Absolutely Sound Testing of Lifted Codes

Elad Haramaty[1] ⋆, Noga Ron-Zewi[1] ⋆⋆, and Madhu Sudan[2]

[1] Department of Computer Science, Technion, Haifa.
{eladh,nogaz}@cs.technion.ac.il
[2] Microsoft Research New England, Cambridge, MA. madhu@mit.edu

Abstract. In this work we present a strong analysis of the testability of a broad, and to date the most interesting known, class of "affine-invariant" codes. Affine-invariant codes are codes whose coordinates are associated with a vector space and are invariant under affine transformations of the coordinate space. Affine-invariant linear codes form a natural abstraction of algebraic properties such as linearity and low-degree, which have been of significant interest in theoretical computer science in the past. The study of affine-invariance is motivated in part by its relationship to property testing: Affine-invariant linear codes tend to be locally testable under fairly minimal and almost necessary conditions.

Recent works by Ben-Sasson et al. (CCC 2011) and Guo et al. (ITCS 2013) have introduced a new class of affine-invariant linear codes based on an operation called "lifting". Given a base code over a t-dimensional space, its m-dimensional lift consists of all words whose restriction to every t-dimensional affine subspace is a codeword of the base code. Lifting not only captures the most familiar codes, which can be expressed as lifts of low-degree polynomials, it also yields new codes when lifting "medium-degree" polynomials whose rate is better than that of corresponding polynomial codes, and all other combinatorial qualities are no worse.

In this work we show that codes derived from lifting are also testable in an "absolutely sound" way. Specifically, we consider the natural test: Pick a random affine subspace of base dimension and verify that a given word is a codeword of the base code when restricted to the chosen subspace. We show that this test accepts codewords with probability one, while rejecting words at constant distance from the code with constant probability (depending only on the alphabet size). This work thus extends the results of Bhattacharyya et al. (FOCS 2010) and Haramaty et al. (FOCS 2011), while giving concrete new codes of higher rate that have absolutely sound testers. In particular we show that there exists codes satisfying the requirements of Barak et al. (FOCS 2012) to construct small set expanders with a large number of eigenvalues close to

⋆ Research was conducted while the author was an intern at Microsoft Research New-England, Cambridge, MA.

⋆⋆ Research was conducted in part while the author was visiting Microsoft Research New-England, Cambridge, MA, and supported in part by a scholarship from the Israel Ministry of Science and Technology. The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 257575.

the maximal one, with rate slightly higher than the ones used in their work.

1 Introduction

In this work we present results on the testability of "affine-invariant linear codes". We start with some basic terminology before describing our work in greater detail.

Let $\mathbb{F}_q$ denote the finite field of q elements and $\{\mathbb{F}_q^n \to \mathbb{F}_q\}$ denote the set of functions mapping $\mathbb{F}_q^n$ to $\mathbb{F}_q$. In this work a code (or a family) will be a subset of functions $\mathcal{F} \subseteq \{\mathbb{F}_q^n \to \mathbb{F}_q\}$. We use $\delta(f, g)$ to denote the normalized Hamming distance between f and g, i.e., the fraction of inputs $x \in \mathbb{F}_q^n$ for which $f(x) \neq g(x)$. We use $\delta(\mathcal{F})$ to denote $\min_{f \neq g, f, g \in \mathcal{F}}\{\delta(f, g)\}$ and $\delta_{\mathcal{F}}(f)$ to denote $\min_{g \in \mathcal{F}}\{\delta(f, g)\}$. A code $\mathcal{F}$ is said to be a linear code if it is an $\mathbb{F}_q$-subspace, i.e., for every $\alpha \in \mathbb{F}_q$ and $f, g \in \mathcal{F}$, we have $\alpha f + g \in \mathcal{F}$. A function $T : \mathbb{F}_q^n \to \mathbb{F}_q^n$ is said to be an affine transformation if there exists a matrix $B \in \mathbb{F}_q^{n \times n}$ and vector $c \in \mathbb{F}_q^n$ such that $T(x) = Bx + c$. The code $\mathcal{F} \subseteq \{\mathbb{F}_q^n \to \mathbb{F}_q\}$ is said to be affine-invariant if for every affine transformation T and every $f \in \mathcal{F}$ we have $f \circ T \in \mathcal{F}$ (where $(f \circ T)(x) = f(T(x))$).

