently with probability δ.

Algorithm 1 Identifying the junta set J'.

1: **Input:** Oracle access to k–submodular function $f : [k+1]^J \to [0,1]$; parameters $\alpha, \delta > 0$

2: $S \leftarrow \emptyset$.

3: **while** there exists $i \notin S$ such that for all partitions P of J we have $\Pr\left[\Delta_f(S(\delta)_P, P(i), i) > \alpha\right] > \frac{1}{2}$ **do**

4: $\quad S \leftarrow S \cup \{i\}$

5: **return** $J' = S$.

First we show that the number of indices picked by Algorithm 1 is small.

▷ Claim 18. The set J' output by Algorithm 1 satisfies $|J'| \leq \frac{2}{\alpha\delta}$.

Proof. Let P be an arbitrary partition of J, and X be an arbitrary subset of J. For any $i \in X$, let $X_{<i}$ denote the set of items in X that were added to S before i in Algorithm 1. Define

$$X_P^+ := \{i \in X \mid \Delta_f(X_{<i,P}, P(i), i) > \alpha\}.$$

Then by *diminishing marginal gains*, at the end of the execution of Algorithm 1, we have

$$\begin{aligned}\mathbb{E}[f(S(\delta)_P^+)] &= f(\emptyset) + \sum_{i \in S(\delta)_P^+} \Delta_f(S(\delta)_{<i,P}^+, P(i), i) \\ &> \mathbb{E}[|S(\delta)_P^+|] \cdot \alpha. && \text{[Assuming } f(\emptyset) = 0 \text{ WLOG]}\end{aligned}$$

Next, we show that $\mathbb{E}[|S(\delta)_P^+|] \geq \frac{\delta \cdot |S|}{2}$. If that is true, then

$$1 \geq \mathbb{E}[f(S(\delta)_P^+)] > \frac{\delta \cdot |S| \cdot \alpha}{2} \implies |S| \leq \frac{2}{\delta\alpha}.$$

To prove our claim, we observe that by construction, for all $u \in S$,

$$\Pr[u \in S(\delta)_P^+] = \Pr[u \in S(\delta)] \cdot \Pr[\Delta_f(S(\delta)_{<u,P}, P(u), u) > \alpha] > \frac{\delta}{2}.$$

Writing $|S(\delta)_P^+| = \sum_{u \in S} X_u$, where X_u is the indicator for whether $u \in S(\delta)_P^+$, we get

$$\mathbb{E}[|S(\delta)_P^+|] = \sum_{u \in S} \Pr[X_u] \geq \frac{\delta \cdot |S|}{2}.$$ ◁

Next, we show that for all partitions P and any "unimportant" coordinate $i \in J \setminus J'$, the probability of the marginal gains on top of J' being large with respect to P when adding i to $P(i)$ is very small if we sample elements from J' with probability $\frac{k}{k+1}$ rather than δ.

▷ Claim 19. For all $i \in J \setminus J'$, all $j \in [k]$, and all partitions $P : J' \to [k]$, the following holds.

$$\Pr_{J'\left(\frac{k}{k+1}\right)}\left[\Delta_f\left(J'\left(\frac{k}{k+1}\right)_P, j, i\right) > \alpha\right] \leq \left(\frac{1}{k+1}\right)^{\frac{1}{2\delta}}.$$

Proof. Consider the following family of points

$$\mathcal{F} = \{X \in [k+1]^n \mid X \preceq J'_P \text{ and } \Delta_f(X, j, i) > \alpha\}$$

This is a down-monotone set – if $Y \preceq X \in \mathcal{F}$ then $Y \in \mathcal{F}$. This is due to *diminishing marginal gains* – $\Delta_f(Y, j, i) \geq \Delta(X, j, i)$. If we let $\sigma_p := \Pr[J'(p)_P \in \mathcal{F}]$ for $p \in (0, 1)$, then by Lemma 12,

$$\sigma_p = (1-p)^{\phi(p)},$$

for some non-decreasing function ϕ. The termination condition of Algorithm 1 guarantees that

$$\sigma_\delta := \Pr[J'(\delta)_P \in \mathcal{F}] = (1-\delta)^{\phi(\delta)} \leq \frac{1}{2},$$

which implies $\phi(\delta) \geq \frac{1}{2\delta}$ for $\delta \leq \frac{1}{2}$. Since ϕ is non-decreasing, we have

$$\phi\left(\frac{k}{k+1}\right) \geq \phi(\delta) \geq \frac{1}{2\delta} \implies$$
$$\sigma_{\frac{k}{k+1}} = \left(1 - \frac{k}{k+1}\right)^{\phi\left(\frac{k}{k+1}\right)} \leq \left(\frac{1}{k+1}\right)^{1/2\delta}. \quad \triangleleft$$

We now define our junta function. Let $\bar{J} := J \setminus J'$. Define $h : [k+1]^n \to [0, 1]$ as

$$h(X) := \mathop{\mathbb{E}}_{Y \in [k+1]^{\bar{J}}}[f(X_{J'}, Y)], \tag{1}$$

where $x_{J'}$ is the point in the hypergrid where the coordinates in J' are assigned as they were in J and the coordinates in $\bar{J}$ are non-assigned.

First, we show that h is k–submodular.

▷ Claim 20. The function h defined in Equation (1) is k–submodular.

