A Fourier-analytic approach to Reed-Muller decoding

Parikshit Gopalan
MSR-Silicon Valley.
Mountain View, CA.
parik@microsoft.com

***Abstract*—We present a Fourier-analytic approach to list-decoding Reed-Muller codes over arbitrary finite fields. We use this to show that quadratic forms over any field are locally list-decodeable up to their minimum distance. The analogous statement for linear polynomials was proved in the celebrated works of Goldreich-Levin [1] and Goldreich-Rubinfeld-Sudan [2]. Previously, tight bounds for quadratic polynomials were known only for $q = 2$ or 3 [3]; the best bound known for other fields was the Johnson radius.**

Departing from previous work on Reed-Muller decoding which relies on some form of self- corrector [2]–[5], our work applies ideas from Fourier analysis of Boolean functions to low-degree polynomials over finite fields, in conjunction with results about the weight- distribution. We believe that the techniques used here could find other applications, we present some applications to testing and learning.

***Keywords*-Polynomials; error-correcting codes; Reed-Muller codes; Fourier analysis.**

I. Introduction

Traditional algorithms to decode error-correcting codes require that the received word is within less than half the minimum distance of a codeword, so that the codeword can be uniquely recovered. In the 1950s, Elias [6] and Wozencraft [7] introduced the notion of list-decoding in order to decode beyond this barrier. Rather than returning a single codeword, a list-decoding algorithm outputs all codewords within a specified radius of a received word. It took over thirty years before Goldreich and Levin [1] and Sudan [8] gave efficient list-decoding algorithms for Hadamard codes and Reed-Solomon codes, respectively. Since these breakthroughs, there has been much progress in devising list-decoders for various codes [9], [10]. Indeed, list-decoding algorithms are the only tools that we have for solving the nearest codeword problem beyond half the minimum distance in the adversarial error model.

Algorithms for list-decoding error-correcting codes have proved tremendously useful in computer science (see [9, Chapter 12] and references therein), with applications ranging from hardness amplification for weakly hard functions, constructions of hard-core predicates from any one-way function, constructions of extractors and pseudorandom generators and the average-case hardness of the permanent. Despite the considerable progress in this area, for several natural and well-studied families of codes including Reed-Solomon and Reed-Muller codes, the *list-decoding* radius, or the largest error radius up to which the list-decoding problem is tractable is as yet unknown. This problem for Reed-Muller codes is the focus of our paper.

Reed-Muller codes (RM codes for short) were discovered by Muller in 1954. The message space of the code $\mathsf{RM}_q(n, d)$ consists of all degree d polynomials in n variables over $\mathbb{F}_q$, the codewords are the evaluations of these polynomials at all points in $\mathbb{F}_q^n$. Let $\delta_q(d)$ denote the normalized minimum distance of $\mathsf{RM}_q(n, d)$. If $d = a(q-1)+b$ where $0 \leqslant b \leqslant q-1$, then

$$\delta_q(d) = \frac{1}{q^a}\left(1 - \frac{b}{q}\right). \tag{1}$$

The case when $d < q$ is the famous Schwartz-Zippel lemma.

Reed-Muller codes are one of the most well-studied families of error-correcting codes in coding theory [11]. They are ubiquitous in computer science, indeed several of the aforementioned applications of list-decoding use Reed-Muller codes. A closely related problem is that of low-degree testing, where we are given a function and asked to test if it is close to a codeword in the Reed-Muller code. This is a problem that has been studied extensively in computer science, and plays in a key role in the (original) proof of the celebrated PCP theorem.

For most applications above, the model of interest is the local-decoding model where we are given an oracle for the received word $R : \mathbb{F}_q^n \to \mathbb{F}_q$ that can be queried at chosen points. The goal is to devise an algorithm whose running time is polynomial in the size of the message (rather than the codeword). The message being a degree d polynomial (d will be constant) in n variables over $\mathbb{F}_q$, our goal is to run in time $\mathrm{poly}(n)$. So we are interested in the settings where the list-size is a constant, or at worst $\mathrm{poly}(n)$. Our running times are typically polynomial in q.

A. Previous Work

For (a family of) codes $\mathcal{C} \subset [q]^n$, let $\ell(\mathcal{C}, \eta)$ denote the maximum list-size at radius η (radius $\eta \in [0, 1]$ denotes normalized Hamming distance). $\mathsf{LDR}(\mathcal{C})$ is the largest η for which $\ell(\mathcal{C}, \eta - \varepsilon)$ can be bounded by a function of ε (independent of n) for every $\varepsilon > 0$.

The study of list-decoding algorithms for Reed-Muller codes was initiated by the seminal work of Goldreich and Levin on list-decoding Hadamard codes over $\mathbb{F}_2$

or equivalently $\mathsf{RM}_2(n,1)$ codes [1]. They showed that $\mathsf{LDR}(\mathsf{RM}_2(n,1)) = 1/2$. Goldreich, Rubinfeld and Sudan generalized this to Hadamard codes over $\mathbb{F}_q$, showing that $\mathsf{LDR}(\mathsf{RM}_q(n,1)) = 1 - 1/q$ [2]. An important development was the discovery of powerful algorithms for list-decoding univariate polynomials over $\mathbb{F}_q$, due to Sudan [8] and Guruswami and Sudan [12]. Sudan, Trevisan and Vadhan used these algorithms to devise a list-decoder that works up to radius $1 - \sqrt{2d/q}$ for [5], improving on work by Arora and Sudan [4] and Goldreich *et al.* [2] (see also [13]).

All of the aforementioned decoding algorithms reach a coding theoretic bound known as the Johnson bound [14], [15]. The Johnson bound guarantees that for *any* code of minimum distance δ over $\mathbb{F}_q$, $\mathsf{LDR}(\mathcal{C}) \geqslant \mathsf{J_q}(\delta) = (1 - 1/q)(1 - \sqrt{1 - q\delta/(q-1)})$. Since the Johnson bound is oblivious to the structure of the code apart from its minimum distance, one does not expect it to be tight for every code, yet examples of codes decodeable beyond the Johnson bound are relatively few and recent (see the discussion in [3], [16]). A tantalizing open problem in this area is whether the Johnson bound is tight for Reed-Solomon codes, this is precisely the radius achieved by the Guruswami-Sudan algorithm [12].

Recently, Gopalan, Klivans and Zuckerman (GKZ) considered the problem of list-decoding Reed-Muller codes over $\mathbb{F}_2$ [3]. They showed that $\mathsf{LDR}(\mathsf{RM}_2(n,d)) = 2^{-d}$ which for $d \geqslant 2$ is much better than the Johnson bound. The GKZ algorithm is a generalization of the Goldreich-Levin algorithm: we assume that we have the correct value of the polynomial given as *advice* on a small random subspace A. This advice allows us to self-correct the values at randomly chosen shifts of A, usin1g a unique decoding algorithm. As pointed out in GKZ, this relies crucially on the coincidence that the ratio of minimum distance to unique decoding radius equals the field size (which is 2), and does not seem to generalize to larger fields (see Section II-B). They propose the following conjecture:

Conjecture 1: [3] For any constants q, d, $\mathsf{LDR}(\mathsf{RM}_q(n,d)) = \delta_q(d)$.

It is easy to show that $\mathsf{LDR}(\mathsf{RM}_q(n,d)) \leqslant \delta_q(d)$, the crux of the conjecture is the matching lower bound. GKZ show that once we bound $\ell(\mathsf{RM}_q(n,d),\eta)$, (a suitable modification of) the [5] algorithm can be used to recover the list of polynomials within radius η. Thus the the algorithmic problem reduces to the combinatorial problem of bounding th list-size. GKZ showed that $\mathsf{LDR}(\mathsf{RM}_q(n,d)) \geqslant \frac{1}{2}\delta_q(d-1)$; by Equation 3 this establishes the conjecture whenever $d \equiv 0 \bmod q-1$. This bound beats the Johnson bound for d sufficiently large. However when $d = 2$, Conjecture 1 states that agreement exceeding $2/q$ guarantees a small list, the Johnson bound guarantees a small list for agreement $\Omega(1/\sqrt{q})$ whereas the GKZ bound requires agreement exceeding $1/2$. Indeed, we believe that the hard(est) case of Conjecture 1 is when d is small, this precisely is where the gap between $\delta_q(d)$ and known bounds is largest.

II. Our Results

Previous work on local decoding of RM codes [2]–[5] relies on the notion of a self-corrector. Starting the correct values at some points as advice, the algorithm self-corrects the values of the polynomial along some low-dimensional subspace. This relies on the locality of the property of being a low-degree polynomial. Our work departs entirely from this paradigm. We seek to explain the good list-decoding properties of RM codes by using the rich structure in the weight distribution of these codes. While the RM code has low-weight codewords, a random codeword is very likely to have weight which is close to $1 - \frac{1}{q}$ (this is in fact true of any linear code). But in RM codes, the low-weight codewords are far from random in a very strong sense: they have very special structure. Results of this form date back to the classical sum-of-squares result for quadratic forms (due to Jacobi and Sylvester) [17], and the work of Kasami and Tokura for the $\mathbb{F}_2$ case [18]. More recently, there has been great progress made in structure versus randomness dichotomies for low-degree polynomials [19]–[21].

