Testing Low-Degree Polynomials over Prime Fields*

Charanjit S. Jutla
IBM Thomas J. Watson Research Center,
Yorktown Heights, NY 10598
csjutla@watson.ibm.com

Anindya C. Patthak†
University of California, Riverside
Riverside, CA 92521
apatthak@ee.ucr.edu

Atri Rudra‡
Dept. of Computer Science & Engineering
University at Buffalo,
The State University of New York
Buffalo, NY 14260
atri@cse.buffalo.edu

David Zuckerman§
Department of Computer Science
1 University Station C0500
University of Texas at Austin
Austin, TX 78712
diz@cs.utexas.edu

June 27, 2008

Abstract

We present an efficient randomized algorithm to test if a given function $f : \mathbb{F}_p^n \to \mathbb{F}_p$ (where p is a prime) is a low-degree polynomial. This gives a local test for Generalized Reed-Muller codes over prime fields. For a given integer t and a given real $\epsilon > 0$, the algorithm queries f at $O\left(\frac{1}{\epsilon} + t \cdot p^{\frac{2t}{p-1}+1}\right)$ points to determine whether f can be described by a polynomial of degree at most t. If f is indeed a polynomial of degree at most t, our algorithm always accepts, and if f has a relative distance at least ϵ from every degree t polynomial, then our algorithm rejects f with probability at least $\frac{1}{2}$. Our result is almost optimal since any such algorithm must query f on at least $\Omega(\frac{1}{\epsilon} + p^{\frac{t+1}{p-1}})$ points.

keywords : Polynomials, Generalized Reed-Muller code, local testing, local correction.

*A preliminary version of this paper appeared in 45th. *Symposium on Foundations of Computer Science, 2004.*

†Most of this work was done while the author was at the University of Texas at Austin. Supported in part by NSF Grant CCR-0310960 and NSF grant CCF-0635339.

‡This work was done while the author was at the University of Texas at Austin.

§Supported in part by NSF Grants CCR-9912428, CCR-0310960, and CCF-0634811 and a David and Lucile Packard Fellowship for Science and Engineering.

1 Introduction

1.1 Background and Context

A *low degree tester* is a probabilistic algorithm which, given a degree parameter t and oracle access to a function f on n arguments (which take values from some finite field $\mathbb{F}$), has the following behavior. If f is the evaluation of a polynomial on n variables with total degree at most t, then the low degree tester must accept with probability one. On the other hand, if f is "far" from being the evaluation of some polynomial on n variables with degree at most t, then the tester must reject with constant probability. The tester can query the function f to obtain the evaluation of f at any point. However, the tester must accomplish its task by using as few probes as possible.

Low degree testers play an important part in the construction of Probabilistically Checkable Proofs (or PCPs). In fact, different parameters of low degree testers (for example, the number of probes and the amount of randomness used) directly affect the parameters of the corresponding PCPs as well as various inapproximability results obtained from such PCPs (starting with the work of Fiege, Goldwasser, Lovasz, Safra and Szegedy [FGL$^+$96] and Arora, Lund, Motwani, Sudan and Szegedy [ALM$^+$98]). Low degree testers also form the core of Babai, Fortnow and Lund's proof of MIP = NEXPTIME in [BFL91].

Blum, Luby, and Rubinfeld designed the first low degree tester, which handled the linear case, i.e., $t = 1$ ([BLR93]). This was followed by a series of works that gave low degree testers that worked for larger values of the degree parameter (e.g., Rubinfeld and Sudan[RS96], Friedl and Sudan [FS95], Arora and Sudan [AS03]). However, these subsequent results as well as others which use low degree testers (such as Gemmell, Lipton, Rubinfeld, Sudan and Wigderson [GLR$^+$91] and [BFL91]) crucially require that the size of the underlying field $\mathbb{F}$ be larger than the degree being tested. One exception is the work of Alon, Kaufman, Krivelevich, Litsyn and Ron which gave a low degree tester for any nontrivial degree parameter over the binary field $\mathbb{F}_2$ [AKK$^+$05].

A natural open problem was to give a low degree tester for all degrees, with the underlying finite fields of size between two and the degree parameter. In this work we (partially) solve this problem by presenting a low degree test for multivariate polynomials over any prime field $\mathbb{F}_p$.

1.1.1 Connection to coding theory

A linear code C over a finite field $\mathbb{F}$ of *dimension* K and *length* N is a K-dimensional subspace of $\mathbb{F}^N$. The code C is said to be *locally testable* if there exists a *tester* that can efficiently distinguish oracles that represent codewords of C from oracles that differ from every *codeword* in C in a "large" fraction of positions.

The evaluations of polynomials in n variables of degree at most t are well known linear codes. In particular, the evaluation of polynomials in n variables of degree at most t over $\mathbb{F}_2$ is the `Reed-Muller code` (or $\mathcal{R}(t, n)$) with parameters t and n. The corresponding code over general fields, called `Generalized Reed-Muller code` (or $\mathrm{GRM}_q(n, t)$) is the vector of (evaluations of) all polynomials in n variables of total degree at most t over $\mathbb{F}_q$. These codes have length q^n and dimension $\binom{n+t}{n}$ (see [DGM70, DK00, AJK98] for more details). Therefore, a function has degree t if and only if (the vector of evaluations of) the function is a valid codeword in $\mathrm{GRM}_q(n, t)$. In other words, low degree testing is equivalent to locally testing Generalized Reed-Muller codes.

1.2 Previous low degree testers

As was mentioned earlier, the study of low degree testing (along with *self-correction*) dates back to the work of Blum, Luby and Rubinfeld ([BLR93]), where an algorithm was required to test whether a given function is linear. The approach in [BLR93] later naturally extended to yield testers for low degree polynomials (but over fields larger than the total degree). Roughly, the idea is to project the given function on to a random line and then test if the projected univariate polynomial has low degree. Specifically, for a purported degree t function $f : \mathbb{F}_q^n \to \mathbb{F}_q$, the test works as follows. Pick vectors y and b from $\mathbb{F}_q^n$ (uniformly at random), and distinct $s_1, \cdots, s_{t+1}$ from $\mathbb{F}_q$ arbitrarily. Query the oracle representing f at the $t+1$ points $b + s_i y$ and extrapolate to a degree t polynomial $P_{b,y}$ in one variable s. Now test for a random $s \in \mathbb{F}_p$ if

$$P_{b,y}(s) = f(b + sy)$$

(for details see [RS96],[FS95]). Similar ideas are also employed to test whether a given function is a low degree polynomial in each of its variables (see [FGL$^+$96, BFLS91, AS98]). Note that this approach does not work when the field size is smaller than the total degree, as $x^q = x$ in $\mathbb{F}_q$.

Alon et al. give a tester over field $\mathbb{F}_2$ for any degree up to the number of inputs to the function (i.e., for any non-trivial degree) [AKK$^+$05]. In other words, their work shows that Reed-Muller codes are locally testable. Under the coding theory interpretation, their tester picks a random codeword u from the dual code and checks if it is orthogonal to the input vector. Since the query complexity depends on the weight of the dual codeword u, u is chosen randomly from a set of minimum-weight codewords that happen to span the dual code.

Specifically their test works as follows: given a function $f : \{0,1\}^n \to \{0,1\}$, to test if the given function f has degree at most t, pick $(t+1)$-vectors $y_1, \cdots, y_{t+1} \in \{0,1\}^n$ and test if

$$\sum_{\emptyset \neq S \subseteq [t+1]} f(\sum_{i \in S} y_i) = 0.$$

As we show later, the test in [RS96] above can also be given this coding theory interpretation.

1.3 Our Result

It is easier to define our tester over $\mathbb{F}_3$. To test if f has degree at most t, set $k = \lceil \frac{t+1}{2} \rceil$, and let $i = (t+1) \mod 2$. Pick k-vectors $y_1, \cdots, y_k$ and b from $\mathbb{F}_3^n$, and test if

$$\sum_{c \in \mathbb{F}_3^k; c=(c_1,\cdots,c_k)} c_1^i f(b + \sum_{j=1}^{k} c_j y_j) = 0,$$

where for notational convenience we use $0^0 = 1$. We prove that a polynomial of degree at most t always passes the test, whereas a polynomial of degree greater than t gets caught with non-negligible probability α. To obtain a constant rejection probability we repeat the test $\Theta(1/\alpha)$ times.

As in [RS96] there are two main parts to the proof. The first step is coming up with an *exact characterization* for functions that have low degree. Following [AKK$^+$05], it is best to view low degree polynomials over $\mathbb{F}_q$ as the Generalized Reed-Muller (GRM) code. As GRM is a linear code, a function is of low degree if and only if it is orthogonal to every codeword in the dual of the corresponding GRM code. The second step of the proof entails showing that the characterization is

a *robust characterization*, that is, the following natural tester is indeed a local tester (see section 2 for a formal definition). Pick a codeword uniformly at random from a set of low-weight codewords that span the dual code and check if it is orthogonal to the given function.

Apart from the obvious difficulty of proving step two, the proof is further complicated by the fact that to obtain a good tester (i.e. one which makes as few queries as possible), we need a sub-collection of the dual GRM code in which each vector has low weight such that it generates the dual code.

Since it is well known that the dual of a GRM code is a GRM code (with different parameters), to obtain a collection of codewords (with low weight) that generate the dual of a GRM code it is enough to do so for the GRM code itself. We present an alternative basis of GRM codes over prime fields that in general differs from the minimum weight basis obtained in the work of Delsarte [DGM70, DK00]. Our basis has a clean geometric structure in terms of *flats* (cf. [AJK98]), and unions of parallel flats (but with different weights assigned to different parallel flats)[1]. This equivalence between the polynomial and geometric representations plays a pivotal role in proving step two. Moreover, our basis consists of codewords with weight within a factor p of the minimal weight of the dual code. This makes the query complexity of our tester almost optimal.

1.3.1 Main Result

Our results may be stated quantitatively as follows. For a given integer $t \geq (p-1)$ and a given real $\epsilon > 0$, our testing algorithm queries f at $O\left(\frac{1}{\epsilon} + t \cdot p^{\frac{2t}{p-1}+1}\right)$ points to determine whether f can be described by a polynomial of degree at most t. If f is indeed a polynomial of degree at most t, our algorithm always accepts, and if f has a relative distance at least ϵ from every degree t polynomial, then our algorithm rejects f with probability at least $\frac{1}{2}$. (In the case $t < (p-1)$, our tester still works but more efficient testers are known). It is folklore that the dual distance (minimum distance of the dual code), which is $p^{\frac{t+1}{p-1}}$ in our case, is a lower bound on the query complexity (cf. [BSHR05]). In fact, a straightforward generalization of a result of Alon, Krivelevich, Newman and Szegedy [AKNS99] implies that our result is almost optimal as any such testing algorithm must query f in at least $\Omega(\frac{1}{\epsilon} + p^{\frac{t+1}{p-1}})$ many points.

Our analysis also enables us to obtain a *self-corrector* (as defined in [BLR93]) for f, in case the function f is reasonably close to a degree t polynomial. Specifically, we show that the value of the function f at any given point $x \in \mathbb{F}_p^n$ may be obtained with good probability by querying f on $\Theta(p^{t/p})$ random points. Using the second moment method and majority logic decoding we can achieve even higher probability by querying f on $p^{O(t/p)}$ random points.

1.4 Related Work and Further Developments

Independently of our work, Kaufman and Ron, generalizing a characterization result of [FS95], gave a tester for low degree polynomials over general finite fields (see [KR06]). They show that a given polynomial is of degree at most t *if and only if* the restriction of the polynomial to every affine subspace of suitable dimension is of degree at most t. Given this characterization, their tester chooses a random affine subspace of a suitable dimension, computes the polynomial restricted to

[1]The natural basis given in [DGM70, DK00] assigns the same weight to each parallel flat.

this subspace, and verifies that the coefficients of the higher degree terms are zero[2]. To obtain constant soundness, the test is repeated many times. An advantage of our approach is that in one round of the test (over the prime field) we test only one linear constraint, whereas their approach needs to test multiple linear constraints.

A basis of GRM (over prime fields) consisting of minimum-weight codewords was considered in [DGM70, DK00]. In fact, following the work of Delsarte (see the complete references in[DK00, AJK98]) the geometric structure of the minimal weight codewords over arbitrary finite fields are well understood. We obtain another exact characterization for low-degree polynomials. Furthermore, it seems that their exact characterization can also be turned into a robust characterization following analysis similar to ours, though we have not worked out the details. However, our basis is cleaner and yields a simpler analysis.

We point out that for degree smaller than the field size, the exact characterization obtained from [DGM70, DK00] coincides with [BLR93, RS96, FS95]. This provides an alternate proof to the exact characterization of [FS95] (for more details, see Remark 3.11 later and [FS95]).

Further Developments In an attempt to generalize our result to arbitrary finite fields, we have obtained an exact characterization of low degree polynomials over general finite fields[3][JPR04]. This provides an alternate proof to the result of Kaufman and Ron [KR06] described earlier. Specifically the result says that a given polynomial is of degree at most t *if and only if* the restriction of the polynomial to every affine subspace of dimension $\lceil \frac{t+1}{q-q/p} \rceil$ (and higher) is of degree at most t. (This characterization was first proved by Cohen [Coh87].) We remark that this gives a basis with weight larger than the minimum weight of the code– this is not surprising as [DK00] showed that in general there exists no minimal weight basis of GRM codes over non-prime finite fields.

1.5 Organization of the paper

The rest of the paper is organized as follows. In Section 2 we introduce notation and mention some preliminary facts. Section 3 contains the exact characterization of the low degree polynomials over prime fields. In Section 4 we formally describe the tester and prove its correctness. In Section 5 we sketch a lower bound that implies that the query complexity of our tester is almost optimal, and suggest how to self-correct a function which agrees with a low degree polynomial on most of its input. Section 6 contains some concluding remarks.

2 Preliminaries

2.1 Facts from Finite Fields

In this section we spell out some facts from finite fields which will be used later. We denote the multiplicative group of $\mathbb{F}_q$ by $\mathbb{F}_q^*$. We begin with a simple lemma.

Lemma 2.1 *For any $t \in [q-1]$, $\sum_{a \in \mathbb{F}_q} a^t \neq 0$ if and only if $t = q-1$.*

[2]Since the coefficients can be written as linear sums of the evaluations of the polynomial, this is equivalent to check several linear constraints

[3]Our alternate proof along with other omitted proofs appear in the second author's doctoral thesis [Pat07]. We also remark that the exact characterization can further be extended to a robust characterization using techniques we develop for prime fields.