Affine-invariant linear codes form a very natural abstraction of the class of low-degree polynomials: The set of polynomials of degree at most d is a linear subspace and is closed under affine transformations. Furthermore, as shown by Kaufman and Sudan [16] affine-invariant linear codes retain some of the "locality" properties of multivariate polynomial codes (or Reed-Muller codes), such as local testability and local decodability, that have found many applications in computational complexity. This has led to a sequence of works exploring these codes, but most of the works led to codes of smaller rate than known ones, or gave alternate understanding of known codes [9, 10, 6, 5, 4]. A recent work by Guo et al. [11] however changes the picture significantly. They study a "lifting" operator on codes and show that it leads to codes with, in some cases dramatic, improvement in parameters compared to Reed-Muller codes. Our work complements theirs by showing that one family of "best-known" tests manages to work abstractly for codes developed by lifting.

We start by describing the lifting operation: Roughly a lifting of a base code leads to a code in more variables whose codewords are words of the base code on every affine subspace of the base dimension. We define this formally next. For $f : \mathbb{F}_q^n \to \mathbb{F}_q$ and $S \subseteq \mathbb{F}_q^n$, let $f|_S$ denote the restriction of f to the set S. A set $A \subseteq \mathbb{F}_q^n$ is said to be a t-dimensional affine subspace, if there exist $\alpha_0, \ldots, \alpha_t \in \mathbb{F}_q^n$ such that $A = \{\alpha_0 + \sum_{i=1}^t \alpha_i x_i | x_1, \ldots, x_t \in \mathbb{F}_q\}$. We use some arbitrary $\mathbb{F}_q$-linear isomorphism from A to $\mathbb{F}_q^t$ to view $f|_A$ as a function from $\{\mathbb{F}_q^t \to \mathbb{F}_q\}$. Given an affine-invariant linear base code $\mathcal{B} \subseteq \{\mathbb{F}_q^t \to \mathbb{F}_q\}$ and integer $n \geq t$, the *n-dimensional lift* of $\mathcal{B}$, denoted $\mathrm{Lift}_n(\mathcal{B})$, is the set $\{f : \mathbb{F}_q^n \to \mathbb{F}_q \mid f|_A \in \mathcal{B}$ for every t-dimensional affine subspace $A \subseteq \mathbb{F}_q^n\}$.

The lifting operation was introduced by Ben-Sasson et al. [5] as a way to build new affine-invariant linear codes that were *not* locally testable. Their codes were also of much lower rate than known affine-invariant linear codes of similar distance. However in more recent work, Guo et al. [11], showed that lifting could be used positively: They used it to build codes with very good locality properties (especially decodability) with rate much better than known affine-invariant linear ones, and matching qualitatively the performance of the best known codes. Our work attempts to complement their work by showing that these codes, over constant sized alphabets, can be "locally tested" as efficiently as polynomial codes.

Testing and Absolutely Sound Testing A code $\mathcal{F} \subseteq \{\mathbb{F}_q^n \to \mathbb{F}_q\}$ is said to be a (k, ϵ, δ)-locally testable code (LTC), if $\delta(\mathcal{F}) \geq \delta$ and there exists a probabilistic oracle algorithm that, on oracle access to $f : \mathbb{F}_q^n \to \mathbb{F}_q$, makes at most k queries to f and accepts $f \in \mathcal{F}$ with probability one, while rejecting $f \notin \mathcal{F}$ with probability at least $\epsilon\delta_{\mathcal{F}}(f)$.

For an ensemble of codes $\{\mathcal{F}_m \subseteq \{\mathbb{F}_q^{n_m} \to \mathbb{F}_q\}\}_m$ for infinitely many m, with $\mathcal{F}_m$ being a $(k(m), \epsilon(m), \delta(m))$-LTC, we say that the code has an *absolutely sound* tester if there exists $\epsilon > 0$ such that $\epsilon(m) \geq \epsilon$ for every m.

Any tester can be converted into an absolutely sound one by repeating the test $1/\epsilon(m)$ times. However this comes with an increase in the query complexity (the parameter $k(m)$) and so it makes sense to ask what is the minimum k one can get for an absolutely sound test.

Previous works by Bhattacharyya et al. [7] and Haramaty et al. [13] raised this question in the context of multivariate polynomial codes (Reed-Muller codes) and showed that the "natural tester" for multivariate polynomial codes is absolutely sound, without any repetitions! The natural test here is derived as follows for prime fields:

> To test if a function f is a polynomial of degree at most d, let t be the smallest integer such that there exist functions of degree greater than d in t variables. Pick a random t-dimensional affine subspace A and verify that $f|_A$ is a degree d polynomial.

The natural test thus makes roughly $q^t = q^{(d+1)/(q-1)}$ queries. This number turns out to be optimal for prime fields in that every function looks like a degree d polynomial if queried at at most q^{t-1} points. Such optimal analyses of low-degree tests turn out to have some uses in computational complexity: In particular one of the many ingredients in the elegant constructions of Barak et al. [3] is the absolutely sound analysis of the polynomial codes over $\mathbb{F}_2$.

