Local Decoding and Testing of Polynomials over Grids[*][†]

Srikanth Srinivasan[1] and Madhu Sudan[2]

1 **Department of Mathematics, IIT Bombay, India**
srikanth@math.iitb.ac.in

2 **Harvard John A. Paulson School of Engineering and Applied Sciences, USA**
madhu@cs.harvard.edu

Abstract

The well-known DeMillo-Lipton-Schwartz-Zippel lemma says that n-variate polynomials of total degree at most d over grids, i.e. sets of the form $A_1 \times A_2 \times \cdots \times A_n$, form error-correcting codes (of distance at least 2^{-d} provided $\min_i\{|A_i|\} \geq 2$). In this work we explore their local decodability and local testability. While these aspects have been studied extensively when $A_1 = \cdots = A_n = \mathbb{F}_q$ are the same finite field, the setting when A_i's are not the full field does not seem to have been explored before.

In this work we focus on the case $A_i = \{0,1\}$ for every i. We show that for every field (finite or otherwise) there is a test whose query complexity depends only on the degree (and not on the number of variables). In contrast we show that decodability is possible over fields of positive characteristic (with query complexity growing with the degree of the polynomial and the characteristic), but not over the reals, where the query complexity must grow with n. As a consequence we get a natural example of a code (one with a transitive group of symmetries) that is locally testable but not locally decodable.

Classical results on local decoding and testing of polynomials have relied on the 2-transitive symmetries of the space of low-degree polynomials (under affine transformations). Grids do not possess this symmetry: So we introduce some new techniques to overcome this handicap and in particular use the hypercontractivity of the (constant weight) noise operator on the Hamming cube.

1998 ACM Subject Classification F.1.2 Modes of Computation, Probabilistic computation

Keywords and phrases Property testing, Coding theory, Low-degree testing, Local decoding, Local testing

Digital Object Identifier 10.4230/LIPIcs.ITCS.2018.26

1 Introduction

Low-degree polynomials have played a central role in computational complexity. (See for instance [27, 8, 5, 21, 23, 19, 28, 3, 2] for some of the early applications.) One of the key properties of low-degree n-variate polynomials underlying many of the applications is the "DeMillo-Lipton-Schwartz-Zippel" distance lemma [10, 26, 30] which upper bounds the number of zeroes that a non-zero low-degree polynomial may have over "grids", i.e., over domains of the form $A_1 \times \cdots \times A_n$. This turns the space of polynomials into an

[*] This work was partially supported by a Simons Investigator Award and NSF Awards CCF 1565641 and CCF 1715187.

[†] A full version of the paper is available at [29], https://arxiv.org/abs/1709.06036

9th Innovations in Theoretical Computer Science Conference (ITCS 2018).
Editor: Anna R. Karlin; Article No. 26; pp. 26:1–26:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

error-correcting code (first observed by Reed [24] and Muller [20]) and many applications are built around this class of codes. These applications have also motivated a rich collection of tools including polynomial time (global) decoding algorithms for these codes, and "local decoding" [4, 18, 9] and "local testing" [25, 1, 15] procedures for these codes.

Somewhat strikingly though, many of these tools associated with these codes don't work (at least not immediately) for all grid-like domains, but work only for the specific case of the domain being the vector space $\mathbb{F}^n$ where $\mathbb{F}$ is the field over which the polynomial is defined and $\mathbb{F}$ is finite. The simplest example of such a gap in knowledge was the case of "global decoding". Here, given a function $f : \prod_{i=1}^{n} A_i \to \mathbb{F}$ as a truth-table, the goal is to find a nearby polynomial (up to half the distance of the underlying code) in time polynomial in $|\prod_i A_i|$. When the domain equals $\mathbb{F}^n$ then such algorithms date back to the 1950s. However the case of general A_i remained open till 2016 when Kim and Kopparty [17] finally solved this problem.

In this paper we initiate the study of local decoding and testing algorithms for polynomials when the domain is not a vector space. As a first step towards this we consider the case of polynomials over hypercubes i.e., when $A_i = \{0, 1\} \subseteq \mathbb{F}$ for every i. (This setting easily extends to the case where $|A_i| = 2$ for all i — see more on this at the end of Section 1.1. The setting of $|A_i| > 2$ seems to offer new challenges that we don't explore in this paper.) We describe the problems formally next and then describe our results.

1.1 Distance, Local Decoding and Local Testing

We start with some brief notation. For finite sets $A_1, \ldots, A_n \subseteq \mathbb{F}$ and functions $f, g : A_1 \times \cdots A_n \to \mathbb{F}$, let the distance between f and g, denoted $\delta(f, g)$ be the quantity $\Pr_a[f(a) \neq g(a)]$ where a is drawn uniformly from $A_1 \times \cdots \times A_n$. We say f is δ-close to g if $\delta(f, g) \leq \delta$, and δ-far otherwise. For a family of functions $\mathcal{F} \subseteq \{h : A_1 \times \cdots \times A_n \to \mathbb{F}\}$, let $\delta(\mathcal{F}) = \min_{f \neq g \in \mathcal{F}}\{\delta(f, g)\}$.

