Electronic Colloquium on Computational Complexity, Revision 4 of Report No. 80 (2016)

Reducing Testing Affine Spaces to Testing Linearity of Functions*

Oded Goldreich†

March 14, 2020

Abstract

For any finite field $\mathcal{F}$ and $k < \ell$, we consider the task of testing whether a function $f : \mathcal{F}^\ell \to \{0,1\}$ is the indicator function of an $(\ell-k)$-dimensional affine space. For the case of $\mathcal{F} = \mathrm{GF}(2)$, an optimal tester for this property was presented by Parnas, Ron, and Samorodnitsky (*SIDMA*, 2002), by mimicking the celebrated linearity tester of Blum, Luby and Rubinfeld (*JCSS*, 1993) and its analysis. We show that the former task (i.e., testing $(\ell-k)$-dimensional affine spaces) can be efficiently reduced to testing the linearity of a related function $g : \mathcal{F}^\ell \to \mathcal{F}^k$. This reduction yields an almost optimal tester for affine spaces (represented by their indicator function).

Recalling that Parnas, Ron, and Samorodnitsky used testing $(\ell-k)$-dimensional affine spaces as the first step in a two-step procedure for testing k-monomials, we also show that the second step in their procedure can be reduced to testing whether the foregoing function g depends on k of its variables.

Contents

*A preliminary version of this work was posted in April 2016 as a guest column on the *Property Testing Review*. It was significantly revised and appeared as TR16-080 of *ECCC*. The current version is the result of an even more extensive revision. In particular, the reduction of testing affine spaces to testing linearity (of functions) is simplified and extended to arbitrary finite fields, the second step in the procedure for testing k-monomials is revised, many of the technical justifications are elaborated, and some crucial typos are fixed. In addition, the title has been augmented for clarity, the brief introduction has been expanded, and the high level structure has been re-organized.

†Department of Computer Science, Weizmann Institute of Science, Rehovot, Israel.

ISSN 1433-8092

1 Introduction

Property Testing is the study of super-fast (randomized) algorithms for approximate decision making. These algorithms are given direct access to items of a huge data set, and determine whether this data set has some predetermined (global) property or is far from having this property, while accessing a small portion of the data set. Thus, property testing is a relaxation of decision problems and it focuses on algorithms, called *testers*, that only read parts of the input. Consequently, the testers are modeled as oracle machines and the inputs are modeled as functions to which the tester has an oracle access.

This paper refers to several basic tasks in property testing, including testing linearity, testing dictatorship, testing (monotone) k-monomials, and testing affine spaces. Whereas the first three tasks refer explicitly to an input function (i.e., the object they test is naturally viewed as a function), in the case of testing affine spaces the object is a set of points and representing this set by an indicator function is a natural choice but not an immediate one. In particular, this is a very redundant representation in the case that the set is sparse, but we will consider the case of relatively dense sets. Furthermore, this representation arises naturally in the study of testing k-monomials.

The problem of testing whether a Boolean function is a (monotone) k-monomial was first studied by Parnas, Ron, and Samorodnitsky [9]. The tester that they presented generalizes their tester of dictatorship (i.e., the case $k = 1$), and does so by following the same two-step strategy and using similar arguments at each step. This raises the question of whether the case of general k can be reduced to the special case of $k = 1$. (We mention that this question occurred to us when writing [7, Sec. 5.2.2], and the first version of the current paper was written at that time.)

Specifically, the first step in the strategy of Parnas, Ron, and Samorodnitsky [9] is *testing whether the input function $f : \{0,1\}^\ell \to \{0,1\}$ describes an $(\ell-k)$-dimensional affine space*, where the space described by f is $f^{-1}(1)$. In the case of dictatorship (i.e., $k = 1$), this amounts to testing whether f itself is an affine function, but in the case of $k > 1$ a more general task arises (i.e., testing whether $f^{-1}(1)$ is an $(\ell-k)$-dimensional affine space is fundamentally different from testing whether f is affine). In the second step, one tests whether this affine space is of the right form (i.e., is a translation by 1^ℓ of a linear space spanned by axis-parallel vectors). In the case of $k = 1$, the latter task amounts to testing whether the affine function depends on a single variable, but in the case of $k > 1$ another more general task arises.

Both these general tasks were solved by Parnas, Ron, and Samorodnitsky [9], but their solutions mimic the solutions used in the case of $k = 1$ (see Section 4.1 for more details). Furthermore, in both cases, the generalization is very cumbersome. We find this state of affairs quite annoying, and believe that it is more appealing to reduce the general case to the special case.

Our contribution. This paper partially achieves this goal by (1) replacing the first step of [9] with a reduction to the (extensively studied) problem of testing linearity of functions, and (2) replacing the second step of [9] with a reduction to the problem of testing k-linearity of functions. Specifically, we first *reduce the problem of testing affine spaces to the problem of testing the linearity of functions.* This reduction actually hold over any finite field, whereas the application to testing monomials only uses the case of the binary field (which is the case treated in [9]). Next, recalling that k-monomials correspond to axis-parallel $(\ell-k)$-dimensional affine spaces, we reduce the testing of such spaces to testing whether a linear function depends on k of its variables. The complexity of the testers that we derive is only slightly inferior to the complexity of the corresponding optimal testers of [9]; specifically, in the relevant case of $\epsilon = O(2^{-k})$, we get a complexity bound of $\widetilde{O}(\log(1/\epsilon)) \cdot 2^k + O(1/\epsilon) = \widetilde{O}(1/\epsilon)$ rather than $O(1/\epsilon)$. (Indeed, the bounds coincide for $\epsilon < 2^{-k}/\widetilde{O}(k)$.)

Organization. In Section 2 we recall the standard definition of property testing and formally define the main properties considered in this paper (i.e., affine spaces and linear functions). The reduction of testing affine spaces to testing linearity of functions is presented in Section 3, whereas the problem of testing monomials is considered in Section 4. In the Appendix we present a simple and direct procedure for testing whether a linear function depends on a given number of variables.

2 Preliminaries

We assume that the reader is familiar with the basic definition of property testing (see, e.g., [7]), but for sake of good order we reproduce it here. The basic definition refers to functions with domain D and range R.

Definition 2.1 (a tester for property Π): *Let Π be a set of functions of the form $f : D \to R$. A* tester for Π *is a probabilistic oracle machine, denoted T, that, on input a* proximity parameter ϵ *and oracle access to a function $f : D \to R$, outputs a binary verdict that satisfies the following two conditions.*

1. T accepts inputs in Π: *For every $\epsilon > 0$, and for every $f \in \Pi$, it holds that $\mathbf{Pr}[T^f(\epsilon) = 1] \geq 2/3$.*
2. T rejects inputs that are ϵ-far from Π: *For every $\epsilon > 0$, and for every function $f : D \to R$ that is ϵ-far from Π it holds that $\mathbf{Pr}[T^f(\epsilon) = 0] \geq 2/3$, where f is* ϵ-far from Π *if for every $g \in \Pi$ it holds that $|\{x \in D : f(x) \neq g(x)\}| > \epsilon \cdot |D|$.*

If the first condition holds with probability 1 (i.e., $\mathbf{Pr}[T^f(\epsilon) = 1] = 1$), *then we say that T has* one-sided error*; otherwise, we say that T has* two-sided error.

We focus on the query complexity of such testers, while viewing $|D|$ as an additional parameter. We seek testers of query complexity that is independent of $|D|$, which means that the complexity will be a function of the proximity parameter ϵ and an auxiliary parameter k (of the two properties that we consider).

The properties we shall consider refer to functions over the domain $\mathcal{F}^\ell$, where $\mathcal{F}$ is a finite field. (In the previous versions of this paper, we confined ourselves to the case that $\mathcal{F}$ is the two-element field GF(2), which is the case treated in [9].)

Definition 2.2 (affine spaces): *For fixed $k, \ell \in \mathbb{N}$ and a finite field $\mathcal{F}$, we say that the function $f : \mathcal{F}^\ell \to \{0,1\}$* describes an $(\ell - k)$-dimensional affine space *if $f^{-1}(1) = \{x \in \mathcal{F}^\ell : f(x) = 1\}$ is an $(\ell - k)$-dimensional affine space; that is, $f^{-1}(1) = \{yG + s : y \in \mathcal{F}^{\ell-k}\}$, where $G \in \mathcal{F}^{(\ell-k)\times\ell}$ is an $(\ell - k)$-by-ℓ full-rank matrix and $s \in \mathcal{F}^\ell$. When $s = 0^\ell$, the* described space is linear.

We mention that, for $\mathcal{F} = \mathrm{GF}(2)$, the set of k-monomials (see Definition 4.1) coincides with the set of functions that describe $(\ell - k)$-dimensional affine spaces that are spanned by unit vectors (i.e., the rows of the matrix G are unit vectors).