Proof: First note that $\sum_{a\in\mathbb{F}_q} a^t = \sum_{a\in\mathbb{F}_q^*} a^t$. Observing that for any $a \in \mathbb{F}_q^*$, $a^{q-1} = 1$, it follows that $\sum_{a\in\mathbb{F}_q^*} a^{q-1} = \sum_{a\in\mathbb{F}_q^*} 1 = -1 \neq 0$.

Next we show that for all $t \neq q-1$, $\sum_{a\in\mathbb{F}_q^*} a^t = 0$. Let α be a generator of $\mathbb{F}_q^*$. The sum can be re-written as $\sum_{i=0}^{q-2} \alpha^{it} = \frac{\alpha^{t(q-1)}-1}{\alpha^t-1}$. The denominator is non-zero for $t \neq q-1$ and thus, the fraction is well defined. The proof is complete by noting that $\alpha^{t(q-1)} = 1$. ■

This immediately implies the following lemma.

Lemma 2.2 *Let $t_1, \cdots, t_\ell \in [q-1]$. Then*

$$\sum_{(c_1,\cdots,c_\ell)\in(\mathbb{F}_q)^\ell} c_1^{t_1} c_2^{t_2} \cdots c_\ell^{t_\ell} \neq 0 \text{ if and only if } t_1 = t_2 = \cdots = t_\ell = q-1. \tag{1}$$

Proof: Note that the left hand side can be rewritten as $\prod_{i\in[\ell]} \left(\sum_{c_i\in\mathbb{F}_q} c_i^{t_i}\right)$. ■

We will need to transform products of variables to powers of linear functions in those variables. With this motivation, we present the following identity.

Lemma 2.3 *For each $kin[p-1]$, there exists a $c_k \in \mathbb{F}_p^*$ such that*

$$c_k \prod_{i=1}^{k} x_i = \sum_{i=1}^{k} (-1)^{k-i} S_i \quad \textit{where} \quad S_i = \sum_{\emptyset\neq I\subseteq[k];|I|=i} \left(\sum_{j\in I} x_j\right)^k. \tag{2}$$

Proof: Consider the right hand side of the (2). Note that all the monomials are of degree exactly k. Also note that $\prod_{i=1}^k x_i$ appears only in S_k and nowhere else. Now consider any other monomial of degree k that has a support of size m, where $0 < m < k$. Further note that the coefficient of any such monomial in the expansion of $(\sum_{j\in I} x_j)^k$ is the same and non-zero. Therefore, summing up the number of times it appears (along with the $(-1)^{k-i}$ factor) in each S_i is enough which is just

$$1 - \binom{k-m}{k-m-1} + \binom{k-m}{k-m-2} + \cdots + (-1)^{k-m}\binom{k-m}{k-m-(k-m)} = (1-1)^{k-m} = 0.$$

Moreover, it is clear that $c_k = k! \mod p$ and $c_k \neq 0$ for the choice of k. ■

2.2 Flats and Pseudoflats

For any integer ℓ, we denote the set $\{1, \cdots, \ell\}$ by $[\ell]$. Throughout we use p to denote a prime and $\mathbb{F}_p$ to denote a prime field of size p. We also use $\mathbb{F}_q$ to denote a finite field of size q, where $q = p^s$ for some positive integer s. In this paper, we mostly deal with prime fields. We therefore restrict most definitions to the prime field setting. For a set $S \subseteq \mathbb{F}_p^n$ and $y \in \mathbb{F}_p^n$, we define $y + S \stackrel{\text{def}}{=} \{x + y | x \in S\}$.

For any $t \in [n(q-1)]$, let $\mathcal{P}_t$ denote the family of all functions over $\mathbb{F}_q^n$ which are polynomials of total degree at most t (and individual degree at most $q-1$) in n variables. In particular $f \in \mathcal{P}_t$ if there exists coefficients $a_{(e_1,\cdots,e_n)} \in \mathbb{F}_q$, for every $i \in [n]$, $e_i \in \{0, \cdots, q-1\}$, $\sum_{i=1}^n e_i \leq t$, such that

$$f = \sum_{(e_1,\cdots,e_n)\in\{0,\cdots,q-1\}^n; 0\leq\sum_{i=1}^n e_i\leq t} a_{(e_1,\cdots,e_n)} \prod_{i=1}^{n} x_i^{e_i}. \tag{3}$$

The codeword corresponding to f will be its evaluation vector. We recall the definition of the Generalized (Primitive) Reed-Muller code as described in [AJK98, DK00].

Definition 2.4 *Let $V = \mathbb{F}_q^n$ be the vector space of n-tuples, for $n \geq 1$, over the field $\mathbb{F}_q$. For any k such that $0 \leq k \leq n(q-1)$, the k^{th} order Generalized Reed-Muller code $\mathrm{GRM}_q(k,n)$ is the subspace of $\mathbb{F}_q^{|V|}$ (with the basis as the characteristic functions of vectors in V) of all n-variable polynomial functions (reduced modulo $x_i^q - x_i$) of degree at most k.*

This implies that the code corresponding to the family of functions $\mathcal{P}_t$ is $\mathrm{GRM}_q(t,n)$. Therefore, a characterization for one will simply translate into a characterization for the other.

For any two functions $f, g : \mathbb{F}_q^n \to \mathbb{F}_q$, the relative distance $\delta(f,g) \in [0,1]$ between f and g is defined as $\delta(f,g) \stackrel{def}{=} \Pr_{x \in \mathbb{F}_q^n}[f(x) \neq g(x)]$. For a function g and a family of functions F (defined over the same domain and range), we say g is ϵ-*close* to F, for some $0 < \epsilon < 1$, if, there exists an $f \in F$, where $\delta(f,g) \leq \epsilon$. Otherwise it is ϵ-*far* from F.

A one sided testing algorithm (*one-sided tester*) for $\mathcal{P}_t$ is a probabilistic algorithm that is given query access to a function f and a distance parameter ϵ, $0 < \epsilon < 1$. If $f \in \mathcal{P}_t$, then the tester should always accept f (perfect completeness), and if f is ϵ-far from $\mathcal{P}_t$, then with probability at least $\frac{1}{2}$ the tester should reject f (a two-sided tester may be defined analogously).

For vectors $x, y \in \mathbb{F}_p^n$, the dot (scalar) product of x and y, denoted $x \cdot y$, is defined to be $\sum_{i=1}^n x_i y_i$, where w_i denotes the i^{th} co-ordinate of the vector w.

To motivate the next notation which we will use frequently, we give a definition.

Definition 2.5 *A k-flat ($k \geq 0$) in $\mathbb{F}_p^n$ is a k-dimensional affine subspace. Let $y_1, \cdots, y_k \in \mathbb{F}_p^n$ be linearly independent vectors and $b \in \mathbb{F}_p^n$ be a point. Then the subset $L = \{\sum_{i=1}^k c_i y_i + b | \forall i \in [k]\ c_i \in \mathbb{F}_p\}$ is a k-flat. We will say that L is generated by $y_1, \cdots, y_k$ at b. The incidence vector of the points in a given k-flat will be referred to as the codeword corresponding to the given k-flat.*

We remark that a 0-flat is just a point.

Given a function $f : \mathbb{F}_p^n \to \mathbb{F}_p$, for $y_1, \cdots, y_\ell, b \in \mathbb{F}_p^n$ we define

$$T_f(y_1, \cdots, y_\ell, b) \stackrel{def}{=} \sum_{c=(c_1,\cdots,c_\ell) \in \mathbb{F}_p^\ell} f(b + \sum_{i \in [\ell]} c_i y_i), \tag{4}$$

which is the sum of the evaluations of function f over an l-flat generated by $y_1, \cdots, y_\ell$, at b. Alternatively, this can also be interpreted as the dot product of the codeword corresponding to the ℓ-flat generated by $y_1, \cdots, y_\ell$ at b and that corresponding to the function f (also see Observation 3.5).

While k-flats are well-known, below we define a new geometric object, called a *pseudoflat*. A k-pseudoflat is a union of $(p-1)$ parallel $(k-1)$-flats. Also, k-pseudoflats can have different exponents ranging from 1 to[4] $(p-2)$. We stress that the point set of a k-pseudoflat remains the same irrespective of its exponent. It is the value assigned to the points that changes with the exponent.

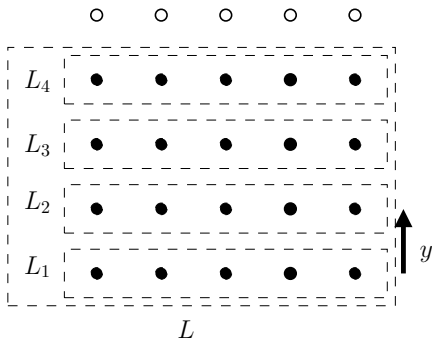
Definition 2.6 *Let $L_1, L_2, \cdots, L_{p-1}$ be parallel $(k-1)$-flats ($k \geq 1$), such that for some $y \in \mathbb{F}_p^n$ and all $t \in [p-2]$, $L_{t+1} = y + L_t$. We define the* `points` *of k-pseudoflat L with exponent r ($1 \leq r \leq p-2$) to be the union of the set of points L_1 to L_{p-1}. Also, let I_j be the incidence vector*

[4]With slight abuse, a k-pseudoflat with exponent zero corresponds to a flat.

of L_j for $j \in [p-1]$. Then the `evaluation vector` *of this k-*`pseudoflat with exponent` *r is defined to be $\sum_{j=1}^{p-1} j^r I_j$. The evaluation vector of a k-pseudoflat with exponent r will be referred to as the codeword corresponding to the given k-pseudoflat with exponent r.*

Let L be a k-pseudoflat with exponent r. Also, for $j \in [p-1]$, let L_j be the $(k-1)$-flat generated by $y_1, \cdots, y_{k-1}$ at $b + j \cdot y$, where $y_1, \cdots, y_{k-1}$ are linearly independent. Then we say that L, a k-pseudoflat with exponent r, is generated by $y, y_1, \cdots, y_{k-1}$ at b exponentiated along y.

See Figure 1 for an illustration of Definition 2.6.



0	0	0	0	0
4	4	4	4	4
3	3	3	3	3
2	2	2	2	2
1	1	1	1	1

Evaluation vector of L with exponent 1

Figure 1: Illustration of a k-pseudoflat L defined over $\mathbb{F}_p^n$ with $k = 2, p = 5$ and $n = 5$. Picture on the left shows the points in L (recall that each of $L_1, \ldots, L_4$ are 1-flats or lines). Each L_i (for $1 \le i \le 4$) has $p^{k-1} = 5$ points in it. The points in L are shown by filled circles and the points in $\mathbb{F}_5^5 \setminus L$ are shown by unfilled circles. The picture on the right is the (partial) evaluation vector of the pseudoflat corresponding to L with exponent 1.

Given a function $f : \mathbb{F}_p^n \to \mathbb{F}_p$, for $y_1, \cdots, y_\ell, b \in \mathbb{F}_p^n$, for all $i \in [p-2]$, we define

$$T_f^i(y_1, \cdots, y_\ell, b) \stackrel{def}{=} \sum_{c=(c_1,\cdots,c_\ell)\in\mathbb{F}_p^\ell} c_1^i \cdot f(b + \sum_{j\in[\ell]} c_j y_j). \tag{5}$$

The above can also be interpreted similarly as the dot product of the codeword corresponding to the ℓ-pseudoflat with exponent i generated by $y_1, \cdots, y_\ell$ at b (exponentiated along y_1) and the codeword corresponding to the function f (also see Observation 3.9). With a slight abuse of notation we will use $T_f^0(y_1, \cdots, y_\ell, b)$ to denote $T_f(y_1, \cdots, y_\ell, b)$, where for notational convenience, we use $0^0 = 1$.

3 Characterization of Low Degree Polynomials over $\mathbb{F}_p$

In this section we present an exact characterization for the family $\mathcal{P}_t$ over prime fields. Specifically we prove the following:

Theorem 3.1 *Let $t = (p-1) \cdot k + R$. (Note $0 \le R \le p-2$.) Let $r = p-2-R$. Then a function f belongs to $\mathcal{P}_t$, if and only if* `for every` *$y_1, \cdots, y_{k+1}, b \in \mathbb{F}_p^n$, we have*

$$T_f(y_1, \cdots, y_{k+1}, b) = 0 \quad \text{if } r = 0; \tag{6}$$

$$T_f^r(y_1, \cdots, y_{k+1}, b) = 0 \quad \text{otherwise.} \tag{7}$$

As mentioned previously, the above characterization is a common generalization of previous special cases such as [FS95, AKK$^+$05]. Further, a characterization for the family $\mathcal{P}_t$ implies a characterization for $\mathrm{GRM}_p(t,n)$ and vice versa. It turns out that it is easier to characterize $\mathcal{P}_t$ when viewed as $\mathrm{GRM}_p(t,n)$. Therefore our goal is to determine whether a given word belongs to the GRM code. Since are we dealing with a linear code, a simple strategy will then be to check whether the given word is orthogonal to all the codewords in the dual code. Though this yields a characterization, this is computationally inefficient. Note however that the dot product is linear in its input. Therefore checking orthogonality with a basis of the dual code suffices. Further, to make it query efficient, we look for a dual basis with small weights. The above theorem essentially is a restatement of this idea.

We recall the following useful lemma that can be found in corollary 5.26 of [AJK98].

Lemma 3.2 $\mathrm{GRM}_q(k,n)$ *is a linear code with block length* q^n *and minimum distance* $(R+1)q^Q$ *where* R *is the remainder and* Q *the quotient resulting from dividing* $(q-1)\cdot n-k$ *by* $(q-1)$*. Denote the dual of a code* $\mathcal{C}$*, i.e. the dual of the subspace* $\mathcal{C}$*, by* $\mathcal{C}^\perp$ *Then* $\mathrm{GRM}_q(k,n)^\perp = \mathrm{GRM}_q((q-1)\cdot n-k-1,n)$*.*

Since the dual of a GRM code is again a GRM (of appropriate order), we therefore need the generators of GRM code (of arbitrary order). We first establish that flats and pseudoflats (of suitable dimension and exponent) indeed generate the Generalized Reed-Muller code (of desired order). We then end the section with a proof of Theorem 3.1 and a few remarks.