To set the context for some of the results on local decoding and testing, we first recall the distance property of polynomials. If $|A_i| \geq 2$ for every i, the polynomial distance lemma asserts that the distance between any two distinct degree d polynomials[1] is at least 2^{-d}. Of particular interest is the fact that for fixed d this distance is bounded away from 0, independent of n or $|\mathbb{F}|$ or the structure of the sets A_i. In turn this behavior effectively has led to "local decoding" and "local testing" algorithms with complexity depending only on d — we define these notions and elaborate on this sentence next.

Given a family of functions $\mathcal{F}$ from the domain $A_1 \times \cdots \times A_n$ to $\mathbb{F}$, we say $\mathcal{F}$ is (δ, q)-*locally decodable* if there exists a probabilistic algorithm that, given $a \in A_1 \times \cdots \times A_n$ and oracle access to a function $f : A_1 \times \cdots \times A_n \to \mathbb{F}$ that is δ-close to some function $p \in \mathcal{F}$, makes at most q oracle queries to f and outputs $p(a)$ with probability at least 3/4. (The existence of a (δ, q)-local decoder for $\mathcal{F}$ in particular implies that $\delta(\mathcal{F}) \geq 2\delta$.) We say that $\mathcal{F}$ is (δ, q)-*locally testable* if there exists a probabilistic algorithm that makes q queries to an oracle for $f : A_1 \times \cdots \times A_n \to \mathbb{F}$ and accepts with probability at least 3/4 if $f \in \mathcal{F}$ and rejects with probability at least 3/4 if f is δ-far from every function in $\mathcal{F}$.

When $A_1 = \cdots = A_n = \mathbb{F}$ (and so $\mathbb{F}$ is finite) it was shown by Kaufman and Ron [15] (with similar results in Jutla et al. [13]) that the family of n-variate degree d polynomials over $\mathbb{F}$ is (δ, q)-locally decodable and (δ, q)-locally testable for some $\delta = \exp(-d)$ and $q = \exp(d)$.

[1] Throughout this paper we only consider the functions represented by degree d polynomials. So, without loss of generality, these maybe viewed as polynomials of degree at most $|A_i| - 1$ in the ith variable, and specifically multilinear polynomials when $|A_i| = 2$.

In particular both q and $1/\delta$ are bounded for fixed d, independent of n and $\mathbb{F}$. Indeed in both cases δ is lower bounded by a constant factor of $\delta(\mathcal{F}(n,d))$ and q is upper bounded by a polynomial in the inverse of $\delta(\mathcal{F}(n,d))$ where $\mathcal{F}(n,d)$ denotes the family of functions corresponding to degree d n-variate polynomials over $\mathbb{F}$, seemingly suggesting that the testability and decodability may be consequences of the distance. If so does this phenomenon should extend to the case of other sets $A_i \neq \mathbb{F}$ - does it? We explore this question in this paper.

In what follows we say that the family of degree d n-variate polynomials is locally decodable (resp. testable) if there is bounded $q = q(d)$ and positive $\delta = \delta(d)$ such that $\mathcal{F}(n,d)$ is (δ,q)-locally decodable (resp. testable) for every n. The specific question we address below is when are the family of degree d n-variate polynomials locally decodable and testable when the domain is $\{0,1\}^n$. (We stress that the choice of $\{0,1\}^n$ as domain is for simplicity and is equivalent to the setting of $|A_i| = 2$ for all i. Working with domains of other (and varying) sizes would lead to qualitative changes and we do not consider that setting in this paper.)

1.2 Main Results

Our first result (Theorem 3.2) shows that even the space of degree 1 polynomials is *not locally decodable* over fields of zero characteristic or over fields of large characteristic. This statement already stresses the main difference between the vector space setting (domain being $\mathbb{F}^n$) and the “grid” setting (domain $= \{0,1\}^n$). One key reason underlying this difference is that the domain $\mathbb{F}^n$ has a rich group of symmetries that preserve the space of degree d polynomials, where the space of symmetries is much smaller when the domain is $\{0,1\}^n$. Specifically the space of degree d polynomials over $\mathbb{F}^n$ is “affine-invariant” (invariant under all affine maps from $\mathbb{F}^n$ to $\mathbb{F}^n$). The richness of this group of symmetries is well-known to lead to local decoding algorithms (see for instance [1]) and this explains the local decodability of $\mathcal{F}(n,d)$ over the domain $\mathbb{F}^n$. Of course the absence of this rich group of symmetries does not rule out local decodability — and so some work has to be done to establish Theorem 3.2. We give an overview of the proof in Section 1.3 and then give the proof in Section 5.

Our second result (Theorem 3.3) shows, in contrast, that the class of *degree d polynomials over fields of small characteristic are locally decodable*. Specifically, we show that there is a $q = q(d,p) < \infty$ and $\delta = \delta(d,p) > 0$ such that $\mathcal{F}(n,d)$ over the domain $\{0,1\}^n$ over a (possibly infinite) field $\mathbb{F}$ of characteristic p is (δ,q)-locally decodable. This is perhaps the first local-decodability result for polynomials over infinite fields. A key technical ingredient that leads to this result, which may be of independent interest, is that when $n = 2p^t$ (twice a power of the characteristic of $\mathbb{F}$) and g is a degree d polynomial for $d < n/2$ then $g(0)$ can be determined from the value of g on the ball on Hamming weight $n/2$ (see Lemma 6.1). Again, we give an overview of the proof in Section 1.3 and then give the actual proof in Section 6.