Definition 2.3 (linear functions): *For fixed $k, \ell \in \mathbb{N}$ and a finite field $\mathcal{F}$, we say that $g : \mathcal{F}^\ell \to \mathcal{F}^k$ is* linear *if $g(x + y) = g(x) + g(y)$ for all $x, y \in \mathcal{F}^\ell$. Equivalently, $g(z) = zT$ for a ℓ-by-k matrix T. We say that f is* affine *if $f(z) = f'(z) + s$ for a linear function f' and some $s \in \mathcal{F}^k$.*

When $k = 1$ and $\mathcal{F} = \mathrm{GF}(2) \equiv \{0,1\}$, it holds that $f : \mathcal{F}^\ell \to \{0,1\}$ describes an $(\ell - k)$-dimensional affine space (resp., linear space) if and only if f is a non-constant affine function (resp., $f + 1$ is a non-constant linear function). However, in the other cases, this does not hold; in particular, for other fields a non-constant affine function must range over $\mathcal{F}$ rather than over $\{0,1\}$, whereas for $\mathcal{F} = \mathrm{GF}(2)$ and $k > 1$ the densities do not match (i.e., an $(\ell - k)$-dimensional affine space over $\mathrm{GF}(2)$ has density 2^{-k}, but $f^{-1}(1)$ has density $1/2$ for any non-constant affine function $f : \mathrm{GF}(2)^\ell \to \mathrm{GF}(2)$).

Conventions. When writing $\mathbf{Pr}_x[\text{event}(x)]$ we refer to the case that x is selected uniformly in a set that is clear from the context; we sometimes spell out this set by writing $\mathbf{Pr}_{x \in S}[\text{event}(x)]$. For sake of simplicity, we often use the phrase "with high probability" (abbrev., "w.h.p."), which mean that we can obtain arbitrary high constant probability smaller 1 (e.g., 0.99). The image of a function $f : D \to R$ is the set $\{f(e) : e \in D\} \subseteq R$. The symbol $\perp$ denotes a special symbol that is not in $\mathcal{F}^k$.

We view $\mathcal{F}$, ℓ and k as parameters, and when using O-notation we refer to universal constants that are independent of $\mathcal{F}$, ℓ and k. However, when stating time-complexity bounds, we shall assume that basic operations on elements of $\mathcal{F}^\ell$ (e.g., addition, selection of a random element, etc) can be performed at unit cost.

3 The reduction (of testing affine spaces to testing linearity of functions)

We start by restating the problem. We are given access to a function $h : \mathcal{F}^\ell \to \{0,1\}$ and wish to test whether $h^{-1}(1)$ is an $(\ell - k)$-dimensional affine subspace by reducing this problem to testing linearity (of a function). We present two reductions: The first (and simpler) reduction increases the complexities by a factor of $|\mathcal{F}|^k$, whereas the second reduction only incurs an overhead of $\widetilde{O}(\log(1/\epsilon))$. The first reduction (presented in Section 3.2) will be used as a subroutine in the second reduction (presented in Section 3.3). Furthermore, the first reduction provides a good warm-up towards the second one. Before describing these reductions, we present and justify some simplifying assumptions.

3.1 Simplifying assumptions

First, note that we may assume that $\epsilon = O(|\mathcal{F}|^{-k})$, which means that $|\mathcal{F}|^k = O(1/\epsilon)$, since the case of $\epsilon > 4 \cdot |\mathcal{F}|^{-k}$ can be handled by merely estimating the density of $h^{-1}(1)$. Specifically, note that any function that describes an $(\ell - k)$-dimensional subspace (over $\mathcal{F}$) is at distance exactly $|\mathcal{F}|^{-k}$ from the all-zero function. Hence, if $\epsilon > 4 \cdot |\mathcal{F}|^{-k}$ and h is 0.75ϵ-close to the all-zero function, then it is ϵ-close to describing a $(\ell - k)$-dimensional subspace, and it is OK to accept it. On the other hand, if $\epsilon > 4 \cdot |\mathcal{F}|^{-k}$ and h is 0.25ϵ-far from the all-zero function, then it cannot describe a $(\ell - k)$-dimensional subspace, and it is OK to reject it. Hence, we have

Claim 3.1 (reducing to the case of $\epsilon \leq 4 \cdot |\mathcal{F}|^{-k}$): *Testing whether a function $h : \mathcal{F}^\ell \to \{0,1\}$ describes a $(\ell - k)$-dimensional affine subspace can be randomly reduced to the case of $\epsilon \leq 4 \cdot |\mathcal{F}|^{-k}$, where the reduction introduces an additive overhead of $O(|\mathcal{F}|^k)$ queries.*

(An alternative justification boils down to resetting the proximity parameter to $\min(\epsilon, 4 \cdot |\mathcal{F}|^{-k})$; that is, on input proximity parameter ϵ and oracle h, we invoke the given tester on proximity parameter $\min(\epsilon, 4 \cdot |\mathcal{F}|^{-k})$ and provide it with oracle access to h.)

Another simplifying assumption is that we are dealing with linear subspaces rather than with affine ones. Actually, we present a reduction of the general case to this special case.[1]

Claim 3.2 (reducing to the linear case): *Testing whether a function $h : \mathcal{F}^\ell \to \{0,1\}$ describes a $(\ell-k)$-dimensional affine subspace can be randomly reduced to testing whether a function $h' : \mathcal{F}^\ell \to \{0,1\}$ describes a $(\ell-k)$-dimensional linear subspace, where the reduction introduces an additive overhead of $O(|\mathcal{F}|^k)$ queries.*

The foregoing randomized reduction has a two-sided error probability; obtaining an analogous one-sided error reduction is left as an open problem. Yet, since the known testers for linear subspaces (i.e., of [9] and of this paper) have two-sided error probability, our use of Claim 3.2 cause no real loss.

Proof: On input parameter $\epsilon > 0$ and oracle access to h, we proceed as follows.

1. Select uniformly a sample of $O(|\mathcal{F}|^k)$ points in $\mathcal{F}^\ell$. If h evaluates to 0 on all these points, then we reject. Otherwise, let u be a point in this sample such that $h(u) = 1$.

2. Invoke the tester for linear subspaces on input parameter ϵ and oracle access to h' defined by $h'(x) \stackrel{\text{def}}{=} h(x+u)$, and output its verdict. That is, each query x to h' is emulated by making the query $x+u$ to h.

The overhead of the reduction is due to Step 1, whereas in Step 2 we just invoke the tester for the special case.

If h describes an $(\ell-k)$-dimensional affine subspace, then, with high probability, Step 1 finds $u \in h^{-1}(1)$, since $h^{-1}(1)$ has density $|\mathcal{F}|^{-k}$, and we proceed to Step 2. But in this case it holds that $h'(x) = 1$ if and only if $x+u \in h^{-1}(1)$, which means that h' describes the $(\ell-k)$-dimensional linear space $h^{-1}(1) - u$, and so the invoked test accepts (w.h.p.).

(Indeed, if $H = \{yG+s : y \in \mathcal{F}^{\ell-k}\}$ is an affine space (as in Definition 2.2) and $u = (zG+s) \in H$, then $H - u = \{yG+s-u : y \in \mathcal{F}\} = \{yG+s-(zG+s) : y \in \mathcal{F}^{\ell-k}\} = \{(y-z)G : y \in \mathcal{F}^{\ell-k}\}$ is a linear space. Likewise, if $H' = \{yG : y \in \mathcal{F}^{\ell-k}\}$ is a linear space, then $H' + u = \{yG+u : y \in \mathcal{F}\}$ is an affine space.)

On the other hand, if h is ϵ-far from being an $(\ell-k)$-dimensional affine subspace, then either Step 1 rejects or else $u \in h^{-1}(1)$. But in this case h' (which was defined by $h'(x) \stackrel{\text{def}}{=} h(x+u)$) must be ϵ-far from describing an $(\ell-k)$-dimensional linear subspace. This is so because if h' is ϵ-close to g' that describes an $(\ell-k)$-dimensional linear subspace (i.e., g' describes the linear space $\{yG : y \in \mathcal{F}^{\ell-k}\}$), then $g(x) \stackrel{\text{def}}{=} g'(x-u)$ (equiv., $g(x+u) = g'(x)$) describes an affine space (i.e., g describes the affine space $\{yG+u : y \in \mathcal{F}^{\ell-k}\}$), whereas h is ϵ-close to g (since $h(x) = h'(x-u)$). Hence, in this case (i.e., h' is ϵ-far from describing an $(\ell-k)$-dimensional linear subspace), Step 2 rejects with high probability. ■

3.2 The first reduction

The pivotal step in the reduction is the definition of a function $g : \mathcal{F}^\ell \to \mathcal{F}^k \cup \{\perp\}$ such that if $H \stackrel{\text{def}}{=} h^{-1}(1)$ is an $(\ell-k)$-dimensional linear space, then g is linear (with image $\mathcal{F}^k$) and $g^{-1}(0^k) = H$. Furthermore, in that case, $g(x)$ indicates one of the $|\mathcal{F}|^k$ translations of H in which x resides;

[1] This reduction somewhat simplifies the presentation in Section 3.2, and more significantly so in Section 3.3.

that is, if $v^{(1)},...,v^{(k)} \in \mathcal{F}^\ell$ form a basis for the k-dimensional space that complements H, then $g(x)$ represents coefficients $(c_1,...,c_k) \in \mathcal{F}^k$ such that $x \in H - \sum_{i\in[k]} c_i v^{(i)}$.