Proof. We have to show that h satisfies the two following conditions.

1. *Diminishing marginal gains.* Let $X, Z \in [k+1]^n$ with $Z \preceq X$ and $e \notin X$. Let $j \in [k]$. Then, we have

$$\Delta_h(Z, j, e) = h(Z_e^i) - h(Z) = \begin{cases} 0, & \text{if } j \notin J' \\ \mathop{\mathbb{E}}_{Y \in [k+1]^{\bar{J}}}[f((Z_e^i)_{J'}, Y) - f(Z_{J'}, Y)], & \text{otherwise} \end{cases}.$$

   If $j \notin J'$, then $\Delta_h(X, j, e) = 0$ as well, so the diminishing marginal gains condition holds. Otherwise, by the k–submodularity of f, we know that for all $y \in [k+1]^{\bar{J}}$,

$$f((Z_e^i)_{J'}, Y) - f(Z_{J'}, Y) \geq f((X_e^i)_{J'}, Y) - f(X_{J'}, Y)$$

   which allows us to conclude that $\Delta_h(Z, j, e) \geq \Delta_h(X, j, e)$, and thus that h satisfies diminishing marginal gains.

2. *Pairwise Monotonicity.* Let $X \in [k+1]^n$, $i \neq j \in [k]$ and $e \in X_{k+1}$. If $e \notin J$, then $\frac{1}{2}\left(h(X_e^i) + h(X_e^j)\right) = h(X)$. Otherwise, by the pairwise monotonicity of f,

$$\frac{1}{2}\left(h(X_e^i) + h(X_e^j)\right) = \mathop{\mathbb{E}}_{Y \in [k+1]^{\bar{J}}} \frac{1}{2}[f((X_e^i)_{J'}, Y) - f((X_e^j)_{J'}, Y)] \geq \mathop{\mathbb{E}}_{Y \in [k+1]^{\bar{J}}}[f(X_{J'}, Y)] = h(X).$$

We can thus conclude by Theorem 9 that h is k–submodular. ◁

Our final step is to show that h is a good approximation to f in the ℓ_2-norm.

▶ **Definition 21** (Bad points). *A point $X_{J'}$ in the restricted hypergrid is called* bad *if there exists $i \in \bar{J}$ and $j \in [k]$ such that*

$$\Delta_f(X_{J'}, j, i) > \alpha,$$

or there exists $i \in X_{J'}$

$$\Delta_f(X_{J'} \setminus i, X_{J'}(i), i) < -\alpha.$$

The ℓ_2 distance can be bounded as below.

$$\begin{aligned}
\|h - f\|_2^2 &= \mathop{\mathbb{E}}_{X \in [k+1]^J}[(f(X) - h(X))^2] \\
&= \mathop{\mathbb{E}}_{X_{J'} \in [k+1]^{J'}}\left[\mathop{\mathbb{E}}_{Y \in [k+1]^{\bar{J}}}\left[(f(X_{J'}, Y) - \mathop{\mathbb{E}}_{Y' \in [k+1]^{\bar{J}}}[f(X_{J'}, Y')])^2\right]\right] \\
&= \mathop{\mathbb{E}}_{X_{J'} \in [k+1]^{J'}}\left[\mathop{\mathrm{Var}}_{Y \in [k+1]^{\bar{J}}}[f(X_{J'}, Y)]\right] \\
&\leq \Pr[X_{J'} \text{ is } \textit{bad}] + \mathop{\mathbb{E}}_{X_{J'} \in [k+1]^{J'} \text{ good}}\left[\mathop{\mathrm{Var}}_{Y \in [k+1]^{\bar{J}}}[f(X_{J'}, Y)]\right]. \qquad \text{[Because } f(X) \leq 1\text{]}
\end{aligned}$$

We first show that for appropriate parameters to Algorithm 1, the probability that a point is bad is small.

▷ Claim 22. If we set $\delta = \dfrac{\log(k+1)}{2\log(8 \cdot k \cdot |J| \cdot \varepsilon^{-2})}$ in Algorithm 1, then

$$\Pr_{X_{J'} \in [k+1]^{J'}}[X_{J'} \text{ is } \textit{bad}] \leq \frac{\varepsilon^2}{8}.$$

Proof. Recall that we consider a uniform distribution over $X_{J'} \in [k+1]^{J'}$. This distribution is the same as sampling a partition $P : J' \to [k]$ uniformly at random and returning $J'\left(\frac{k}{k+1}\right)_P$. To see this, fix an arbitrary point $X \in [k+1]^{J'}$. Let $A := \{i \in J' : x_i = k+1\}$ and $B = J' \setminus A$. The probability of generating this particular X with the latter sampling procedure is

$$\Pr[X] = \left(\frac{1}{k+1}\right)^{|A|} \cdot \left(\frac{k}{k+1}\right)^{|B|} \cdot \left(\frac{1}{k}\right)^{|B|} = \frac{1}{(k+1)^{|J'|}}$$

So, via a union bound, Claim 19 and by plugging in $\delta = \frac{\log(k+1)}{2\log(8\cdot k\cdot|J|\cdot\varepsilon^{-2})}$, we get that

$$\begin{aligned}\Pr[X_{J'} \text{ is } \mathit{bad}] &= \Pr_{J'\left(\frac{k}{k+1}\right), P: J'\to[k]}\left[\bigcup_{j\in[k]}\bigcup_{i\in\bar{J}}\left(\Delta_f\left(J'\left(\frac{k}{k+1}\right)_P, j, i\right) > \alpha\right)\right]\\ &\le |\bar{J}|\cdot k\cdot\left(\frac{1}{k+1}\right)^{1/2\delta}\\ &\le \frac{\varepsilon^2}{8}. \qquad\triangleleft\end{aligned}$$

Finally, we show that for appropriate parameters to Algorithm 1, the variance of $f(X_{J'}, \cdot)$ when $X_{J'}$ is good is small.