Our final, and main technical, result (Theorem 3.1) shows somewhat surprisingly that *$\mathcal{F}(n,d)$ is always (i.e., over all fields) locally testable*. This leads to perhaps the simplest natural example of a locally testable code that is not locally decodable. We remark there are of course many examples of such codes (see, for instance, the locally testable codes of Dinur [11]) but these are results of careful constructions and in particular not very symmetric. On the other hand $\mathcal{F}(n,d)$ over $\{0,1\}^n$ does possess moderate symmetry and in particular the automorphism group is transitive. We remark that for both our positive results (Theorems 3.3 and 3.1), the algorithms themselves are not obvious and the analysis leads to further interesting questions. We elaborate on these in the next section.

1.3 Overview of proofs

1.3.1 Impossibility of local decoding over fields of large characteristic

In Section 5 we show that even the family of affine functions over $\{0,1\}^n$ is not locally decodable. The main idea behind this construction and proof is to show that the value of a affine function $\ell : \{0,1\}^n \to \mathbb{F}$ at 1^n can not be determined from its values on any set S if $|S|$ is small (specifically $|S| = o(\log n/\log\log n)$) and S contains only "balanced" elements (i.e., $x \in S \Rightarrow |\sum_i x_i - (n/2)| = O(\sqrt{n})$. Since the space of affine functions from $\{0,1\}^n$ to $\mathbb{F}$ forms a vector space, this in turn translates to showing that no set of up to $|S|$ balanced vectors contain the vector 1^n in their affine span (over $\mathbb{F}$) and we prove this in Lemma 5.2.

Going from the above statement to Theorem 3.2 is relatively standard in the case of finite fields. We show that if one picks a random linear function and simply erase its values on imbalanced inputs, this leads to only a small fraction of error, but its value at 1^n is not decodable with $o(\log n/\log\log n)$ queries. (Indeed many of the ingredients go back to the work of [6], who show that a canonical non-adaptive algorithm is effectively optimal for linear codes, though their results are stated in terms of local testing rather than local decoding.) In the case of infinite fields one has to be careful since one can not simply work with functions that are chosen uniformly at random. Instead we work with random linear functions with bounded coefficients. The bound on the coefficients leads to mild complications due to border effects that need care. In the proof of Theorem 5.3, we show how to overcome these complications using a counting (or encoding) argument.

The technical heart of this part is thus the proof of Lemma 5.2 and we give some idea of this proof next. Suppose $S = \{x^1, \ldots, x^t\}$ contained $x^0 = 1^n$ in its affine span and suppose $|\sum_{j=1}^n x_j^i - (n/2)| \le n/s$ for all i. Let $a_1, \ldots, a_t \in \mathbb{F}$ be coefficients such that $x^0 = \sum_i a_i x^i$ with $\sum_i a_i = 1$. Our proof involves reasoning about the size of the coefficients $a_1, \ldots, a_t$. To get some intuition why this may help, note that

$$\frac{n}{2} = \left|\sum_{j=1}^n x_j^0 - \frac{n}{2}\right| = \left|\sum_{i=1}^t a_i \cdot \left(\sum_{j=1}^n x_j^i - \frac{n}{2}\right)\right| \le \sum_{i=1}^t |a_i| \cdot \left|\sum_{j=1}^n x_j^i - \frac{n}{2}\right| \le \frac{n}{s} \cdot \sum_j |a_j|.$$

So in particular if the a_j's are small, specifically if $|a_j| \le 1$ then we conclude $t = \Omega(s)$. But what happens if large a_j's are used? To understand this, we first show that the coefficients need not be too large (as a function of t) - see Lemma 5.1, and then use this to prove Lemma 5.2. The details are in Section 5.1.

1.3.2 Local decodability over fields of small characteristic

The classical method to obtain a q-query local decoder is to find, given a target point $x^0 \in \mathbb{F}^n$, a distribution on queries $x^1, \ldots, x^q \in \mathbb{F}^n$ such that (1) $P(x^0)$ is determined by $P(x^1), \ldots, P(x^q)$ for every degree d polynomial P, and (2) the query x^i is independent of x^0 (so that an oracle f that usually equals P will satisfy $P(x^i) = f(x^i)$ for all i, with probability at least 3/4). Classical reductions used the "2-transitivity" of the underlying space of automorphisms to guarantee that x^i is independent of x^j for every pair $i \neq j \in \{0, \ldots, q\}$ — a stronger property than required! Unfortunately, our automorphism space is not "2-transitive" but it turns out we can still find a distribution that satisfies the minimal needs.

Specifically, in our reduction we identify a parameter $k = k(p,d)$ and map each variable x_ℓ to either y_j or $1 - y_j$ for some $j = j(\ell) \in [k]$. This reduces the n-variate decoding task with oracle access to $f(x_1, \ldots, x_k)$ to a k-variate decoding task with access to the function $g(y_1, \ldots, y_k)$. Since there are only 2^k distinct inputs to g, decoding can solved with at most

2^k queries (if it can be solved at all). The choice of whether x_ℓ is mapped to y_j or $1 - y_j$ is determined by x_j^0 so that $f(x^0) = g(0^k)$. Thus given x^0, the only randomness is in the choice of $j(\ell)$. We choose $j(\ell)$ uniformly and independently from $[k]$ for each ℓ. For $y \in \{0,1\}^k$, x^y denote the corresponding query in $\{0,1\}^n$ (i.e., $g(y) = f(x^y)$). Given our choices, x^y is not independent of x^0 for every choice of y. Indeed if y has Hamming weight 1, then x^y is very likely to have Hamming distance $\approx n/k$ from x^0 which is far from independent. However if $y \in \{0,1\}^k$ is a balanced vector with exactly $k/2$ 1s (so in particular we will need k to be even), then it turns out x^y is indeed independent of x^0. So we query only those x^y for which y is balanced. But this leads to a new challenge: can $P(0^k)$ be determined from the values of $P(y)$ for balanced ys? It turns out that for a careful choice of k (and this is where the small characteristic plays a role) the value of a degree d polynomial at 0 is indeed determined by its values on balanced inputs (see Lemma 6.1) and this turns out to be sufficient to build a decoding algorithm over fields of small characteristic. Details may be found in Section 6.