We begin with few simple observations about flats. Note that an ℓ-flat L is the intersection of $(n-\ell)$ hyperplanes in general position. Equivalently, it consists of all points v that satisfy $(n-\ell)$ linear equations over $\mathbb{F}_p$ (i.e., one equation for each hyperplane): $\forall i \in [n-\ell] \quad \sum_{j=1}^{n} c_{ij}x_j = b_i$ where c_{ij}, b_i defines the i^{th} hyperplane (i.e., v satisfies $\sum_{j=1}^{n} c_{ij}v_j = b_i$). General position means that the matrix $\{c_{ij}\}$ has rank $(n-\ell)$. Note that then the incidence vector of L can be written as

$$\prod_{i=1}^{n-\ell}(1-(\sum_{j=1}^{n} c_{ij}x_j - b_i)^{p-1}) = \begin{cases} 1 & \text{if } (v_1,\cdots,v_\ell)\in L \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

We record a lemma here that will be used later in this section. We leave the proof as a straightforward exercise.

Lemma 3.3 *For* $\ell \geq k$*, the incidence vector of any* ℓ*-flat is a linear sum of the incidence vectors of* k*-flats.*

As mentioned previously, we give an explicit basis for $\mathrm{GRM}_p(r,n)$. For the special case of $p=3$, our basis coincides with the min-weight basis given in [DK00].[5] However, in general, our basis differs from the min-weight basis provided in [DK00].

The following Proposition originated in the work of Delsarte (see [DK00],[AJK98]) and has at least two known proofs. It shows that the incidence vectors of flats form a basis for the Generalized Reed-Muller code of orders that are multiples of $(p-1)$. We give an alternative elementary proof for completeness.

Proposition 3.4 $\mathrm{GRM}_p((p-1)(n-\ell),n)$ *is generated by the incidence vectors of the* ℓ*-flats.*

[5]The equations of the hyperplanes are slightly different in our case; nonetheless, both of them define the same basis generated by the min-weight codewords.

Proof: We first show that the incidence vectors of the ℓ-flats are in $\mathrm{GRM}_p((p-1)(n-\ell), n)$. Recall that L is the intersection of $(n-\ell)$ independent hyperplanes. Therefore using (8), L can be represented by a polynomial of degree at most $(n-\ell)(p-1)$ in $x_1, \cdots, x_n$. Therefore the incidence vectors of ℓ-flats are in $\mathrm{GRM}_p((p-1)(n-\ell), n)$.

We prove that $\mathrm{GRM}_p((p-1)(n-\ell), n)$ is generated by ℓ-flats by induction on $n-\ell$. When $n-\ell = 0$, the code consists of constants, which is clearly generated by n-flats i.e., the whole space.

To prove the result for an arbitrary $(n-\ell) > 0$, we show that any monomial of total degree $d \leq (p-1)(n-\ell)$ can be written as a linear sum of the incidence vectors of ℓ-flats. Let the monomial be $x_1^{e_1} \cdots x_t^{e_t}$. Rewrite the monomials as $\underbrace{x_1 \cdots x_1}_{e_1 \text{ times}} \cdots \underbrace{x_t \cdots x_t}_{e_t \text{ times}}$. Group into products of $(p-1)$ (not necessarily distinct) variable as much as possible. Rewrite each group using Lemma 2.3 setting $k = (p-1)$. For any incomplete group of size $d' < p-1$, use the same lemma by setting the last $(p-1-d')$ variables to the constant 1. After expansion, the monomial can be seen to be a sum of product of at most $(n-\ell)$ degree $(p-1)^{\text{th}}$ powered linear terms. We can add to it a polynomial of degree at most $(p-1)(n-\ell-1)$ so as to represent the resulting polynomial as a sum of polynomials, each polynomial as in (8). Each such non-zero polynomial is generated by a t flat, $t \geq \ell$. By induction, the polynomial we added is generated by $(\ell+1)$ flats. Thus, by Lemma 3.3 our given monomial is generated by ℓ-flats. ■

This leads to the following observation:

Observation 3.5 *Consider an ℓ-flat generated by $y_1, \cdots, y_\ell$ at b. Denote the incidence vector of this flat by I. Then the right hand side of (4) may be identified as $I \cdot f$, where I and f denote the vector corresponding to respective codewords and $\cdot$ is the dot (scalar) product.*

To generate GRM codes of arbitrary order, we need pseudoflats. Note that the points in a k-pseudoflat may alternatively be viewed as the space given by union of intersections of $(n-k)$ hyperplanes, where the union is parameterized by another hyperplane that does not take one particular value. Concretely, it is the set of points v which satisfy the following constraints over $\mathbb{F}_p$:

$$\forall i \in [n-k] \sum_{j=1}^{n} c_{ij} x_j = b_i; \text{ and } \quad \sum_{j=1}^{n} c_{n-k+1,j} x_j \neq b_{n-k+1}.$$

Thus the values taken by the points of a k-pseudoflat with exponent r is given by the polynomial

$$\prod_{i=1}^{n-k} (1 - (\sum_{j=1}^{n} c_{ij} x_j - b_i)^{(p-1)}) \cdot (\sum_{j=1}^{n} c_{n-k+1,j} x_j - b_{n-k+1})^r \tag{9}$$

Remark 3.6 *Note the difference between (9) and the basis polynomial in [DK00], which along with the action of the affine general linear group yields the min-weight codewords:*

$$h(x_1, \cdots, x_n) = \prod_{i=1}^{n-k} (1 - (x_i - w_i)^{(p-1)}) \prod_{j=1}^{r} (x_{n-k+1} - u_j),$$

where $w_1, \cdots, w_{n-k}, u_1, \cdots, u_r \in \mathbb{F}_p$.

The next lemma shows that the code generated by the incidence vectors of l-flats is a subcode of the code generated by the evaluation vectors of l-pseudoflats with exponent r.

Claim 3.7 *The evaluation vectors of ℓ-pseudoflats ($\ell \geq 1$) with exponent r ($r \in [p-2]$) generate a code containing the incidence vectors of ℓ-flats.*

Proof: Let W be the incidence vector of an ℓ-flat generated by $y_1, \cdots, y_\ell$ at b. Clearly $W = \langle 1, \cdots, 1 \rangle$, where the i^{th} ($i \in [p-1] \cup \{0\}$) coordinate denotes the values taken by the characteristic functions of $(\ell-1)$-flats generated by $y_2, \cdots, y_\ell$ at $b + i \cdot y_1$.[6] Let this denote the standard basis. Let L_j be a pseudoflat with exponent r generated by $y_1, \cdots, y_\ell$ exponentiated along y_1 at $b + j \cdot y_1$, for each $j \in \mathbb{F}_p$, and let V_j be the corresponding evaluation vector. By Definition 2.6, V_j assign a value i^r to the $(\ell-1)$-flat generated by $y_2, \cdots, y_\ell$ at $b + (j+i)y$. Rewriting them in the standard basis yields that $V_j = \langle (p-j)^r, (p-j+1)^r, \cdots, (p-j+i)^r, \cdots, (p-j-1)^r \rangle \in \mathbb{F}_p^p$. Let λ_j denote p variables for $t = 0, 1, \cdots, (p-1)$, each taking values in $\mathbb{F}_p$. Then a solution to the following system of equations

$$\forall i \in [p-1] \cup \{0\} \quad 1 = \sum_{j \in \mathbb{F}_p} \lambda_j (i-j)^r$$

implies that $W = \sum_{j=0}^{p-1} \lambda_j V_j$, which suffices to establish the claim. Consider the identity

$$1 = (-1) \sum_{j \in \mathbb{F}_p} (j+i)^r j^{p-1-r}$$

which may be verified by expanding and applying Lemma 2.1. Setting λ_j to $(-1)(-j)^{p-1-r}$ establishes the claim. ∎

The next Proposition complements Proposition 3.4. Together they say that by choosing dimension and exponent appropriately, Generalized Reed-Muller code of any given order can be generated. This gives an equivalent representation of Generalized Reed-Muller code. An exact characterization then follows from this alternate representation.

Proposition 3.8 *For every $r \in [p-2]$, the linear code generated by the evaluation vectors of ℓ-pseudoflats with exponent r is equivalent to* $\mathrm{GRM}_p((p-1)(n-\ell)+r, n)$.

Proof: For the forward direction, consider an l-pseudoflat L with exponent r. Its evaluation vector is given by an equation similar to (9). Thus the codeword corresponding to the evaluation vector of this flat can be represented by a polynomial of degree at most $(p-1)(n-\ell)+r$. This completes the forward direction.

To prove the other direction, we restrict our attention to monomials of degree at least $(p-1)(n-\ell)+1$ and show that these monomials are generated by ℓ-pseudoflats with exponent r. Since monomials of degree at most $(p-1)(n-\ell)$ is generated by ℓ-flats, Claim 3.7 will establish the Proposition. Now consider any such monomial. Let the degree of the monomial be $(p-1)(n-\ell)+r'$ ($1 \leq r' \leq r$). Rewrite it as in Proposition 3.4. Since the degree of the monomial is $(p-1)(n-\ell)+r'$, we will be left with an incomplete group of degree r'. We make any incomplete group complete (i.e. of size r) by adding 1's (as necessary) to the product. We then use Lemma 2.3 to rewrite this group as a linear sum of r^{th} powered terms. After expansion, the monomial can be seen to be a sum of product of at most $(n-\ell)$ degree $(p-1)^{th}$ powered linear terms and a r^{th} powered linear term. Each such polynomial is generated either by an ℓ-pseudoflat with exponent r or an ℓ-flat. Claim 3.7 completes the proof. ∎

The following is analogous to Observation 3.5.

[6] Recall that a ℓ-pseudoflat (as well as a flat) assigns the same value to all points in the same $(\ell-1)$-flat.

Observation 3.9 *Consider an ℓ-pseudoflat with exponent r, generated by $y_1, \cdots, y_\ell$ at b exponentiated along y_1. Let E be the evaluation vector of this pseudoflat with exponent r. Then the right hand side of (5) may be interpreted as $E \cdot f$.*

Now we prove the exact characterization.

Proof of Theorem 3.1: The proof directly follows from Lemma 3.2, Proposition 3.4, Proposition 3.8, Observation 3.5 and Observation 3.9. Indeed by Observation 3.5 and Observation 3.9, (6) and (7) are essentially tests to determine whether the dot product of the function with every vector in the dual space of $\mathrm{GRM}(t, n)$ evaluates to zero. ■

Remark 3.10 *One can obtain an alternate characterization from Remark 3.6 which we state here without proof.*

Let $t = (p-1) \cdot k + R$ (note $0 < R \leq (p-2)$). Let $r = (p-1) - R - 1$. Let $W \subseteq \mathbb{F}_p$ with $|W| = r$. Define the polynomial $g(x) \stackrel{def}{=} \prod_{\alpha \in W}(x - \alpha)$ if W is non-empty; and $g(x) = 1$ otherwise. Then a function f belongs to $\mathcal{P}_t$ if and only if for every $y_1, \cdots, y_{k+1}, b \in \mathbb{F}_p^n$, we have

$$\sum_{c_1 \in \mathbb{F}_p \setminus W} g(c_1) \sum_{(c_2, \cdots, c_{k+1}) \in \mathbb{F}_p^k} f(b + \sum_{i=1}^{k+1} c_i \cdot y_i) = 0.$$

Moreover, this characterization can also be extended to certain degrees for more general fields, i.e., $\mathbb{F}_{p^s}$ (see the next remark).

Ding and Key [DK00] showed that minimal weight bases in general do not generate GRM codes. In a nutshell, this happens because certain transformations between monomials of fixed degree do not act transitively. These transformations involve binomial coefficients, and some indices get annihilated by Lucas's theorem.

We mention here that we do not know whether our exact characterization can be worked out over arbitrary finite fields $\mathbb{F}_q$. The difficulty essentially seems to arise from our failure to estimate sums of the form $\sum_{i \in I} c_i \alpha^i$ where

$$I \stackrel{\text{def}}{=} \{r \mid \binom{n}{r} \neq 0 \text{ over } \mathbb{F}_q \text{ where } p < n \leq q-1\},$$

and $c_i \in \mathbb{F}_q$.

Remark 3.11 *The exact characterization of low degree polynomials as claimed in [FS95] may be proved using duality. Note that their proof works as long as the dual code has a min-weight basis (see [DK00]). Suppose that the polynomial has degree $d \leq q - q/p - 1$, then the dual of $\mathrm{GRM}_q(d, n)$ is $\mathrm{GRM}_q((q-1)n - d - 1, n)$ and therefore has a min-weight basis. Note that then the dual code has min-weight $(d+1)$. Therefore, assuming the minimum weight codewords constitute a basis, any $d+1$ evaluations of the original polynomial on a line are dependent and vice-versa. We leave the details as an exercise for the interested readers.*

4 A Tester for Low Degree Polynomials over $\mathbb{F}_p^n$

In this section we present and analyze a one-sided tester for $\mathcal{P}_t$. The analysis of the algorithm roughly follows the proof structure given in [RS96, AKK$^+$05]. We emphasize that the generalization from [AKK$^+$05] to our case is not straightforward. As in [RS96, AKK$^+$05] we first define a self-corrected version of the (possibly corrupted) function being tested. The straightforward adoption of the analysis given in [RS96] gives reasonable bounds. However, the better bound is achieved by following the techniques developed in [AKK$^+$05]. In there, they show that the self-corrector function can be interpolated with overwhelming probability. However their approach appears to use special properties of $\mathbb{F}_2$ and it is not clear how to generalize their technique for arbitrary prime fields. We give a clean formulation which relies on the flats being represented through polynomials as described earlier. In particular, Claims 4.10, 4.12 and their generalization appear to require our new polynomial based view.

4.1 Tester in $\mathbb{F}_p$

In this subsection we describe the algorithm when the underlying field is $\mathbb{F}_p$. In what follows, ϵ denotes the distance between f and $\mathcal{P}_t$.

Algorithm Test-$\mathcal{P}_t$ in $\mathbb{F}_p$

0. Let $t = (p-1)\cdot k + R$, $0 \le R < (p-1)$. Denote $r = p - 2 - R$.
1. Uniformly and independently at random select $y_1, \cdots, y_{k+1}, b \in \mathbb{F}_p^n$.
2. If $T_f^r(y_1, \cdots, y_{k+1}, b) \neq 0$, then `reject`, else `accept`.