▷ Claim 23. If we set $\alpha = \dfrac{\varepsilon^2}{16}$ in Algorithm 1, then

$$\mathop{\mathbb{E}}_{X_{J'}\in[k+1]^{J'} \text{ good}} \mathrm{Var}\left[f(X_{J'}, \cdot)\right] \le \frac{\varepsilon^2}{8}.$$

Proof. When $X_{J'}$ is good, we have that the marginal gain $\Delta_f(X_{J'}, j, i)$ of adding any $i \in \bar{J}$ to any bucket $j \in [k]$ is at most α. Because of *diminishing marginal gains*, this bounds the marginal gains of $f(X_{J'}, \cdot)$ over its entire domain by α. Also, the marginal gains of $f(X_{J'}, \cdot)$ are at least $-\alpha$, because if there existed some $Y \ge X_{J'}$, $i \in \bar{J}$ and $j \in [k]$ with $\Delta_f(Y, j, i) < -\alpha$, then by *pairwise monotonicity* we must have for all $j' \ne j$ that $\Delta_f(Y, j', i) \ge \alpha$ as

$$\Delta_f(Y, j, i) + \Delta_f(Y, j', i) \ge 0.$$

But this contradicts the upper bound of α to the marginal gains. Thus, $f(X_{J'}, \cdot)$ is α–Lipschitz and we can appeal to Lemma 14 to get

$$\mathrm{Var}[f(x_{J'}, \cdot)] \le 2\alpha\cdot\mathbb{E}[f(x_{J'}, \cdot)] \le \frac{\varepsilon^2}{8}. \qquad\triangleleft$$

We are now ready to finish the proof of Theorem 17.

Proof of Theorem 17. We let J' by the output of Algorithm 1 with input f and parameters

$$\delta = \frac{\log(k+1)}{2\log(8\cdot k\cdot|J|\cdot\varepsilon^{-2})}, \qquad \alpha = \frac{\varepsilon^2}{16}.$$

Then the function h defined by Equation (1) is the desired junta function. Indeed, combining (22) and (23), we get

$$\|h - f\|_2^2 \le \frac{\varepsilon^2}{8} + \frac{\varepsilon^2}{8} = \frac{\varepsilon^2}{4}.$$

To bound the size of J', we use Claim 18 to get

$$|J'| \le \frac{2}{\alpha\delta} = 32\cdot\frac{1}{\log(1+k)}\cdot\frac{1}{\varepsilon^2}\cdot\log\frac{8\cdot k\cdot|J|}{\varepsilon^2} \le \frac{32}{\varepsilon^2}\cdot\log\frac{8|J|}{\varepsilon^2}. \qquad\blacktriangleleft$$

Using the weak junta approximation, we are able arrive at our final junta approximation.

3.3 Proof of Theorem 2

Let $f : [k+1]^n \to [0,1]$ be a k–submodular function. We will prove a bound of $|J| \le \frac{4000}{\varepsilon^2} \log \frac{1}{\varepsilon^2}$ for the size of the approximating junta.

Observe that this bound holds trivially for $\varepsilon < n^{-1/2}$, because then we are allowed to choose $J = [n]$. For contradiction, suppose that there is $\varepsilon \in (n^{-1/2}, 1/2)$ for which the statement of Theorem 2 does not hold. Let $\mathcal{E} \subset (n^{-1/2}, 1/2)$ be the set of all ε for which the statement does not hold, and pick an $\varepsilon \in \mathcal{E}$ such that $\varepsilon < 2 \inf \mathcal{E}$. Then, the statement still holds for $\varepsilon_2 = \varepsilon^2 < \frac{1}{2}\varepsilon$.

By the statement of Theorem 2 for ε_2, there is a subset of variables J' of size $|J'| \le \frac{4000}{\varepsilon^2} \log \frac{1}{\varepsilon} \le \frac{2^{13}}{\varepsilon^2}$ and a k–submodular function g depending only on J, such that $\|f - g\|_2 \le \varepsilon_2 \le \frac{1}{2}\varepsilon$.

Now we apply Theorem 17 to g with parameter ε. Thus, there exists a k–submodular function h such that $\|g - h\|_2 \le \frac{1}{2}\varepsilon$, and h depends only on a subset of variables $J'' \subseteq J'$, $|J''| \le \frac{32}{\varepsilon^2} \log \frac{8|J'|}{\varepsilon}$. We have $|J'| \le \frac{2^{13}}{\varepsilon^2}$, and therefore $|J''| \le \frac{32}{\varepsilon^2} \log \frac{2^{16}}{\varepsilon^3} \le \frac{32}{\varepsilon^2} \log \frac{1}{\varepsilon^3}$ (using $\varepsilon \le \frac{1}{2}$). We conclude that $|J''| \le \frac{32}{\varepsilon^2} \log \frac{1}{\varepsilon^3} \le \frac{4000}{\varepsilon^2} \log \frac{1}{\varepsilon}$ as required in Theorem 2. By the triangle inequality, we have $\|f - h\|_2 \le \|f - g\|_2 + \|g - h\|_2 \le \frac{1}{2}\varepsilon + \frac{1}{2}\varepsilon = \varepsilon$.