Returning to the natural test above, it ends being a little less natural, and not quite optimal when dealing with non-prime fields. Turns out one needs to use a larger value of t than the one in the definition above (specifically, $t = q^{(d+1)/(q-q/p)}$ where p is the characteristic of the field $\mathbb{F}_q$). While it is unclear if sampling all the points in the larger dimensional space is really necessary for absolutely sound testing the results so far seem to suggest working with prime fields is a better option.

1.1 Our work: motivation and results

The motivation for our work is two-fold: Our first motivation is to understand "low-degree testing" better. Low-degree testing has played a fundamental role in computational complexity and yet its proofs are barely understood. They tend to involve a mix of probabilistic, algebraic, and geometric arguments, and the only setting where the mix of these features seems applicable seems to be the setting of low-degree polynomials. Affine-invariant codes naturally seperate the geometry of subspaces in high-dimensional spaces, from the algebra of polynomials of low-degree. Thus extending a proof or analysis method from the setting of low-degree polynomials to the setting of generic geometric arguments has the nice feature that it has the potential to separate the geometric arguments from the algebraic ones.

Within the theme of low-degree testing, the previous works have revealed interesting analyses. And several of these variations in the resulting theorems have played a role in construction of efficient PCPs or more recently in other searches for explicit objects. In particular the literature includes tests such as those originally given by Blum, Luby and Rubinfeld [8] for testing linearity and followed by [20, 1, 15, 14] for testing higher degree polynomials. The aspects of this family of tests are well abstracted in Kaufman and Sudan [16]. But the literature contains other very interesting theorems, such as those of Raz and Safra [18] and Arora and Sudan [2] which tend to work in the "list-decoding" regime. The analysis of the former in particular seems especially amenable to a "generic proof" in the affine-invariant setting and yet such a proof is not yet available. Our work explores a third such paradigm in the analysis of low-degree tests, which was introduced in the above-mentioned "absolutely-sound testers" of Bhattacharyya et al. and Haramaty et al.

Our work starts by noticing that the natural tests above are really "lifting tests": Namely, the test could be applied to any code that is defined as the lift of a base code with the test checking if a given function is a codeword of the base code when restricted to a random small dimensional affine subspace of the base dimension. Indeed this is the natural way of interpreting almost all the previous results in low-degree testing (with the exception of that of [19]). If so, it is natural to ask if the analysis can be carried out to show the absolute soundness of such tests.

The second, more concrete, motivation for our work is the work of Guo et al. [11]. Over prime fields, it was well-known that lifts of low-degree polynomials lead only to polynomials of the same degree (in more variables). Guo et al. show that lifting over non-prime fields leads to better codes than over prime fields! (Prior to their work, it seemed that working with non-prime fields was worse than working with prime fields.) The improved rate gives motivation to study lifted codes in general, and in particular one class of results that would have been nice to extend was the absolutely-sound tester of [13].

In this work we show that the natural test of lifted codes is indeed absolutely sound. The following theorem spells this statement out precisely.

Theorem 1 (Main). *For every prime power q, there exists $\epsilon_q > 0$ such that the following holds: Let $t \leq n$ be positive integers and let $\mathcal{B} \subsetneq \{\mathbb{F}_q^t \to \mathbb{F}_q\}$ be any affine-invariant linear code. Then $\mathcal{F} = \mathrm{Lift}_n(\mathcal{B})$ is $(q^t, \epsilon_q, q^{-t})$-locally testable.*

We stress that the importance of the above is in the absolute soundness, i.e., the fact that ϵ_q does not depend on t or $\mathcal{B}$. If one is willing to let ϵ_q depend on t and $\mathcal{B}$ then such a result follows from the main theorem of [16].

Our result also sets into proper light the previous work of Haramaty et al. [13] who show that the "natural test" for degree d polynomials over the field $\mathbb{F}_q$ of characteristic p makes $q^{(d+1)/(q-q/p)}$ queries and is absolutely sound. Our result does not mention any dependence on p, the characteristic of the field. It turns out that such a dependence comes due to the following proposition.

Let $\mathrm{RM}(n, d, q)$ denote the set of polynomials over $\mathbb{F}_q$ of degree at most d in n variables.

Proposition 1. *For positive integers d and q where q is a power of a prime p, let $t = t_{d,q} = \lceil \frac{d+1}{q-q/p} \rceil$. Then for every $n \geq t$, the Reed-Muller code $\mathrm{RM}(n, d, q)$ equals the code $\mathrm{Lift}_n(\mathrm{RM}(t, d, q))$.*

Applying Theorem 1 to $\mathrm{RM}(n, d, q)$ we immediately obtain the main results of [7] and [13]. And the somewhat cumbersome dependence on the characteristic of q can be blamed on the proposition above, rather than any weakness of the testing analysis. Furthermore, as is exploited by Guo et al. [11] if one interprets the proposition above correctly, then one should use lifts of Reed-Muller codes over non-prime fields with dimension being smaller than $t_{d,q}$. These will yield codes of higher rate while our main theorem guarantees that testability does not suffer.