1.3.3 Local testability over all fields

We now turn to the main technical result of the paper, namely the local testability of polynomials over grids. All previous analyses of local testability of polynomials with query complexity independent of the number of variables have relied on symmetry either implicitly or explicitly. (See for example [16] for further elaboration.) Furthermore many also depend on the local decodability explicitly; and in our setting we seem to have insufficient symmetry and definitely no local decodability. This forces us to choose the test and analysis quite carefully.

It turns out that among existing approaches to analyses of local tests, the one due to Bhattacharyya et al [7] (henceforth BKSSZ) seems to make the least use of local decodability and our hope is to be able to simulate this analysis in our case — but the question remains: “which tester should we use?". This is a non-trivial question since the BKSSZ test is a natural one in a setting with sufficient symmetry; but their analysis relies crucially on the ability to view their test as a sequence of restrictions: Given a function $f : \mathbb{F}^n \to \mathbb{F}$ they produce a sequence of functions $f = f_n, f_{n-1}, \ldots, f_k$, where the function f_r is an r-variate function obtained by restricting f_{r+1} to a codimension one affine subspace. Their test finally checks to see if f_k is a degree d polynomial. To emulate this analysis, we design a somewhat artificial test: We also produce a sequence of functions $f_n, f_{n-1}, \ldots, f_k$ with f_r being an r-variate function. Since we do not have the luxury to restrict to arbitrary subspaces, we instead derive f_r from $f_{r+1}(z_1, \ldots, z_{r+1})$ by setting $z_i = z_j$ or $z_i = 1 - z_j$ for some random pair i, j (since these are the only simple affine restrictions that preserve the domain). We stop when the number of variables k is small enough (and hopefully a number depending on d alone and not on n or $\mathbb{F}$). We then test that the final function has degree d.

The analysis of this test is not straightforward even given previous works, but we are able to adapt the analyses to our setting. Two new ingredients that appear in our analyses are the hypercontractivity of hypercube with the constant weight noise operator (analyzed by Polyanskiy [22]) and the intriguing stochastics of a random set-union problem. We explain our analysis and where the above appear next.

We start with the part which is more immediate from the BKSSZ analysis. This corresponds to a key step in the BKSSZ analysis where it is shown that if f_{r+1} is far from degree d polynomials then, with high probability, so also is f_r. This step is argued via contradiction. If f_r is close to the space of degree d polynomials for many restrictions, then from the many polynomials that agree with f_r (for many of the restrictions) one can glue together an $r + 1$-variate polynomial that is close to f_{r+1}. This step is mostly algebraic and works out in our case also; though the actual algebra is different and involves more cases.

The new part in our analysis is in the case where f_n is moderately close to some low-degree polynomial P. In this case we would still like to show that the test rejects f_n with positive probability. In both BKSSZ and in our analysis this is shown by showing the the 2^k queries into f_n (that given the entire truth table of the function f_k) satisfy the property that exactly f_n is not equal to P on exactly one of the queried points. Note that the value of $f_k(y)$ is obtained by querying f at some point, which we denote x^y. In the BKSSZ analysis x^a and x^b are completely independent given $a \neq b \in \{0,1\}^k$. (Note that the mapping from y to x^y is randomized and depends on the random choices of the tester.) In our setting the behavior of x^a and x^b is more complex and depends on both the set of coordinates j such that where $a_j \neq b_j$ and on the number of indices $i \in [n]$ such that the variable x_i is mapped to variable y_j. Our analysis ends up depending on two new ingredients: (1) The number of variables x_i that map to any particular variable y_j is $\Omega(n/k)$ with probability at least $2^{-O(k)}$. This part involves the analysis of a random set-union process elaborated on below. (2) Once the exact number of indices i such that x_i maps to y_j is fixed for every $j \in [k]$ and none of the sets is too small, the distribution of x^a and x^b is sufficiently independent to ensure that the events $f(x^a) = P(x^a)$ and $f(x^b) = P(x^b)$ co-occur with probability much smaller than the individual probabilities of these events. This part uses the hypercontractivity of the hypercube but under an unusual noise operator corresponding to the “constant weight operator”, fortunately analyzed by Polyanskiy [22]. Invoking his theorem we are able to conclude the proof of this section.