However, this would mean that the statement of Theorem 2 holds for ε as well, which is a contradiction. ◀

4 Towards testing k–submodularity in the ℓ_0-norm

In this section, we develop the local-witness theory for the two constraints that characterize k–submodularity in Hamming distance. For diminishing marginal gains, the witnesses are generalized violated squares. For pairwise monotonicity, the witnesses are violated triangles. We prove that if either constraint is far from being satisfied, then the corresponding witnesses are dense enough to be found by a non-adaptive one-sided tester. The key step in both cases is a repair argument over ideals and filters in the hypergrid.

4.1 Setup

An inductive argument shows that absence of violating squares is equivalent to the diminishing marginal gains property:

▶ **Proposition 24.** *A function $f : [k+1]^n \to \mathbb{R}$ satisfies the diminishing marginal gains property if and only if it has no violating squares.*

Proof. If a function satisfies the diminishing marginal gains property, then it is easy to see that it has no violating squares: take $Y = X_g^j$. We show the converse by considering $X \preceq Y$, $e \notin Y$ and $i \in [k]$. Suppose $e_1, \ldots, e_\ell$ are the elements of Y not in $X \sqcap Y$ that have $Y(e) \ne k+1$. Then we have that:

$$f(X_e^i) - f(X) \ge f(X_{e,e_1}^{i,Y(e_1)}) - f(X_{e_1}^{Y(e_1)})$$

By induction we can telescopically build the partial partition Y on the right hand side and recover the diminishing marginal gains property. ◀

▶ **Definition 25** (Violating triangle). *Let $X \in [k+1]^n$, $i \ne j \in [k]$, and $e \in X_{k+1}$. We call $\{x, X_e^i, X_e^j\}$ a* triangle. *For a function $f : [k+1]^n \to \mathbb{R}$, this is called an f–*violating triangle *if*

$$\frac{f(X_e^i) + f(X_e^j)}{2} < f(X),$$

that is, it violates the pairwise monotonicity condition.

4.2 Testing diminishing marginal gains

Algorithm 2 A tester for diminishing marginal gains.

Input: Parameters k, n, a query oracle for function $f : [k+1]^n \to \mathbb{R}$, and an error parameter $\varepsilon > 0$.
for $n^2 \cdot (1/\varepsilon)^{\Theta(\sqrt{n \log n})}$ iterations **do**
  sample the following uniformly at random.
    A point $X \in [k+1]^n$ from the hypercube,
    two (possibly same) indices $i, j \in [k]$, and
    two elements $e \neq g \in X_{k+1}$.
  if $\{X, X_e^i, X_g^j, X_{(e,g)}^{(i,j)}\}$ is an f–violating square **then**
    reject.
accept.

We prove the following theorem.

▶ **Theorem 26.** *Algorithm 2 has the following properties.*

1. *If f has* diminishing marginal gains*, then Algorithm 2* **accepts** *with probability* 1*.*
2. *If f is ε-far from* diminishing marginal gains*, then Algorithm 2* **rejects** *with probability at least* $2/3$*.*
3. *Algorithm 2 makes $q = 4n^2 \cdot (1/\varepsilon)^{\Theta(\sqrt{n \log n})}$ queries.*

Items 1 and 3 are easy to verify – they are direct consequences of Theorem 9 and the description of Algorithm 2, respectively. The rest of this section will be dedicated to proving Item 2.

Intuitively, our proof proceeds as follows. We first show that if a function f has a "low density" of violating squares, then it can be made to have the *diminishing marginal gains* property by changing the values of only a "few" points. By contrapositive, this means that a function that is "far" from *diminishing marginal gains* must have a "high density" of witnesses, which implies the correctness of our tester. To "repair" a function f towards *diminishing marginal gains*, we modify its values over a union of *ideals* and *filters*, defined as follows.

▶ **Definition 27** (Ideal)**.** *Let $X \in [k+1]^n$. The* Ideal *of X is defined as*

$$I(X) := \{Y \mid Y \preceq X\}.$$

▶ **Definition 28** (Filter)**.** *Let $X \in [k+1]^n$. The* Filter *of X is defined as*

$$F(X) := \{Y \mid X \preceq Y\}.$$

We start with the following "repair" lemma for violating squares.

▶ **Lemma 29** (Square repair lemma)**.** *Let $f : [k+1]^n \to \mathbb{R}$ be a function in which $\{X, X_e^i, X_g^j, X_{(e,g)}^{(i,j)}\}$ is an f–violating square. There is a way to change only the values of the ideal $I(X)$ or the filter $F(X_{(e,g)}^{(i,j)})$ to obtain a function f' in which $\{X, X_e^i, X_g^j, X_{(e,g)}^{(i,j)}\}$ is no longer an f'–violating square, and no additional f'–violating squares have been introduced.*

Proof. Because $\{X, X^i_e, X^j_g, X^{(i,j)}_{(e,g)}\}$ is an f–violating square, we have

$$\Delta_X := f(X) + f(X^{(i,j)}_{(e,g)}) - f(X^i_e) - f(X^j_g) > 0.$$

Fix any constant $\Delta \geq \Delta_X$. We obtain f' by subtracting Δ from $f(Y)$ for all $Y \in I(X)$. Clearly, this repairs the f–violating square itself, but we must also argue that it does not introduce any new f'–violating squares.