Indeed, the definition of g is based on any fixed sequence of linearly independent vectors $v^{(1)},...,v^{(k)} \in \mathcal{F}^\ell$ such that for every non-zero sequence of coefficients $(c_1,...,c_k) \in \mathcal{F}^k$ it holds that $\sum_{i\in[k]} c_i v^{(i)} \notin H$. Such sequences of vectors exist[2] if H is an $(\ell-k)$-dimensional linear space, and we can find such a sequence in this case (by random sampling and querying h). Failure to find such a sequence will provide good justification for ruling that H is not an $(\ell-k)$-dimensional linear space.

Fixing such a sequence of $v^{(i)}$'s, we define $g : \mathcal{F}^\ell \to \mathcal{F}^k \cup \{\perp\}$ such that $g(x) = (c_1,...,c_k)$ if $(c_1,...,c_k) \in \mathcal{F}^k$ is the unique sequence that satisfies $x + \sum_{i\in[k]} c_i v^{(i)} \in H$ and let $g(x) = \perp \notin \mathcal{F}^k$ otherwise. Indeed, a unique sequence $(c_1,...,c_k) \in \mathcal{F}^k$ exists for each $x \in \mathcal{F}^\ell$ if H is an $(\ell-k)$-dimensional linear space, and in that case $g(x) \in \mathcal{F}^k$ for every $x \in \mathcal{F}^\ell$. But when H is not an $(\ell-k)$-dimensional linear space, it may happen that for some (or even all) x's there is no sequence $(c_1,...,c_k) \in \mathcal{F}^k$ such that $x + \sum_{i\in[k]} c_i v^{(i)} \in H$; similarly, it may happen that there are several different sequences $(c_1,...,c_k) \in \mathcal{F}^k$ that satisfy $x + \sum_{i\in[k]} c_i v^{(i)} \in H$. Anyhow, using matrix notation, we restate the foregoing definition next (where the $v^{(i)}$'s are the rows of the matrix V).

Definition 3.3 (the function $g = g_{H,V}$): *Let V be a k-by-ℓ full-rank matrix over $\mathcal{F}$ such that $cV \notin H$ for every $c \in \mathcal{F}^k \setminus \{0^k\}$. Then, $g_{H,V} : \mathcal{F}^\ell \to \mathcal{F}^k \cup \{\perp\}$ is defined such that $g_{H,V}(x) = c$ if $c \in \mathcal{F}^k$ is the unique vector that satisfies $x + cV \in H$, and $g_{H,V}(x) = \perp$ if the number of such k-long vectors is not one.*

Note that $g(x) = c$ implies $x + cV \in H$; hence, in particular, $g_{H,V}(x) = 0^k$ implies $x \in H$; that is, $g^{-1}_{H,V}(0^k) \subseteq H$.

3.2.1 Key observations

The most important observation is that if H is an $(\ell-k)$-dimensional linear space then g is a linear function, whereas if H is far from being an $(\ell-k)$-dimensional linear space then g is far from any linear function. Whenever we say that g is linear, we mean, in particular, that it never assumes the value $\perp$. (Indeed, when emulating g for the linearity tester, we shall reject if we ever encounter the value $\perp$.)

Claim 3.4 (H versus $g_{V,H}$): *Let H, V and $g = g_{H,V}$ be as in Definition 3.3. Then, H is an $(\ell-k)$-dimensional linear space if and only if g is a linear function with image $\mathcal{F}^k$.*

Actually, it turns out that if g is linear, then it has image $\mathcal{F}^k$; a more general statement is proved in Claim 3.6. Furthermore, if g is ϵ-close to being a linear function with image $\mathcal{F}^k$, then $g^{-1}(0^k)$ is ϵ-close to being an $(\ell-k)$-dimensional linear space (i.e., the indicator functions of these sets are ϵ-close). To see this, consider a linear g' that is ϵ-close to g, and note that the $(\ell-k)$-dimensional linear space $H' = \{x \in \mathcal{F}^\ell : g'(x) = 0^k\}$ is ϵ-close to $g^{-1}(0^k)$, since $g'(x) \neq g(x)$ for any x that resides in the symmetric difference of these sets.

Proof: Recall that $g^{-1}(0^k) \subseteq H$ always holds. Furthermore, equality (i.e., $g^{-1}(0^k) = H$) holds if g never assumes the value $\perp$, since in this case $x + cV \in H$ implies that $g(x) = c$ (and so $x \in H$ implies $g(x) = 0^\ell$).

[2] Actually, the density of suitable k-long sequences in H is $\prod_{i\in[k]}(1 - |\mathcal{F}|^{-i}) > 1/4$.

Now, on the one hand, if g is a linear function with image $\mathcal{F}^k$ (i.e., $g(x) = xT$ for some full-rank ℓ-by-k matrix T), then $H = g^{-1}(0^k)$ (i.e., $H = \{x \in \mathcal{F}^\ell : xT = 0^k\}$), which implies that H is an $(\ell - k)$-dimensional linear subspace (since $H = \{yG : y \in \mathcal{F}^{\ell-k}\}$ for any G that is a basis of the space orthogonal to $T^\top$).[3]

On the other hand, if H is an $(\ell - k)$-dimensional linear space, then, for some full-rank $(\ell - k)$-by-ℓ matrix G, it holds that $H = \{yG : y \in \mathcal{F}^{\ell-k}\}$. In this case, for every $x \in \mathcal{F}^\ell$ there exists a *unique* representation of x as $yG - cV$, since V is a basis for a k-dimensional linear space that complements the $(\ell - k)$-dimensional linear space H. Hence, for every $x \in \mathcal{F}^\ell$, there exists a unique $(c, y) \in \mathcal{F}^k \times \mathcal{F}^{\ell-k}$ such that $x + cV = yG \in H$, and $g(x) = c$ follows. We now observe that the image of g equals $\mathcal{F}^k$, since $g(0^\ell - cV) = c$ for every $c \in \mathcal{F}^k$, and that g is linear, since for every $x = yG - cV$ and $x' = y'G - c'V$ in $\mathcal{F}^\ell$, it holds that $g(x) + g(x') = c + c'$ and $c + c' = g(y''G - (c + c')V)$ holds for every $y'' \in \mathcal{F}^{\ell-k}$ (and in particular for $y'' = y + y'$, which implies that $c + c' = g(x + x')$). ■

Claim 3.5 (finding V): *Let $h : \mathcal{F}^\ell \to \{0,1\}$. If $H = h^{-1}(1)$ is an $(\ell - k)$-dimensional linear space, then a matrix V as underlying the definition of $g_{H,V}$ can be found* (w.h.p.) *by making $O(|\mathcal{F}|^k)$ queries to h.*

Proof: The matrix V can be found in k iterations as follows. In the i^{th} iteration we try to find a vector $v^{(i)}$ such that $\sum_{j\in[i]} c_j v^{(j)} \notin H$ holds for every $(c_1, ..., c_i) \in \mathcal{F}^i \setminus \{0^i\}$. In each trial, we pick $v^{(i)}$ at random, while noting that the probability of success is $1 - |\mathcal{F}|^{i-1} \cdot |\mathcal{F}|^{-k} \geq 1/2$, since for every $(c_1, ..., c_i) \in \mathcal{F}^i \setminus \{0^i\}$ it holds that $\mathbf{Pr}_{v^{(i)}}[\sum_{j\in[i]} c_j v^{(j)} \in H] = \mathbf{Pr}_{v^{(i)}}[c_i v^{(i)} \in H - \sum_{j\in[i-1]} c_j v^{(j)}] \leq |\mathcal{F}|^{-k}$, where equality holds if $c_i \neq 0$. Lastly, observe that the foregoing condition can be checked by making $|\mathcal{F}|^i - 1$ queries to h. (Actually, $|\mathcal{F}|^{i-1}$ queries suffice for checking in the i^{th} iteration, since it suffices to check the cases in which $c_i = 1$.)[4] ■

Claim 3.6 (on the linear function closest to $g_{V,H}$):[5] *Let H, V and $g = g_{H,V}$ be as in Definition 3.3. If g is 0.499-close to a linear function, then this linear function has image $\mathcal{F}^k$.*

The constant 0.499 can be replaced by any quantity that is smaller than $1 - |\mathcal{F}|^{-1} \geq 1/2$.

Proof: We consider a partition of $\mathcal{F}^\ell$ into $|\mathcal{F}|^{\ell-k}$ equivalence classes such that x and y are in the same class if $x - y$ is spanned by the rows of V; that is, x resides in the class $C_x \stackrel{\text{def}}{=} \{x + cV : c \in \mathcal{F}^k\}$ and $C_x = C_{x+c'V}$ for every $c' \in \mathcal{F}^k$. A class is considered good if it contains a single element of H, which happens if and only if $g(x) \in \mathcal{F}^k$. The key observation is that if C_x is good (equiv., $g(x) \in \mathcal{F}^k$), then, for every $c \in \mathcal{F}^k$, it holds that $g(x + cV) = g(x) - c$, since $C_x \cap H = \{x + g(x)V\} = \{(x + cV) + (g(x) - c)V\}$.