Our approach is to reduce the problem of RM decoding to list-decoding low-weight codewords using the Deletion Lemma from [3], [16]. We then use the structure of low-weight codewords to bound the list size. We note that the work of [3], [22] also uses the weight-distribution to bound the list-size for RM codes over $\mathbb{F}_2$. However, these papers only require a bound on the number of low-weight codewords, whereas we make crucial use of the structure of these codewords. The structural property that we use is that of being low-dimensional. A k-dimensional function is one that can be expressed as a k-junta (a function of at most k variables) under a suitable change of basis for $\mathbb{F}$. The choice of low-dimensional codewords is natural for a couple of reasons: firstly, the examples we know for exhibiting large lists at radius $\delta_q(d)$ are all low-dimensional [3, Theorem 12]. Secondly, there are classical results showing that in the cases $d = 2$ and $q = 2$ respectively, all low-weight codewords in RM codes are low-dimensional [17], [18].

Definition 1: The dimension of $F : \mathbb{F}_q^n \to \mathbb{F}_q$ denoted $\dim(F)$ is the smallest k for which there exist linear functions $\alpha_1, \ldots, \alpha_k : \mathbb{F}_q^n \to \mathbb{F}_q$ such that F can be expressed as a function of $\alpha_1, \ldots, \alpha_k$.

Let $\mathsf{RM}_q^k(n,d)$ be the sub-code of $\mathsf{RM}_q(n,d)$ consisting of all polynomials of dimension at most k (where k is constant).

Theorem 2.1: For all q, k and d it holds that $\mathsf{LDR}(\mathsf{RM}_q^k(n,d)) = \delta_q(d)$.

We prove this bound by designing a new Fourier-based algorithm for list-decoding low-dimensional polynomials. This algorithm and its analysis are the principal contributions of this work.

In the case of quadratic forms, our notion of dimension coincides with the classical notion of the rank of a quadratic form. It is well known that as the rank of a quadratic form increases, the distribution of its values approaches the uniform distribution over $\mathbb{F}_q$ [17]. We use this to prove:

Theorem 2.2: For all q, it holds that $\mathsf{LDR}(\mathsf{RM}_q(n,2)) = \delta_q(2)$. Further, for any q and $\varepsilon > 0$, we have $\ell(\mathsf{RM}_q(n,2), \delta_q(2)-\varepsilon) = \mathrm{poly}(q, \varepsilon^{-1})$.

This gives a tight bound on the list-decoding radius of quadratic forms, resolving what is a special, but important case of the GKZ conjecture, given the rich history of quadratic forms in mathematics and coding theory [11], [17]. In fact their conjecture was only for constant q, whereas our bound is reasonable even for $q = \mathrm{poly}(n)$. Using the local list-decoder from GKZ, we get an algorithm to recover all quadratic polynomials that have agreement $\frac{2}{q} + \varepsilon$ in time $\mathrm{poly}(n, q, \varepsilon^{-1})$. This improves on both the Johnson bound, which requires agreement $\frac{1}{\sqrt{q}}$ and the GKZ bound which requires $\frac{1}{2}$. Concretely, for $q = 256$, Theorem 2.2 guarantees constant list-size for agreement exceeding $\frac{1}{128}$, whereas Johnson and GKZ require agreement more than $\frac{1}{16}$ and $\frac{1}{2}$ respectively.

In the case of $\mathbb{F}_2$, classical results of Kasami and Tokura [18] imply that deletion of low- dimensional codewords doubles the distance of RM codes . This gives an alternate proof of the GKZ result that $\mathsf{LDR}(\mathsf{RM}_2(n,d)) = 2^{-d}$.

When d and q are arbitrary, we propose a conjecture quantifying how the deletion of low-dimensional codewords improves the distance of RM codes (see Conjecture 2 and Theorem 6.2 in Section VI). If the conjecture holds true, then with Theorem 2.1, we get an improvement on the best known current bounds for all d and q, which however falls short of the GKZ conjecture for $d > 3$. Nevertheless, Theorem 2.1 shows that low-dimensional polynomials are not an obstacle to the GKZ conjecture. Since the tight examples with large list-size at radius $\delta_q(d)$ stem from low-dimensional polynomials [3], this might be considered evidence in its favor.

A. Our Techniques

All previous work on Reed-Muller decoding [2]–[5] relies on the notion of a self-corrector. Starting the correct values at some point(s) as advice, the algorithm self-corrects the values of the polynomial along some low-dimensional subspace. Our work departs entirely from the self-correction paradigm and draws on ideas from Fourier analysis of Boolean functions; notably (a generalization of) the notion of influence of a variable. Fourier analytic methods are extensively used in learning, typically for concept classes such as halfspaces or decision trees [23] whose Fourier spectra show good *concentration*. Reed-Muller decoding is equivalent to (agnostically) learning low-degree polynomials over $\mathbb{F}_q$. It is not at all clear that Fourier analysis ought to be useful even for $d = 2$, since quadratic forms over $\mathbb{F}_2$ are the canonical examples of *bent* functions whose Fourier spectrum is maximally anti-concentrated [11]. However, the deletion lemma allows us to focus on low-degree polynomials which are additionally low-dimensional (dimension at most 6 for quadratic forms). The Fourier spectrum of a k-dimensional polynomial P is supported on a k-dimensional subspace which we denote by $\mathsf{Spec}(P)$. Our key insight is that *within* $\mathsf{Spec}(P)$*, the Fourier mass is anti-concentrated, which makes it possible to identify this subspace via Hadamard decoding, even after the adversary has corrupted the codeword.* We outline the main steps in our proof below:

1) Finding $\mathsf{Spec}(P)$**:** Fix $q = 2$ for simplicity. The Fourier mass of a k-dimensional polynomial P lies entirely on a k-dimensional subspace $\mathsf{Spec}(P)$. It is easy to recover P if we know $\mathsf{Spec}(P)$, by enumerating over all degree d polynomials in k variables and replacing the variables by linear forms (recall that k is constant). Our goal is to show for any received word F where $\Delta(F,P) \leqslant \delta_q(d)$, the *large* Fourier coefficients of F contain a basis for $\mathsf{Spec}(P)$. Equivalently, the large Fourier coefficients α of F that lie in $\mathsf{Spec}(P)$ should not all fall in a low-dimensional subspace $B \subset \mathsf{Spec}(P)$ satisfying an additional equation $b \cdot \alpha = 0$. One can try and prove this using the Fourier expression for ℓ_2 distance, but this approach fails. This suggests that one needs to use the *discreteness* of F.

2) The Influence of a Direction: Given a function F, the Fourier mass that lies in the set $S_b = \{\alpha : b \cdot \alpha \neq 0\}$ captures the influence of direction b, which is defined as $\Pr_{x \in \mathbb{F}_2^n}[F(x) \neq F(x+b)]$. This generalizes the notion of the influence of a variable [24]. Influences in low-degree polynomials P show a dichotomy: they are 0 over a subspace $\mathsf{Spec}(P)^{\perp}$, and large for all other b. We use this to show that if $\Delta(F,P) \leqslant \delta_q(d)$, and if b is influential in P, then it has noticeable influence on F. Hence, a noticeable fraction of the Fourier mass of F lies in the set S_b. But it falls short of the claim we really wish to prove, which is that there is noticeable Fourier mass lying in $\mathsf{Spec}(P) \cap S_b$, since F (unlike P) need not be low-dimensional.

3) Folding the Received word: The crucial step of our analysis is to go from F to a randomized function $\mathbf{F}$, which is F folded over the subspace $\mathsf{Spec}(P)$. While we defer the formal definition of folding, the following example is illustrative: if P depends only on $X_1, \ldots, X_k$, then so does $\mathbf{F}$; for each setting of $x_1, \ldots, x_k$, $\mathbf{F}(x_1, \ldots, x_k)$ equals $F(x_1, \ldots, x_n)$ where $x_{k+1}, \ldots, x_n$ are set randomly. From the viewpoint of P, $\mathbf{F}$ is a received word where the noise added at each point is randomized. The crucial observation is that the noise rate stays the same, so $\Delta(\mathbf{F}, P) \leqslant \delta_q(d)$, hence every influential direction b of P still has influence on $\mathbf{F}$. But since $\mathbf{F}$ is obtained by folding F over $\mathsf{Spec}(P)$, the Fourier spectrum of $\mathbf{F}$ if just the spectrum of F projected on to $\mathsf{Spec}(P)$. Thus we conclude that $\mathbf{F}$ (and hence F) has noticeable Fourier mass lying in $\mathsf{Spec}(P) \cap S_b$. Note that folding is just introduced for the sake of analysis, it plays no role in the algorithm.