Let $\{Y, Y^{i'}_{e'}, Y^{j'}_{g'}, Y^{(i',j')}_{(e',g')}\}$ be an f'–violating square that is not f–violating. If $Y \not\preceq X$, then none of the values in the square were modified. So we can assume $Y \preceq X$. We consider the following cases.

1. Neither $Y^{i'}_{e'} \not\preceq X$ nor $Y^{j'}_{g'} \not\preceq X$. Then the points $Y^{i'}_{e'}, Y^{j'}_{g'}, Y^{(i',j')}_{(e',g')}$ are not modified. Thus, the value of this square is
$$(f(Y)-\Delta)+f(Y^{(i',j')}_{(e',g')})-f(Y^{j'}_{g'})-f(Y^{i'}_{e'}) = f(Y)+f(Y^{(i',j')}_{(e',g')})-f(Y^{j'}_{g'})-f(Y^{i'}_{e'})-\Delta \leq -\Delta.$$
2. Only $Y^{i'}_{e'} \not\preceq X$ and $Y^{j'}_{g'} \preceq X$. Then the points $Y^{i'}_{e'}, Y^{(i',j')}_{(e',g')}$ are not modified. Thus, the value of this square is
$$(f(Y)-\Delta)+f(Y^{(i',j')}_{(e',g')})-(f(Y^{j'}_{g'})-\Delta)-f(Y^{i'}_{e'}) = f(Y)+f(Y^{(i',j')}_{(e',g')})-f(Y^{j'}_{g'})-f(Y^{i'}_{e'}) \leq 0.$$
The case of $Y^{i'}_{e'} \preceq X$ and $Y^{j'}_{g'} \not\preceq X$ is symmetric and identical.
3. Both $Y^{i'}_{e'} \preceq X$ and $Y^{j'}_{g'} \preceq X$. Then $Y^{(i',j')}_{(e',g')} \preceq X$, and all the points are modified. Thus, the value of this square is
$$\begin{aligned}(f(Y) - \Delta) + (f(Y^{(i',j')}_{(e',g')}) - \Delta) - (f(Y^{j'}_{g'}) - \Delta) - (f(Y^{i'}_{e'}) - \Delta)\\ = f(Y) + f(Y^{(i',j')}_{(e',g')}) - f(Y^{j'}_{g'}) - f(Y^{i'}_{e'}) \leq 0.\end{aligned}$$

As a result, we have shown that no new violating squares are introduced after changing all the values in $I(X)$, and the original violating square rooted in X is repaired. An identical argument suffices to prove that subtracting Δ from the filter $F(X^{(i,j)}_{(e,g)})$ has the same consequence. ◀

The proof of this lemma also implies the following corollary.

▶ **Corollary 30.** *Let $f : [k+1]^n \to \mathbb{R}$ be a function and $X \in [k+1]^n$ be a domain point. For any constant $\Delta > 0$, the function f' obtained by subtracting Δ from $f(Y)$ for all $Y \in I(X)$ or all $Y \in F(X)$ does not introduce any new f'–violating squares.*

Next, we prove the following density lemma.

▶ **Lemma 31.** *Let $f : [k+1]^n \to \mathbb{R}$ be a function with m violating squares. If $m \leq \varepsilon^{\sqrt{n}\log n} \cdot (k+1)^n$ for $\varepsilon \in (0, e^{-5})$, then f can be made to have* diminishing marginal gains *by changing at most $\varepsilon \cdot (k+1)^n$ values.*

Proof. The proof is a counting argument on the number of points we will need to change by the square repair procedure above. We enumerate all m violating squares and arrange them in order of item-weight (equal to $\sum_{i=1}^k |X_i|$) of the bottom point. We split the hypercube evenly based on item-weight. It can be verified that this split occurs at weight $\frac{nk}{k+1}$.

1. For violating squares where the bottom vertex has item-weight $\leq \frac{nk}{k+1}$, we repair them by modifying values in the ideal. Let I denote the bottom vertex in these violating squares. Then the total number of modified points is $|\cup_{X \in I} I(X)|$.

2. For violating squares where the bottom vertex has item-weight $> \frac{nk}{k+1}$, we repair them by modifying values in the filter. Let F denote the top vertex in these violating squares. Then the total number of modified points is $|\cup_{X\in F} F(X)|$.