Now, let $f : \mathcal{F}^\ell \to \mathcal{F}^k$ be an arbitrary linear function that has an image that is partial to $\mathcal{F}^k$, and note that this image has size at most $|\mathcal{F}|^{k-1}$ (since the image of f must be a linear subspace).

[3] Alternatively, if $g(x + x') = g(x) + g(x')$ for every $x, x' \in \mathcal{F}^\ell$, then $x, x' \in H$ implies $x + x' \in H$ (for every $x, x' \in \mathcal{F}^\ell$), since $g(x) = g(x') = 0^k$ implies $g(x + x') = 0^k$. Hence, $H = g^{-1}(0^k)$ is a linear subspace. Lastly, we note that this subspace has dimension $\ell - k$, since the image of g equals $\mathcal{F}^k$ and $|g^{-1}(0^k)| = |g^{-1}(c)|$ holds for every $c \in \mathcal{F}^k$.

[4] First note that there is no need to check the cases in which $c_i = 0$. As for the other cases, by linearity of H, it holds that $\sum_{j\in[i]} c_j v^{(j)} \in H$ if and only if $\sum_{j\in[i]} (c_j/c_i) v^{(j)} \in H$.

[5] This observation was missed in prior versions of this work, leading to unnecessary checks in the original testers.

Noting that $g(x) = f(x)$ implies $g(x) \in \mathcal{F}^k$, we get

$$\begin{aligned}\mathbf{Pr}_{x\in\mathcal{F}^\ell}[g(x)=f(x)] &= \mathbf{Pr}_{x\in\mathcal{F}^\ell,c\in\mathcal{F}^k}[g(x)\in\mathcal{F}^k \;\&\; g(x+cV)=f(x+cV)] \\ &\leq \max_{x\in\mathcal{F}^\ell:g(x)\in\mathcal{F}^k}\{\mathbf{Pr}_{c\in\mathcal{F}^k}[g(x+cV)=f(x+cV)]\} \\ &\leq |\mathcal{F}|^{-1},\end{aligned}$$

since for any such $x \in \mathcal{F}^\ell$ (i.e., $g(x) \in \mathcal{F}^k$) and uniformly distributed $c \in \mathcal{F}^k$ it holds that $g(x + cV) = g(x) - c$ is uniformly distributed over $\mathcal{F}^k$, whereas the image of f contains at most $|\mathcal{F}|^{k-1}$ elements. It follows that g is at distance at least $1 - |\mathcal{F}|^{-1} \geq 1/2$ from any linear function that has image that is partial to $\mathcal{F}^k$. ■

3.2.2 The actual reduction

Combining the above three claims, the desired reduction follows (as detailed next). Note that this reduction has two-sided error, and that the resulting tester has query complexity $O(|\mathcal{F}|^k/\epsilon)$ (rather than $O(1/\epsilon)$, all in case $\epsilon < 4 \cdot |\mathcal{F}|^{-k}$).[6]

Algorithm 3.7 (testing whether H is an $(\ell - k)$-dimensional linear space): *On input a proximity parameter $\epsilon \in (0, 0.5)$ and oracle access to $h : \mathcal{F}^\ell \to \{0,1\}$, specifying $H = h^{-1}(1)$, proceed as follows.*

1. Find an adequate matrix V: *Using $O(|\mathcal{F}|^k)$ queries to h, try to find a k-by-ℓ full-rank matrix V such that for any non-zero $c \in \mathcal{F}^k$ it holds that $cV \notin H$. If such a matrix V is found, then proceed to the next step. Otherwise, reject.*

2. Test whether the function $g = g_{H,V}$ is linear: *Invoke a linearity test with proximity parameter ϵ, while providing it with oracle access to the function $g = g_{H,V}$. When the tester queries g at x, query h on $x + cV$ for all $c \in \mathcal{F}^k$, and answer accordingly; that is, the answer is c if c is the unique vector satisfying $h(x + cV) = 1$, otherwise* (i.e., $g(x) = \perp$) *the execution is suspended and the algorithm rejects. If the execution of the linearity tester is not suspended, then output its verdict.*

Recalling that linearity testing (with proximity parameter ϵ) has complexity $O(1/\epsilon)$, the complexity of the foregoing algorithm is $O(|\mathcal{F}|^k) + O(1/\epsilon) \cdot |\mathcal{F}|^k$, where the two terms correspond to the two steps.

Proposition 3.8 (analysis of Algorithm 3.7): *For any $\epsilon \in (0, 0.5)$, the following holds.*

1. *Algorithm 3.7 accepts every function h that describes an $(\ell - k)$-dimensional linear space with high probability.*

2. *Algorithm 3.7 rejects every function h that is ϵ-far from describing an $(\ell - k)$-dimensional linear space with high probability.*

[6] Needless to say, we would welcome a one-sided error reduction. Recall that the case $\epsilon \geq 4 \cdot |\mathcal{F}|^{-k}$ can be handled by density estimation. A complexity improvement for the main case (of $\epsilon < 4 \cdot |\mathcal{F}|^{-k}$) appears in Section 3.3.

Hence, Algorithm 3.7 constitutes a tester for $(\ell-k)$-dimensional linear spaces over $\mathcal{F}$ with query complexity $O(|\mathcal{F}|^k/\epsilon)$.

Proof: Suppose that H is an $(\ell-k)$-dimensional linear space. Then, by Claim 3.5, with high probability, a suitable matrix V will be found (in Step 1) and Step 2 will accept, since (by Claim 3.4) the function $g_{H,V}$ is linear. This establishes Part 1.

Turning to Part 2, if h is ϵ-far from describing an $(\ell-k)$-dimensional linear space, then either no suitable matrix V is found in Step 1, and the algorithm rejects, or such V is found. In the latter case, we consider the corresponding function $g_{H,V}$. We shall prove that $g_{H,V}$ is ϵ-far from the set of linear functions, and it follows that Step 2 rejects with high probability.

Using Claim 3.6, it suffices to show that $g = g_{H,V}$ is ϵ-far from being a linear function with image $\mathcal{F}^k$, since (for $\epsilon < 0.5$) if g is ϵ-close to a linear function then this linear function has image $\mathcal{F}^k$. Now, suppose towards the contradiction that g is ϵ-close to a linear function g' with image $\mathcal{F}^k$. Then, $H' = \{x \in \mathcal{F}^\ell : g'(x) = 0^k\}$ is an $(\ell-k)$-dimensional linear space, since $x, x' \in H'$ implies $x + x' \in H'$ (because $g'(x+x') = g'(x) + g'(x') = 0^k + 0^k = 0^k$), and $|H'| = |\mathcal{F}|^\ell/|\mathcal{F}|^k$ (because each image of g' has the same number of preimages). Next, letting $h' : \mathcal{F}^\ell \to \{0,1\}$ describe H' (i.e., $h'(x) = 1$ iff $x \in H'$), we show that $g'(x) = g(x)$ implies $h'(x) = h(x)$. This is because $g'(x) = g(x) = 0^k$ implies $h'(x) = 1$ and $h(x) = 1$ (since $g^{-1}(0^k) \subseteq h^{-1}(1)$), whereas $g'(x) = g(x) \notin \{0^\ell, \perp\}$ implies $h'(x) = 0$ and $h(x) \neq 1$ (since $h(x) = 1$ implies $g(x) \in \{0^\ell, \perp\}$). It follows that h is ϵ-close to h', which contradicts our hypothesis that h is ϵ-far from describing an $(\ell-k)$-dimensional linear space. ■

Remark 3.9 (extension to affine spaces): *In light of Claim 3.2 there is no real need to extend Algorithm 3.7 to the affine case, but let us outline such an extension nevertheless.*

- *The definition of g will be as in the linear case* (i.e., $g(x) = c$ iff $x + cV \in H$)*, except that it will be based on a full-rank k-by-ℓ matrix V such that for some $u \in H$ and every non-zero $c \in \mathcal{F}^k$ it holds that $u + cV \notin H$.* (Indeed, finding such a $u \in H$ is moved from Claim 3.2 to the revised algorithm.)
- *Claim 3.4 is extended to show that H is an $(\ell-k)$-dimensional affine space if and only if g is a affine function with image $\mathcal{F}^k$.*
- *Claim 3.6 is extended to show that if g is* 0.499*-close to an affine function, then this affine function has image $\mathcal{F}^k$.*

Recall that testing the affinity of $g : \mathcal{F}^\ell \to \mathcal{F}^k$ can be reduced to testing the linearity of the mapping $x \mapsto g(x) - g(0^\ell)$. Alternatively, one can just use the natural extension of the linearity test of [3] that selects $x, y, z \in \mathcal{F}^\ell$ uniformly at random and checks that $g(x+y) - g(y) = g(x+z) - g(z)$.