We now briefly expand on the “random set-union” process alluded to above. Recall that our process starts with n variables, and at each stage a pair of remaining variables is identified and given the same name. (We may ignore the complications due to the complementation of the form $z_i = 1 - z_j$ for this part.) Equivalently we start with n sets $X_1, \ldots, X_n$ with $X_i = \{i\}$ initially. We then pick two random sets and merge them. We stop when there are k sets left and our goal is to understand the likelihood that one of the sets turn out to be too tiny. (The expected size of a set is n/k and too tiny corresponds to being smaller than $n/(4k)$.) It turns out that the distribution of set sizes produced by this process has a particularly clean description as follows: Randomly arrange the elements 1 to n on a cycle and consider the partition into k sets generated by the set of elements that start with a special element and end before the next special element as we go clockwise around the cycle, where the elements in $\{1, \ldots, k\}$ are the special ones. The sizes of these partitions are distributed identically to the sizes of the sets S_j! For example, when $k = 2$ the two sets have sizes distributed uniformly from 1 to $n - 1$. In particular the sets size are not strongly concentrated around n/k - but nevertheless the probability that no set is tiny is not too small and this suffices for our analysis.

Details of this analysis may be found in Section 4.

Organization

In Section 2 we start with some preliminaries including the main definitions and some of the tools we will need later. In Section 3 we give a formal statement of our results. In Section 4 we present the local tester over all fields. In Section 5 we sketch our proof that over fields of large (or zero) characteristic, local decoding is not possible. Finally in Section 6 we give a local decoder over fields of small characteristic. Most analysis is omitted from this version and included in the full version of this paper [29].

2 Preliminaries

2.1 Basic notation

Fix a field $\mathbb{F}$ and an $n \in \mathbb{N}$. We consider functions $f : \{0,1\}^n \to \mathbb{F}$ that can be written as *multilinear* polynomials of total degree at most d. We denote this space by $\mathcal{F}(n, d; \mathbb{F})$. The space of all functions from $\{0,1\}^n$ to $\mathbb{F}$ will be denoted simply as $\mathcal{F}(n; \mathbb{F})$. (We will simplify these to $\mathcal{F}(n, d)$ and $\mathcal{F}(n)$ respectively, if the field $\mathbb{F}$ is clear from context.)

Given $f, g \in \mathcal{F}(n)$, we use $\delta(f, g)$ to denote the fractional Hamming distance between f and g. I.e.,

$$\delta(f, g) := \Pr_{x \in \{0,1\}^n} [f(x) \neq g(x)]$$

For a family $\mathcal{F}' \subseteq \mathcal{F}(n)$, we use $\delta(f, \mathcal{F}')$ to denote $\min_{g \in \mathcal{F}'}\{\delta(f, g)\}$. Given an $f \in \mathcal{F}(n)$ and $d \geq 0$, we use $\delta_d(f)$ to denote $\delta(f, \mathcal{F}(n, d))$.

2.2 Local Testers and Decoders

Let $\mathbb{F}$ be any field. We define the notion of a local tester and local decoder for subspaces of $\mathcal{F}(n)$. These notions go back at least to the works of Goldreich and Sudan [12] and Katz and Trevisan [14], though the exact definitions and paramaters may differ here.

▶ **Definition 2.1** (Local tester). Fix $q \in \mathbb{N}$ and $\delta \in (0, 1)$. Let $\mathcal{F}'$ be any subspace of $\mathcal{F}(n)$.

We say that a randomized algorithm T is a (δ, q)-*local tester for* $\mathcal{F}'$ if on an input $f \in \mathcal{F}(n)$, the algorithm does the following.

- T makes at most q queries to f and either accepts or rejects.
- (Completeness) If $f \in \mathcal{F}'$, then T accepts with probability at least $3/4$.
- (Soundness) If $\delta(f, \mathcal{F}') \geq \delta$, then T rejects with probability at least $3/4$.

We say that a tester is *adaptive* if the queries it makes to the input f depend on the answers to its earlier queries. Otherwise, we say that the tester is *non-adaptive*.

▶ **Definition 2.2** (Local decoder). Fix $q \in \mathbb{N}$ and $\delta \in (0, 1)$. Let $\mathcal{F}'$ be any subspace of $\mathcal{F}(n)$.

We say that a randomized algorithm T is a (δ, q)-*local decoder for* $\mathcal{F}'$ if on an input $f \in \mathcal{F}(n)$ and $x \in \{0,1\}^n$, the algorithm does the following.

- T makes at most q queries to f and outputs $b \in \mathbb{F}$.
- If $\delta(f, \mathcal{F}') \leq \delta$, then the output $b = f(x)$ with probability at least $3/4$.

We say that a decoder is *adaptive* if the queries it makes to the input f depend on the answers to its earlier queries. Otherwise, we say that the tester is *non-adaptive*.

2.3 Some basic facts about binomial coefficients

▶ **Fact 2.3.** *For integer parameters* $0 \leq b \leq a$, *let* $\binom{a}{\leq b}$ *denote the size of a Hamming ball of radius* b *in* $\{0,1\}^a$*; equivalently,* $\binom{a}{\leq b} = \sum_{j \leq b} \binom{a}{j}$. *Then, we have*

$$\binom{a}{\leq b} \leq 2^{aH(b/a)}$$

where $H(\cdot)$ *is the binary entropy function.*

2.4 Hypercontractivity theorem for spherical averages.

In this section, let $\mathbb{R}$ be the underlying field. Let $\eta \in (0,1)$ be arbitrary. We define a smoothing operator T_η, which maps $\mathcal{F}(r) = \{f : \{0,1\}^r \to \mathbb{R}\}$ to itself. For $F \in \mathcal{F}(r)$, we define $T_\eta F$ as follows

$$T_\eta F(x) = \mathop{\mathbf{E}}_{J \in \binom{[r]}{\eta r}} [F(x \oplus J)]$$

where $x \oplus J$ is the point $y \in \{0,1\}^r$ obtained by flipping x at exactly the coordinates in J.