4) Fourier analysis over $\mathbb{F}_q$: Implementing the above scheme over $\mathbb{F}_q$ is fairly challenging, since it is unclear what *the* Fourier expansion of $F : \mathbb{F}_q^n \to \mathbb{F}_q$ should mean. Our main technical innovation is to associate $q-1$ Fourier polynomials with every such F, this allows us to exactly arithmetize Hamming distance over $\mathbb{F}_q$ and handle randomized functions which is crucial in our setting.

We believe the Fourier analytic techniques here will find other applications. We use them prove an equivalence between learning parity with worst-case noise and weaker noise models over $\mathbb{F}_q$, generalizing a result of [25] for $\mathbb{F}_2$. Working with many Fourier polynomials as opposed to a single one is crucial for this result.

B. Comparison to Previous Work

It is interesting to contrast our approach to that of [3]. While GKZ the bound also involves a *dimension reduction* step, the term refers to restricting the received word to a random low-dimensional subspace, which is very different from what we do. The GKZ algorithm is based on a self-corrector that works correctly given the right advice. The self-correction argument already shows that the list-size at radius $2^{-d} - \varepsilon$ is quasi-polynomial in ε^{-1}. The deletion lemma is used only to improve the bounds to polynomial in ε^{-1}. As remarked earlier though, this self-corrector does not seem to generalize well to larger fields. Our approach is in fact inspired by the list-decoding algorithms of [16] for tensor products and interleaved codes, which reduce bounding the list-size to the low-rank case (in their setting, codewords are matrices and rank refers to the rank of these matrices).

Organization: We present Fourier-analytic preliminaries in Section III. The proofs for this section can be found in the full version of this paper [26]. The decoding algorithm for low-dimensional polynomials and its analysis are in Section IV, with the proof of Theorem 2.2. We present further reductions to the low-dimensional case in Section V, and a discussion of the case $d \geqslant 3$ in Section VI. We apply our techniques to give a worst-case noise to average-case noise reduction for the Noisy Parity problem in Section VII, the proofs for this are in the full version.

III. Low-Dimensional Functions, Folding and Influences

All proofs for this Section can be found in [26].

Fourier analysis: Let $p = \mathsf{char}(q)$ and let $q = p^h$. Let ω be a primitive p^{th} root of unity. Given a random variable Z taking values in $\mathbb{F}_q$, we define the quantities $z^c = \mathbb{E}_Z[\omega^{\mathsf{Tr}(cZ)}]$, which we call the (un-normalized) Fourier coefficients of Z. For two such random variables Y, Z, let $\mathsf{SD}(Y, Z)$ denote their statistical distance. The following relation to the Fourier transform is folklore:

Fact 3.1: For two random variables Y, Z taking values in $\mathbb{F}_q$, we have

$$\mathsf{SD}(Y, Z) \leqslant \frac{1}{2}\left(\sum_{c \in \mathbb{F}_q^\star} |y^c - z^c|^2\right)^{\frac{1}{2}}.$$

Let $\mathsf{Tr}(x) = \sum_{i=0}^{h-1} x^{p^i}$ denote the trace map from $\mathbb{F}_q$ to $\mathbb{F}_p$. The set of all linear functions $\mathbb{F}_q \to \mathbb{F}_p$ is given by $\{\mathsf{Tr}(cx)\}_{c \in \mathbb{F}_q}$. The character group $\hat{\mathbb{F}_q}^n$ of $\mathbb{F}_q^n$ of all homomorphisms $\chi : \mathbb{F}_q^n \to \mathbb{C}$ comprises all functions of the form $\chi_\alpha(x) = \omega^{\mathsf{Tr}(\alpha(x))}$ where $\alpha : \mathbb{F}_q^n \to \mathbb{F}_q$ is a linear function. It is easy to show that the functions χ_α form an orthonormal basis for all functions $f : \mathbb{F}_q^n \to \mathbb{C}$ under the inner-product $\langle f, g\rangle = \mathbb{E}_{x \in \mathbb{F}_q^n} f(x)\overline{g(x)}$. Thus every such f has a Fourier expansion given by

$$f(x) = \sum_{\alpha \in \hat{\mathbb{F}_q}^n} \hat{f}(\alpha)\chi_\alpha(x).$$

We also have $\|f\|_2 = \langle f, f\rangle = \sum_\alpha |\hat{f}(\alpha)|^2$. Given a polynomial $F : \mathbb{F}_q^n \to \mathbb{F}_q$, we associate it with $q-1$ Fourier polynomials mapping $\mathbb{F}_q^n \to \mathbb{C}$, one for every $c \in \mathbb{F}_q^\star$, given by

$$f^c(x) := \omega^{\mathsf{Tr}(cP(x))} = \sum_{\alpha \in \hat{\mathbb{F}_q}^n} \hat{f}^c(\alpha)\chi_\alpha(x).$$

The reason for using $q-1$ polynomials is that we can exactly arithmetize agreement and Hamming distance; this is crucial in our applications.

Fact 3.2: Given functions F, G that map $\mathbb{F}_q^n \to \mathbb{F}_q$,

$$\begin{aligned}\mathrm{Ag}(F, G) &= \frac{1}{q}(1 + \sum_{c \in \mathbb{F}_q^\star} \langle f^c, g^c\rangle) \\ &= \frac{1}{q}(1 + \sum_{c \in \mathbb{F}_q^\star}\sum_\alpha \hat{f}^c_\alpha \overline{\hat{g}^c_\alpha}) \qquad (2)\\ \Delta(F, G) &= \frac{1}{2q}\sum_{c \in \mathbb{F}_q^\star} \|f^c - g^c\|_2^2 \\ &= \frac{1}{2q}\sum_{c \in \mathbb{F}_q^\star}\sum_{\alpha \in \hat{\mathbb{F}_q}^n} |\hat{f}^c_\alpha - \hat{g}^c_\alpha|^2 \qquad (3)\end{aligned}$$

Randomized Functions: We consider randomized functions $\mathbf{F} : \mathbb{F}_q^n \to \mathbb{F}_q$, where each $\mathbf{F}(x)$ is a random variable taking values in $\mathbb{F}_q$. We define the Fourier polynomials associated with $\mathbf{F}$:

Definition 2: Given a randomized function $\mathbf{F} : \mathbb{F}_q^n \to \mathbb{F}_q$, for each $c \in \mathbb{F}_q^\star$, we define the polynomial $\mathbf{f}^c : \mathbb{F}_q^n \to \mathbb{C}$ by $\mathbf{f}^c(x) = \mathbb{E}_{\mathbf{F}}[\omega^{\mathsf{Tr}(c\mathbf{F}(x))}]$.

Note that $\mathbf{f}^c$ is a (deterministic) function from $\mathbb{F}_q^n \to \mathbb{C}$ and the values $\{\mathbf{f}^c(x)\}_{c \in \mathbb{F}_q^\star}$ give us the Fourier transform of $\mathbf{F}(x)$. Given two randomized functions $\mathbf{F}, \mathbf{G} : \mathbb{F}_q^n \to \mathbb{F}_q$, we define

$$\begin{aligned} d(\mathbf{F},\mathbf{G}) &= \mathbb{E}_{x\in\mathbb{F}_q^n}[\mathsf{SD}(\mathbf{F}(x),\mathbf{G}(x))], \\ \mathrm{Ag}(\mathbf{F},\mathbf{G}) &= 1-d(\mathbf{F},\mathbf{G}). \end{aligned}$$

generalizing the definitions for deterministic functions.

Fact 3.3: Given randomized function $\mathbf{F},\mathbf{G}$ that map $\mathbb{F}_q^n \to \mathbb{F}_q$, we have

$$\begin{aligned} d(\mathbf{F},\mathbf{G}) &\leqslant \frac{1}{2}\left(\sum_{c\in\mathbb{F}_q^\star}\mathbb{E}_x[|f^c(x)-g^c(x)|^2]\right)^{\frac{1}{2}} \\ &= \frac{1}{2}\left(\sum_{c\in\mathbb{F}_q^\star}\sum_{\alpha\in\hat{\mathbb{F}_q}^n}|\hat{f^c}(\alpha)-\hat{g^c}(\alpha)|^2\right)^{\frac{1}{2}}. \end{aligned} \quad (4)$$

Low-Dimensional Functions: Low dimensional deterministic functions are defined in Definition 1. We generalize the definition to randomized functions:

Definition 3: A randomized function $\mathbf{F}:\mathbb{F}_q^n \to \mathbb{F}_q$ is k dimensional if there exist k linear forms $\alpha_1,\ldots,\alpha_k:\mathbb{F}_q^n \to \mathbb{F}_q$ such that knowing $\alpha_1(x),\ldots,\alpha_k(x)$ fixes the distribution of $\mathbf{F}(x)$.