Theorem 4.1 *The algorithm* **Test-$\mathcal{P}_t$ in $\mathbb{F}_p$** *is a one-sided tester for $\mathcal{P}_t$ with a success probability at least min$(\Omega(p^{k+1}\epsilon), \frac{1}{2(k+7)p^{k+2}})$.*

Corollary 4.2 *Repeating the algorithm* **Test-$\mathcal{P}_t$ in $\mathbb{F}_p$** $\Theta(\frac{1}{p^{k+1}\epsilon} + kp^k)$ *times, the probability of error can be reduced to less than* $1/2$.

We will provide a general proof framework. However, for the ease of exposition we prove the main technical lemmas for the case of $\mathbb{F}_3$. The proof idea in the general case is similar and the details are omitted. Therefore we will essentially prove the following.

Theorem 4.3 *The algorithm* **Test-$\mathcal{P}_t$ in $\mathbb{F}_3$** *is a one-sided tester for $\mathcal{P}_t$ with success probability at least min$(\Omega(3^{k+1}\epsilon), \frac{1}{2(t+7)3^{t/2+1}})$.*

4.1.1 Intuition for the proof

As was mentioned earlier, the analysis of the above algorithm roughly follows the proof structure given in [RS96, AKK$^+$05]. Recall that our task is to catch functions that are not in the family $\mathcal{P}_t$. Of course, the exact characterization implies that our tester can only have one-sided error. This means, if f, the function being tested, somehow passes our test with high probability, we then need to justify that f is indeed close to the family $\mathcal{P}_t$. To prove this, we essentially prove that in this case f can be "uniquely" decoded, confirming that the function is indeed not very far given that the rejection probability, say η, of our algorithm is small.

As in [RS96, AKK+05] we first define a self-corrected version, say g, of the function f (see (10)). In Lemma 4.4, it is shown that if η is small then the distance between f and g is small. We then show that the value of g at any point can be obtained with good probability by interpolating the values of f on a random k-flat or k-pseudoflat as appropriate. The straightforward adoption of the analysis given in [RS96] gives Lemma 4.5 which in turn gives reasonable bounds on the probability. However, the better bound is achieved by following the techniques developed in [AKK+05] and is given in Lemma 4.6. The proof of the lemma in turn crucially uses Claims 4.10 and 4.12.

Next in Lemma 4.7 we show that if the rejection probability is sufficiently low, then g indeed belongs to the family $\mathcal{P}_t$, i.e. it satisfies the exact characterization of the family $\mathcal{P}_t$. The proof of Lemma 4.7 in turn uses Lemma 4.6.

If η is sufficiently large, then we have nothing to prove (this gives the $\frac{1}{2(t+7)3^{t/2+1}}$ term in Theorem 4.3). Otherwise, by Lemma 4.7 we know it can be "decoded" to a function g that belongs to the family $\mathcal{P}_t$. We also know that g and f are sufficiently close (this follows from Lemma 4.4). This by Lemma 4.8 in turn will imply that η is large enough in terms of ϵ (this gives the $3^{k+1}\epsilon$ term in Theorem 4.3).

4.2 Analysis of Algorithm *Test-*$\mathcal{P}_t$

In this subsection we analyze the algorithm described in Section 4.1. From Claim 3.1 it is clear that if $f \in \mathcal{P}_t$, then the tester accepts. Thus, the bulk of the proof is to show that if f is ϵ-far from $\mathcal{P}_t$, then the tester rejects with significant probability. Our proof structure follows that of the analysis of the test in [AKK+05]. In what follows, we will denote $T_f(y_1, \cdots, y_l, b)$ by $T_f^0(y_1, \cdots, y_l, b)$ for the ease of exposition. In particular, let f be the function to be tested for membership in $\mathcal{P}_t$. Assume we perform Test T_f^i for an appropriate i as required by the algorithm described in Section 4.1. For such an i, we define $g_i : \mathbb{F}_p^n \to \mathbb{F}_p$ as follows: For $y \in F_p^n, \alpha \in \mathbb{F}_p$, denote $p_{y,\alpha} = \Pr_{y_1,\cdots,y_{k+1}}[f(y) - T_{f_i}(y - y_1, y_2, \cdots, y_{k+1}, y_1) = \alpha]$. Define $g_i(y) = \alpha$ such that $\forall \beta \neq \alpha \in \mathbb{F}_p, p_{y,\alpha} \geq p_{y,\beta}$ with ties broken arbitrarily. With this meaning of plurality, for all $i \in [p-2] \cup \{0\}$, g_i can be written as:

$$g_i(y) \;=\; \text{plurality}_{y_1,\cdots,y_{k+1}} \left[f(y) - T_f^i(y - y_1, y_2, \cdots, y_{k+1}, y_1) \right]. \tag{10}$$

Further define

$$\eta_i \stackrel{def}{=} Pr_{y_1,\cdots,y_{k+1},b}[T_f^i(y_1, \cdots, y_{k+1}, b) \neq 0], \tag{11}$$

which is typically very small, i.e., at most $\frac{1}{p^{k+2}}$. The next lemma follows from a Markov-type argument.

Lemma 4.4 *For a fixed $f : \mathbb{F}_p^n \to \mathbb{F}_p$, let g_i, η_i be defined as above. Then, $\delta(f, g_i) \leq 2\eta_i$.*

Proof: Consider the set of elements y such that $\Pr_{y_1,\cdots,y_{k+1}}[f(y) = f(y) - T_f^i(y - y_1, y_2, \cdots, y_{k+1}, y_1)] < 1/2$. If the fraction of such elements is more than $2\eta_i$ then that contradicts the condition that

$$\begin{aligned}
\eta_i &= \Pr_{y_1,\cdots,y_{k+1},b}[T_f^i(y_1, \cdots, y_{k+1}, b) \neq 0] \\
&= \Pr_{y_1,y_2,\cdots,y_{k+1},b}[T_f^i(y_1 - b, y_2, \cdots, y_{k+1}, b) \neq 0] \\
&= \Pr_{y,y_1,\cdots,y_{k+1}}[f(y) \neq f(y) - T_f^i(y - y_1, y_2, \cdots, y_{k+1}, y_1)].
\end{aligned}$$

Therefore, we obtain $\delta(f, g_i) \leq 2\eta_i$. ∎

Note that $\Pr_{y_1,\cdots,y_{k+1}}[g_i(y) = f(y) - T_f^i(y - y_1, y_2, \cdots, y_{k+1}, y_1)] \geq \frac{1}{p}$. We now show that this probability is actually much higher. The next lemma gives a weak bound in that direction following the analysis in [RS96]. For the sake of completeness, we present a proof in the appendix.

Lemma 4.5 *For all* $y \in \mathbb{F}_p^n$, $\Pr_{y_1,\cdots,y_{k+1}\in\mathbb{F}_p^n}[g_i(y) = f(y) - T_f^i(y - y_1, y_2, \cdots, y_{k+1}, y_1)] \geq 1 - 2p^{k+1}\eta_i$.

However, when the degree being tested is larger than the field size, we can improve the above lemma considerably. The following lemma strengthens Lemma 4.5 when $t \geq (p-1)$ or equivalently $k \geq 1$. We now focus on the $\mathbb{F}_3$ case. The proof appears in Section 4.2.1.

Lemma 4.6 *For all* $y \in \mathbb{F}_3^n$, $\Pr_{y_1,\cdots,y_{k+1}\in\mathbb{F}_3^n}[g_i(y) = f(y) - T_f^i(y - y_1, y_2, \cdots, y_{k+1}, y_1)] \geq 1 - (4k + 14)\eta_i$.

Lemma 4.6 will be instrumental in proving the next lemma, which shows that sufficiently small η_i implies that g_i is the self-corrected version of the function f (the proof appears in Section 4.2.2).

Lemma 4.7 *Let* $k \geq 1$ *be an integer. Over* $\mathbb{F}_3$, *if* $\eta_i < \frac{1}{2(2k+7)3^{k+1}}$, *then the function* g_i *belongs to* $\mathcal{P}_t$.

By combining Lemma 4.4 and Lemma 4.7 we obtain that if f is $\Omega(1/(k3^k))$-far from $\mathcal{P}_t$ then $\eta_i = \Omega(1/(k3^k))$. We next consider the case in which η_i is small. By Lemma 4.4, in this case, the distance $\delta = \delta(f, g)$ is small. The next lemma shows that in this case the test rejects f with probability that is close to $3^{k+1}\delta$. This follows from the fact that in this case, the probability over the selection of $y_1, \cdots, y_{k+1}, b$, that among the 3^{k+1} points $\sum_i c_i y_i + b$, the functions f and g differ in precisely one point, is close to $3^{k+1} \cdot \delta$. Observe that if they do, then the test rejects.

Lemma 4.8 *Suppose* $0 \leq \eta_i \leq \frac{1}{2(2k+7)3^{k+1}}$. *Let* δ *denote the relative distance between* f *and* g, $\ell = 3^{k+1}$, *and* $Q \stackrel{def}{=} (\frac{1-\ell\delta}{1+\ell\delta}) \cdot \ell\delta$. *Then, when* $y_1, \cdots, y_{k+1}, b$ *are chosen randomly, the probability that for exactly one point* v *among the* ℓ *points* $\sum_i C_i y_i + b$, $f(v) \neq g(v)$ *is at least* Q.

Observe that $\eta_i = \Omega(Q) = \Omega(3^{k+1}\delta)$. The proof of Lemma 4.8 is deferred to Section 4.2.3

Proof of Theorem 4.3: Clearly if f belongs to $\mathcal{P}_t$, then by Claim 3.1 the tester accepts f with probability 1.

Therefore let $\delta(f, \mathcal{P}_t) \geq \epsilon$. Let $d = \delta(f, g_r)$, where r is as in algorithm **Test-$\mathcal{P}_t$**. If $\eta_r < \frac{1}{2(2k+7)3^{k+1}}$ then by Lemma 4.7 $g_r \in \mathcal{P}_t$ and, by Lemma 4.8, $\eta_r = \Omega(3^{k+1} \cdot d) = \Omega(3^{k+1}\epsilon)$. Hence $\eta_r \geq min\left(\Omega(3^{k+1}\epsilon), \frac{1}{2(2k+7)3^{k+1}}\right)$. ■

Remark 4.9 *Theorem 4.1 follows from a similar argument.*

4.2.1 Proof of Lemma 4.6

Observe that the goal of the lemma is to show that at any fixed point y, if g_i is interpolated out of a random hyperplane, then w.h.p. the interpolated value is the most popular vote. To ensure this we show that if g_i is interpolated on two independently random hyperplanes, then the probability that these interpolated values are same, that is the collision probability, is large. To estimate this collision probability, we show that the difference of the interpolation values can be rewritten as a sum of T_f^i on small number of random hyperplanes. Thus if the test passes often (that is, T_f^i

evaluates to zero w.h.p.), then this sum (by a simple union bound) evaluates to zero often, which proves the high collision probability.

The improvement will arise because we will express differences involving $T_f^i(\cdots)$ as a telescoping series to essentially reduce the number of events in the union bound. To do this we will need the following claims. They can easily be verified by expanding the terms on both sides like the proof of Claim 4 in [AKK$^+$05]. However, this does not give much insight into the general case i.e., for $\mathbb{F}_p$. We provide an alternate proof that can be generalized to get similar claims and has a much cleaner structure based on the underlying geometric structure, i.e., flats or pseudoflats.

Claim 4.10 *For every $\ell \in \{2, \cdots, k+1\}$, for every $y(= y_1), z, w, b, y_2, \cdots, y_{\ell-1}, y_{\ell+1}, \cdots, y_{k+1} \in \mathbb{F}_3^n$, let*

$$S_f(y, z) \stackrel{def}{=} T_f(y, y_2, \cdots, y_{\ell-1}, z, y_{\ell+1}, \cdots, y_{k+1}, b).$$

(Note that $T_f(\cdot)$ is a symmetric function in all but its last input. Therefore to enhance readability, we omit the reference to index ℓ in S.) Then the following holds:

$$S_f(y, w) - S_f(y, z) = S_f(y + w, z) + S_f(y - w, z) - S_f(y + z, w) - S_f(y - z, w).$$

Proof: Assume y, z, w are linearly independent. If not then both sides are equal to 0 and hence the equality is trivially satisfied. To see why this claim is true for the left hand side, recall the definition of $T_f(\cdot)$ and note that the sets of points in the flat generated by $y, y_2, \cdots, y_{\ell-1}, w, y_{\ell+1}, \cdots, y_{k+1}$ at b and the flat generated by $y, y_2, \cdots, y_{\ell-1}, z, y_{\ell+1}, \cdots, y_{k+1}$ at b are the same. A similar argument works for the expression on the right hand side of the equality.

We first prove the claim for the special case of $k = 1$ and $b = \mathbf{0}$. Consider the space $\mathcal{H}$ generated by y, z and w at $\mathbf{0}$. Thus, every point in $\mathcal{H}$ can be written as $\hat{y} \cdot y + \hat{z} \cdot z + \hat{w} \cdot w$, with $\hat{y}, \hat{z}$ and $\hat{w}$ in $\mathbb{F}_3$. Note that $S_f(y, w)$ (with $b = \mathbf{0}$) is just $f \cdot 1_L$, where 1_L is the incidence vector of the 2-flat given by the equation $\hat{z} = 0$. Therefore 1_L is equivalent to the polynomial $(1 - \hat{z}^2)$. Similarly $S_f(y, z) = f \cdot 1_{L'}$ where L' is given by the polynomial $(1 - \hat{w}^2)$. When it is clear from context, we will identify the coordinates $\hat{y}$ with y itself, etc.

We use the following polynomial identity (in $\mathbb{F}_3$)

$$w^2 - z^2 = [1 - (y - w)^2 + 1 - (y + w)^2] - [1 - (y + z)^2 + 1 - (y - z)^2].$$

Now observe that the equation $(1 - (y - w)^2)$ is the incidence vector of the flat generated by $y + w$ and z. Similar observations hold for other terms. Therefore, interpreting the above equation in terms of incidence vectors of flats, we complete the proof for the case of $k = 1$ and $b = \mathbf{0}$ with Observation 3.5.