Recall that for any $F \in \mathcal{F}(r)$ and any $p \geq 1$, $\|F\|_p$ denotes $\mathbf{E}_{x \in \{0,1\}^r} [|F(x)|^p]^{1/p}$.

We will use the following hypercontractivity theorem of Polanskiy [22].

▶ **Theorem 2.4** (Follows from Theorem 1 in [22])**.** *Assume that $\eta \in [1/20, 19/20]$ and $\eta_0 = 1/20$. For any $F \in \mathcal{F}(r)$, we have*

$$\|T_\eta F\|_2 \leq C \cdot \|F\|_p$$

for $p = 1 + (1 - 2\eta_0)^2$ and C is an absolute constant.

▶ **Corollary 2.5.** *Assume that η_0, η are as in the statement of Theorem 2.4 and let $\delta \in (0,1)$ be arbitrary. Say $E \subseteq \{0,1\}^r$ s.t. $|E| \leq \delta \cdot 2^r$. Assume that $(x', x'') \in \{0,1\}^r$ are chosen as follows: $x' \in \{0,1\}^r$ and $I' \in \binom{[r]}{\eta r}$ are chosen i.u.a.r., and we set $x'' = x' \oplus I'$. Then we have*

$$\Pr_{x', I'} [x' \in E \wedge x'' \in E] \leq C \cdot \delta^{1 + (1/40)}$$

where C is the constant from Theorem 2.4.

Proof. Let $F : \{0,1\}^n \to \{0,1\} \subseteq \mathbb{R}$ be the indicator function of the set E. Note that we have

$$\Pr_{x', I'} [x' \in E \wedge x'' \in E] = \mathop{\mathbf{E}}_{x', I'} [F(x') F(x' \oplus I')] = \mathop{\mathbf{E}}_{x'} [F(x') T_\eta F(x')] \, .$$

By the Cauchy-Schwarz inequality and Theorem 2.4 we get

$$\mathop{\mathbf{E}}_{x'} [F(x') T_\eta F(x')] \leq \|F\|_2 \cdot C \cdot \|F\|_p \tag{1}$$

for $p = 1 + (1 - 2\eta_0)^2$. Note that we have

$$\begin{aligned} \|F\|_p &\leq \delta^{1/p} = \delta^{\frac{1}{1 + (1 - 2\eta_0)^2}} \\ &= \delta^{\frac{1}{2(1 - 2\eta_0(1 - \eta_0))}} \leq (\sqrt{\delta})^{1 + \min\{\eta_0, 1 - \eta_0\}} = \sqrt{\delta}^{1 + (1/20)} \end{aligned}$$

where for the last inequality we have used the fact that for $\eta_0 \in [0,1]$ we have

$$\frac{1}{1 - 2\eta_0(1 - \eta_0)} \geq 1 + 2\eta_0(1 - \eta_0) \geq 1 + \min\{\eta_0, 1 - \eta_0\}.$$

Putting the upper bound on $\|F\|_p$ together with the fact that $\|F\|_2 \leq \sqrt{\delta}$ and (1), we get the claim. ◀

3 Results

We show upper and lower bounds for testing and decoding polynomial codes over grids. All our upper bounds hold in the non-adaptive setting, while our lower bounds hold in the stronger adaptive setting.

Our first result is that for any choice of the field $\mathbb{F}$ (possibly even infinite), the space of functions $\mathcal{F}(n, d)$ is locally testable. More precisely, we show the following.

▶ **Theorem 3.1** ($\mathcal{F}(n, d)$ has a local tester for any field). *There exists a constant $c < \infty$ and polynomial $p_0(x)$ such that the following holds for every field $\mathbb{F}$, every non-negative integer d, every positive integer n and every real number $\varepsilon > 0$: The space $\mathcal{F}(n, d; \mathbb{F})$ has a non-adaptive (ε, q)-local tester for $q \leq 2^{c \cdot d} \cdot p_0(1/\varepsilon)$.*

In contrast, we show that the space $\mathcal{F}(n, d)$ is *not* locally *decodable* over fields of large characteristic, even for $d = 1$.

▶ **Theorem 3.2** ($\mathcal{F}(n, d)$ does not have a local decoder for large characteristic). *For every $\varepsilon > 0$ there exists $c_\varepsilon > 0$ such that the following holds: Let $n \in \mathbb{N}$ and let $\mathbb{F}$ be a field such that either $\mathrm{char}(\mathbb{F}) = 0$ or $\mathrm{char}(\mathbb{F}) \geq n^2$. Then any adaptive (ε, q)-local decoder for $\mathcal{F}(n, 1; \mathbb{F})$ must satisfy $q \geq c_\varepsilon \cdot \log n / \log \log n$.*

Complementing the above result, we can show that if $\mathrm{char}(\mathbb{F})$ is a constant, then in fact the space $\mathcal{F}(n, d)$ does have a local decoding procedure.