Hence $\mathbf{F}$ is a (randomized) function of $\alpha_1,\ldots,\alpha_k$, generalizing Definition 1. Facts 3.4 and 3.5 below are proved in [27], [28] for deterministic functions.

Fact 3.4: For each $c\in\mathbb{F}_q^\star$, let $\mathsf{Supp}(\mathbf{f}^c)\subseteq\hat{\mathbb{F}_q}^n$ denote the set of non-zero Fourier coefficients of $\mathbf{f}^c(x)$. Let $\mathsf{Spec}(\mathbf{F}) = \mathsf{Span}(\cup_{c\in\mathbb{F}_q^\star}\mathsf{Supp}(\mathbf{f}^c))$. Then $\dim(\mathbf{F})=\dim(\mathsf{Spec}(\mathbf{F}))$.

Alternatively, low-dimensional functions can be defined via invariant spaces.

Definition 4: Given $h\in\mathbb{F}_q^n$, if $\mathbf{F}:\mathbb{F}_q^n\to\mathbb{F}_q$ satisfies $\mathsf{SD}(\mathbf{F}(x+\lambda h),\mathbf{F}(x)=0$ for all $x\in\mathbb{F}_q^n,\lambda\in\mathbb{F}_q$ we say that $\mathbf{F}$ is *h-invariant*. We define $\mathsf{Inv}(\mathbf{F})=\{h:\mathbf{F}\text{ is }h\text{-invariant}\}$.

$\mathsf{Inv}(\mathbf{F})$ is clearly a subspace of $\mathbb{F}_q^n$, and is in fact dual to $\mathsf{Spec}(\mathbf{F})$.

Fact 3.5: We have $\mathsf{Spec}(\mathbf{F}) = \mathsf{Inv}(\mathbf{F})^\perp$. Hence $\dim(\mathbf{F})=\mathrm{codim}(\mathsf{Inv}(\mathbf{F}))$.

Folding: Folding over subspaces was introduced in [25] (in the $\mathbb{F}_2$ case). Folding maps high-dimensional functions to lower-dimensional randomized functions.

Definition 5: Let H be a subspace of $\mathbb{F}_q^n$ and let $F:\mathbb{F}_q^n\to\mathbb{F}_q$. Define the randomized function $\mathbf{F}(x)=F(x+h)$ where $h\in H$ is chosen randomly. We call $\mathbf{F}$ the folding of F over H.

Given an oracle for F, we can simulate an oracle for $\mathbf{F}$: on query x, choose a random point $x+h$ in the coset $x+H$ and return $F(x+h)$. Thus $\mathbf{F}$ is invariant on H. In fact, its Fourier spectrum is obtained by projecting the spectrum of F onto $H^\perp$.

Lemma 3.6: [25] Let $\mathbf{F}$ be the folding of F over H. For any $c\in\mathbb{F}_q^\star$, we have $\hat{\mathbf{f}^c}(\alpha)=\hat{f^c}(\alpha)$ if $\alpha\in H^\perp$ and $\hat{\mathbf{f}^c}(\alpha)=0$ otherwise.

The Influence of a Direction: We define the influence of a direction, which is a generalization of the notion of influence of a variable. Given a vector $b\in\mathbb{F}_q^n\setminus\{0^n\}$, we partition $\mathbb{F}_q^n$ into lines along the direction b, which are the equivalence classes for the relation $x\sim y$ if $x-y=\lambda b$ for some $\lambda\in\mathbb{F}_q$. This partition is nothing but $\mathbb{F}_q^n/\{b\}$, and it is isomorphic to $\mathbb{F}_q^{n-1}$.

Definition 6: (Influence of a direction) Given $b\in\mathbb{F}_q^n$, and a function $F:\mathbb{F}_q^n\to\mathbb{F}_q$ we define

$$\mathsf{Inf}_b(F)=\Pr_{x\in\mathbb{F}_q^n,\lambda\in\mathbb{F}_q}[F(x)\neq F(x+\lambda b)].$$

One can relate $\mathsf{Inf}_b(F)$ to the Fourier mass lying outside the subspace of $\hat{\mathbb{F}_q}^n$ given by $b\cdot\alpha=0$:

Fact 3.7: Given $b\in\mathbb{F}_q^n$, we have

$$\mathsf{Inf}_b(F)=\frac{1}{q}\sum_{c\in\mathbb{F}_q^\star}\sum_{\alpha:b\cdot\alpha\neq 0}|\hat{f^c}(\alpha)|^2. \quad (5)$$

We extend the notion of influences to randomized functions generalizing the above notion. To compute the influence of b for a deterministic function, we pick sample two points on a line in the direction b and compute their Hamming distance. For randomized function, we sample two such points and compute their statistical distance.

Definition 7: Given a randomized function $\mathbf{F}:\mathbb{F}_q^n\to\mathbb{F}_q$ and $b\in\mathbb{F}_q^n$, we define $\mathsf{Inf}_b(\mathbf{F})$ as

$$\mathsf{Inf}_b(\mathbf{F})=\mathbb{E}_{x\in\mathbb{F}_q^n,\lambda\in\mathbb{F}_q}[\mathsf{SD}(\mathbf{F}(x),\mathbf{F}(x+\lambda b)].$$

One can again bound the influence in terms of the Fourier mass that lies outside the subspace $b\cdot\alpha=0$ (though the bound is no longer exact, owing to the application of Cauchy-Schwartz).

Lemma 3.8: Given $b\in\mathbb{F}_q^n$, we have

$$\mathsf{Inf}_b(\mathbf{F})\leqslant\frac{1}{\sqrt{2}}\left(\sum_{c\in\mathbb{F}_q^\star}\sum_{\alpha:\ b\cdot\alpha\neq 0}|\hat{\mathbf{f}^c}(\alpha)|^2\right)^{\frac{1}{2}}.$$

IV. List-Decoding Low-Dimensional Polynomials

In this section, we prove Theorem 2.1. Assume that we have an efficient procedure Had for finding large Fourier coefficients over $\mathbb{F}_q^n$. Given oracle access to $f:\mathbb{F}_q^n\to\mathbb{C}$ and a parameter μ, $\mathsf{Had}(f,\mu)$ returns all $\alpha\in\hat{\mathbb{F}_q}^n$ so that $|\hat{f}(\alpha)|^2\geqslant\mu$. The list-size is bounded by $\|f\|_2^2/\mu$. Theorem 2.1 is proved by arguing that the polynomial P will be in the list of polynomials that is returned by the following algorithm.

Algorithm 1: LIST-DECODING LOW-DIMENSIONAL POLYNOMIALS
Input: d, k, ε, oracle for $F : \mathbb{F}_q^n \to \mathbb{F}_q$.
Output: All $P : \mathbb{F}_q^n \to \mathbb{F}_q$ s.t. $\deg(P) \leqslant d, \dim(P) \leqslant k$ and $\Delta(P, F) \leqslant \delta_q(d)(1-\varepsilon)$.

1. Set $\mu = \varepsilon^4 \delta_q(d)^2/(64q^{k+1})$.
2. Run $\mathsf{Had}(f^c, \mu)$ for all $c \in \mathbb{F}_q^\star$.
3. Let $\mathcal{L}$ be the list of all linear functions α returned.
4. Pick $\alpha_1, \ldots, \alpha_k$ from $\mathcal{L}$.
5. Return all $P(\alpha_1, \ldots, \alpha_k)$ s.t. $\deg(P) \leqslant d$ and $\Delta(P, F) \leqslant \delta_q(d)(1-\varepsilon)$.

In the last step, we enumerate over all polynomials P in k variables of degree d, after replacing the variables by $\alpha_1, \ldots, \alpha_k$.

A. Correctness of the Algorithm

Fix a polynomial P with $\deg(P) \leqslant d$, $\dim(P) \leqslant k$ and $\Delta(F, P) = \eta \leqslant \delta_q(d)(1-\varepsilon)$. Our goal is to prove that the list $\mathcal{L}$ contains a basis for $\mathsf{Spec}(P)$, which implies that P one of the polynomials returned by our algorithm. For the analysis, we work with the randomized function $\mathbf{F}$ obtained by folding F over $\mathsf{Inv}(P)$. Folding over $\mathsf{Inv}(P)$ projects the Fourier spectrum of F on to $\mathsf{Spec}(P)$, which is a *small* subspace with only q^k vectors in it. Our main lemma states that all directions that were influential in P continue to have some influence even in $\mathbf{F}$.