One concrete consequence of our result is in the use of Reed-Muller codes in the work of Barak et al. [3]. They show how to construct small-set expander graphs with many large eigenvalues and one of the ingredients in their result is a tester of Reed-Muller codes over $\mathbb{F}_2$ (codes obtained by lifting an appropriate family of base codes over $\mathbb{F}_2$). Till this work, the binary Reed-Muller code seemed to be the only code with performance good enough to derive their result. Our work shows that using codes over $\mathbb{F}_4$ or $\mathbb{F}_8$ (or any constant power of two) would serve their purpose at least as well, and even give slight (though really negligible) improvements. We elaborate on these codes and their exact parameters in Section 3. (In particular, see Theorem 3.)

Finally, unlike the works of Bhattacharyya et al., and Haramaty et al., we can not claim that our testers are "optimal". This is not because of a weakness in our analysis, rather it is due to the generality of our theorem. For some codes, including the codes considered in the previous works, our theorem is obviously optimal (being the same test and more or less same analysis as previously). Other codes however may possess special properties making them testable much better. In such cases we can not rule out better tests, though we hope our techniques will still be of some use in analyzing tests for such codes.

Future research directions As noted earlier, the field of low-degree testing has seen several different themes in the analyses. Combined with the work of Kaufman and Sudan [17] our work points to the possibility that much of that study can be explained in terms of the geometry of affine-invariance, and the role of algebra can be encapsulated away nicely. One family of low-degree tests that would be very nice to include in this general view would be that of Raz and Safra [18]. Their work presents a very general proof technique that uses really little algebra; and seems ideally amenable to extend to the affine-invariant setting. We hope that future work will address this.

We also hope that future work improve the dependence of ϵ_q on q in Theorem 1 (which is unfortunately outrageous). Indeed it is not clear why there should be any dependence at all and it would be nice to eliminate it if possible.

Organization We give an overview of the proof of Theorem 1 in Section 2, where we also introduce the main technical theorem of this paper (Theorem 2). We also describe our technical contributions in this section, contrasting the current proof with those of [7, 13], which we modify. In Section 3 we give examples of family of lifted codes for which our main theorem applies. Some of the details are omitted from this version due to space considerations. A full version of this paper is available as [12].

2 Overview of Proof

2.1 Some natural tests

Our proof of Theorem 1 follows the paradigm used in [7] and [13]. Both works consider a natural family of tests (and not just the "most" natural test), and analyze their performance by studying the behavior of functions when restricted to "hyperplanes". We introduce the family of tests first.

From now onwards all codes we consider will be linear and affine-invariant unless we explicitly say otherwise. Given a base code $\mathcal{B} \subseteq \{\mathbb{F}_q^t \to \mathbb{F}_q\}$ and $n \geq \ell \geq t$, we let $\mathcal{L}_\ell = \mathrm{Lift}_\ell(\mathcal{B})$, with $\mathcal{F} = \mathcal{L}_n$. The *$\ell$-dimensional test* for membership in $\mathcal{F}$ works as follows: Pick a random ℓ-dimensional affine subspace A in $\mathbb{F}_q^n$ and accept f if and only if $f|_A \in \mathcal{L}_\ell$.

Let $\mathrm{Rej}_\ell(f)$ denote the probability with which the ℓ-dimensional test rejects. Our main theorem aims to show that $\mathrm{Rej}_\ell(f) = \Omega(\delta_{\mathcal{F}}(f))$ when $\ell = t$. As in previous works, our analysis will first lower bound $\mathrm{Rej}_\ell(f)$ for $\ell = t + O(1)$ and then relate the performance of this test to the performance of the t-dimensional test.

2.2 Overview of proof of Main Theorem 1

The analysis of the performance of the ℓ-dimensional tests is by induction on the number of variables n and based on the behaviour of functions when restricted to "hyperplanes". A *hyperplane* in $\mathbb{F}_q^n$ is an affine subspace of dimension $n - 1$.

In many future calculations it will be useful to know the number of hyperplanes in $\mathbb{F}_q^n$. We note that this number is $q^n + q^{n-1} + \cdots + 1 = q^n(1 + o(1))$.