3.3 The second reduction

In Section 3.2, for $h : \mathcal{F}^\ell \to \{0,1\}$, we reduced ϵ-testing whether $h^{-1}(1)$ is an $(\ell-k)$-dimensional linear subspace to ϵ-testing the linearity of a function $g : \mathcal{F}^\ell \to \mathcal{F}^k \cup \{\perp\}$, where the value of g at any point can be computed by making $|\mathcal{F}|^k$ queries to h. (Indeed, in order to define the function g, the reduction made $O(|\mathcal{F}|^k)$ additional queries to h.) This yields an ϵ-tester of time complexity $O(|\mathcal{F}|^k/\epsilon)$ for testing $(\ell-k)$-dimensional linear subspaces. In this section we improve this time bound.

Our starting point is the fact that, for every $\epsilon < 1/4$, if g is ϵ-close to being a linear function, then it is ϵ-close to a unique linear function g', which can be computed by self-correction of g

(where each invocation of the self-corrector makes two queries to g and is correct with probability at least $1-2\epsilon$). Furthermore, if g' has image $\mathcal{F}^k$, then the corresponding Boolean function h' (i.e., $h'(x)=1$ iff $g'(x)=0^k$) describes an $(\ell-k)$-dimensional linear space, whereas if h describes an $(\ell-k)$-dimensional linear space then $g'=g$ and $h'=h$. This suggests a two-step algorithm in which we first invoke Algorithm 3.7 with constant proximity parameter ϵ_0 (e.g., $\epsilon_0=0.1$ will do), and then test equality between h and h'.

The key observation is that if h is ϵ-far from describing an $(\ell-k)$-dimensional linear space, then, with high probability, either the first step rejects or it yields a function $g=g_{H,V}$ that is ϵ_0-close to a linear function g' with image $\mathcal{F}^k$. In the latter case, the corresponding h', which describes an $(\ell-k)$-dimensional linear space (i.e., $\{x\in\mathcal{F}^\ell : h'(x)=1\}=\{x\in\mathcal{F}^\ell : g'(x)=0^k\}$), must be ϵ-far from h (by the foregoing hypothesis). (On the other hand, if h describes an $(\ell-k)$-dimensional linear space, then $h'=h$.)

High level structure of the new algorithm. Using the foregoing observation, we spell out the resulting algorithm, while leaving a crucial detail to later.

Step I: Invoke Algorithm 3.7 with proximity parameter set to ϵ_0, where $\epsilon_0\in(0,0.5)$ is a constant (e.g., $\epsilon_0=0.1$). If the said invocation rejects, then reject. Otherwise, let V be the matrix found in Step 1 of that invocation, and let $g=g_{H,V}$ be the corresponding function.

Let g' denote the *linear function closest to* g, and note that g is ϵ_0-close to g' (or else Algorithm 3.7 would have rejected with high probability). Furthermore, by Claim 3.6, the image of g' equals $\mathcal{F}^k$. Defining $h':\mathcal{F}^\ell\to\{0,1\}$ such that $h'(x)=1$ if and only if $g'(x)=0^k$, it follows that h' describes an $(\ell-k)$-dimensional linear subspace. Hence, *if h is ϵ-far from describing an $(\ell-k)$-dimensional linear subspace, then h is ϵ-far from h'.*

Step II (overview)**:** Test whether h equals h' by using a sample of $O(1/\epsilon)$ points. For each sample point, the value of h is obtained by querying h, whereas the value of h' on all sample points is obtained by obtaining the values of g' on these points (since $h'(x)=1$ iff $g'(x)=0^k$), where the values of g' on these points are computed via self-correction of g.

The problem is that each query to g is implemented by making $|\mathcal{F}|^k$ queries to h. Hence a straightforward implementation of the foregoing procedure will result in making $O(|\mathcal{F}|^k/\epsilon)$ queries to h, which is no better than Algorithm 3.7. Instead, we shall use a sample of $O(1/\epsilon)$ *pairwise-independent* points such that their g'-values can be determined by the value of g' at $O(\log(1/\epsilon))$ points, which in turn are computed by self-correction of g that uses $|\mathcal{F}|^k$ queries to h per each point. The details are given in Algorithm 3.10.

Note that if h describes an $(\ell-k)$-dimensional linear subspace, then $g=g'$, and the current step accepts if reached (which happens with high probability). On the other hand, if h is ϵ-far from this property and the current step is reached (which implies that g, g' and h' are well-defined), then h is ϵ-far from h', and a sample of $O(1/\epsilon)$ pairwise-independent points will contain a point of disagreement (w.h.p.).

The key observation here is that Step II can be implemented in complexity $\widetilde{O}(1/\epsilon)$ by taking a sample of $m=O(1/\epsilon)$ *pairwise independent points* in $\mathcal{F}^\ell$ such that evaluating g' on these m points only requires time $O(m+|\mathcal{F}|^k\cdot\widetilde{O}(\log m))$ rather than $O(|\mathcal{F}|^k\cdot m)$ time. This is done as follows.[7]

[7] This procedure is inspired by [8] (as presented in [6, Sec. 7.1.3] for $\mathcal{F}=\mathrm{GF}(2)$).

The pairwise independent sample points. For $t' = \lceil \log_{|\mathcal{F}|}(m+1) \rceil$, select uniformly $s^{(1)}, ..., s^{(t')} \in \mathcal{F}^\ell$, compute each $g'(s^{(j)})$ via self-correcting g, with error probability $o(1/t')$, and use the sample points $r^{(L)} = L(s^{(1)}, ..., s^{(t')})$ for m non-zero linear function $L : \mathcal{F}^{t'} \to \mathcal{F}$. The key observations are that (1) the $r^{(L)}$'s are pairwise independent, and (2) the values of g' at all $r^{(L)}$'s can be determined based on the values of g' on the $s^{(j)}$'s. This determination is based on the fact that $g'(r^{(L)}) = L(g'(s^{(1)}), ..., g'(s^{(t')})$, by linearity of g'. Hence, the values of g' on t' random points (i.e., the $s^{(j)}$'s) determines the value of g' on $m \leq |\mathcal{F}|^{t'} - 1$ pairwise independent points (i.e., the $r^{(L)}$'s). This yields the following —

Algorithm 3.10 (implementing Step II): *For $m = O(1/\epsilon)$ and $t' = \lceil \log_{|\mathcal{F}|}(m+1) \rceil$, set $t'' = O(\log_2 t')$, and proceed as follows.*

1. *Select uniformly $s^{(1)}, ..., s^{(t')} \in \mathcal{F}^\ell$.*

2. *For each $j \in [t']$, compute the value of $g'(s^{(j)})$ by using self-correction on g, which in turn queries h on $|\mathcal{F}|^k$ points per each query to g. The self-correction procedure is invoked t'' times so that the correct value is obtained with probability $1 - o(1/t')$.*

   Specifically, select uniformly $w^{(1)}, ..., w^{(t'')} \in \mathcal{F}^\ell$, and set $\sigma^{(j)}$ to equal the majority vote of $g(s^{(j)} + w^{(1)}) - g(w^{(1)}), ..., g(s^{(j)} + w^{(t'')}) - g(w^{(t'')})$, where the values of g at each point x is determined according to the value of h at the points $\{x + cV : c \in \mathcal{F}^k\}$.

   Recall that $g(x) = c$ if c is the unique point in $\mathcal{F}^k$ such that $h(x + cV) = 1$, and is set to $\perp$ otherwise. If the value of g at any point is set to $\perp$, then we can abort this algorithm and reject h. Alternatively, we can set $\sigma^{(j)}$ to $\perp$, and define the $z + \perp = \perp$ for every $z \in \mathcal{F}^k$.

   (Indeed, $\sigma^{(j)}$ is our sound guess for $g'(s^{(j)})$, and this guess is correct with probability $1 - \exp(-\Omega(t'')) = 1 - o(1/t')$).

3. *For each of m non-zero linear function $L : \mathcal{F}^{t'} \to \mathcal{F}$, let $r^{(L)} = L(s^{(1)}, ..., s^{(t')})$ and check whether $h(r^{(L)})$ equals our guess for $h'(r^{(L)})$, where the later value is set to 1 if and only if $L(\sigma^{(1)}, ..., \sigma^{(t')}) = 0^k$. Accept if and only if all checks were successful* (i.e., equality holds in all).

   (Recall that $g'(r^{(L)}) = g'(L(s^{(1)}, ..., s^{(t')})) = L(g'(s^{(1)}), ..., g'(s^{(t')}))$. Hence, $L(\sigma^{(1)}, ..., \sigma^{(t')})$ is our sound guess for $g'(r^{(L)})$, and this guess is correct if all guesses for the $g'(s^{(j)})$'s are correct, which happens with probability $1 - o(1)$).