Lemma 4.1: **(Main)** For the function $\mathbf{F}$ defined above and any $b \notin \mathsf{Inv}(P)$,

$$\mathsf{Inf}_b(\mathbf{F}) \geqslant \frac{\varepsilon^2}{4}\delta_q(d).$$

Proof: Consider the vector space $V = \mathbb{F}_q^n/\mathsf{Inv}(P) \sim \mathbb{F}_q^k$. We can view P as a function $P : V \to \mathbb{F}_q$. Similarly, we can view $\mathbf{F}$ as a randomized function $\mathbf{F} : V \to \mathbb{F}_q$, obtained by adding random noise of rate η to P. Formally, for each $y \in V$, define the noise rate

$$\eta(y) = \Pr_{\mathbf{F}}[\mathbf{F}(y) \neq P(y)] = \Pr_{x \in y + \mathsf{Inv}(P)}[F(x) \neq P(y)]$$

and note that

$$\begin{aligned}\mathbb{E}_{y \in V}\, \eta(y) &= \Pr_{y \in V, x \in y+\mathsf{Inv}(P)}[F(x) \neq P(y)] \\ &= \Pr_{x \in \mathbb{F}_q^n}[F(x) \neq P(x)] = \eta.\end{aligned}$$

Our goal is to show that any $b \notin \mathsf{Inv}(P)$ has non-negligible influence on $\mathbf{F}$. Recall that for a randomized function $\mathbf{F} : \mathbb{F}_q^n \to \mathbb{F}_q$ and $b \in \mathbb{F}_q^n$, we defined $\mathsf{Inf}_b(\mathbf{F})$ as

$$\mathsf{Inf}_b(\mathbf{F}) = \mathbb{E}_{x \in \mathbb{F}_q^n, \lambda \in \mathbb{F}_q}[\mathsf{SD}(\mathbf{F}(x), \mathbf{F}(x + \lambda b))].$$

Since $\mathbf{F}$ is invariant on $\mathsf{Inv}(P)$, this is equivalent to

$$\mathsf{Inf}_b(\mathbf{F}) = \mathbb{E}_{y \in V, \lambda \in \mathbb{F}_q}[\mathsf{SD}(\mathbf{F}(y), \mathbf{F}(y + \lambda b))]. \tag{6}$$

Consider $V/\{b\}$, the partition of V into lines along b. We can rewrite Equation 6 as

$$\mathsf{Inf}_b(\mathbf{F}) = \mathbb{E}_{\substack{L \in V/\{b\} \\ x,y \in L}}[\mathsf{SD}(\mathbf{F}(x), \mathbf{F}(y))]. \tag{7}$$

Let us fix a basis containing the vector b for V: call it $\{a_1, \ldots, a_{k-1}, b\}$. Every vector $y \in V$ can be written in this basis as $y = \sum_{i=1}^{k-1} a_i y_i + b y_k$. The polynomial P can we written as $P(y_1, \ldots, y_k)$ of degree d. Assume that y_k occurs with degree $d_2 \leqslant q-1$ (this might depend on the choice of basis). So we can write

$$\begin{aligned}P(y_1, \ldots, y_k) = Q(y_1, \ldots, y_{k-1})y_k^{d_2} \\ + \sum_{e<d_2} Q_e(y_1, \ldots, y_{k-1})y_k^e.\end{aligned}$$

for some Q such that $\deg(Q) = d_1 \leqslant d - d_2$. Fixing values for $(y_1, \ldots, y_{k-1})$ specifies a line in $V/\{b\}$, while fixing y_k specifies a point on that line. Thus we can rewrite

$$\mathsf{Inf}_b(\mathbf{F}) = \mathbb{E}[\mathsf{SD}(\mathbf{F}(y_1, \ldots, y_{k-1}, y_k), \mathbf{F}(y_1, \ldots, y_{k-1}, y_k'))].$$

where the expectation is over $y_1, \ldots, y_{k-1}, y_k, y_k'$. We say that a line $\ell = (y_1, \ldots, y_{k-1}) \in V/\{b\}$ is good if $Q(y_1, \ldots, y_{k-1}) \neq 0$. Since $\deg(Q) \leqslant d_1$, $\Pr_\ell[\ell \text{ is good}] \geqslant \delta_q(d_1)$. Conditioning on the event that ℓ is good, $P|_\ell$ is a univariate polynomial of degree d_2. Hence, it takes on any particular value in $\mathbb{F}_q$ no more than d_2 times. In contrast, if ℓ is bad, then $P|_\ell$ is constant.

Define the noise rate $\eta(\ell)$ for a line as $\eta(\ell) = \mathbb{E}_{y \in \ell}[\eta(y)]$. We have $\mathbb{E}_{\ell \in V/\{b\}}[\eta(\ell)] = \eta$. We say that a good line is *quiet* if the noise rate along the line is low:

$$\eta(\ell) \leqslant \left(1 - \frac{d_2}{q}\right)\left(1 - \frac{\varepsilon}{2}\right).$$

We claim that at least $\varepsilon/2$ fraction of good lines are quiet; else we have

$$\begin{aligned}\mathbb{E}_\ell[\eta(\ell)] &\geqslant \delta_q(d_1)\left(1 - \frac{\varepsilon}{2}\right)\left(1 - \frac{d_2}{q}\right)\left(1 - \frac{\varepsilon}{2}\right) \\ &> \ \delta_q(d_1)\left(1 - \frac{d_2}{q}\right)(1-\varepsilon) \ \geqslant \ \delta_q(d)(1-\varepsilon).\end{aligned}$$

where the last inequality follows from the following property of $\delta_q(d)$: for all d_1, d_2 s.t. $d_1 + d_2 \leqslant d, 0 \leqslant d_2 \leqslant q-1$,

$$\delta_q(d) \leqslant \delta_q(d_1)\left(1 - \frac{d_2}{q}\right).$$

Fix a quiet line ℓ. We have a polynomial $P|_\ell : \ell \to F_q$ of degree $d_2 \leqslant q-1$ and $\mathbf{F}|_\ell$ such that

$$\begin{aligned}d(P|_\ell, \mathbf{F}|_\ell) &= \mathbb{E}_{x \in \ell}[\mathsf{SD}(P(x), \mathbf{F}(x))] \\ &= \mathbb{E}_{x \in \ell}[\eta(x)] \leqslant \delta_q(d_2) - \varepsilon'\end{aligned}$$

where $\delta_q(d_2) = 1 - \frac{d_2}{q}$ and $\varepsilon' = \frac{1}{2}\delta_q(d_2)\varepsilon$. The final piece of the argument is to show that for every quiet line, $\mathsf{Inf}_b(\mathbf{F})$ is high, which is a claim about univariate polynomials.

Claim 4.2: For a quiet line ℓ, we have $\mathbb{E}_{x,y\in\ell}[\mathsf{SD}(\mathbf{F}(x),\mathbf{F}(y))] \geqslant \varepsilon'$.

Let us defer the proof of this claim and finish the proof of Lemma 4.1. We have argued that

$$\Pr_{\ell\in V/\{b\}}[\ell \text{ is quiet}] \geqslant \frac{1}{2}\varepsilon\delta_q(d_1) \tag{8}$$

Conditioned on the event that ℓ is quiet, we have proved that

$$\mathop{\mathbb{E}}_{x,y\in\ell}[\mathsf{SD}(\mathbf{F}(x),\mathbf{F}(y))] \geqslant \frac{1}{2}\varepsilon\delta_q(d_2) \tag{9}$$

Plugging this into Equation 7 gives

$$\begin{aligned}\mathsf{Inf}_b(\mathbf{F}) &= \mathbb{E}_{\substack{\ell\in V/\{b\}\\ x,y\in\ell}}[\mathsf{SD}(\mathbf{F}(x),\mathbf{F}(y))] \\ &\geqslant \frac{\varepsilon^2}{4}\delta_q(d_1)\delta_q(d_2) \geqslant \frac{\varepsilon^2}{4}\delta_q(d) \end{aligned}\tag{10}$$

which completes the proof of Lemma 4.1. ■

Proof of Claim 4.2: For the purposes of this claim, we use P and $\mathbf{F}|_\ell$ to denote $P|_\ell$ and $\mathbf{F}_\ell$ respectively. Similarly d will denote distance between randomized functions on the line ℓ.