To complete the proof, we "reduce" the $k > 1$ and $b \neq \mathbf{0}$ case to the $k = 1$ and $b = \mathbf{0}$ case . A linear transform (or renaming the co-ordinate system appropriately) reduces the case of $k = 1$ and $b \neq \mathbf{0}$ to the case of $k = 1$ and $b = \mathbf{0}$. We now show how to "reduce" the case of $k > 1$ to the $k = 1$ case. Fix some values $c_2, \cdots, c_{\ell-1}, c_{\ell+1}, \cdots, c_{k+1}$ and note that one can write $c_1 y + c_2 y_2 + \cdots c_{\ell-1} y_{\ell-1} + c_\ell w + c_{\ell+1} y_{\ell+1} + c_{k+1} y_{k+1} + b$ as $c_1 y + c_\ell w + b'$, where $b' = \sum_{j \in \{2, \cdots, \ell-1, \ell+1, \cdots, k+1\}} c_j y_j + b$. Thus, $S_f(y, w) = \sum_{(c_2, \cdots, c_{\ell-1}, c_{\ell+1}, \cdots, c_{k+1}) \in \mathbb{F}_3}^{k-1} \sum_{(c_1, c_\ell) \in \mathbb{F}_3^2} f(c_1 y + c_\ell w + b')$, where b' is as defined earlier. One can rewrite the other $S_f(\cdot)$ terms similarly. Note that for a fixed vector $(c_2, \cdots, c_{\ell-1}, c_{\ell+1}, \cdots, c_{k+1})$, the value of b' is the same. Finally note that the equality (in the $k > 1$ case) is satisfied if 3^{k-1} similar equalities hold (in the $k = 1$ case). ■

We have the following analogue[7] of Claim 4.10 in $\mathbb{F}_p$:

Claim 4.11 *For every $\ell \in \{2, \cdots, k+1\}$, for every $y(= y_1), z, w, b, y_2, \cdots, y_{\ell-1}, y_{\ell+1}, \cdots, y_{k+1} \in \mathbb{F}_p^n$, with notation used from the previous lemma, it holds that*

$$S_f(y, w) - S_f(y, z) = \sum_{e \in \mathbb{F}_p^*} \left[S_f(y + ew, z) - S_f(y + ez, w) \right].$$

Proof: (**Sketch**) If the following identity

$$w^{(p-1)} - z^{(p-1)} = \sum_{e \in \mathbb{F}_p^*} \left[[1 - (ew + y)^{(p-1)}] - [1 - (ez + y)^{(p-1)}] \right], \tag{12}$$

is true then we can prove the claim along the same lines as the proof of Claim 4.10 above. We complete the proof by proving (12). Consider the sum: $\sum_{e \in \mathbb{F}_p^*} (ew + y)^{(p-1)}$. Expanding the terms and rearranging the sums we get $\sum_{j=0}^{p-1} \binom{p-1}{j} w^{(p-1)-j} y^j \sum_{e \in \mathbb{F}_p^*} e^{p-1-j}$. By Lemma 2.1 the sum evaluates to $(-w^{(p-1)} - y^{(p-1)})$. Similarly, $\sum_{e \in \mathbb{F}_p^*} (ez + y)^{(p-1)} = (-z^{(p-1)} - y^{(p-1)})$ which proves (12). ■

Claim 4.12 *For every $\ell \in \{2, \cdots, k+1\}$, for every $y(= y_1), z, w, b, y_2, \cdots, y_{\ell-1}, y_{\ell+1}, \cdots, y_{k+1} \in \mathbb{F}_3^n$, denote*

$$S_f^1(y, w) \stackrel{def}{=} T_f^1(y, y_2, \cdots, y_{\ell-1}, w, y_{\ell+1}, \cdots, y_{k+1}, b).$$

Then[8] *the following holds:*

$$S_f^1(y, w) - S_f^1(y, z) = S_f^1(y + z, w) + S_f^1(y - z, w) - S_f^1(y + w, z) - S_f^1(y - w, z).$$

Proof: Note here that the defining equation of $S_f^1(y, z)$ is $y(1 - w^2)$. Now consider the following identity in $\mathbb{F}_3$:

$$\begin{aligned} y(z^2 - w^2) &= (y + w)[1 - (y - w)^2] + (y - w)[1 - (y + w)^2] \\ &\quad -(y + z)[1 - (y - z)^2] - (y - z)[1 - (y + z)^2] \end{aligned}$$

for variables $y, z, w \in \mathbb{F}_3$. Rest of the proof is similar to the proof of Claim 4.10 (the proof replaces flats by pseudoflats) and is omitted. ■

We now prove the following analogue in $\mathbb{F}_p$:

Claim 4.13 *For every $i \in \{1, \cdots, p-2\}$, for every $\ell \in \{2, \cdots, k+1\}$ and for every $y(= y_1), z, w, b, y_2, \cdots, y_{\ell-1}, y_{\ell+1}, \cdots, y_{k+1} \in \mathbb{F}_p^n$, denote*

$$S_f^i(y, w) \stackrel{def}{=} T_f^i(y, y_2, \cdots, y_{\ell-1}, w, y_{\ell+1}, \cdots, y_{k+1}, b).$$

Then there exists c_i such that

$$S_f^i(y, w) - S_f^i(y, z) = c_i \sum_{e \in \mathbb{F}_p^*} \left[S_f^i(y + ew, z) - S_f^i(y + ez, w) \right].$$

[7] This claim can be extended to $\mathbb{F}_q$ in a straightforward manner. We mention here that this lemma over $\mathbb{F}_q$ allows one to prove a similar version of Lemma 4.6 over $\mathbb{F}_q$. That lemma along with versions of Lemma 4.7 and Lemma 4.8 can be used to get a robust characterization as is done in [KR06].

[8] Note that $T_f^i(\cdot)$ is a symmetric function in its all but last and first input. Therefore to enhance readability, we omit the reference to index ℓ.

Proof: Observe that $T^i_f(y,z) = f \cdot E_{L_i}$, where E_{L_i} denotes the evaluation vector of the pseudoflat L with exponent i, generated by y, z at b exponentiated along y. Note that the polynomial defining E_{L_i} is just $y^i(w^{(p-1)}-1)$. We now give an identity similar to that of (12) that completes the proof. We claim that the following identity holds

$$y^i(w^{(p-1)} - z^{(p-1)}) = c_i \sum_{e \in \mathbb{F}_p^*} \left[(y+ew)^i[1-(y-ew)^{(p-1)}] - (y+ez)^i[1-(y-ez)^{(p-1)}] \right]. \quad (13)$$

where $c_i = 2^{-i}$. Before we prove the identity, note that $(-1)^j \binom{p-1}{j} = 1$ in $\mathbb{F}_p$. This is because for $1 \le m \le j$, $m = (-1)(p-m)$. Therefore $j! = (-1)^j \frac{(p-1)!}{(p-j-1)!}$ holds in $\mathbb{F}_p$. Substitution yields the desired result. Also note that $\sum_{e \in \mathbb{F}_p^*} (y+ew)^i = -y^i$ (expand and apply Lemma 2.1). Now consider the sum

$$\begin{aligned}
\sum_{e \in \mathbb{F}_p^*} (y+ew)^i (y-ew)^{(p-1)} &= \sum_{e \in \mathbb{F}_p^*} \sum_{0 \le j \le i; 0 \le m \le (p-1)} (-1)^m \binom{i}{j} \binom{p-1}{m} y^{(p-1)+i-j-m} w^{j+m} e^{j+m} \\
&= \sum_{0 \le j \le i; 0 \le m \le (p-1)} (-1)^m \binom{i}{j} \binom{p-1}{m} y^{(p-1)+i-j-m} w^{j+m} \sum_{e \in \mathbb{F}_p^*} e^{j+m} \\
&= (-1)[y^{(p-1)+i} + (-1)^{(p-1)} \sum_{j=0}^{i} \binom{i}{j} \underbrace{\binom{p-1}{p-1-j} (-1)^j}_{=1} y^i w^{(p-1)}] \\
&= (-1)[y^i + y^i w^{(p-1)} 2^i]. \quad (14)
\end{aligned}$$

Similarly one has $\sum_{e \in \mathbb{F}_p^*} (y+ez)^i (y-ez)^{(p-1)} = (-1)[y^i + y^i z^{(p-1)} 2^i]$. Substituting and simplifying one gets (13). ∎

We will also need the following claims.

Claim 4.14 *For every $\ell \in \{2, \cdots, k+1\}$, $y(=y_l), z, w, b, y_2, \cdots, y_{\ell-1}, y_{\ell+1}, \cdots, y_{k+1} \in \mathbb{F}_3^n$, with notation used in the previous claim, it holds that*

$$S^1_f(w,y) - S^1_f(z,y) = S^1_f(z+w, y-z) - S^1_f(z+w, y-w) + S^1_f(y+z, w) + S^1_f(y-z, w) - S^1_f(y+w, z) - S^1_f(y-w, z).$$

Proof: The above follows from the identity

$$w(1-z^2) - z(1-w^2) = (z+w)[1-(z+y)^2 - 1 + (y+w)^2] + y(w^2 - z^2).$$

Also we can expand $y(w^2 - z^2)$ as in the proof of Claim 4.12. ∎

We have the following analogue in $\mathbb{F}_p$.

Claim 4.15 *For every $i \in \{1, \cdots, p-2\}$, for every $\ell \in \{2, \cdots, k+1\}$ and for every $y(=y_\ell), z, w, b, y_2, \cdots, y_{\ell-1}, y_{\ell+1}, \cdots, y_{k+1} \in \mathbb{F}_p^n$, there exists $c_i \in \mathbb{F}_p^*$ such that*

$$\begin{aligned}
S^i_f(w,y) - S^i_f(z,y) = \sum_{e \in \mathbb{F}_p^*} \big[& S^i_f(y+ew, y-ew) - S^i_f(w+ey, w-ey) + S^i_f(z+ey, z-ey) \\
& - S^i_f(y+ez, y-ez) + c_i \left[S^i_f(y+ew, z) - S^i_f(y+ez, w) \right] \big]
\end{aligned}$$

Proof: The above follows from the identity

$$w^i(1-z^{(p-1)}) - z^i(1-w^{(p-1)}) = (w^i-y^i)(1-z^{(p-1)}) - (z^i-y^i)(1-w^{(p-1)}) + y^i(w^{(p-1)}-z^{(p-1)}).$$

We also use that $\sum_{e\in\mathbb{F}_p^*}(w+ey)^i = -w^i$ and Claim 4.13 to expand the last term. Note that $c_i = 2^{-i}$ as before. ■

Proof of Lemma 4.6: We first prove the lemma for $g_0(y)$. We fix $y \in \mathbb{F}_3^n$ and let $\gamma \stackrel{def}{=} \Pr_{y_1,\cdots,y_{k+1}\in\mathbb{F}_3^n}[g_0(y) = f(y) - T_f(y-y_1, y_2, \cdots, y_{k+1}, y_1)]$. Recall that we want to lower bound γ by $1-(4k+14)\eta_0$. In that direction, we bound a slightly different but related probability. Let

$$\mu \stackrel{def}{=} \Pr_{y_1,\cdots,y_{k+1},z_1,\cdots,z_{k+1}\in\mathbb{F}_3^n}[T_f(y-y_1,y_2,\cdots,y_{k+1},y_1) = T_f(y-z_1,z_2,\cdots,z_{k+1},z_1)]$$

Denote $\langle y_1,\cdots,y_{k+1}\rangle$ by Y and $\langle z_1,\cdots,z_{k+1}\rangle$ by Z. Then by the definitions of μ and γ we have, $\gamma \geq \mu$. (This is because for a probability vector $v \in [0,1]^n$, $\left\|v\right\|_\infty = \max_{i\in[n]}\{v_i\} \geq \max_{i\in[n]}\{v_i\}\cdot(\sum_{i=1}^n v_i) = \sum_{i=1}^n v_i\cdot\max_{i\in[n]}\{v_i\} \geq \sum_{i=1}^n v_i^2 = \left\|v\right\|_2^2$.)

We have $\mu = \Pr_{y_1,\cdots,y_{k+1},z_1,\cdots,z_{k+1}\in\mathbb{F}_3^n}[T_f(y-y_1,y_2,\cdots,y_{k+1},y_1) - T_f(y-z_1,z_2,\cdots,z_{k+1},z_1) = 0]$.

Now, for any choice of $y_1,\cdots,y_{k+1}$ and $z_1,\cdots,z_{k+1}$:

$$\begin{array}{lcll}
T_f(y-y_1,y_2,\cdots,y_{k+1},y_1) & - & T_f(y-z_1,z_2,\cdots,z_{k+1},z_1) & = \\
T_f(y-y_1,y_2,\cdots,y_{k+1},y_1) & - & T_f(y-y_1,y_2,\cdots,y_k,z_{k+1},y_1) & + \\
T_f(y-y_1,y_2,\cdots,y_k,z_{k+1},y_1) & - & T_f(y-y_1,y_2,\cdots,y_{k-1},z_k,z_{k+1},y_1) & + \\
T_f(y-y_1,y_2,\cdots,y_{k-1},z_k,z_{k+1},y_1) & - & T_f(y-y_1,y_2,\cdots,y_{k-2},z_{k-1},z_k,z_{k+1},y_1) & + \\
\vdots & & & \\
T_f(y-y_1,z_2,z_3,\cdots,z_{k+1},y_1) & - & T_f(y-z_1,z_2,\cdots,z_{k+1},y_1) & + \\
T_f(y-z_1,z_2,z_3,\cdots,z_{k+1},y_1) & - & T_f(y-y_1,z_2,\cdots,z_{k+1},z_1) & + \\
T_f(y-y_1,z_2,z_3,\cdots,z_{k+1},z_1) & - & T_f(y-z_1,z_2,\cdots,z_{k+1},z_1) &
\end{array}$$

Consider any pair $T_f(y-y_1,y_2,\cdots,y_\ell,z_{\ell+1},\cdots,z_{k+1},y_1) - T_f(y-y_1,y_2,\cdots,y_{\ell-1},z_\ell,\cdots,z_{k+1},y_1)$ that appears in the first k "rows" in the sum above. Note that $T_f(y-y_1,y_2,\cdots,y_\ell,z_{\ell+1},\cdots,z_{k+1},y_1)$ and $T_f(y-y_1,y_2,\cdots,y_{\ell-1},z_\ell,\cdots,z_{k+1},y_1)$ differ only in a single parameter. We apply Claim 4.10 and obtain:

$$\begin{array}{l}
T_f(y-y_1,y_2,\cdots,y_\ell,z_{\ell+1},\cdots,z_{k+1},y_1) - T_f(y-y_1,y_2,\cdots,y_{\ell-1},z_\ell,\cdots,z_{k+1},y_1) = \\
T_f(y-y_1+y_l,y_2,\cdots,y_{\ell-1},z_\ell,\cdots,z_{k+1},y_1) + T_f(y-y_1-y_l,y_2,\cdots,y_{\ell-1},z_\ell,\cdots,z_{k+1},y_1) \\
-T_f(y-y_1+z_l,y_2,\cdots,y_\ell,z_{\ell+1},\cdots,z_{k+1},y_1) - T_f(y-y_l-z_l,y_2,\cdots,y_\ell,z_{\ell+1},\cdots,z_{k+1},y_1).
\end{array}$$

Recall that y is fixed and $y_2,\cdots,y_{k+1},z_2,\cdots,z_{k+1} \in \mathbb{F}_3^n$ are chosen uniformly at random, so all the parameters on the right hand side of the equation are independent and uniformly distributed. Similarly one can expand the pairs $T_f(y-y_1,z_2,z_3,\cdots,z_{k+1},y_1) - T_f(y-z_1,z_2,\cdots,z_{k+1},y_1)$ and $T_f(y-y_1,z_2,z_3,\cdots,z_{k+1},z_1) - T_f(y-z_1,z_2,\cdots,z_{k+1},z_1)$ into four T_f with all parameters being independent and uniformly distributed[9]. Finally notice that the parameters in both $T_f(y-z_1,z_2,z_3,\cdots,z_{k+1},y_1)$ and $T_f(y-z_1,z_2,\cdots,z_{k+1},y_1)$ are independent and uniformly distributed. Further recall that by the definition of η_0, $\Pr_{r_1,\cdots,r_{k+1}}[T_f(r_1,\cdots,r_{k+1}) \neq 0] \leq \eta_0$ for independent and uniformly distributed r_i's. Thus, by the union bound, we have:

$$\Pr_{y_1,\cdots,y_{k+1},z_1,\cdots,z_{k+1}\in\mathbb{F}_3^n}[T_f(y_1,\cdots,y_{k+1}) - T_f(z_1,\cdots,z_{k+1}) \neq 0] \leq (4k+10)\eta_0 \leq (4k+14)\eta_0. \quad (15)$$

[9]Since $T_f(\cdot)$ is symmetric.