The time complexity of Algorithm 3.10 is $O(t' \cdot t'' \cdot |\mathcal{F}|^k + m) = \widetilde{O}(\log_{|\mathcal{F}|}(1/\epsilon)) \cdot |\mathcal{F}|^k + O(1/\epsilon)$. Hence, the time complexity of Step II dominates the time complexity of Step I, which is $O(|\mathcal{F}|^k/\epsilon_0) = O(|\mathcal{F}|^k)$. Assuming $\epsilon = O(|\mathcal{F}|^{-k})$, the resulting algorithm has time complexity $\widetilde{O}(\log(1/\epsilon)) \cdot \epsilon^{-1} = \widetilde{O}(1/\epsilon)$. Recall that for $\epsilon > 4 \cdot |\mathcal{F}|^{-k}$ there is an almost trivial tester of complexity $O(1/\epsilon)$, and note that for $\epsilon < |\mathcal{F}|^{-k-1.01 \cdot \log_{|\mathcal{F}|} k}$ it holds that $\widetilde{O}(\log_{|\mathcal{F}|}(1/\epsilon)) \cdot |\mathcal{F}|^k = O(1/\epsilon)$. Hence, our complexity bound is (slightly) inferior to the optimal bound of $O(1/\epsilon)$ only for a narrow range of parameters (i.e., for $\epsilon \in [k^{-1.01} \cdot |\mathcal{F}|^{-k}, 4 \cdot |\mathcal{F}|^{-k}]$).

Theorem 3.11 (analysis of the foregoing algorithm): *Consider an algorithm that invokes the algorithm captured by the foregoing Steps I and II, where Step II is implemented as detailed in Algorithm 3.10. Then, the resulting algorithm constitutes a tester for $(\ell - k)$-dimensional linear subspaces.*

Note that this tester has two-sided error probability. Obtaining an analogous one-sided error tester is left as an open problem. It is indeed possible that such a tester does not exist.

Proof: If h describes an $(\ell-k)$-dimensional linear subspace, then (w.h.p.) the execution reaches Step II, which always accepts, since in this case $h' = h$. On the other hand, if h is ϵ-far from describing an $(\ell-k)$-dimensional linear subspace, then we consider two cases.

1. If h is ϵ_0-far from describing an $(\ell-k)$-dimensional linear subspace, then Step I rejects (w.h.p).

2. Otherwise, assuming that Step II is reached, we consider the corresponding functions $g = g_{H,V}$ and g'. Recall g' is a linear function that is ϵ_0-close to g, since otherwise Step I rejects with high probability (due to its linearity test), and that the image of g' equals $\mathcal{F}^k$ (by Claim 3.6). Hence, h must be ϵ-far from h', since in this case h' describes an $(\ell-k)$-dimensional linear subspace.

   In this case, with probability at least $1-o(1)$, the tester obtains the correct values of g' at all $s^{(j)}$'s and hence determined correctly the values of g' at all the $r^{(L)}$'s. Since these $r^{(L)}$ are uniformly distributed in $\mathcal{F}^\ell$ in a pairwise independent manner, with high probability (i.e., w.p. at least $1-\frac{m\epsilon}{(m\epsilon)^2}$), the sample contains a point on which h and h' disagree, and Step II rejects.

In conclusion, if h is ϵ-far from the tested property, then the foregoing algorithm rejects with high probability. The theorem follows. ■

Remark 3.12 (extension to affine spaces): *Again, there is no real need to extend the foregoing to the affine case, but we outline such an extension nevertheless. We first note that self-correction of g that is close to an affine g' requires querying g at three random locations rather than at two; specifically, to obtain $g'(s)$, we select uniformly $r, r' \in \mathcal{F}^\ell$ and query g at $s+r+r', r, r'$, while relying on $g'(s) = g'(s+r-r') - g'(r) + g'(r')$. Likewise, the equality $g'(r^{(L)}) = L(g'(s^{(1)}), ..., g'(s^{(t')})$ is replaced by $g'(r^{(L)}) = L(g'(s^{(1)}), ..., g'(s^{(t')}) - L(v, ..., v) + v$, where $v = g'(0^\ell)$ is also obtained by self-correction of g, which calls for making $3 \cdot |\mathcal{F}|^k$ additional queries.*

4 On testing monotone monomials

As stated in the introduction, the problem that motivated our study is trying to reduce testing monomials to testing dictatorships. Such a reduction is preferable to an extension of the ideas that underly the tests of (monotone) dictatorship towards testing the set of functions that are (monotone) k-monomials, for any $k \geq 1$. In this section, we first review the said extension, as performed by Parnas, Ron, and Samorodnitsky [9], and then we present our alternative. We start with the definition of the relevant properties.

Definition 4.1 (monomial and monotone monomial): *A Boolean function $f : \{0,1\}^\ell \to \{0,1\}$ is called a k-*monomial *if for some k-subset $I \subseteq [\ell]$ and $\sigma = \sigma_1 \cdots \sigma_\ell \in \{0,1\}^\ell$ it holds $f(x) = \wedge_{i \in I}(x_i \oplus \sigma_i)$. It is called a* monotone k-monomial *if $\sigma = 0^\ell$.*

Indeed, the definitions of (regular and monotone) 1-monomials coincide with the notions of (regular and monotone) dictatorships. We focus on the task of testing monotone k-monomials, while recalling that the task of testing k-monomials is reducible to it (see [9] or [7, Sec. 5.2.2.1]). We also recall that this testing problem is of interest only when the proximity parameter, denoted ϵ, is small (in relation to 2^{-k}). In contrast, when $\epsilon > 2^{-k+2}$, we may just estimate the density of $f^{-1}(1)$ and accept if and only if the estimate is below $\epsilon/2$.

4.1 The tester of Parnas, Ron, and Samorodnitsky

We start by interpreting the dictatorship tester of [1, 9] in a way that facilitates its generalization. Recall that these works perform a dictatorship test by first testing that the function is linear and then performing a "conjunction check" (i.e., checking that $f(x \wedge y) = f(x) \wedge f(y)$). Now, if f is a monotone dictatorship, then $f^{-1}(1)$ is an $(\ell-1)$-dimensional affine subspace (of the ℓ-dimensional space $\{0,1\}^\ell$), where $\{0,1\}$ is associated with the two-element field GF(2). Specifically, if $f(x) = x_i$, then this subspace is $\{x \in \{0,1\}^\ell : x_i = 1\}$. In this case, the *linearity tester* could be thought of as testing that $f^{-1}(1)$ is an arbitrary $(\ell-1)$-dimensional affine subspace, whereas the "conjunction check" verifies that this subspace is an affine translation by 1^ℓ of a linear space that is spanned by $\ell - 1$ unit vectors (i.e., vectors of Hamming weight 1).[8]

When generalizing the treatment for abitrary k, we observe that if f is a monotone k-monomial, then $f^{-1}(1)$ is an $(\ell-k)$-dimensional affine subspace. So the foregoing two-step procedure generalizes to first testing that $f^{-1}(1)$ is an $(\ell-k)$-dimensional affine subspace, and then testing that it is an affine subspace of the right form (i.e., it has the form $\{x \in \{0,1\}^\ell : (\forall i \in I)\ x_i = 1\}$, for some k-subset I). Following are outlines of the treatment of these two tasks in [9].

Testing affine subspaces. Supposed that the alleged affine subspace H is presented by a Boolean function h such that $h(x) = 1$ if and only if $x \in H$. (Indeed, in our application, $h = f$.) We wish to test that H is indeed an affine subspace.

(Actually, we are interested in testing that H has a given dimension, but this extra condition can be checked easily by estimating the density of H in $\{0,1\}^\ell$, since we are willing to have complexity that is inversely proportional to the designated density (i.e., 2^{-k}).)[9]

This task is related to linearity testing and it was indeed solved in [9] using a tester and an analysis that resembles the standard linearity tester of [3]. Specifically, the tester selects uniformly $x, y \in H$ and $z \in \{0,1\}^\ell$ and checks that $h(x-y+z) = h(z)$ (i.e., that $x - y + z \in H$ if and only if $z \in H$). Indeed, we uniformly sample H by repeatedly sampling $\{0,1\}^\ell$ and checking whether the sampled element is in H.

Note that, for co-dimension $k > 1$, the function h is not affine (i.e., $h(x) = h(y) = h(z) = 0$, which means $x, y, z \notin H$, does not determine the value of $h(x - y + z)$ (i.e., whether or not $x - y + z \in H$)).[10] Still, testing whether h describes an affine subspace can be reduced to testing linearity of functions (albeit not of h but rather of a related function), providing an alternative to the tester of [9]. Presenting such a reduction is the core of this paper (see Section 3, which handles an arbitrary finite field $\mathcal{F}$, whereas the current application only requires $\mathcal{F} = \mathrm{GF}(2)$).

Testing that an affine subspace is a translation by 1^ℓ of a linear subspace spanned by unit vectors. Suppose that an affine subspace H' is presented by a Boolean function, denoted

[8] That is, we requires that this subspace has the form

$$\left\{ 1^\ell + \sum_{j \in ([\ell] \setminus \{i\})} c_j e_j : c_1, ..., c_\ell \in \{0,1\} \right\}$$

where $e_1, ..., e_\ell \in \{0,1\}^\ell$ are the ℓ unit vectors (i.e., vectors of Hamming weight 1).

[9] Recall that if $\epsilon < 2^{-k+2}$, then $O(2^k) = O(1/\epsilon)$, and otherwise (i.e., for $\epsilon \geq 2^{-k+2}$) testing affinity of H reduces to estimating the density of $h^{-1}(1)$.