For every distribution $\mathcal{D}$ on $\mathbb{F}_q$, we can define the (constant) randomized function $\mathcal{D}^q : \ell \to \mathbb{F}_q$ where $\mathcal{D}^q(x) = \mathcal{D}$ for every $x \in \ell$. We claim that $d(P, \mathcal{D}^q) \geqslant \delta_q(d_2)$ for every such distribution $\mathcal{D}$. In the case where $\mathcal{D} = \mathcal{D}_y$ is concentrated at a single point $y \in \mathbb{F}_q$, this holds since $P(x)$ is a univariate polynomial with $\deg(P) = d_2$ and so $\Pr_x[P(x) = y] \leqslant d_2/q$. More generally, we have

$$\begin{aligned} d(P,\mathcal{D}^q) = \mathbb{E}_x[\mathsf{SD}(P(x),\mathcal{D})] &= \sum_{x\in F_q}\frac{1}{q}(1-\mathcal{D}(P(x)) \\ &= \sum_{y\in F_q}\Pr[P(x)=y](1-\mathcal{D}(y)) \\ &\geqslant 1-\frac{d_2}{q}\end{aligned}$$

where the last inequality uses $\Pr_x[P(x) = y] \leqslant d_2/q$ as $\deg(P) \leqslant d_2$. By the triangle inequality

$$d(\mathbf{F},\mathcal{D}^q) \geqslant d(P,\mathcal{D}^q) - d(\mathbf{F},P) \geqslant \delta_q(d_2) - (\delta_q(d_2)-\varepsilon') = \varepsilon'.$$

We compute $\mathbb{E}_{x,y\in\ell}[\mathsf{SD}(\mathbf{F}(x),\mathbf{F}(y))]$ by first sampling $x \in \ell$ and then computing the distance between $\mathbf{F}$ and the distribution $\mathcal{D}^q$ where $\mathcal{D} = \mathbf{F}(x)$.

$$\begin{aligned}\mathbb{E}_{x,y\in\ell}[\mathsf{SD}(\mathbf{F}(x),\mathbf{F}(y))] &= \mathbb{E}_{x\in\ell}[\mathbb{E}_{y\in\ell}[\mathsf{SD}(\mathbf{F}(x),\mathbf{F}(y))]] \\ &= \mathbb{E}_{x\in\ell}[d(\mathbf{F}(x)^q,\mathbf{F})] \geqslant \varepsilon'.\end{aligned}$$

This finishes the proof of Claim 4.2. ■

With the Main lemma in hand, Theorem 2.1 follows easily.

Lemma 4.3: The list $\mathcal{L}$ returned contains a basis for $\mathsf{Spec}(P)$.

Proof: Assume that the Fourier coefficients in $\mathcal{L} \cap \mathsf{Spec}(P)$ do not span all of $\mathsf{Spec}(P)$, rather they span a subspace B of it that satisfies the additional constraint $b\cdot\alpha = 0$ for $b \notin \mathsf{Inv}(P)$. We have

$$2\left(\sum_{c\in\mathbb{F}_q^\star}\sum_{\alpha:b\cdot\alpha\neq 0}|\hat{\mathbf{f}}^c_\alpha|^2\right)^{\frac{1}{2}} \geqslant \mathsf{Inf}_b(\mathbf{F}) \geqslant \frac{1}{4}\varepsilon^2\delta_q(d) \tag{11}$$

where the first inequality is from Lemma 3.8 and the second from Lemma 4.1. Applying Lemma 3.6 to the function $\mathbf{F}$ which is F folded over $\mathsf{Inv}(P)$, we get $\hat{\mathbf{f}}^c(\alpha) = \hat{f}^c(\alpha)$ for $\alpha \in \mathsf{Spec}(P)$ and $\hat{\mathbf{f}}^c(\alpha) = 0$ otherwise. Combining these equations, we get

$$\sum_{c\in\mathbb{F}_q^\star}\sum_{\alpha\in\mathsf{Spec}(P)\setminus B}|\hat{f}^c{}_\alpha|^2 \geqslant \frac{1}{64}\varepsilon^4\delta_q(d)^2$$

Since we sum over $(q^k - q^{k-1})(q-1) < q^{k+1}$ Fourier coefficients on the LHS, at least one of them is as large as the average. Thus, there exist $c \in \mathbb{F}_q^\star$ and $\alpha \in \mathsf{Spec}(P) \setminus B$ so that

$$|\hat{f}^c(\alpha)|^2 > \frac{1}{64}\frac{\varepsilon^4\delta_q(d)^2}{q^{k+1}}.$$

This coefficient α must belong to the list $\mathcal{L}$, which contradicts the assumption that $\mathcal{L} \cap \mathsf{Spec}(P)$ is contained within B. ■

A simple calculation which we omit gives the following bound on the list-size for $\mathsf{RM}_q^k(n,d)$ (we have not attempted to optimize this bound). There exists a constant $c > 0$ such that

$$\ell(\mathsf{RM}_q^k(n,d),\delta_q(d)(1-\varepsilon)) \leqslant \frac{c^k q^{k^d+k^2+2k}}{\varepsilon^{4k}\delta_q(d)^{2k}}. \tag{12}$$

The running time of Algorithm 1 is polynomial in n^d, q and the list-size.

V. Reduction to the low-dimensional case.

A. Quadratic Forms

We use the deletion lemma from [3]. The following version of the lemma appears in [16].

Lemma 5.1: [3], [16] *(Deletion Lemma)* Let $\mathcal{C} \subset \mathbb{F}_q^n$ be a linear code over $\mathbb{F}_q$. Let $\mathcal{C}' \subseteq \mathcal{C}$ be a (possibly non-linear) subset of codewords so that $c' \in \mathcal{C}'$ iff $-c' \in \mathcal{C}'$, and every codeword $c \in \mathcal{C} \setminus \mathcal{C}'$ has $\mathsf{wt}(c) \geqslant \delta^h$. Let $\eta = \mathsf{J_q}(\delta^h) - \gamma$ for $\gamma > 0$. Then $\ell(\mathcal{C},\eta) \leqslant \gamma^{-2}\ell(\mathcal{C}',\eta)$.

For quadratic forms $Q : \mathbb{F}_q^n \to \mathbb{F}_q$, $\dim(Q)$ coincides with the well-studied notion of the rank of a quadratic form. Theorems 6.26, 6.27 and 6.32 from Chapter 6 of [17] give the following bound:

Lemma 5.2: Let $Q : \mathbb{F}_q^n \to \mathbb{F}_q$ be a quadratic form such that $\dim(P) = k$. Then

$$\mathsf{wt}(Q) \geqslant 1 - \frac{1}{q} - \frac{1}{q^{k/2}}.$$

We use this to complete the proof of Theorem 2.2.

Proof of of Theorem 2.2.: By Lemma VI, if $\dim(Q) \geqslant 6$, then we have

$$\mathsf{wt}(Q) \geqslant 1-\frac{1}{q}-\frac{1}{q^3}; \quad \mathsf{J_q}\left(1-\frac{1}{q}-\frac{1}{q^3}\right) > 1-\frac{2}{q}.$$

Hence we can apply Lemma 5.1 with $\mathcal{C}' = \mathsf{RM}_q^6(n,2)$ to conclude that there exists c so that

$$\ell(\mathsf{RM}_q(n,2), \delta_q(2)-\varepsilon) \leqslant \frac{1}{\varepsilon^2}\ell(\mathsf{RM}_q^6(n,2), \delta_q(2)-\varepsilon) \leqslant c\frac{q^{84}}{\varepsilon^{26}}.$$

■

B. The $\mathbb{F}_2$ case revisited

Using our techniques, we can give an alternate proof of the GKZ result that $\mathsf{LDR}(\mathsf{RM}_2(n,d) = 2^{-d}$. A classical result of Kasami and Tokura allows us to bound the dimension of any codeword of $\mathsf{RM}_2(n,d)$ which has weight less than $2\delta_2(d)$.

Lemma 5.3: [18] Let $d \geqslant 2$. Let $P : \mathbb{F}_2^n \to \mathbb{F}_2$ with $\deg(P) \leqslant d$ and $\mathsf{wt}(P) < 2\delta_2(d)$. Then P is of one of the following two types:

1. $P(\alpha_1,\ldots,\alpha_{d+t}) = \alpha_1\cdots\alpha_{d-t}(\alpha_{d-t+1}\cdots\alpha_d + \alpha_{d+1}\cdots\alpha_{d+t}) \quad 3 \leqslant t < d$.
2. $P(\alpha_1,\ldots,\alpha_{d+2t-2}) = \alpha_1\cdots\alpha_{d-2}(\alpha_{d-1}\alpha_d + \alpha_{d+1}\alpha_{d+2}+\cdots+\alpha_{d+2t-3}\alpha_{d+2t-2})$.

where the α_is are independent linear forms.

Strictly speaking, the α_is are affine rather than linear, but we can safely ignore this issue.

Corollary 5.4: Let $P : \mathbb{F}_2^n \to \mathbb{F}_2$ be a degree d polynomial with $\dim(P) = k \geqslant 2d$. Then $\mathsf{wt}(P) \geqslant 2\delta_2(d) - 2^{-(k+d)/2}$.

Proof: Assume that $\mathsf{wt}(P) < 2\delta_2(d)$, else the claim is trivial. Now applying Lemma 5.3, P must be of type (2), since polynomials of type (1) have dimension less than $2d$. A simple calculation shows that for polynomials of type (2), if $\dim(P) = k$, then $\mathsf{wt}(P) \geqslant 2\delta_2(d) - 2^{-(k+d)/2}$. ■

We can now reprove the main result from [3]. Our list-size bound is polynomial in ε^{-d}, though the exact bound is inferior to GKZ, who also showed a lower bound of $\varepsilon^{-\Omega(d)}$.

Theorem 5.5: [3] For all $d \geqslant 1$, it holds that $\mathsf{LDR}(\mathsf{RM}_2(n,d)) = 2^{-d}$.