Therefore $\gamma \geq \mu \geq 1 - (4k+14)\eta_0$. A similar argument, modulo the following caveats, proves the Lemma for $g_1(y)$. $T_{f_1}(.)$ is not symmetric and needs some work. We use another identity as given in Claim 4.14 to resolve the issue and get four extra terms than in the case of g_0. In other words, the proof for $g_1(y)$ is same as the proof for $g_0(y)$ except it also needs Claim 4.14. ■

Analogously, for $\mathbb{F}_p$ we have:

Lemma 4.16 *For every $y \in \mathbb{F}_p^n$, $\Pr_{y_1,y_2,\cdots,y_{k+1}\in\mathbb{F}_p^n}[g_i(y) = f(y) - T_f^i(y - y_1, y_2, \cdots, y_{k+1}, y_1) + f(y)] \geq 1 - 2((p-1)k + 6(p-1) + 1)\eta_i$.*

The proof is similar to that of Lemma 4.6 where it can be shown $\mu_i \geq 1-2((p-1)k+6(p-1)+1)\eta_i$, for each μ_i defined for $g_i(y)$.

Remark 4.17 *Using Lemma 4.6, we can get a slightly stronger version of Lemma 4.4 following the proof of Lemma 2 in [AKK+05]. For a fixed function $f : \mathbb{F}_p^n \to \mathbb{F}_p$, let g_i, η_i be defined as in (10) and (11). Then, $\delta(f, g_i) \leq \min(2\eta_i, \frac{\eta_i}{1-2((p-1)k+6(p-1)+1)\eta_i})$.*

4.2.2 Proof of Lemma 4.7

From Theorem 3.1, it suffices to prove that if $\eta_i < \frac{1}{2(2k+7)3^{k+1}}$ then $T_{g_i}^i(y_1, \cdots, y_{k+1}, b) = 0$ for every $y_1, \cdots, y_{k+1}, b \in \mathbb{F}_3^n$. Fix the choice of $y_1, \cdots, y_{k+1}, b$. Define $Y = \langle y_1, \cdots, y_{k+1}\rangle$. We will express $T_{g_i}^i(Y, b)$ as the sum of $T_f^i(\cdot)$ with random arguments. We uniformly select $(k+1)^2$ random variables $z_{i,j}$ over $\mathbb{F}_3^n$ for $1 \leq i \leq k+1$, and $1 \leq j \leq k+1$. Define $Z_i = \langle z_{i,1}, \cdots, z_{i,k+1}\rangle$. We also select uniformly $(k+1)$ random variables r_i over $\mathbb{F}_3^n$ for $1 \leq i \leq k+1$. We use $z_{i,j}$ and r_i's to set up the random arguments. Now by Lemma 4.6, for every $I \in \mathbb{F}_3^{k+1}$ (i.e. think of I as an ordered $(k+1)$-tuple over $\{0,1,2\}$), with probability at least $1 - 2(2k+7)\eta_i$ over the choice of $z_{i,j}$ and r_i,

$$g_i(I \cdot Y + b) = f(I \cdot Y + b) - T_f^i(I \cdot Y + b - I \cdot Z_1 - r_1, I \cdot Z_2 + r_2, \cdots, I \cdot Z_{k+1} + r_{k+1}, I \cdot Z_1 + r_1), \quad (16)$$

where for vectors $X \in \mathbb{F}_3^{k+1}, Y \in (\mathbb{F}_3^n)^{k+1}$, we define $Y \cdot X \stackrel{\text{def}}{=} \sum_{i=1}^{k+1} Y_i X_i$, where the operations are over $\mathbb{F}_3^n$.

Let E_1 be the event that (16) holds for all $I \in \mathbb{F}_3^{k+1}$. By the union bound:

$$\Pr[E_1] \geq 1 - 3^{k+1} \cdot 2(2k+7)\eta_i. \quad (17)$$

Assume that E_1 holds. We now need the following claims. Let $J = \langle J_1, \cdots, J_{k+1}\rangle$ be a $(k+1)$ dimensional vector over $\mathbb{F}_3$, and denote $J' = \langle J_2, \cdots, J_{k+1}\rangle$.

Claim 4.18 *If (16) holds for all $I \in \mathbb{F}_3^{k+1}$, then*

$$\begin{aligned} T_{g_0}^0(Y, b) &= \sum_{0 \neq J' \in \mathbb{F}_3^k} \left[-T_f(y_1 + \sum_{t=2}^{k+1} J_t z_{t,1}, \cdots, y_{k+1} + \sum_{t=2}^{k+1} J_t z_{t,(k+1)}, b + \sum_{t=2}^{k+1} J_t r_t) \right] \\ &+ \sum_{J' \in \mathbb{F}_3^k} \left[-T_f(2y_1 - z_{1,1} + \sum_{t=2}^{k+1} J_t z_{t,1}, \cdots, 2y_{k+1} - z_{1,(k+1)} + \sum_{t=2}^{k+1} J_t z_{t,(k+1)}, 2b - r_1 + \sum_{t=2}^{k+1} J_t r_t) \right. \\ &+ \left. T_f(z_{1,1} + \sum_{t=2}^{k+1} J_t z_{t,1}, \cdots, z_{1,k+1} + \sum_{t=2}^{k+1} J_t z_{t,(k+1)}, r_1 + \sum_{t=2}^{k+1} J_t r_t) \right]. \quad (18) \end{aligned}$$

Claim 4.19 *If (16) holds for all $I \in \mathbb{F}_3^{k+1}$, then*

$$\begin{aligned} T^1_{g_1}(Y,b) &= \sum_{0\neq J'\in\mathbb{F}_3^k}\left[-T^1_f(y_1+\sum_{t=2}^{k+1}J_tz_{t,1},\cdots,y_{k+1}+\sum_{t=2}^{k+1}J_tz_{t,(k+1)},b+\sum_{t=2}^{k+1}J_tr_t)\right] \\ &+ \sum_{J'\in\mathbb{F}_3^k}\left[T^1_f(2y_1-z_{1,1}+\sum_{t=2}^{k+1}J_tz_{t,1},\cdots,2y_{k+1}-z_{1,(k+1)}+\sum_{t=2}^{k+1}J_tz_{t,(k+1)},2b-r_1+\sum_{t=2}^{k+1}J_tr_t)\right]. \end{aligned} \tag{19}$$

The proofs of Claim 4.18 and Claim 4.19 are deferred to the appendix. Let E_2 be the event that for every $J' \in \mathbb{F}_3^k, T^i_f(y_1+\sum_t J_tz_{t,1},\cdots,y_{k+1}+\sum_t J_tz_{t,(k+1)},b+\sum_{t=2}k+1J_tr_t)=0$, $T^i_f(2y_1-z_{1,1}+\sum_{t=2}^{k+1}J_tz_{t,1},\cdots,2y_{k+1}-z_{1,k+1}+\sum_{t=2}^{k+1}J_tz_{t,(k+1)},2b-r_1+\sum_{t=2}^{k+1}J_tr_t)=0$, and $T_f(z_{1,1}+\sum_{t=2}^{k+1}J_tz_{t,1},\cdots,z_{1,k+1}+\sum_{t=2}^{k+1}J_tz_{t,k+1},r_1+\sum_{t=2}^{k+1}J_tr_t)=0$. By the definition of η_i and the union bound, we have:

$$\Pr[E_2] \geq 1-3^{k+1}\eta_i. \tag{20}$$

Suppose that $\eta_i \leq \frac{1}{2(2k+7)3^{k+1}}$ holds. Then by (17) and (20), the probability that E_1 and E_2 hold is strictly positive. In other words, there exists a choice of the $z_{i,j}$'s and r_i's for which all summands in either Claim 4.18 or in Claim 4.19, whichever is appropriate, is 0. This implies that $T^i_{g_i}(y_1,\cdots,y_{k+1},b)=0$. In other words, if $\eta_i \leq \frac{1}{2(2k+7)3^{k+1}}$, then g_i belongs to $\mathcal{P}_t$. ■

Remark 4.20 *Over $\mathbb{F}_p$ we have: if $\eta_i < \frac{1}{2((p-1)k+6(p-1)+1)p^{k+1}}$, then g_i belongs to $\mathcal{P}_t$ (if $k \geq 1$).*

In case of $\mathbb{F}_p$, we can generalize (16) in a straightforward manner. Let E_1' denote the event that all such events holds. We can similarly obtain

$$\Pr[E_1'] \geq 1-p^{k+1}\cdot 2((p-1)k+6(p-1)+1)\eta_i. \tag{21}$$

Claim 4.21 *Assume equivalent of (16) holds for all $I \in \mathbb{F}_p^{k+1}$, then*[10]

$$\begin{aligned} T^i_{g_i}(Y,b) &= \sum_{0\neq J'\in\mathbb{F}_p^k}\left[-T^i_f(y_1+\sum_{t=2}^{k+1}J_tz_{t,1},\cdots,y_{k+1}+\sum_{t=2}^{k+1}J_tz_{t,(k+1)},b+\sum_{t=2}^{k+1}J_tr_t)\right] \\ &+ \sum_{J'\in\mathbb{F}_p^k}\left[\sum_{J_1\in\mathbb{F}_p;J_1\neq 1}J_1^i\left[-T^i_f(J_1y_1-(J_1-1)z_{1,1}+\sum_{t=2}^{k+1}J_tz_{t,1},\cdots,J_1y_{k+1}-(J_1-1)z_{1,(k+1)}\right.\right. \\ &\left.\left.+\sum_{t=2}^{k+1}J_tz_{t,(k+1)},J_1b-(J_1-1)r_1+\sum_{t=2}^{k+1}J_tr_t)\right]\right] \end{aligned} \tag{22}$$

Let E_2' be the event analogous to the event E_2 in Claim 4.19. Then by the definition of η_i and the union bound, we have

$$\Pr[E_2'] \geq 1-2p^{k+1}\eta_i. \tag{23}$$

Then if we are given that $\eta_i < \frac{1}{2((p-1)k+6(p-1)+1)p^{k+1}}$, then the probability that E_1' and E_2' hold is strictly positive. Therefore, this implies $T^i_{g_i}(y_1,\cdots,y_{k+1},b)=0$.

[10]Recall that we are using the convention $0^0=1$.

4.2.3 Proof of Lemma 4.8

For each $C \in \mathbb{F}_3^{k+1}$, let X_C be the indicator random variable whose value is 1 if and only if $f(C \cdot Y + b) \neq g(C \cdot Y + b)$. Clearly, $\Pr[X_C = 1] = \delta$ for every C. It follows that the random variable $X = \sum_C X_C$ which counts the number of points v of the required form in which $f(v) \neq g(v)$ has expectation $\mathrm{E}[X] = 3^{k+1}\delta = \ell \cdot \delta$. It is not difficult to check that the random variables X_C are pairwise independent, since for any two distinct C_1 and C_2, the sums $\sum_{i=1}^{k+1} C_{1,i}+b$ and $\sum_{i=1}^{k+1} C_{2,i}+b$ attain each pair of distinct values in $\mathbb{F}_3^n$ with equal probability when the vectors are chosen randomly and independently. Since X_C's are pairwise independent, $\mathrm{Var}[X] = \sum_C \mathrm{Var}[X_C]$. Since X_C's are boolean random variables, we note

$$\mathrm{Var}[X_C] = \mathrm{E}[X_C^2] - (\mathrm{E}[X_C])^2 = \mathrm{E}[X_C] - (\mathrm{E}[X_C])^2 \leq \mathrm{E}[X_C].$$

Thus we obtain $\mathrm{Var}[X] \leq \mathrm{E}[X]$, so $\mathrm{E}[X^2] \leq \mathrm{E}[X]^2 + \mathrm{E}[X]$. Next we use the following inequality from [AKK$^+$05] which holds for a random variable X taking nonnegative, integer values,

$$\Pr[X > 0] \geq \frac{(\mathrm{E}[X])^2}{\mathrm{E}[X^2]}.$$

In our case, this implies

$$\Pr[X > 0] \geq \frac{(\mathrm{E}[X])^2}{\mathrm{E}[X^2]} \geq \frac{(\mathrm{E}[X])^2}{\mathrm{E}[X] + (\mathrm{E}[X])^2} = \frac{\mathrm{E}[X]}{1 + \mathrm{E}[X]}.$$

Therefore,

$$\mathrm{E}[X] \;\geq\; \Pr[X = 1]+2\Pr[X \geq 2] \;=\; \Pr[X = 1]+2\left(\frac{\mathrm{E}[X]}{1 + \mathrm{E}[X]} - \Pr[X = 1]\right) = \frac{2\mathrm{E}[X]}{1 + \mathrm{E}[X]} - \Pr[X = 1].$$

After simplification we obtain,

$$\Pr[X = 1] \geq \frac{1 - \mathrm{E}[X]}{1 + \mathrm{E}[X]} \cdot \mathrm{E}[X].$$

The proof is complete by recalling that $\mathrm{E}[X] = \ell \cdot \delta$. ■

5 A Lower Bound and Improved Self-correction

5.1 A Lower Bound

The next theorem is a simple modification of a theorem in [AKK$^+$05] and essentially implies that our result is almost optimal.