The inductive strategy to analyzing $\mathrm{Rej}_\ell(f)$ is based on the observation that $\mathrm{Rej}_\ell(f) = \mathbb{E}_H[\mathrm{Rej}_\ell(f|_H)]$ where H is a uniform hyperplane. If we know that on most hyperplanes $\delta_{\mathcal{L}_{n-1}}(f|_H)$ is large, then we can prove the right hand side above is large by induction. Thus the inductive strategy relies crucially on showing that if f is far from $\mathcal{F}$, then $f|_H$ can not be too close to $\mathcal{L}_{n-1}$ on too many hyperplanes. We state this technical result in the contrapositive form below.

Theorem 2 (Main technical). *For every q there exists $\tau < \infty$ such that the following holds: Let $\mathcal{B} \subseteq \{\mathbb{F}_q^t \to \mathbb{F}_q\}$ be an affine-invariant linear code and for $\ell \geq t$ let $\mathcal{L}_\ell = \mathrm{Lift}_\ell(\mathcal{B})$. For $n > t$, let $f : \mathbb{F}_q^n \to \mathbb{F}_q$ be a function and $H_1, \ldots, H_k$ be hyperplanes in $\mathbb{F}_q^n$ such that $\delta_{\mathcal{L}_{n-1}}(f|_{H_i}) \leq \delta$ for every $i \in [k]$ for $\delta < \frac{1}{2}q^{-(t+1)}$. Then, if $k \geq q^{t+\tau}$, we have $\delta_{\mathcal{L}_n}(f) \leq 2\delta + 4(q-1)/k$.*

The theorem thus states that if f is sufficiently close to a lift of $\mathcal{B}$ on a sufficiently large number of hyperplanes, yet a very small number (independent of n) of hyperplanes, then f is close to a lift of $\mathcal{B}$. The dependence of the number of hyperplanes on q and t is actually important to our (and previous) analysis. The fact that it is some fixed multiple of q^t, where the multiple depends only on q and not on t, is crucial to the resulting performance.

Going from Theorem 2 above to Theorem 1 is relatively straightforward. In particular using Theorem 2 we can get a lower bound on $\mathrm{Rej}_{t+\tau}(f)$ without any changes to the proof of [13]. However going from such an analysis to a lower bound on $\mathrm{Rej}_t(f)$ involves some extra work, with complications similar to (but simpler than), those in the proof of Theorem 2 so we omit a discussion here.

The main contribution of this paper is the proof of Theorem 2. Here, the previous proofs, both in [7] and [13] crucially relied on properties of polynomials and in particular the first step in both proofs, when testing degree d polynomials, is to consider the case of f being a degree $d+1$ (or a degree $d+q$) polynomial. In our case there is no obvious candidate for the notion of a degree $d+1$ polynomial and it is abstracting such properties that forms the bulk of our work. In what follows we give an overview of some of the issues arising in such steps and how we deal with them.

2.3 Overview of proof of Theorem 2

To understand our proof of Theorem 2 we need to give some background, specifically to the proofs from the previous work of [13]. Recall the analogous statement in [13] attempted to show that if f was far from being a polynomial of degree d, then the number of hyperplanes where f turns out to be close to being a degree d polynomial is at most $O(q^t)$ (where $t \approx d/q$, the exact number will not be important to us). [13] reasoned about this in a sequence of steps: (1) They first showed that any function of degree greater than d, stays of degree greater than d on at least $1/q$ fraction of all hyperplanes (provided $n > t$). (2) Next they reasoned about functions of degree $d + 1$ and showed that such a function reduces its degree on at most $O(q^t)$ hyperplanes. (3) In the third step they consider a general function f that is *far* from being of degree d and show that the number of hyperplanes on which f becomes a degree d polynomial *exactly* is $O(q^t)$. (This is the step where the big-Oh becomes a really big-Oh.) (4) Finally, they show that for functions of the type considered in the previous step the number of hyperplanes where they even get *close* to being of degree d is at most $O(q^t)$, thus yielding the analog of Theorem 2.

In implementing the program above (which is what we will end up doing) in our more general/abstract setting, our first bottleneck is that, for instance in Step (2) above, we don't have a notion of degree $d+1$ or some notion of functions that are "just outside our good set $\mathcal{F}$". Natural notions of things outside our set do exist, but they don't necessarily satisfy our needs. To understand this issue better, let us see why polynomials of degree $d + O(1)$ appear in the analysis of a theorem such as Theorem 2. Consider a simple case where $H_1, \ldots, H_q$ are parallel hyperplanes completely covering $\mathbb{F}_q^n$ and $\delta = 0$ so f is known to be a good function (member of $\mathcal{F}$, or degree d) when restricted to these hyperplanes. So, in the setting of testing polynomials of degree at most d, the hypothesis asserts that f restricted to these hyperplanes is a polynomial of degree at most d. For notational simplicity we assume that H_i is the hyperplane given by $x_1 = \eta_i$ where $\mathbb{F}_q = \{\eta_1, \ldots, \eta_q\}$. Then $f|_{H_i} = P_i(x_2, \ldots, x_n)$ for some polynomial P_i of degree d. By polynomial interpolation, it follows that f can be described as a degree $d+q-1$ polynomial in $x_1, \ldots, x_n$. The bulk of the analysis in [7, 13] now attempts to use the remaining $K - q$ hyperplanes on which f reduces to degree at most d, in conjunction with the fact that f is a polynomial of degree at most $d + q - 1$ to argue that f is of degree at most d.