We first prove the counting argument for Item 1. We estimate this cardinality by combining two different bounds across levels of the hypergrid. Let $L_j := \{X \in [k+1]^n : |X| \leq j\}$. Then we can write

$$\left|\bigcup_{X\in I} I(X)\right| = \sum_{j=0}^{\frac{nk}{k+1}} \left|\bigcup_{X\in I} (I(X)\cap L_j)\right|.$$

We can bound these terms in two ways. First, we have

$$\left|\bigcup_{X\in I} (I(X)\cap L_j)\right| \leq \sum_{X\in I} |I(X)\cap L_j| = \sum_{X\in I} \binom{|X|}{j} \leq |I|\cdot \binom{\frac{nk}{k+1}}{j}.$$

Second, there is the more trivial bound of

$$\left|\bigcup_{X\in I} (I(X)\cap L_j)\right| \leq |L_j| = \binom{n}{j}\cdot k^j.$$

This second bound is reminiscent of the binomial distribution $\mathrm{Bin}\left(n, \frac{k}{k+1}\right)$. Combining this with a Chernoff bound, we can get an upper bound on the number of points with item-weight far from the mean. Now we just have to use the correct bound depending on j.

1. If $j \leq \frac{nk}{k+1} - \alpha\sqrt{\frac{nk}{k+1}}$ for some constant α, then

$$\begin{aligned}
\sum_{j=0}^{\frac{nk}{k+1}-\alpha\sqrt{\frac{nk}{k+1}}} \binom{n}{j}\cdot k^j &= (k+1)^n \sum_{j=0}^{\frac{nk}{k+1}-\alpha\sqrt{\frac{nk}{k+1}}} \binom{n}{j}\cdot\left(\frac{k}{k+1}\right)^j\left(\frac{1}{k+1}\right)^{n-j} \\
&= (k+1)^n\cdot \Pr\left[Z \leq \frac{nk}{k+1} - \alpha\sqrt{\frac{nk}{k+1}}\right] && \left[Z\sim \mathrm{Bin}\left(n, \frac{k}{k+1}\right)\right] \\
&= (k+1)^n\cdot \Pr\left[Z \leq (1-\delta)\frac{nk}{k+1}\right] && \left[\text{Letting } \delta = \frac{\alpha}{\sqrt{\frac{nk}{k+1}}}\right] \\
&\leq (k+1)^n\cdot e^{-\delta^2 nk/2(k+1)} && [\text{Chernoff bound}] \\
&= (k+1)^n\cdot e^{-\alpha^2/2}.
\end{aligned}$$

2. If $j > \frac{nk}{k+1} - \alpha\sqrt{\frac{nk}{k+1}}$, then

$$\begin{aligned}
\sum_{j=\frac{nk}{k+1}-\alpha\sqrt{\frac{nk}{k+1}}}^{\frac{nk}{k+1}} |I|\cdot\binom{\frac{nk}{k+1}}{j} &= |I|\cdot\sum_{j=0}^{\alpha\sqrt{\frac{nk}{k+1}}}\binom{\frac{nk}{k+1}}{j} \\
&\leq |I|\cdot\left(\frac{nk}{k+1}\right)^{\alpha\sqrt{\frac{nk}{k+1}}} \\
&\leq |I|\cdot n^{\alpha\sqrt{n}} \\
&= |I|\cdot e^{\alpha\sqrt{n}\ln n}.
\end{aligned}$$

Finally, setting $\alpha = \frac{1}{2}\ln\varepsilon^{-1}$ yields

$$
\begin{aligned}
\left|\bigcup_{X\in I} I(X)\right| &= \sum_{j=0}^{\frac{nk}{k+1}} \left|\bigcup_{X\in I}(I(X)\cap L_j)\right| && \text{[Splitting by item-weight]}\\
&= \sum_{j=0}^{\frac{nk}{k+1}-\alpha\sqrt{\frac{nk}{k+1}}} \left|\bigcup_{X\in I}(I(X)\cap L_j)\right| + \sum_{j=\frac{nk}{k+1}-\alpha\sqrt{\frac{nk}{k+1}}}^{\frac{nk}{k+1}} \left|\bigcup_{X\in I}(I(X)\cap L_j)\right| && \text{[Splitting the sum]}\\
&\le \sum_{j=0}^{\frac{nk}{k+1}-\alpha\sqrt{\frac{nk}{k+1}}} \binom{n}{j}\cdot k^j + \sum_{j=\frac{nk}{k+1}-\alpha\sqrt{\frac{nk}{k+1}}}^{\frac{nk}{k+1}} |I|\cdot\binom{\frac{nk}{k+1}}{j} && \text{[Discussed bounds]}\\
&\le (k+1)^n\cdot e^{-\alpha^2/2} + |I|\cdot e^{\alpha\sqrt{n}\ln n} && \text{[From case analysis]}\\
&\le (k+1)^n\cdot e^{-\alpha^2/2} + m\cdot e^{\alpha\sqrt{n}\ln n} && [|I|\le m]\\
&\le (k+1)^n\cdot e^{-\alpha^2/2} + (k+1)^n\cdot\varepsilon^{\sqrt{n}\ln n}\cdot e^{\alpha\sqrt{n}\ln n} && [m\le (k+1)^n\cdot\varepsilon^{\sqrt{n}\ln n}]\\
&= (k+1)^n\cdot\varepsilon^{\frac{1}{8}\ln\varepsilon^{-1}} + (k+1)^n\cdot\varepsilon^{\sqrt{n}\ln n}\cdot\varepsilon^{-\frac{1}{2}\sqrt{n}\ln n} && [\alpha = 1/2\cdot\ln\varepsilon^{-1}]\\
&\le (k+1)^n\cdot\left(\varepsilon^{\frac{1}{8}\ln\varepsilon^{-1}} + \varepsilon^{\frac{1}{2}\sqrt{n}\ln n}\right) && \text{[Simplifying]}\\
&\le \frac{\varepsilon}{2}\cdot(k+1)^n. && [\varepsilon\in(0,e^{-5})]
\end{aligned}
$$

This concludes the counting argument for Item 1.