Proof: Pick $k = 3d$. Take $\mathcal{C}' = \mathsf{RM}_2^k(n,d)$. By Equation 12, we have $\ell(\mathcal{C}', \delta_2(d)-\varepsilon) \leqslant c\varepsilon^{-12d}$ for some constant $c = c(d)$ that depends on d. By Corollary 5.4, if $\dim(P) \geqslant 3d$,

$$\mathsf{wt}(P) \geqslant 2\cdot 2^{-d} - 2^{-2d}; \quad \mathsf{J}_2(2\cdot 2^{-d} - 2^{-2d}) > 2^{-d}.$$

Hence applying Lemma 5.1, we get

$$\ell(\mathsf{RM}_2(n,d), \delta_2(d)-\varepsilon) \leqslant c\varepsilon^{-(12d+2)}$$

which completes the proof. ■

VI. The Case of arbitrary d and q.

For cubic forms and higher, codewords of weight $1-\frac{1}{q}-\varepsilon$ need not be low-dimensional. However the results of [19], [20] show that when q is prime, such codewords must be expressible as functions of a few polynomials of degree $d-1$. Define $\mathrm{Rank}_d(P)$ to be the smallest number of degree d polynomials $Q_1,\ldots,Q_t$ such that $P = f(Q_1,\ldots,Q_t)$ for some function f. Note that $\mathrm{Rank}_1(P) = \dim(P)$.

Theorem 6.1: [19], [20] Let q be prime. For every degree d, there exists a function $r(\varepsilon)$ such that if $\deg(P) = d$ and $\mathsf{wt}(P) \leqslant 1-\frac{1}{q}-\varepsilon$, then $\mathrm{Rank}_{d-1}(P) \leqslant r(\varepsilon)$.

This suffices to show that over prime fields, $\ell(\mathsf{RM}_q(n,d), 1-\frac{1}{q}-\varepsilon) \leqslant q^{O_\varepsilon(n^{d-1})}$ as opposed to the trivial $q^{O(n^d)}$, by using the Deletion lemma. For $d = 2$ Lemma suffices, and it holds for all fields. This question (for the case $d = 2$) was raised by Tim Gowers in a blog-post titled "A conversation about complexity lower bounds, continued". Further one can find the list of all such polynomials in similar running time using Theorem 21 from [3].

To extend the approach taken in this work, one would need a good list-size bound for degree d polynomials where $\mathrm{Rank}_{d-1}(P) \leqslant k$. This seems fairly challenging given current techniques. As a first step one would require the combining function f to be made explicit. This is done for $d = 3,4$ in recent work of Haramaty and Shpilka [21].

Nevertheless, we believe that with Theorem 2.1 in hand, it is possible to improve on currently known bounds for all degrees. For cubic forms and higher, this leads to the question of how much the distance improves by deleting all low-dimensional polynomials. To formalize this, we define $\delta_q^h(d)$ which is the smallest weight at which codewords of unbounded dimension appear. Let

$$\delta_q^k(d) = \min\{\mathsf{wt}(P) : P \ s.t.\ \deg(P) \leqslant d, \ \ \dim(P) = k\},$$
$$\delta_q^h(d) = \liminf_{k\to\infty} \delta_q^k(d). \tag{13}$$

In defining $\delta_q^k(d)$, we minimize over the infinite set of all degree d polynomials P with $\dim(P) = k$, the number of variables n could be arbitrary. But since $\dim(P) = k$, we may assume that P is on exactly k variables. Thus we are in effect minimizing over the finite set of $P : \mathbb{F}_q^k \to \mathbb{F}_q$ s.t. $\deg(P) = d$ and $\dim(P) = k$, so $\delta_q^k(d)$ is well-defined. Our interest in $\delta_q^h(d)$ stems from the following theorem.

Theorem 6.2: For all d and q it holds that $\mathsf{LDR}(\mathsf{RM}_q(n,d)) \geqslant \min(\mathsf{J_q}(\delta_q^h(d)), \delta_q(d))$.

Proof: Let $\eta = \min(\delta_q(d), \mathsf{J_q}(\delta_q^h(d))) - \varepsilon$. Our goal is to show that for any $\varepsilon > 0$, $\ell(\mathsf{RM}_q(n,d),\eta)$ which is the list-size at radius η can be bounded independent of n.

Since $\eta \leqslant \delta_q(d)-\varepsilon$, by Theorem 2.1 $\ell(\mathsf{RM}_q^k(n,d),\eta) \leqslant \ell(d,k,q,\varepsilon)$.

We choose k large enough that $\mathsf{J_q}(\delta_q^k(d)) > \mathsf{J_q}(\delta_q^h(d)) - \varepsilon/2$ so that $\eta \leqslant \mathsf{J_q}(\delta_q^h(d)) - \varepsilon \leqslant \mathsf{J_q}(\delta_q^k(d)) - \varepsilon/2$. Every codeword outside of $\mathsf{RM}_q^k(n,d)$ has $\dim(P) \geqslant k$, and hence $\mathsf{wt}(P) \geqslant \delta_q^k(d)$. Thus we can invoke Lemma 5.1 with $\mathcal{C}' = \mathsf{RM}_q^k(n,d)$ to conclude that

$$\ell(\mathsf{RM}_q(n,d),\eta) \leqslant \frac{4}{\varepsilon^2}\ell(d,k,q,\varepsilon) = \ell'(d,k,q,\varepsilon).$$

This shows that the list-size at radius $\min(\delta_q(d), \mathsf{J}_\mathsf{q}(\delta_q^h(d))) - \varepsilon$ is bounded independent of n for every $\varepsilon > 0$, which proves the claim. ■

While it is *a priori* unclear if $\delta_q^h(d) > \delta_q(d)$, we conjecture that it is in fact substantially larger.

Conjecture 2: For all d and q it holds that $\delta_q^h(d) \geqslant \left(1 - \frac{1}{q}\right)\delta_q(d-2)$.

It is easy to see that $\delta_h^q(d)$ is at most the claimed bound, by taking the product of a large rank quadratic form and a minimum weight polynomial of degree $d-2$. In the case of $\mathbb{F}_2$, Conjecture 2 is implied by classical results of Kasami and Tokura [18]. For degree 3 polynomials, Amir Shpilka observed that it follows from the results of [21].

Observe that $\left(1 - \frac{1}{q}\right)\delta_q(d-2) \geqslant \delta_q(d-1)$. Hence if Conjecture 2 holds, then Theorem 6.2 gives

$$\mathsf{LDR}(\mathsf{RM}_q(n,d)) \geqslant \min(\mathsf{J}_\mathsf{q}(\delta_q(d-1)), \delta_q(d))$$

which improves on the bound of $\max(\frac{1}{2}\delta_q(d-1), \mathsf{J}_\mathsf{q}(\delta_q(d)))$ from GKZ for all d and q where their bound is less than $\delta_q(d)$.

Claim 6.3: For all d and q, it holds that

$$\min(\mathsf{J}_\mathsf{q}(\delta_q(d-1)), \delta_q(d)) \geqslant \max(\mathsf{J}_\mathsf{q}(\delta_q(d)), \frac{1}{2}\delta_q(d-1)).$$

The inequality is strict except when $d = 1$ and $d \equiv 0 \mod q-1$, and in both those cases the RHS equals $\delta_q(d)$.

Proof: Note that for all $\eta \in [0, 1-1/q]$, we have $\eta/2 \leqslant \mathsf{J}_\mathsf{q}(\eta) \leqslant \eta$ with $\mathsf{J}_\mathsf{q}(\eta) = \eta$ iff $\eta = 1 - \frac{1}{q}$ and $\mathsf{J}_\mathsf{q}(\eta) = \eta/2$ iff $\eta = 0$.

Further, if $d = a(q-1)+b$ for $1 \leqslant b \leqslant q-1$, $\delta_q(d-1) = \delta_q(d)\left(1 + \frac{1}{q-b}\right)$ hence

$$\frac{q}{q-1}\delta_q(d) \ \leqslant\ \delta_q(d-1) \ \leqslant\ 2\delta_q(d).$$

The former is tight when $d \equiv 1 \mod (q-1)$, the latter when $d \equiv q-1 \mod (q-1)$.