[10] Note that $x, y \notin H$ does not determine whether or not $x - y$ is in the linear space $H - H$, let alone that a negative answer does not allow to related $h(x-y+z)$ to $h(z)$. In contast, $x, y \in H$ implies that $h(x-y+z) = h(z)$ for every $z \in \{0,1\}^\ell$.

h', and that we wish to test that H' has the form

$$\left\{ 1^\ell + \sum_{i \in [\ell] \setminus I} c_i e_i : c_1, ..., c_\ell \in \{0,1\} \right\}$$

where $e_1, ..., e_\ell \in \{0,1\}^\ell$ are unit vectors, and $I \subseteq [\ell]$ is arbitrary. That is, we wish to test that $h'(x) = \wedge_{i \in I} x_i$.

This can be done by picking uniformly $x \in H'$ and $y \in \{0,1\}^\ell$, and checking that $h'(x \wedge y) = h'(y)$ (i.e., $x \wedge y \in H'$ if and only if $y \in H'$). Note that if H' has the form $1^\ell + L$, where L is a linear subspace spanned by the unit vectors $\{e_i : i \in [\ell] \setminus I\}$ for some I, then $h'(z) = \wedge_{i \in I} z_i$ holds for all $z \in \{0,1\}^\ell$, and $h'(x \wedge y) = h'(x) \wedge h'(y)$ holds for all $x, y \in \{0,1\}^\ell$. On the other hand, as shown in [9], if H' is an affine subspace that does not have the foregoing form, then the test fails with probability at least 2^{-k-1}.

However, as in the case of $k = 1$, we do not have access to h' but rather to a Boolean function h that is (very) close to h'. So we need to obtain the value of h' at specific points by querying h at uniformly distributed points. Specifically, the value of h' at z is obtained by uniformly selecting $r, s \in h^{-1}(1)$ and using the value $h(z + r - s)$. In other words, we self-correct h at any desired point z by using the value of h at a point obtained by shifting z by the difference between two random elements of $h^{-1}(1)$, while hoping that these points actually reside in the affine subspace H' (so that their difference is in the linear space $H' - H'$). This hope is likely to materialize when h is $0.01 \cdot 2^{-k}$-close to h'.

The foregoing is indeed related to the conjunction check performed as part of the dictatorship tester of [1, 9], and the test and the analysis in [9] (for the case of $k > 1$) resemble the corresponding parts in [1, 9] (which handle the case of $k = 1$).

In contrast, building on the main idea of Section 3, in Section 4.2, we present a simple reduction from the general case (of any $k \geq 1$) to testing that a linear function depends on at most k of its variables. Note that the later problem generalizes the problem that was solved by the conjunction check when $k = 1$.

4.2 An alternative tester of monotone monomials

As outlined in Section 4.1, the function $f : \{0,1\}^\ell \to \{0,1\}$ is a monotone k-monomial if and only if f describes an $(\ell - k)$-dimensional affine space that is a translation by 1^ℓ of an $(\ell - k)$-dimensional axis-parallel linear space; that is, if $f^{-1}(1)$ has the form $\{yG + 1^\ell : y \in \{0,1\}^{\ell-k}\}$, where G is a full-rank $(\ell - k)$-by-ℓ Boolean matrix that contains k all-zero columns. Hence, we may focus on testing that the function $h : \{0,1\}^\ell \to \{0,1\}$ defined by $h(x) \stackrel{\text{def}}{=} f(x + 1^\ell)$ describes an $(\ell - k)$-dimensional *axis-parallel* linear space. (Indeed, the reduction of Section 3.1 is instantiated here by mandating $u = 1^\ell$.)

Following [9], we first test that the Boolean function h describes an $(\ell - k)$-dimensional linear space, and next test that this linear space has the right form. As detailed Section 3, the first task is reduced to testing the linearity of a corresponding function $g : \{0,1\}^\ell \to \{0,1\}^k \cup \{\perp\}$, and if this test passes (w.h.p.) then g must be close to a linear function $g' : \{0,1\}^\ell \to \{0,1\}^k$, which has image $\{0,1\}^k$. In this case, h is closed to the Boolean function h' that describes the $(\ell - k)$-dimensional linear space $\{x \in \{0,1\}^\ell : g'(x) = 1\}$.

The key observation is that h' describes an axis-parallel linear space (i.e., the set $\{x \in \{0,1\}^\ell : h'(x) = 1\}$ equals the linear space $\{yG : y \in \{0,1\}^{\ell-k}\}$ for some full-rank $(\ell - k)$-by-ℓ matrix G

with k all-zero columns) if and only if g' depends on k variables.[11] In general:

Claim 4.2 (on axis-parallel linear spaces): *Let $g' : \{0,1\}^\ell \to \{0,1\}^k$ be a linear function with image $\{0,1\}^k$, and $H' = \{x \in \{0,1\}^\ell : g'(x) = 0^k\}$. Then, for every k-subset I, it holds that $H' = \{x \in \{0,1\}^\ell : x_I = 0^k\}$ if and only if g' depends only on the bits in locations I* (equiv., there exists an invertible linear function $T : \{0,1\}^k \to \{0,1\}^k$ such that $g'(x) = T(x_I)$).

In case of $k = 1$, Claim 4.2 coincides with the assertion that $H' = \{x \in \{0,1\}^\ell : x_i = 0\}$ *if and only if g' is the i^{th} dictatorship function* (i.e., $g'(x) = x_i$).

Proof: First note that if g' depends only on the bits in locations I, then $H' = \{x : g'(x) = 0^k\}$ is an axis-parallel linear space; specifically, if $g'(x) = T(x_I)$ for some linear function $T : \{0,1\}^k \to \{0,1\}^k$, which must be invertible (since the image of g' equals $\{0,1\}^k$), then $H' = \{x : T(x_I) = 0^k\} = \{x : x_I = 0^k\}$, since $T(z) = 0^k$ if and only if $z = 0^k$.

Turning to the opposite direction, suppose that $H' = \{x : x_I = 0^k\}$, and recall that $H' = \{x : g'(x) = 0^k\}$. Hence, $g'(x) = 0^k$ if and only if $x_I = 0^k$. To see that g' necessarily depends only on variables in locations I, suppose that some bit of $g'(x)$ depends on variable $j \notin I$, and derive a contradiction by considering $e^{(j)} = 0^{j-1}10^{\ell-j}$ (i.e., evidently $e^{(j)}_I = 0^k$, but $g'(e^{(j)}) \neq 0^k$ due to the bit in the value of g' that depends on location j). ■

It seems natrual to refer to a linear function that depends only on k of its variables by the term k-linear; this term was used before when referring to Boolean functions (i.e., functions from $\{0,1\}^\ell$ to $\{0,1\}$). Recall that, for constant proximity parameter, testing whether a Boolean function is k-linear can be performed in time $O(k \log k)$ by using either a junta tester (e.g., Blais's [2]) or a general function isomorphism tester (cf. [4]). It seems that these testers extend also to the case of functions from $\{0,1\}^\ell$ to $\{0,1\}^m$, but much simpler testers can be applied when we are guaranteed that the tested function is linear. For sake of self-containment, we present such a tester in the appendix.

However, as in Section 3, we do not have access to g', but rather can obtain its values at desired points by applying self-correction to g, which is close to g'. We can afford to compute g at any desired point, since we intend to do so only for $\widetilde{O}(k)$ points. Specifically, each query x of the k-linearity tester is answered by $g(x + w) - g(w)$, where w is selected uniformly in $\{0,1\}^\ell$. To wrap-up, we obtain the following tester, where we assume (for simplicity and w.l.o.g.) that $\epsilon = O(2^{-k}) = o(1/\widetilde{O}(k))$.

Algorithm 4.3 (testing whether f is a monotone k-monomial): *On input a proximity parameter $\epsilon \in (0, O(2^{-k})]$ and oracle access to $f : \{0,1\}^\ell \to \{0,1\}$, the algorithm proceeds as follows.*

1. *Apply the tester of Section 3.3 to test whether $h : \{0,1\}^\ell \to \{0,1\}$, defined by $h(x) \stackrel{\text{def}}{=} f(x+1^\ell)$, describes an $(\ell - k)$-dimensional linear space over* GF(2). *The said tester is invoked with proximity parameter ϵ, and each query x is answered by the value $f(x + 1^\ell)$. If the foregoing tester rejects, then the current algorithm reject.*

2. *Find a matrix V as in Step 1 of Algorithm 3.7, and let $g = g_{h^{-1}(1),V} : \{0,1\}^\ell \to \{0,1\}^k$ denote the corresponding function.* (Alternatively, we may use the matrix V that is found by the fortegoing invocation of Algorithm 3.7.)

   Letting g' be the linear function closest to g, we check whether the function g' is k-linear.

[11] In some previous versions of this work it was stated that h' describes an axis-parallel linear space if and only if g' is a projection function (i.e., $g'(x) = x_I$ for some k-subset I). As shown next, this statement is wrong.