The counting argument for Item 2 is almost identical. We split the filters by level, and for $j \ge \frac{nk}{k+1} + \alpha\sqrt{\frac{nk}{k+1}}$, the Chernoff-inequality based upper bound still holds. For $j < \frac{nk}{k+1} + \alpha\sqrt{\frac{nk}{k+1}}$, we get the following bound.

$$
\begin{aligned}
\sum_{j=\frac{nk}{k+1}}^{\frac{nk}{k+1}+\alpha\sqrt{\frac{nk}{k+1}}} \left|\bigcup_{X\in F}(F(X)\cap L_j)\right| &\le \sum_{X\in F}|F(X)\cap L_j|\\
&= \sum_{j=\frac{nk}{k+1}}^{\frac{nk}{k+1}+\alpha\sqrt{\frac{nk}{k+1}}} \sum_{X\in F}\binom{n-|X|}{j-|X|}k^{j-|X|}\\
&\le \sum_{j=\frac{nk}{k+1}}^{\frac{nk}{k+1}+\alpha\sqrt{\frac{nk}{k+1}}} \sum_{X\in F}\binom{\frac{n}{k+1}}{j-|X|}k^{j-|X|}\\
&\le \sum_{j=\frac{nk}{k+1}}^{\frac{nk}{k+1}+\alpha\sqrt{\frac{nk}{k+1}}} \sum_{X\in F}\left(\frac{n}{k+1}\right)^{j-|X|}k^{j-|X|}\\
&\le \sum_{j=\frac{nk}{k+1}}^{\frac{nk}{k+1}+\alpha\sqrt{\frac{nk}{k+1}}} \sum_{X\in F} n^{j-|X|}\\
&\le \sum_{j=\frac{nk}{k+1}}^{\frac{nk}{k+1}+\alpha\sqrt{\frac{nk}{k+1}}} |F|\cdot n^{j-\frac{nk}{k+1}}\\
&\le |F|\cdot n^{\alpha\sqrt{\frac{nk}{k+1}}}\\
&\le |F|\cdot n^{\alpha\sqrt{n}}\\
&= |F|\cdot e^{\alpha\sqrt{n}\ln n}.
\end{aligned}
$$

The rest of the counting argument is exactly the same and thus omitted.

In conclusion, we need to modify $\leq \varepsilon \cdot (k+1)^n$ values to make f have *diminishing marginal gains.* ◀

We are now ready to finish the analysis of our tester.

Proof of Theorem 26. We prove each part separately.

1. By Theorem 9, a function with *diminishing marginal gains* does not have any f–violating squares. As a result, the tester will always **accept**.
2. By the contrapositive of Lemma 31, a function that is ε-far from *diminishing marginal gains* has at least $(k+1)^n \cdot \varepsilon^{\sqrt{n}\ln n}$ violating squares. The total number of squares is $\binom{n}{2} \cdot k^2 \cdot (k+1)^{n-2} = \Theta(n^2(k+1)^n)$. Thus, by the witness lemma, $n^2 \cdot (1/\varepsilon)^{\Theta(\sqrt{n \log n})}$ queries suffice to catch an f–violating square with high probability.
3. This follows trivially from the design of the algorithm.

This completes the analysis of Algorithm 2. ◀

4.3 An obstruction to component-wise local repair

The component testers of the previous two sections each rest on a dense witness theorem and a local repair argument. A natural attempt at a tester for full k–submodularity is to apply both component testers in parallel and union-bound the failure probability over the two repair arguments. We now show that this attempt cannot succeed: there exist configurations on which the repair direction prescribed by a triangle witness is forced to oppose the repair direction prescribed by an overlapping square witness on a shared vertex.

For ideals rooted at item-weight $\leq \frac{nk}{k+1}$, the repairs for both witness types proceed downward and can be aligned. The obstruction lives in the upward direction. On filters rooted at item-weight $> \frac{nk}{k+1}$, repairing a violated square requires subtracting $\Delta > 0$ from a vertex, whereas repairing a violated triangle in the same filter requires *adding* $\Delta > 0$ to that vertex. The configuration in Figure 1 realizes both demands simultaneously on the vertices x_2 and x_3.