For us, the main challenge is that in the generic setting of the lift of some code $\mathcal{B}$, we don't have a ready notion of a degree $d + q - 1$ polynomial and so we have to define one. Thus the first step in this work is to define such a code. For our current discussion it suffices to say that there is an affine-invariant linear code, which we denote $\mathcal{F}^+$, which contains all "interpolating functions" of elements of $\mathcal{F}$ (so $\mathcal{F}^+$ contains every function f for which there exist some q parallel hyperplanes $H_1, \ldots, H_q$ such that $f|_{H_i}$ is a function in $\mathcal{L}_{n-1}$ for all i). Of course such a set is not useful if it does not have some nice structure. The key property of our definition of $\mathcal{F}^+$ is that it is the lift of a non-trivial code on at most $t+q-1$ dimensions. This definition of $\mathcal{F}^+$ and its analysis rely centrally on some of the structural understanding of affine-invariant linear codes derived in previous works [16, 9, 10, 6, 5, 4]. Our analysis shows that $\mathcal{F}^+$ is almost as nice as $\mathcal{F}$, roughly analogous to the way the set of degree $d + q - 1$ polynomials is almost as nice as the set of degree d polynomials.

The notion of $\mathcal{F}^+$ turns out to be easy enough to use to be able to carry out the steps (3) and (4) in the program above by directly mimicking the proofs of [13], assuming Steps (1) and (2) hold. But Steps (1) and (2) turn out to be more tricky. So we turn to these, and in particular Step (2) next.

Our next barrier in extending the proofs of [13] is a notion of "canonical monomials" which play a crucial role in Step (2) of [13]. For a function of degree

$d+1$, the canonical monomial is a monomial of degree $d+1$ supported on very few variables. The fact that the number of variables in the support is small, while the monomial remains a "forbidden one" turns out to be central to their analysis and allows them to convert questions of the form: "Does f become a polynomial of smaller degree on the hyperplane H?"(which are typically not well-understood) to questions of the form "Does g become the zero polynomial when restricted to H?" (which is a very well-studied question).

In our case, we need to work with some function f in $\mathcal{F}^+$ which is not a function of $\mathcal{F}$. The fact that $\mathcal{F}^+$ is a lift of "few-dimensional" code, in principle ought to help us find a monomial supported on few variables that is not in $\mathcal{F}$. But isolating the "right one" to work with for f turns out to be a subtle issue and we work hard, and come up with a definition that is very specific to each function $f \in \mathcal{F}^+ \setminus \mathcal{F}$. (In contrast the canonical monomials of [13] were of similar structure for every function f.) Armed with this definition and some careful analysis we are able to simulate Step (2) in the program above. We give a few more details into this step below. Full details may be found in the full version of this paper.

Let $t^+ = t + q - 1$, and let $\mathcal{B}^+$ be a family on t^+ variables such that $\mathcal{F}^+$ is a lift of $\mathcal{B}^+$. Let $f \in \mathcal{F}^+ \setminus \mathcal{F}$. We first show that for every such f there exists an invertible affine transformation T and monomial $M \notin \mathcal{F}$ supported on the first t^+ variables such that $f \circ T$ is supported on M. We further assume that T is such that the degree of M is maximal. Without loss of generality we may assume T is just the identity transformation and so f is supported on M. Next we partition the space of all possible hyperplanes into $q^{t^+ + 1}$ sets (based on their coefficients on the first t^+ variables). Our goal is to show that in each set in the partition there are at most some constant (depending on q) number of hyperplanes such that f restricted to that hyperplane becomes a member of $\mathcal{F}$. To do so we extract from f a non-zero low-degree function g. (this function g depends on M and the set in the partition under consideration). We show that for the correct definition of g, it is the case that $f|_H \in \mathcal{F}$ only if $g|_H \equiv 0$. This brings us to the final task: to bound the number of hyperplanes on which $g|_H$ can be identically zero. For this part we show a simple lemma (see Lemma 4.8 in the full version) that shows that a low-degree function can only be zero on a small number of hyperplanes (bounded by a function of q and the degree, but independent of n). Putting the above ingredients together gives us a bound (of desired quality) on the number of hyperplanes H for which $f|_H \in \mathcal{F}$.