▶ **Theorem 3.3** ($\mathcal{F}(n, d)$ has a local decoder for constant characteristic). *There exists a constant $c < \infty$ such that for every field $\mathbb{F}$ of characteristic p, every non-negative integer d and every positive integer n, the space $\mathcal{F}(n, d; \mathbb{F})$ has a non-adaptive $(2^{-c \cdot p \cdot d}, 4^{p \cdot d})$-local decoder.*

4 A local tester for $\mathcal{F}(n, d)$ over any field

We now present our local tester and its analysis. The reader may find the overview from Section 1.3 helpful while reading the below.

We start by introducing some notation for this section. Throughout, fix any field $\mathbb{F}$. We consider functions $f : \{0, 1\}^I \to \mathbb{F}$ where I is a finite set of positive integers and indexes into the set of variables $\{X_i \mid i \in I\}$. We denote this space as $\mathcal{F}(I)$. Similarly, $\mathcal{F}(I, d)$ is defined to be the space of functions of degree at most d over the variables indexed by I.

The following is the test we use to check if a given function $f : \{0, 1\}^I \to \mathbb{F}$ is close to $\mathcal{F}(I, d)$.

Test $T_{k,I}(f_I)$

Notation. Given two variables X and Y and $a \in \{0, 1\}$, “replacing X by $a \oplus Y$” refers to substituting X by Y if $a = 0$ and by $1 - Y$ if $a = 1$.

- If $|I| > k$, then
  - Choose a random $a \in \{0, 1\}$ and distinct $i_0, j_0 \in I$ at random and replace X_{j_0} by $a \oplus X_{i_0}$. Let f'_I denote the resulting restriction of f_I.
  - Run $T_{k,I \setminus \{j_0\}}(f'_I)$ and output what it outputs.
- If $|I| = k$ then
  - Choose a uniformly random bijection $\sigma : I \to [k]$.

- Choose an $a \in \{0,1\}^k$ uniformly at random.
- Replace each X_i ($i \in I$) with $Y_{\sigma(i)} \oplus a_i$.
- Check if the restricted function $g(Y_1, \ldots, Y_k) \in \mathcal{F}(k,d)$ by querying g on all its inputs. Accept if so and reject otherwise.

▶ Remark. It is not strictly necessary to choose a *random* bijection σ in the test $T_{k,I}$ and a fixed bijection $\sigma : I \to [k]$ would do just as well. However, the above leads to a cleaner reformulation of the test.

▶ **Observation 4.1.** *Test $T_{k,I}$ has query complexity 2^k.*

▶ **Observation 4.2.** *If $f_I \in \mathcal{F}(I,d)$, then $T_{k,I}$ accepts with probability 1.*

The following theorem is the main result of this section and implies Theorem 3.1 from Section 3.

▶ **Theorem 4.3.** *For each positive integer d, there is a $k = O(d)$ and $\varepsilon_0 = 1/2^{O(d)}$ such that for any I of size at least $k+1$ and any $f_I \in \mathcal{F}(I)$,*

$$\Pr\left[\textit{Test } T_{k,I} \textit{ rejects } f_I\right] \geq \frac{1}{2^{O(d)}} \cdot \min\{\delta_d(f_I), \varepsilon_0\}.$$

Theorem 3.1 immediately follows from Theorem 4.3 since to get an $(\varepsilon, 2^{O(d)})$-tester, we repeat the test $T_{k,[n]}$ $t = 2^{O(d)} \cdot \mathrm{poly}(1/\varepsilon)$ many times and accept if and only if each iteration of the test accepts. If the input function $f \in \mathcal{F}(n)$ is of degree at most d, this test accepts with probability 1. Otherwise, this test rejects with probability at least 3/4 for suitably chosen t as above. The number of queries made by the test is $2^k \cdot t = 2^{O(d)} \cdot \mathrm{poly}(1/\varepsilon)$.

5 Impossibility of local decoding when $\mathrm{char}(\mathbb{F})$ is large

In this section, we prove Theorem 5.3 which is a more detailed version of Theorem 3.2. Again we remind the reader that an overview may be found in Section 1.3.

Let n be a growing parameter and $\mathbb{F}$ a field of characteristic 0 or positive characteristic greater than n^2. For the results in this section, it will be easier to deal with the domain $\{-1,1\}^n$ rather than $\{0,1\}^n$. Since there a natural invertible linear map that maps $\{0,1\}$ to $\{-1,1\}$ (i.e. $a \mapsto 1-2a$), this change of input space is without loss of generality.

5.1 Local linear spans of balanced vectors

Let $u \in \mathbb{F}^n$ and $U \subseteq \mathbb{F}^n$. For any integer $t \in \mathbb{N}$, we say that u is in the t-span of U if it can be written as a linear combination of at most t elements of U. For $x \in \{-1,1\}^n$, we use $|x|$ to denote the sum of the entries of x over $\mathbb{Z}$. In this section, we wish to show that if the vector 1^n is in the t-span of balanced vectors, i.e., vectors x with $|x| \leq n/s$ then t is must be growing as a function of s.

As explained earlier we first establish a bound on the size of the solutions of linear equations in systems over $\mathbb{Q}$ with few variables or few constraints. This fact is well-known, but we prove it here for completeness.