(Recall g' is computed by self-reduction of g, whereas the value of g at x is computed by querying h at the points $\{x + cV : c \in \{0,1\}^k\}$.)

The query complexity of Algorithm 4.3 equals $(\widetilde{O}(\log(1/\epsilon)) \cdot 2^k + O(1/\epsilon)) + \widetilde{O}(k) \cdot 2^k$, where the first (resp., second) term is due to Step 1 (resp., Step 2).

Proposition 4.4 (analysis of Algorithm 4.3): *Assuming that $\epsilon \in (0, O(2^{-k})]$, the following holds.*

1. *Algorithm 4.3 accepts any monotone k-monomial with high probability.*
2. *Algorithm 4.3 rejects any function that is ϵ-far from being a monotone k-monomial with high probability.*

Hence, Algorithm 4.3 constitutes a tester for monotone k-monomials.

Proof: The small error probability in the case that f is a monotone k-monomial is due to Step 1. Furthermore, in this case h describes an $(\ell - k)$-dimensional axis-parallel linear space, and $g' = g$ depends on k variables, and so Step 2 (if reached) accepts with probability 1.

The analysis of the case of functions that are ϵ-far from being monotone k-monomials reduces to the analysis of Step 2, in which we may assume that g is ϵ-close to a linear function g' with image $\{0,1\}^k$, since otherwise Step 1 rejects (w.h.p.). By Claim 4.2, g' must depend on more than k variables, since otherwise h is ϵ-close to describing an $(\ell - k)$-dimensional axis-parallel linear space. But in this case Step 2 rejects with high probability, because the probability that any invocation of the self-corrector is wrong is at most $\widetilde{O}(k) \cdot 2\epsilon = o(1)$. ■

Comparison to [9]. Algorithm 4.3 differs from the tester of [9] in two aspects. Firstly, Step 1 uses the tester of Section 3.3, which is based on a reduction of testing affine spaces to testing linear functions. Specifically, we reduce testing that $f : \{0,1\}^\ell \to \{0,1\}$ describes an affine space to testing that a related $h : \{0,1\}^\ell \to \{0,1\}$ describes a linear space, which in turn is reduced to testing that a related $g : \{0,1\}^\ell \to \{0,1\}^k$ is linear. In contrast, testing affine spaces is performed in [9] by modifying the linearity tester of [3] and mimicking the known analysis of this tester.

Second, Step 2 uses the foregoing function g (of Step 1), and reduces testing that the linear space described by h has the right form to testing that g depends on k of its variables, which generalizes the condition used in the case of $k = 1$. In contrast, as can be seen in Section 4.1, the path taken by [9] involves a modification of the procedure and analysis used in the case of $k = 1$.

Acknowledgements

I am grateful to Roei Tell for reading prior versions of this text and pointing out numerous inaccuracies and gaps. I also wish to thank Clement Canonne and Tom Gur for their comments. Lastly, I am grateful to Eric Blais for helpful discussions regarding the problem of testing k-linearity, and for permission to include his ideas in the appendix. This research was partially supported by the Israel Science Foundation (grant No. 671/13).

Appendix: On testing k-linearity

For an arbitrary finite field $\mathcal{F}$ and integers $k, \ell, m \in \mathbb{N}$, we say that $f : \mathcal{F}^\ell \to \mathcal{F}^m$ is k-linear if f is a linear function that depends on at most k (out of its ℓ) variables. Assuming that the tested function f is linear, we present a simple test that always accepts k-linear functions and rejects (w.h.p.) linear functions that are not k-linear.

The tester uses the key idea of the k-junta test of Fischer *et al.* [5], which is to randomly partition the ℓ variables to $O(k^2)$ sets, hoping that (in the case of a yes-instance) each set contains at most one variable that influences the function's value, and detecting which of the sets contain such a variable, hereafter called an influential variable. We accept if and only if at most k sets are detected as containing influential variables. To check whether a set $S \subset [\ell]$ contains influential variables, we select an arbitrary (e.g., random) assignment to the variables in $\overline{S} = [\ell] \setminus S$ and two random (and independent) assignments to the variables in S, and check whether the function's value is the same under both assignments.[12]

Given that the tested function is linear, the analysis of this procedure is quite straightforward (and, indeed, much simpler than in [5]): If S contains some influential variables of f, then the function's value changes with probability at least $1 - |\mathcal{F}|^{-1}$ (e.g., it is exactly $1 - |\mathcal{F}|^{-1}$ in case $m = 1$), whereas the value does not change if S contains no influential variable. The complexity of this procedure is $\Omega(k^2)$, but we can improve it by using two levels of partitions. Specifically, we suggest the following procedure (where the stated expectations refer to the case that the function is $O(k)$-linear).[13]

1. Select a random partition of $[\ell]$ into $t = k/\log k$ parts, expecting $O(\log k)$ influential variables in each part.
2. For each part S in this t-partition, perform the following (three-step) trial for $t' = O(\log k)$ times.
   (a) Select a random partition of S to $O(\log k)^2$ sub-parts, expecting at most a single influential variable in each sub-part.
   (b) For each sub-part, check whether it contains influential variables (by selecting two random assignments as described above). Actually, we perform this check $O(\log k)$ times such that a sub-part that contains influential variables is detected to be so with probability at least $1 - o(1/k)$.

[12] We can set the values of the variables in $\overline{S}$ arbitrarily since we are testing a linear function. In contrast, when testing for k-junta (as in [5]) it is essential to assign the variables in $\overline{S}$ at random, and perform this test of influence many times.

[13] In the general case, these expectations may not be satisfied, but such violation provides statistical evidence for rejection.

(c) Set the vote of the current trial to equal the number of sub-parts that were detected as containing influential variables.

(Note that if the sub-partition selected in Step (a) is good (i.e., each sub-part contains at most one influential variable) and all relevant sub-parts were detected as containing influential variables, then the current vote equals the number of influential variables in S.)

For each S, set the verdict for number of influential variables in S to equal the largest vote obtained in the t' corresponding trials, expecting the answer to be correct with probability $1 - o(1/t)$.

3. Accept if and only if the sum of the verdicts is at most k.

The foregoing test has complexity $t \cdot t' \cdot O(\log k)^3 = \widetilde{O}(k)$, but this can be improved using an adaptive procedure. Specifically, the following recursive procedure was suggested to us by Eric Blais.

1. Select a random partition of $[\ell]$ into k parts, expecting $\Omega(k)$ parts with a single variable.
2. For each part S in this k-partition, determine whether it contains exactly one variable by performing the following (two-step) trial for $t' = O(\log k)$ times.
   (a) Select a random 2-partition of S and check whether each sub-part contains influential variables. (We stress that each of these two checks is performed once.)
   (b) Set the vote of the current trial to equal the number of sub-parts that were detected as containing influential variables.

   For each S, declare this part as having a single influential variable if and only if the maximum vote for it equals 1.

   (Note that, with probability $1 - o(1/k)$, the declaration regarding S is correct; that is, S is declared as as having a single influential variable if and only if this is actually the case.)
3. Let U be the union of the parts declared to have a single influential variable, and k' be their number. If more than k parts were found to contain influential varaibles (e.g., $k' > k$) then reject; if $k' \leq k$ and no part was found to contain more than one influential variable then accept. Otherwise, recurse with the function restricted to $[\ell] \setminus U$ and $k \leftarrow k - k'$.

Observing that for k-linear functions, with high probability, the number of parts in all recursion calls is $O(k)$, it follows that this procedure has complexity $O(k \log k)$.

References

[1] M. Bellare, O. Goldreich and M. Sudan. Free Bits, PCPs and Non-Approximability – Towards Tight Results. *SIAM Journal on Computing*, Vol. 27, No. 3, pages 804–915, 1998. Extended abstract in *36th FOCS*, 1995.

[2] E. Blais. Testing Juntas Nearly Optimally. In *41st ACM Symposium on the Theory of Computing*, pages 151–158, 2009.

[3] M. Blum, M. Luby and R. Rubinfeld. Self-Testing/Correcting with Applications to Numerical Problems. *Journal of Computer and System Science*, Vol. 47, No. 3, pages 549–595, 1993. Extended abstract in *22nd STOC*, 1990.

[4] S. Chakraborty, D. Garcia-Soriano, and A. Matsliah. Nearly Tight Bounds for Testing Function Isomorphism. In *22nd ACM-SIAM Symposium on Discrete Algorithms*, pages 1683–1702, 2011.

[5] E. Fischer, G. Kindler, D. Ron, S. Safra and A. Samorodnitsky. Testing juntas. *Journal of Computer and System Science*, Vol. 68 (4), pages 753–787, 2004. Extended abstract in *44th FOCS*, 2002.

[6] O. Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 2008.

[7] O. Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017.

[8] O. Goldreich and L.A. Levin. A Hard-Core Predicate for all One-Way Functions. In the proceedings of *21st ACM Symposium on the Theory of Computing*, pages 25–32, 1989.

[9] M. Parnas, D. Ron, and A. Samorodnitsky. Testing Basic Boolean Formulae. *SIAM Journal on Disc. Math. and Alg.*, Vol. 16 (1), pages 20–46, 2002.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il