Finally, Step (1) is also dealt with similarly, using some of the same style of ideas as in the proof of Step (2).

3 New testable codes

In this section, we give some examples of codes with "nice" parameters that are testable with absolute soundness based on our main theorem (Theorem 1).

The need for such codes is motivated by the work of Barak et al. [3]. Their work used appropriate Reed-Muller codes over $\mathbb{F}_2$. Our work gives the second family of codes that is known to satisfy their requirements. We point out that Guo et al. [11] also give codes motivated by the work of [3], but their codes are not, thus far, known to be testable with absolute soundness and so fail to meet all the requirements of [3]. Our codes fall within the class of "lifted" codes studied by [11], but were not analyzed there. Here we use analysis similar to their to analyze the rate and distance of our codes, while the testing follows from our main theorem.

The code. Our codes are defined by three parameters: a real number $\epsilon > 0$ and two integers s and n. The code $\mathcal{F} = \mathcal{F}_{\epsilon,s,n}$ is obtained as follows: Let $q = 2^s$, and let $\ell = \lfloor \frac{1}{s} \log 1/\epsilon \rfloor$. Let $\mathcal{B} = \{f : \mathbb{F}_q^{n-\ell} \to \mathbb{F}_2 | \sum_{\boldsymbol{x} \in \mathbb{F}_q^{n-\ell}} f(\boldsymbol{x}) = 0\}$. Let $\mathcal{F} = \mathrm{Lift}_n(\mathcal{B})$.

Basic parameters:

Proposition 2. *For every ϵ, s and n the code $\mathcal{F} = \mathcal{F}_{\epsilon,s,n}$ has block length $N = 2^{sn}$, (absolute, non-normalized) distance at least $1/\epsilon$ and dimension at least $2^{sn} - \left(\binom{n}{\ell}^s + \sum_{i=0}^{s\ell-1} \binom{ns}{i}\right)$.*

Proof. The size of the block length can be easily verified and the distance follows from the general properties of lifting (see full version for details. Lemmas 3.11. and 3.12. in Guo et. al. [11] analyzed the dimension of the code $\mathcal{F}_{\epsilon,s,n}$ for the case in which $s = \log(1/\epsilon)$ (so $\ell = 1$). More specifically, given a degree pattern $a = (a_1, \ldots, a_n)$ with $\{a_i\}_{i=1}^n \subseteq \mathbb{Z}_q$, let $a_i^{(j)}$ denote the j-th bit of the binary expansion of a_i. Let $M(a)$ denote the $n \times s$ matrix with entries $M(a)_{i,j} = a_i^{(j)}$. Guo et. al. show that in the special case in which $\ell = 1$ the code $\mathcal{F}_{\epsilon,s,n}$ contains in its support all monomials with degree pattern $a = (a_1, \ldots, a_n)$ such that there exists a column in $M(a)$ with at least two zeroes. This readily implies a bound of $2^{sn} - (n+1)^\ell$ on the dimension of their code.

A similar analysis shows that our code $\mathcal{F}_{\epsilon,s,n}$ contains all monomials with degree pattern $a = (a_1, \ldots, a_n)$ where the matrix $M(a)$ has at least $s\ell + 1$ zeroes, or the matrix has $s\ell$ zeroes and there exists a column in $M(a)$ with at least $\ell + 1$ zeros. The lower bound on the dimension follows.

Testability. The following is an immediate application of Theorem 1.

Proposition 3. *For every s there exists a constant $\tau > 0$ such that for every ϵ and n the code $\mathcal{F} = \mathcal{F}_{\epsilon,s,n}$ is testable by a test that makes ϵN queries, accepts codewords with probability one, while rejecting all functions $f : \mathbb{F}_q^n \to \mathbb{F}_2$ with probability at least $\tau \cdot \delta(f, \mathcal{F})$.*

We remark that the dimension of our codes, for any choice of N and ϵ is strictly better than that of the codes used in [3] which have dimension $2^{sn} - \sum_{i=0}^{s\ell} \binom{sn}{i} \approx 2^{sn} - \frac{1}{\sqrt{2\pi s\ell}}(en/\ell)^{s\ell}$. An important parameter for them is the "co-dimension" of their code (block length minus the dimension, or the dimension of the dual code), which thus turns out to be roughly $\frac{1}{\sqrt{2\pi s\ell}}(en/\ell)^{s\ell}$ from the above expression. (A smaller codimension is better for their application.) Simplifying the dimension of our code from Proposition 2, we see that the codimension of our code is smaller by a multiplicative factor of roughly $O(\ell^{s/2-1})$, making our codes noticeably better. Unfortunately such changes do not alter the essential relationship between $N = 2^{sn}$, the parameter ϵ (which determines the locality of the tester) and the codimension of the code. The following theorem summarizes the performance of our codes.