Proposition 5.1 *Let $\mathcal{F}$ be any family of functions $f : \mathbb{F}_p^n \to \mathbb{F}_p$ that corresponds to a linear code $\mathcal{C}$. Let d denote the minimum distance of the code $\mathcal{C}$ and let $\bar{d}$ denote the minimum distance of the dual code of $\mathcal{C}$.*
Every one-sided testing algorithm for the family $\mathcal{F}$ must perform $\Omega(\bar{d})$ queries, and if the distance parameter ϵ is at most d/p^{n+1}, then $\Omega(1/\epsilon)$ is also a lower bound for the necessary number of queries.

Lemma 3.2 and Proposition 5.1 gives us the following corollary.

Corollary 5.2 *Every one-sided tester for testing $\mathcal{P}_t$ with distance parameter ϵ must perform $\Omega(\max(\frac{1}{\epsilon}, (1+((t+1) \mod (p-1)))p^{\frac{t+1}{p-1}}))$ queries.*

5.2 Improved Self-correction

The following corollary follows from Theorem 3.1 and an application of the union bound.

Corollary 5.3 *Consider a function $f : \mathbb{F}_3^n \to \mathbb{F}_3$ that is ϵ-close to a degree-t polynomial $g : \mathbb{F}_3^n \to \mathbb{F}_3$, where $\epsilon < \frac{2}{3^{k+2}}$. (Assume $k \geq 1$.) Then the function f can be self-corrected. That is, for any given $x \in \mathbb{F}_3^n$, it is possible to obtain the value $g(x)$ with probability at least $1 - 3^{k+1}\epsilon$ by querying f on $3^{k+1} - 1$ points on $\mathbb{F}_3^n$.*

An analogous result may be obtained for the general case. If $\epsilon < \frac{1}{2 \cdot p^{k+1}}$, then repeating the above s number of times and taking the plurality, one can retrieve the correct value with probability at least $1 - 2^{-\Omega(s)}$ (this follows from the Chernoff bound). The query complexity is sp^{k+1}. Further, note that the corrector uses $\Theta(k \cdot s \cdot n \log p)$ many random bits.

Recall that the relative distance of the GRM code is $\delta = (1 - R/p)p^{-k}$ where $t = (p-1) \cdot k + R$ and $0 \leq R \leq (p-2)$. Thus, the self-corrector discussed above does *not* work for $\epsilon \geq \delta/4$. Below, we show how to locally self-correct from error rate $\delta/2 - \epsilon'$ for any arbitrary $\epsilon' > 0$. Further, the result below can also use less amounts of randomness than the self-corrector above for certain setting of parameter. For example, if one is shooting for a success probability of $1 - p^{-k}$, ϵ is polynomial in p^{-k} and ϵ' is polynomially related to ϵ; then in the first case we will need $\Theta(k^2 n \log^2 p)$ many random bits (as we need $s = \Theta(k \log p)$) while the corrector below uses $\Theta(kn \log^2 p)$ many random bits (as $K = \Theta(k \log p)$).

The improvement comes from the following observation. The corrector above does not allow any error in the p^{k+1} points it queries. We obtain a stronger result by querying on a slightly larger flat H, but allowing more errors. Errors are handled by decoding the induced Generalized Reed-Muller code on H.

Proposition 5.4 *Consider a function $f : \mathbb{F}_p^n \to \mathbb{F}_p$ that is ϵ-close, where $\epsilon < \delta/2 - \epsilon'$ for any $\delta/2 >> \epsilon' > 0$, to a degree-$t$ polynomial $g : \mathbb{F}_p^n \to \mathbb{F}_p$. Then the function f can be self-corrected. That is, assume $K > 1 + \log\left(\frac{\epsilon}{(\epsilon')^2}\right)$, then for any given $x \in \mathbb{F}_p^n$, the value of $g(x)$ can be obtained with probability at least $1 - \frac{\epsilon}{(\epsilon')^2} \cdot p^{-(K-1)}$ with p^K queries to f.*

Proof: Our goal is to correct the f at the point x. Recall that our local tester requires a $(k+1)$-flat, i.e., it tests $\sum_{c_1,\cdots,c_{k+1} \in \mathbb{F}_p} c_1^{p-2-R} f(y_0 + \sum_{i=1}^{k+1} c_i y_i) = 0$, where $y_i \in \mathbb{F}_p^n$.

We choose a slightly larger flat, i.e., a K-flat with $K > 1 + \log(\epsilon/(\epsilon')^2)$. We consider the code restricted to this K-flat with point x being the origin. We query f on this K-flat. It is known that a majority logic decoding algorithm exists that can decode Generalized Reed-Muller code up to half the minimum distance for any choice of parameters (see [Sud01]). Thus if the number of error is small we can recover $g(x)$. We now present the details.

Let the relative distance of f from $\mathrm{GRM}_p(t, m)$ be ϵ and let S be the set of points where it disagrees with the closest codeword. Consider a K-flat $H = \{x + \sum_{i=1}^{K} t_i u_i | t_i \in \mathbb{F}, u_i \in_R \mathbb{F}_p^n\}$. Let $D = \mathbb{F}^K \setminus \{\mathbf{0}\}$ and $U = \langle u_1, \cdots, u_K \rangle$. Let the indicator variable $Y_{U,\langle t_1,\cdots,t_K\rangle}$ take the value 1 if $x + \sum_{i=1}^{K} u_i t_i \in S$ and 0 otherwise. Define $Y_U = \sum_{\langle t_1,\cdots,t_K\rangle \in D} Y_{U,\langle t_1,\cdots,t_K\rangle}$ and $\ell = (p^K - 1)$. We would like to bound the probability

$$\Pr_U[Y_U \geq \ell \cdot \delta/2] \leq \Pr_U[|Y_U - \epsilon\ell| \geq \epsilon'\ell].$$

Since $\Pr_U[Y_{U,\langle t_1,\cdots,t_K\rangle} = 1] = \epsilon$, by linearity of expectation, we get $\mathrm{E}_U[Y_U] = \epsilon\ell$. Let $T = \langle t_1, \cdots, t_K\rangle$. Since U will be clear from context, we drop it from the subscripts. Now

$$\begin{aligned} Var[Y] &= \sum_{T\in D} Var[Y_T] + \sum_{T\neq T'} Cov[Y_T, Y_{T'}] \\ &= \ell(\epsilon-\epsilon^2) + \sum_{T\neq\lambda T'} Cov[Y_T, Y_{T'}] \\ &\quad + \sum_{T=\lambda T';1\neq\lambda\in\mathbb{F}^*} Cov[Y_T, Y_{T'}] \\ &\leq \ell(\epsilon-\epsilon^2) + \ell\cdot(p-2)(\epsilon-\epsilon^2) \\ &= \ell(\epsilon-\epsilon^2)(p-1) \end{aligned}$$

The above follows from the fact that when $T \neq \lambda T'$ then the corresponding events Y_T and $Y_{T'}$ are almost independent, and in fact $Cov[Y_T, Y_{T'}] = -(\epsilon-\epsilon^2)/l$. Also, when $T = \lambda T'$, Y_T and $Y_{T'}$ may be dependent. Nevertheless, $Cov[Y_T, Y_{T'}] = \mathrm{E}_U[Y_T Y_{T'}] - \mathrm{E}_U[Y_T]\mathrm{E}_U[Y_{T'}] \leq \epsilon - \epsilon^2$.
Therefore, by Chebyshev's inequality we have

$$\Pr_U[|Y - \epsilon\ell| \geq \epsilon'\ell] \leq \frac{\ell\epsilon(1-\epsilon)(p-1)}{(\epsilon')^2\ell^2}$$

We thus have

$$\begin{aligned} \Pr_U[|Y - \epsilon\ell| \geq \epsilon'\ell] &\leq \frac{\epsilon p}{(\epsilon')^2(\ell+1)} \\ &= \frac{\epsilon}{(\epsilon')^2}\cdot p^{-(K-1)}. \end{aligned}$$

Thus with probability at least $1 - \frac{\epsilon}{(\epsilon')^2}\cdot p^{-(K-1)}$ the function can be self-corrected at x. ■

6 Conclusions

The lower bound in Corollary 5.2 implies that our upper bound is almost tight. We resolved the question posed in [AKK+05] for all prime fields. Independently in [KR06] the question has been resolved for all fields. We mention that later we found an alternate proof of their characterization of polynomials over arbitrary finite fields.

Recently there has been some interest in *tolerant testing*. In our setting this requires a tester to also accept received words that are "close" to some codeword in addition to rejecting received words that are "far" away received words [GR05]. Note that the "standard" testers (such as those considered in this paper) satisfy the second requirement but are only required to satisfy the first requirement when there is no error. Our work unfortunately does not imply anything non-trivial about tolerant testing of GRM codes. However, designing a "standard" tester that either has the optimal query complexity or is the so-called "robust" tester (cf. [BSS06]), will by the simple observations in [GR05], imply a tolerant tester for GRM codes. We remark that there exists robust testers for GRM codes when the degree parameter is smaller than the alphabet size, which in turn implies a tolerant tester for such codes [GR05]. However, the problem of designing a tolerant tester for GRM codes over all alphabets is still open.

Kaufman and Litsyn ([KL05]) have shown that the dual of BCH codes are locally testable (this result was later extended by Kaufman and Sudan to hold for a larger set of "sparse" codes [KS07]). They also give a sufficient condition for a code to be locally testable. The condition roughly says that if the number of fixed length codewords in the dual of the union of the code and its ϵ-far coset is suitably smaller than the same in the dual of the code, then the code is locally testable. Their argument is more combinatorial in nature and needs the knowledge of weight-distribution of the code and thus differs from the self-correction approach used in this work.

Alon et al. [AKK$^+$05], made the following general conjecture. The conjecture claims that any linear code with small (constant) dual distance that has a doubly transitive group acting on the co-ordinates of the codewords mapping the dual code to itself, is locally testable. [KL05] resolved this conjecture in the affirmative for the dual BCH codes. However the general conjecture was very recently shown to be false by Grigorescu, Kaufman and Sudan [GKS08].

6.1 Acknowledgment

The second author wishes to thank Felipe Voloch for motivating discussions in an early stage of the work. We thank the anonymous reviewers for several useful comments.

References

[AJK98] E. F. Assumus Jr. and J. D. Key. *Polynomial codes and Finite Geometries in Handbook of Coding Theory, Vol II , Edited by V. S. Pless Jr., and W. C. Huffman*, chapter 16. Elsevier, 1998.

[AKK$^+$05] N. Alon, T. Kaufman, M. Krivelevich, S. Litsyn, and D. Ron. Testing Reed-Muller codes. *IEEE Transactions on Information Theory*, 51(11):4032–4039, November 2005. Preliminary version "Testing Low-Degree Polynomials over $GF(2)$" appeared in RANDOM 2003.

[AKNS99] N. Alon, M. Krivelevich, I. Newman, and M. Szegedy. Regular languages are testable with a constant number of queries. In *Proc. of Fortieth Annual Symposium on Foundations of Computer Science*, pages 645–655, 1999.

[ALM$^+$98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the intractibility of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998.

[AS03] Sanjeev Arora and Madhu Sudan. Improved low-degree testing and its applications. *Combinatorica*, 23(3):365–426, 2003.

[BFL91] L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two prover interactive protocols. In *Computational Complexity*, pages 3–40, 1991.

[BFLS91] L. Babai, L. Fortnow, L. Levin, and M. Szegedy. Checking computations in polylogarithmic time. In *Proc. of Symposium on the Theory of Computing*, pages 21–31, 1991.

[BLR93] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47:549–595, 1993.

[BSHR05] Eli Ben-Sasson, Prahladh Harsha, and Sofya Raskhodnikova. Some 3CNF properties are hard to test. *SIAM Journal on Computing*, 35(1):1–21, 2005.

[BSS06] Eli Ben-Sasson and Madhu Sudan. Robust locally testable codes and products of codes. *Random Structures and Algorithms*, 28(4):387–402, 2006.

[Coh87] S. D. Cohen. Functions and polynomials in vector spaces. *Archiv der Mathematik*, IT-14(2):409–419, March 1987.

[DGM70] P. Delsarte, J. M. Goethals, and F. J. MacWillams. On generalized Reed-Muller codes and their relatives. *Information and Control*, 16:403–442, 1970.

[DK00] P. Ding and J. D. Key. Minimum-weight codewords as generators of generalized Reed-Muller codes. *IEEE Trans. on Information Theory.*, 46:2152–2158, 2000.

[FGL+96] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *Journal of the ACM*, 43(2):268–292, 1996.

[FS95] K. Friedl and M. Sudan. Some improvements to total degree tests. In *Proceedings of the 3rd Annual Israel symposium on Theory of Computing and Systems*, pages 190–198, 1995. Corrected version available at http://theory.lcs.mit.edu/∼madhu/papers/friedl.ps.

[GKS08] Elena Grigorescu, Tali Kaufman, and Madhu Sudan. 2-transitivity is insufficient for local testability. In *Proceedings of the 23rd IEEE Conference on Computational Complexity (CCC)*, 2008. To Appear.

[GLR+91] P. Gemmell, R. Lipton, R. Rubinfeld, M. Sudan, and A. Wigderson. Self-testing/correcting for polynomials and for approxiamte functions. In *Proc. of Symposium on the Theory of Computing*, 1991.

[GR05] Venkatesan Guruswami and Atri Rudra. Tolerant locally testable codes. In *Proceedings of the 9th International Workshop on Randomization and Computation (RANDOM)*, pages 306–317, 2005.

[JPR04] C. S. Jutla, A. C. Patthak, and A. Rudra. Testing polynomials over general fields. manuscript, 2004.

[KL05] T. Kaufman and S. Litsyn. Almost orthogonal linear codes are locally testable. In *To appear in Proc. of IEEE Symposium of the Foundation of Computer Science*, 2005.

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

[KS07] Tali Kaufman and Madhu Sudan. Sparse random linear codes are locally decodable and testable. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 590–600, 2007.