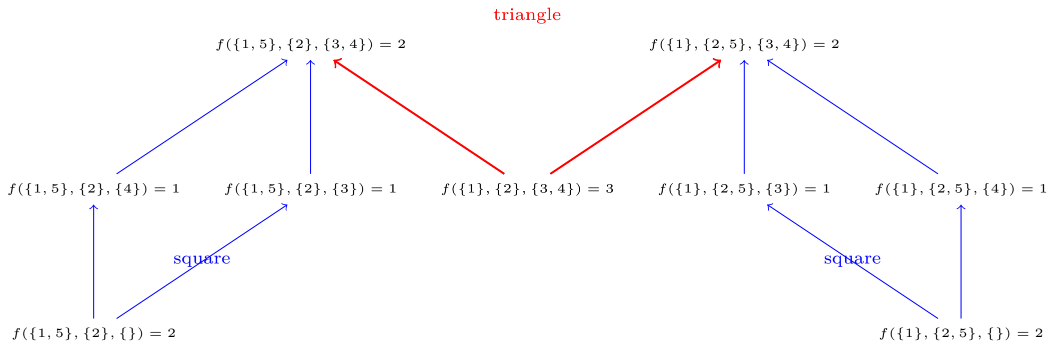

Figure 1 A configuration on a high item-weight filter realizing two incompatible local violations. The red edges form a violating *triangle* on item 5 above x_1: assigning item 5 to part 1 (giving x_2) or to part 2 (giving x_3) both yield marginal -1, summing to $-2 < 0$. The blue edges form two violating *squares* for diminishing marginal gains on items 3 and 4 in part 3, above x_8 (left) and x_9 (right). Repairing the triangle requires raising $f(x_2)$ or $f(x_3)$; repairing the squares requires lowering the same values.

The configuration is not isolated to a small region of the hypergrid. Embedding the same value pattern into any high-item-weight filter on a larger domain produces an instance of the same conflict, so an ideal-only repair strategy that avoids upward filters cannot route around it without losing the balance argument that makes the component testers work.

▶ **Observation 32** (Component-wise repair is insufficient). *For every $k \geq 2$ there exist functions $f : [k+1]^n \to \mathbb{R}$ and high item-weight filters on which a violated triangle witness and an overlapping violated square witness prescribe local repairs of opposite sign on a common vertex. In particular, the standard component-wise repair argument – repair triangles and squares independently and combine via union bound – does not yield a tester for k-submodularity.*

Observation 32 does not rule out a Hamming-distance tester for k-submodularity; it shows only that obtaining one will require either a richer witness vocabulary that simultaneously certifies both component constraints, or a non-local repair scheme that mediates between square and triangle repairs across the hypergrid.

References

1. Eric Blais and Abhinav Bommireddi. Testing submodularity and other properties of valuation functions. *arXiv preprint arXiv:1611.07879*, 2016. arXiv:1611.07879.
2. Stéphane Boucheron, Gábor Lugosi, and Olivier Bousquet. Concentration inequalities. In *Summer school on machine learning*, pages 208–240. Springer, 2003. doi:10.1007/978-3-540-28650-9_9.
3. Deeparnab Chakrabarty and Zhiyi Huang. Recognizing coverage functions. *SIAM Journal on Discrete Mathematics*, 29(3):1585–1599, 2015. doi:10.1137/140964072.
4. Alina Ene and Huy Nguyen. Streaming algorithm for monotone k-submodular maximization with cardinality constraints. In *International Conference on Machine Learning*, pages 5944–5967. PMLR, 2022. URL: https://proceedings.mlr.press/v162/ene22a.html.
5. Jon Feldman, Monika Henzinger, Nitish Korula, Vahab S Mirrokni, and Cliff Stein. Online stochastic packing applied to display ad allocation. In *European Symposium on Algorithms*, pages 182–194. Springer, 2010. doi:10.1007/978-3-642-15775-2_16.
6. Vitaly Feldman and Jan Vondrák. Optimal bounds on approximation of submodular and xos functions by juntas. *SIAM Journal on Computing*, 45(3):1129–1170, 2016. doi:10.1137/140958207.
7. Michel X Goemans and Jan Vondrák. Covering minimum spanning trees of random subgraphs. *Random Structures & Algorithms*, 29(3):257–276, 2006. doi:10.1002/RSA.20115.
8. Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653–750, 1998. doi:10.1145/285055.285060.
9. Themistoklis Haris and Diptaksho Palit. Testing k-submodularity. *CoRR*, abs/2606.30433, 2026. doi:10.48550/arXiv.2606.30433.
10. Pooya Hatami and Jan Vondrák. Testing real-valued modularity and submodularity, 2016. URL: https://www.ias.edu/sites/default/files/math/csdm/15-16/HatamiVondrak.pdf.
11. Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the fourier spectrum. *SIAM J. Comput.*, 22(6):1331–1348, 1993. doi:10.1137/0222080.
12. Naoto Ohsaka and Yuichi Yoshida. Monotone k-submodular function maximization with size constraints. *Advances in Neural Information Processing Systems*, 28, 2015.
13. Michal Parnas, Dana Ron, and Ronitt Rubinfeld. On testing convexity and submodularity. *SIAM Journal on Computing*, 32(5):1158–1184, 2003. doi:10.1137/S0097539702414026.
14. Sofya Raskhodnikova. CS 599: Sublinear Algorithms (Spring 2025). https://cs-people.bu.edu/sofya/sublinear-course/, 2025. Retrieved on 10 October, 2025. URL: https://cs-people.bu.edu/sofya/sublinear-course/.

15 Sofya Raskhodnikova and Grigory Yaroslavtsev. Learning pseudo-boolean k-dnf and submodular functions. In Sanjeev Khanna, editor, *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1356–1368. SIAM, 2013. doi:10.1137/1.9781611973105.98.

16 Comandur Seshadhri and Jan Vondrák. Is submodularity testable? *Algorithmica*, 69(1):1–25, 2014. doi:10.1007/S00453-012-9719-2.

17 Justin Ward and Stanislav Živnỳ. Maximizing k-submodular functions and beyond. *ACM Transactions on Algorithms (TALG)*, 12(4):1–26, 2016. doi:10.1145/2850419.