Theorem 3. *For every positive s there exists a constant τ such that for every sufficiently small ϵ and sufficiently large N there exists a code of block length N, codimension $\left(\log\frac{1}{\epsilon}\right)^{-s}\cdot\left(\frac{e\log N}{\log\frac{1}{\epsilon}}\right)^{\log\frac{1}{\epsilon}}$ that is testable with a tester that makes $\epsilon\cdot N$ queries accepting codewords with probability one, while rejecting words at distance δ with probability at least $\tau\cdot\delta$.*

To contrast, the corresponding result in [3] would assert the existence of a positive constant s for which the above held.

References

1. Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron. Testing Reed-Muller codes. *IEEE Transactions on Information Theory*, 51(11):4032–4039, 2005.
2. Sanjeev Arora and Madhu Sudan. Improved low degree testing and its applications. *Combinatorica*, 23(3):365–426, 2003.
3. Boaz Barak, Parikshit Gopalan, Johan Håstad, Raghu Meka, Prasad Raghavendra, and David Steurer. Making the long code shorter, with applications to the unique games conjecture. In *FOCS*. IEEE Computer Society, 2012.
4. Eli Ben-Sasson, Elena Grigorescu, Ghid Maatouk, Amir Shpilka, and Madhu Sudan. On sums of locally testable affine invariant properties. In Leslie Ann Goldberg, Klaus Jansen, R. Ravi, and José D. P. Rolim, editors, *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques, volume 6845 of LNCS*, volume 6845 of *Lecture Notes in Computer Science*, pages 400–411. IEEE Computer Society, 2011.
5. Eli Ben-Sasson, Ghid Maatouk, Amir Shpilka, and Madhu Sudan. Symmetric LDPC codes are not necessarily locally testable. In *IEEE Conference on Computational Complexity*, pages 55–65. IEEE Computer Society, 2011.
6. Eli Ben-Sasson and Madhu Sudan. Limits on the rate of locally testable affine-invariant codes. In *APPROX-RANDOM*, volume 6845 of *Lecture Notes in Computer Science*, pages 412–423. Springer, 2011.
7. Arnab Bhattacharyya, Swastik Kopparty, Grant Schoenebeck, Madhu Sudan, and David Zuckerman. Optimal testing of reed-muller codes. In *FOCS*, pages 488–497. IEEE Computer Society, 2010.
8. Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. In *STOC*, pages 73–83. ACM, 1990.
9. Elena Grigorescu, Tali Kaufman, and Madhu Sudan. 2-transitivity is insufficient for local testability. In *IEEE Conference on Computational Complexity*, pages 259–267. IEEE Computer Society, 2008.

10. Elena Grigorescu, Tali Kaufman, and Madhu Sudan. Succinct representation of codes with applications to testing. In *APPROX-RANDOM*, volume 5687 of *Lecture Notes in Computer Science*, pages 534–547. Springer, 2009.
11. Alan Guo, Swastik Kopparty, and Madhu Sudan. New affine-invariant codes from lifting. *Proceedings of ITCS 2013*, (to appear), 2013.
12. Elad Haramaty, Noga Ron-Zewi, and Madhu Sudan. Absolutely sound testing of lifted codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 20:30, 2013.
13. Elad Haramaty, Amir Shpilka, and Madhu Sudan. Optimal testing of multivariate polynomials over small prime fields. In *FOCS*, pages 629–637. IEEE Computer Society, 2011.
14. Charanjit S. Jutla, Anindya C. Patthak, Atri Rudra, and David Zuckerman. Testing low-degree polynomials over prime fields. *Random Struct. Algorithms*, 35(2):163–193, 2009.
15. Tali Kaufman and Dana Ron. Testing polynomials over general fields. *SIAM Journal of Computing*, 36(3):779–802, 2006.
16. Tali Kaufman and Madhu Sudan. Algebraic property testing: The role of invariance. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(111), 2007.
17. Tali Kaufman and Madhu Sudan. Algebraic property testing: the role of invariance. In *STOC*, pages 403–412. ACM, 2008.
18. Ran Raz and Shmuel Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability pcp characterization of np. In *STOC*, pages 475–484. ACM, 1997.
19. Noga Ron-Zewi and Madhu Sudan. A new upper bound on the query complexity for testing generalized reed-muller codes. In *APPROX-RANDOM*, volume 7408 of *Lecture Notes in Computer Science*, pages 639–650. Springer, 2012.
20. Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. on Computing*, 25(2):252–271, 1996.