We now prove the above claim. Firstly, note that from the above inequalities, we have

$$\mathsf{J}_\mathsf{q}(\delta_q(d-1)) > \frac{1}{2}\delta_q(d-1) \quad\text{and}\quad \mathsf{J}_\mathsf{q}(\delta_q(d-1)) > \mathsf{J}_\mathsf{q}(\delta_q).$$

Secondly, we also have

$$\delta_q(d) \geqslant \frac{1}{2}\delta_q(d-1) \quad\text{and}\quad \delta_q(d) \geqslant \mathsf{J}_\mathsf{q}(\delta_q(d)).$$

The first inequality is strict, except when $d \equiv q-1 \mod (q-1)$. In this case, the GKZ bound is already tight. Similarly, the second inequality is strict except when $\delta_q(d) = 1 - \frac{1}{q}$, which holds when $d = 1$ or Hadamard codes, in which case the Johnson bound is tight. ■

VII. Learning Parity with Noise over Arbitrary Fields.

The Noisy Parity problem is a central problem in learning theory [25], [29], with connections to coding and cryptography. There are cryptosystems whose security is based on the assumption that learning parity with random noise is hard over large fields. The two natural noise models for this problem are random noise and adversarial noise, which we define below. Unlike the $\mathbb{F}_2$ case, there are many possible models for random noise over $\mathbb{F}_q$ of varying sophistication [30]. Feldman *et al.* [25] showed that over $\mathbb{F}_2$ the (seemingly harder) adversarial model reduces to the random noise model. No analogue of their result was previously known for other fields. We note that for cryptography, the case of interest is the large field case.

We show that over any field, the learning parity with adversarial noise reduces to learning parity in a weaker noise model called the Discrete Memoryless Channel (DMC) noise model [30]. This is a noise model that lies inbetween the adversarial model and the additive random noise model.

In the DMC model, we are required to learn some linear function $\alpha : \mathbb{F}_q^n \to \mathbb{F}_q$, from samples of the form $\langle x, \mathbf{F}(x)\rangle$, The noise is modeled by a $q \times q$ stochastic matrix W, where $w_{ij} = \Pr[\mathbf{F}(x) = j \mid \alpha(x) = i]$. Thus the noise added may depend on $\alpha(x)$ but not on x itself, unlike the adversarial model. But the DMC model is stronger than the additive noise model where the noise added is a random variable that is independent of the label. The matrix W is not known to the algorithm, but we assume that

$$\sum_{i \leqslant q} w_{ii} \geqslant 1 + q\eta.$$

This is analogous to assuming a bound on the overall noise rate since

$$\begin{aligned}\Pr_{x\in\mathbb{F}_q^n}[\mathbf{F}(x) = \alpha(x)] &= \sum_{i\in\mathbb{F}_q}\Pr[\alpha(x) = i]w_{ii} \\ &= \frac{1}{q}\sum_i w_{ii} \geqslant \frac{1}{q} + \eta.\end{aligned}$$

The adversarial channel model seems harder, being a generalization of the DMC model. In the adversarial setting there could be $\frac{1}{\eta^2}$ linear function with agreement $\frac{1}{q} + \eta$, whereas we show that in the DMC model, one can uniquely recover linear functions (up to scalar multiplication). We prove the following equivalence between the two models:

Theorem 7.1: Assume there is an algorithm $\mathcal{A}$ that solves the noisy parity problem over $\mathbb{F}_q$ in the DMC model in time $T(\eta,n)$ using $S(\eta,n) \leqslant T(\eta,n)$ samples. Then there is an algorithm $\mathcal{B}$ that solves the noisy parity problem over $\mathbb{F}_q$ in the adversarial noise model in time $\mathrm{poly}(q, T(\eta,n))$.

The proof is deferred to the full version [26].

Acknowledgments

I thank Prasad Raghavendra and Venkatesan Guruswami for numerous discussions about this problem, without which this paper would not exist. I also thank Sergey Yekhanin, David Zuckerman, Jaikumar Radhakrishnan, Madhu Sudan, Amir Shpilka, Irit Dinur, Ran Raz and Alex Samorodnitsky for many helpful discussions.

References

[1] O. Goldreich and L. Levin, "A hard-core predicate for all one-way functions." in *Proc.* 21^{st} *ACM Symposium on the Theory of Computing (STOC'89)*, 1989, pp. 25–32.

[2] O. Goldreich, R. Rubinfeld, and M. Sudan, "Learning polynomials with queries: The highly noisy case." *SIAM J. Discrete Math.*, vol. 13, no. 4, pp. 535–570, 2000.

[3] P. Gopalan, A. Klivans, and D. Zuckerman, "List-decoding Reed-Muller codes over small fields," in *Proc.* 40^{th} *ACM Symposium on the Theory of Computing (STOC'08)*, 2008.

[4] S. Arora and M. Sudan, "Improved low-degree testing and its applications." *Combinatorica*, vol. 23, no. 3, pp. 365–426, 2003.

[5] M. Sudan, L. Trevisan, and S. P. Vadhan, "Pseudorandom generators without the XOR lemma." *J. Comput. Syst. Sci.*, vol. 62, no. 2, pp. 236–266, 2001.

[6] P. Elias, "List decoding for noisy channels," Research Laboratory of Electronics, MIT, Tech. Rep. 335, 1957.

[7] J. Wozencraft, "List decoding," Quarterly Progress Report, Research Laboratory of Electronics, MIT, Tech. Rep. 48:90-95, 1958.

[8] M. Sudan, "Decoding of Reed-Solomon codes beyond the error-correction bound." *Journal of Complexity*, vol. 13, no. 1, pp. 180–193, 1997.

[9] V. Guruswami, *List Decoding of Error-Correcting Codes*, ser. Lecture Notes in Computer Science. Springer, 2004, vol. 3282.

[10] M. Sudan, "List decoding: Algorithms and applications," *SIGACT News*, vol. 31, no. 1, pp. 16–27, 2000.

[11] F. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*. North-Holland, 1977.

[12] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and Algebraic-Geometric codes." *IEEE Transactions on Information Theory*, vol. 45(6), pp. 1757–1767, 1999.

[13] R. Pellikaan and X. Wu, "List decoding of q-ary Reed-Muller codes," *IEEE Transactions on Information Theory*, vol. 50(4), pp. 679–682, 2004.

[14] S. M. Johnson, "A new upper bound for error-correcting codes," *IEEE Transactions on Information Theory*, vol. 8, pp. 203–207, 1962.

[15] ——, "Improved asymptotic bounds for error-correcting codes," *IEEE Transactions on Information Theory*, vol. 9, pp. 198–205, 1963.

[16] P. Gopalan, V. Guruswami, and P. Raghavendra, "List-decoding tensor products and interleaved codes," in *Proc.* 41^{st} *ACM Symposium on the Theory of Computing (STOC'09)*, 2009.

[17] R. Lidl and H. Neiderreiter, *Finite Fields*. Cambridge University Press, 1997.

[18] T. Kasami and N. Tokura, "On the weight structure of Reed-Muller codes," *IEEE Transactions on Information Theory*, vol. 16(6), pp. 752–759, 1970.

[19] B. Green and T. Tao, "The distribution of polynomials over finite fields, with applications to the Gowers norms," *Contrib. Discrete Math*, no. 4(2), pp. 1–36, 2009.

[20] T. Kaufman and S. Lovett, "Worst-case to average-case reductions for polynomials," in *Proc.* 49^{th} *IEEE Symp. on Foundations of Computer Science (FOCS'08)*, 2008.

[21] E. Haramaty and A. Shpilka, "On the structure of cubic and quartic polynomials," in *Proc.* 42^{nd} *ACM Symposium on the Theory of Computing (STOC'10)*, 2010, pp. 331–340.

[22] T. Kaufman and S. Lovett, "Weight distribution and list-decoding size of Reed-Muller codes," in *Proc.* 1^{st} *Symposium on Innovations in Computer Science (ICS 2010)*, 2010.

[23] E. Kushilevitz and Y. Mansour, "Learning decision trees using the Fourier spectrum," *SIAM Journal of Computing*, vol. 22(6), pp. 1331–1348, 1993.

[24] J. Kahn, G. Kalai, and N. Linial, "The influence of variables on Boolean functions," in *Proc.* 29^{th} *IEEE Symp. on Foundations of Computer Science (FOCS'88)*, 1988, pp. 68–80.

[25] V. Feldman, P. Gopalan, S. Khot, and A. K. Ponnuswami, "New results for learning noisy parities and halfspaces," in *Proc.* 47^{th} *IEEE Symp. on Foundations of Computer Science (FOCS'06)*, 2006.

[26] P. Gopalan, "A Fourier-analytic approach to Reed-Muller decoding," 2010. [Online]. Available: http://research.microsoft.com/∼parik

[27] P. Gopalan, S. Khot, and R. Saket, "Hardness of reconstructing multivariate polynomials over finite fields," in *Proc.* 48^{th} *IEEE Symp. on Foundations of Computer Science (FOCS'07)*, 2007, pp. 349–359.

[28] P. Gopalan, R. O'Donnell, A. Shpilka, R. Servedio, and K. Wimmer, "Testing Fourier dimensionality and sparsity," in *ICALP (1)*, 2009, pp. 500–512.

[29] A. Blum, A. Kalai, and H. Wasserman, "Noise-tolerant learning, the parity problem, and the statistical query model," *J. ACM*, vol. 50, no. 4, pp. 506–519, 2003.

[30] A. Lapidoth and P. Narayan, "Reliable communication under channel uncertainty," *IEEE Transactions on Information Theory*, vol. 44(6), pp. 2148–2177, 1998.