[Pat07] A. C. Patthak. *Error-correcting Codes : Local Testing, List Decoding, and Applications.* PhD thesis, University of Texas at Austin, 2007.

[RS96] R. Rubinfeld and M. Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

[Sud01] M. Sudan. Lecture notes on algorithmic introduction to coding theory, Fall 2001. Lecture 15.

A Omitted Proofs from Section 4

Proof of Lemma 4.5: We will use I, J, I', J' to denote $(k+1)$ dimensional vectors over $\mathbb{F}_p$. Now note that

$$\begin{aligned} g_i(y) &= \text{Plurality}_{y_1,\cdots,y_{k+1}\in\mathbb{F}_p^n}[-\sum_{I\in\mathbb{F}_p^{k+1};I\neq\langle 1,0,\cdots,0\rangle} I_1^i f(I_1(y-y_1)+\sum_{t=2}^{k+1} I_t y_t + y_1)] \\ &= \text{Plurality}_{y-y_1,y_2,\cdots,y_{k+1}\in\mathbb{F}_p^n}[-\sum_{I\in\mathbb{F}_p^{k+1};I\neq\langle 0,\cdots,0\rangle} (I_1+1)^i f(I_1(y-y_1)+\sum_{t=2}^{k+1} I_t y_t + y)] \\ &= \text{Plurality}_{y_1,\cdots,y_{k+1}\in\mathbb{F}_p^n}[-\sum_{I\in\mathbb{F}_p^{k+1};I\neq\langle 0,\cdots,0\rangle} (I_1+1)^i f(\sum_{t=1}^{k+1} I_t y_t + y)] \end{aligned} \tag{24}$$

Let $Y = \langle y_1, \cdots, y_{k+1}\rangle$ and $Y' = \langle y'_1, \cdots, y'_{k+1}\rangle$. Also we will denote $\langle 0, \cdots, 0\rangle$ by $\vec{0}$. Now note that

$$\begin{aligned} 1-\eta_i &\leq \text{Pr}_{y_1,\cdots,y_{k+1},b}[T_f^i(y_1,\cdots,y_{k+1},b)=0] \qquad (25) \\ &= \text{Pr}_{y_1,\cdots,y_{k+1},b}[\sum_{I\in\mathbb{F}_p^{k+1}} I_1^i f(b+I\cdot Y)=0] \\ &= \text{Pr}_{y_1,\cdots,y_{k+1},b}[f(b+y_1)+\sum_{I\in\mathbb{F}_p^{k+1};I\neq\langle 1,0,\cdots,0\rangle} I_1^i f(b+I\cdot Y)=0] \\ &= \text{Pr}_{y_1,\cdots,y_{k+1},y}[f(y)+\sum_{I\in\mathbb{F}_p^{k+1};I\neq\langle 1,0,\cdots,0\rangle} I_1^i f(y-y_1+I\cdot Y)=0] \\ &= \text{Pr}_{y_1,\cdots,y_{k+1},y}[f(y)+\sum_{I\in\mathbb{F}_p^{k+1};I\neq\langle 0,\cdots,0\rangle} (I_1+1)^i f(y+I\cdot Y)=0] \qquad (26) \end{aligned}$$

Therefore for any given $I \neq \vec{0}$ we have the following:

$$\text{Pr}_{Y,Y'}[f(y+I\cdot Y) = \sum_{J\in\mathbb{F}_p^{k+1};J\neq\vec{0}} -(J_1+1)^i f(y+I\cdot Y+J\cdot Y')] \geq 1-\eta_i$$

and for any given $J \neq \vec{0}$,

$$\text{Pr}_{Y,Y'}[f(y+J\cdot Y') = \sum_{I\in\mathbb{F}_p^{k+1};I\neq\vec{0}} -(I_1+1)^i f(y+I\cdot Y+J\cdot Y')] \geq 1-\eta_i.$$

Combining the above two and using the union bound we get,

$$\begin{aligned}
\Pr_{Y,Y'}[\sum_{I\in\mathbb{F}_p^{k+1};I\neq\vec{0}}(I_1+1)^i f(y+I\cdot Y) &= \sum_{I\in\mathbb{F}_p^{k+1};I\neq\vec{0}}\sum_{J\in\mathbb{F}_p^{k+1};J\neq\vec{0}} -(I_1+1)^i(J_1+1)^i f(y+I\cdot Y+J\cdot Y') \\
&= \sum_{J\in\mathbb{F}_p^{k+1};J\neq\vec{0}}(J_1+1)^i f(y+J\cdot Y')] \\
&\geq 1-2(p^{k+1}-1)\eta \geq 1-2p^{k+1}\eta_i \qquad (27)
\end{aligned}$$

The lemma now follows from the observation that the probability that the same object is drawn from a set in two independent trials lower bounds the probability of drawing the most likely object in one trial: Suppose the objects are ordered so that p_i is the probability of drawing object i, and $p_1 \geq p_2 \geq \cdots$. Then the probability of drawing the same object twice is $\sum_i p_i^2 \leq \sum_i p_1 p_i \leq p_1$. ■

Proof of Claim 4.18:

$$\begin{aligned}
T_g(Y,b) &= \sum_{I\in\mathbb{F}_3^{k+1}} g(I\cdot Y+b) \\
&= \sum_{I\in\mathbb{F}_3^{k+1}} [-T_f(I\cdot Y+b-I\cdot Z_1-r_1, I\cdot Z_2+r_2,\cdots,I\cdot Z_{k+1}+r_{k+1}, I\cdot Z_1+r_1) \\
&\quad +f(I\cdot Y+b)] \\
&= -\sum_{I\in\mathbb{F}_3^{k+1}}\left[\left[\sum_{\emptyset\neq J'\in\mathbb{F}_3^k} f(I\cdot Y+b+\sum_{t=2}^{k+1}J_t I\cdot Z_t+\sum_{t=2}^{k+1}J_t r_t)\right]\right. \\
&\quad +\left[\sum_{J'\in F_3^k}\left(f(2I\cdot Y+2b-I\cdot Z_1-r_1+\sum_{t=2}^{k+1}J_t I\cdot Z_t+\sum_{t=2}^{k+1}J_t r_t)\right.\right. \\
&\quad \left.\left.\left.+f(I\cdot Z_1+r_1+\sum_{t=2}^{k+1}J_t I\cdot Z_t+\sum_{t=2}^{k+1}J_t r_t)\right)\right]\right] \\
&= -\sum_{0\neq J'\in\mathbb{F}_3^k}\left[\sum_{I\in\mathbb{F}_3^{k+1}} f(I\cdot Y+b+\sum_{t=2}^{k+1}J_t r_t+\sum_{t=2}^{k+1}J_t I\cdot Z_t)\right] \\
&\quad -\sum_{J'\in\mathbb{F}_3^k}\left[\left[\sum_{I\in\mathbb{F}_3^{k+1}} f(2I\cdot Y+2b-I\cdot Z_1-r_1+\sum_{t=2}^{k+1}J_t I\cdot Z_t+\sum_{t=2}^{k+1}J_t r_t)\right]\right. \\
&\quad \left.+\left[\sum_{I\in\mathbb{F}_3^{k+1}} f(I\cdot Z_1+r_1+\sum_{t=2}^{k+1}J_t I\cdot Z_t+\sum_{t=2}^{k+1}J_t r_t)\right]\right] \\
&= \sum_{0\neq J'\in\mathbb{F}_3^k}\left[-T_f(y_1+\sum_{t=2}^{k+1}J_t z_{t,1},\cdots,y_{k+1}+\sum_{t=2}^{k+1}J_t z_{t,(k+1)}, b+\sum_{t=2}^{k+1}J_t r_t)\right]
\end{aligned}$$

$$+ \quad \sum_{J' \in \mathbb{F}_3^k} \Bigg[-T_f(2y_1 - z_{1,1} + \sum_{t=2}^{k+1} J_t z_{t,1}, \cdots, 2y_{k+1} - z_{1,(k+1)} + \sum_{t=2}^{k+1} J_t z_{t,(k+1)}, 2b - r_1 + \sum_{t=2}^{k+1} J_t r_t)$$

$$+ \quad T_f(z_{1,1} + \sum_{t=2}^{k+1} J_t z_{t,1}, \cdots, z_{1,k+1} + \sum_{t=2}^{k+1} J_t z_{t,(k+1)}, r_1 + \sum_{t=2}^{k+1} J_t r_t) \Bigg] \tag{28}$$

■

Proof of Claim 4.19:

$$\begin{aligned}
T^1_{g_1}(Y,b) &= \sum_{I \in \mathbb{F}_3^{k+1}} I_1 g_1(I \cdot Y + b) \\
&= \sum_{I \in \mathbb{F}_3^{k+1}} I_1 \left[-T^1_f(I \cdot Y + b - I \cdot Z_1 - r_1, I \cdot Z_2 + r_2, \cdots, I \cdot Z_{k+1} + r_{k+1}, I \cdot Z_1 + r_1) \right. \\
&\qquad \left. + f(I \cdot Y + b) \right] \\
&= - \sum_{I \in \mathbb{F}_3^{k+1}} I_1 \left[\left[\sum_{\emptyset \neq J' \in \mathbb{F}_3^k} f(I \cdot Y + b + \sum_{t=2}^{k+1} J_t I \cdot Z_t + \sum_{t=2}^{k+1} J_t r_t) \right] \right. \\
&\qquad \left. + \left[\sum_{J' \in F_3^k} f(2I \cdot Y + 2b - I \cdot Z_1 - r_1 + \sum_{t=2}^{k+1} J_t I \cdot Z_t + \sum_{t=2}^{k+1} J_t r_t) \right] \right] \\
&= - \sum_{0 \neq J' \in \mathbb{F}_3^k} \left[\sum_{I \in \mathbb{F}_3^{k+1}} I_1 f(I \cdot Y + b + \sum_{t=2}^{k+1} J_t r_t + \sum_{t=2}^{k+1} J_t I \cdot Z_t) \right] \\
&\qquad - \sum_{J' \in \mathbb{F}_3^k} \left[\sum_{I \in \mathbb{F}_3^{k+1}} I_1 f(2I \cdot Y + 2b - I \cdot Z_1 - r_1 + \sum_{t=2}^{k+1} J_t I \cdot Z_t + \sum_{t=2}^{k+1} J_t r_t) \right] \\
&= \sum_{0 \neq J' \in \mathbb{F}_3^k} \left[-T^1_f(y_1 + \sum_{t=2}^{k+1} J_t z_{t,1}, \cdots, y_{k+1} + \sum_{t=2}^{k+1} J_t z_{t,(k+1)}, b + \sum_{t=2}^{k+1} J_t r_t) \right] \\
&+ \quad \sum_{J' \in \mathbb{F}_3^k} \left[T^1_f(2y_1 - z_{1,1} + \sum_{t=2}^{k+1} J_t z_{t,1}, \cdots, 2y_{k+1} - z_{1,(k+1)} + \sum_{t=2}^{k+1} J_t z_{t,(k+1)}, 2b - r_1 + \sum_{t=2}^{k+1} J_t r_t) \right]
\end{aligned} \tag{29}$$

■

Proof of Claim 4.21:

$$\begin{aligned}
T^i_{g_i}(Y,b) &= \sum_{I \in \mathbb{F}_p^{k+1}} I^i_1 g_i(I \cdot Y + b) \\
&= \sum_{I \in \mathbb{F}_p^{k+1}} I^i_1 \left[-T^i_f(I \cdot Y + b - I \cdot Z_1 - r_1, I \cdot Z_2 + r_2, \cdots, I \cdot Z_{k+1} + r_{k+1}, I \cdot Z_1 + r_1) \right.
\end{aligned}$$

$$
\begin{aligned}
& + f(I \cdot Y + b)] \\
= & - \sum_{I \in \mathbb{F}_p^{k+1}} I_1^i \left[\left[\sum_{\emptyset \neq J' \in \mathbb{F}_p^k} f(I \cdot Y + b + \sum_{t=2}^{k+1} J_t I \cdot Z_t + \sum_{t=2}^{k+1} J_t r_t) \right] \right. \\
+ & \left. \left[\sum_{J_1 \in \mathbb{F}_p, J_1 \neq 1} J_1^i \left[\sum_{J' \in \mathbb{F}_p^k} f(J_1 I \cdot Y + J_1 b - (J_1 - 1) I \cdot Z_1 - (J_1 - 1) r_1 + \sum_{t=2}^{k+1} J_t I \cdot Z_t + \sum_{t=2}^{k+1} J_t r_t) \right] \right] \right] \\
= & - \sum_{0 \neq J' \in \mathbb{F}_p^k} \left[\sum_{I \in \mathbb{F}_p^{k+1}} I_1^i f(I \cdot Y + b + \sum_{t=2}^{k+1} J_t r_t + \sum_{t=2}^{k+1} J_t I \cdot Z_t) \right] \\
- & \sum_{J' \in \mathbb{F}_p^k} \left[\sum_{J_1 \in \mathbb{F}_p; J_1 \neq 1} J_1^i \left[\sum_{I \in \mathbb{F}_p^{k+1}} I_1^i f(J_1 I \cdot Y + J_1 b - (J_1 - 1) I \cdot Z_1 - (J_1 - 1) r_1 \right. \right. \\
& \left. \left. + \sum_{t=2}^{k+1} J_t I \cdot Z_t + \sum_{t=2}^{k+1} J_t r_t) \right] \right] \\
= & \sum_{0 \neq J' \in \mathbb{F}_p^k} \left[-T_f^i (y_1 + \sum_{t=2}^{k+1} J_t z_{t,1}, \cdots, y_{k+1} + \sum_{t=2}^{k+1} J_t z_{t,(k+1)}, b + \sum_{t=2}^{k+1} J_t r_t) \right] \\
+ & \sum_{J' \in \mathbb{F}_p^k} \left[\sum_{J_1 \in \mathbb{F}_p; J_1 \neq 1} J_1^i \left[-T_f^i (J_1 y_1 - (J_1 - 1) z_{1,1} + \sum_{t=2}^{k+1} J_t z_{t,1}, \cdots, J_1 y_{k+1} - (J_1 - 1) z_{1,(k+1)} \right. \right. \\
& \left. \left. + \sum_{t=2}^{k+1} J_t z_{t,(k+1)}, J_1 b - (J_1 - 1) r_1 + \sum_{t=2}^{k+1} J_t r_t) \right] \right]
\end{aligned} \tag{30}
$$

■