▶ **Lemma 5.1.** *Let $r, s \in \mathbb{N}$ and let $t = \min\{r,s\}$. Let $Mx = u$ be a system of linear equations with $M \in \{-1,0,1\}^{r \times s}$ and $u \in \{-1,0,1\}^r$.*

- *If $\mathbb{F}$ is a field of characteristic zero and the system has a solution in $\mathbb{F}^s$, then there exist integers $a_1, \ldots, a_s, b \in \mathbb{Z}$ with $|a_i|, |b| \leq t!$ such that $x_i = a_i/b$ is a solution to $Mx = u$. In particular, there is a solution in $\mathbb{Q}^s$.*

- *If $\mathbb{F}$ is a field of characteristic p and if the system has a solution in $\mathbb{F}^s$, then there exist integers $a_1, \ldots, a_s, b \in \mathbb{Z}$ with $|a_i|, |b| \leq t!$ such that $x_i = a_i/b \pmod p$ is a solution to $Mx = u$. In particular, there is a solution in $\mathbb{F}_p^s$.*

Proof omitted from this version.

We now turn to the main technical lemma of this section showing that 1^n is not in linear span of a small number of nearly balanced elements of $\{-1, 1\}^n$.

▶ **Lemma 5.2.** *Let $n, s = s(n) \in \mathbb{N}$ with $s(n) \leq n$. Let $S = \{x \in \{-1,1\}^n \mid |x| \leq n/s)\}$. Then $x^0 = 1^n$ is not in the t-span of S unless $t \geq \log s / \log\log s$ provided $\mathbb{F}$ is field of zero characteristic or of characteristic $p \geq 2n^2$.*

Proof. We first consider the case when $\mathbb{F}$ is of zero characteristic. Note that in this case $\mathbb{Q} \subseteq \mathbb{F}$. Suppose $x^0 \in \mathsf{Span}\{x^1, \ldots, x^t\}$ with $x^0 = \sum_{i=1}^n c_i x^i$. Note that the c_i's are expressible as the solution to a linear system whose $Mz = u$ where M and u have entries in $\{-1, 0, 1\}$ and M is a $n \times t$ matrix. By Lemma 5.1 we have that $c_i \in \mathbb{Q}$ with $|c_i| \leq t!$ (more specifically we have $c_i = a_i/b$ with $|a_i| \leq t!$ and this implies $|c_i| \leq t!$). We thus have

$$n = \left|\sum_{j=1}^n x_j^0\right| = \left|\sum_{i=1}^t c_i \sum_{j=1}^n x_j^i\right| \leq \sum_{i=1}^t |c_i| \cdot \left|\sum_{j=1}^n x_j^i\right| \leq \sum_{i=1}^t (t!) \cdot (n/s) \leq (t+1)! \cdot (n/s).$$

We thus conclude that $(t+1)! \geq s$ and thus $t \geq \log s / \log\log s$.

In the case of finite field $\mathbb{F}$, we proceed as above and let $x^0 = \sum_{i=1}^t c_i x^i$. By Lemma 5.1 we have that there are integers a_i, b with $|a_i|, |b| \leq t!$ such that $c_i = a_i/b \pmod p$ is a solution to $x^0 = \sum_{i=1}^t c_i x^i$. Now consider $b \cdot n$ and we get $b \cdot n = \sum_{i=1}^t a_i \sum_{j=1}^n x_j^i \pmod p$. We now show that this implies $(t+1)! \geq \min\{p/(2n), s\} = s$ (where the equality follows from $p \geq 2n^2$ and $s \leq n$). Assume $(t+1)! \leq p/(2n)$. Then we have $n \leq |b \cdot n| \leq t! \cdot n < p/2$ over the integers, and $\left|\sum_{i=1}^t a_i \sum_{j=1}^n x_j^i\right| \leq (t+1)!(n/s) < p/2$ also over the integers. We again conclude that $n \leq (t+1)!(n/s)$ and so $(t+1)! \geq s$ as claimed. The lemma follows. ◀

We now state the main result of this section which immediately implies Theorem 3.2.

▶ **Theorem 5.3.** *Let $n \in \mathbb{N}$ be a growing parameter and $\varepsilon \in (0,1)$ such that $\varepsilon \geq 2\exp(-n/2s^2)$ for some $s \in \mathbb{N}$ with $100 \leq s \leq \sqrt{n}/100$. Let $\mathbb{F}$ be any field such that either $\mathrm{char}(\mathbb{F}) = 0$ or $\mathrm{char}(\mathbb{F}) \geq n^2$. Then any adaptive (ε, q)-local decoder for $\mathcal{F}(n,1)$ that corrects an ε fraction of errors must satisfy $q = \Omega(\log s / \log\log s)$.*

Proof omitted from this version.

6 Local decoding when char($\mathbb{F}$) is small

In this section, we give a local decoder over fields of small characteristic. An overview of this construction may be found in Section 1.3.

Let p be a prime of constant size and let $\mathbb{F}$ be any (possibly infinite) field of characteristic p. Let d be the degree parameter and k be the smallest power of p that is strictly greater than d. Note that $k \leq pd$. We show that the space $\mathcal{F}(n,d)$ has a $(1/(4 \cdot \binom{2k}{k}), \binom{2k}{k})$-local decoder, hence proving Theorem 3.3.

The main technical tool we use is a suitable linear relation on the space $\mathcal{F}(2k, d)$, which we describe now. We say that a set $S \subseteq \{0,1\}^{2k}$ is *useful* if for every polynomial $G \in \mathcal{F}(2k, d)$, $G(0^{2k})$ is determined by the restriction of the function G to the inputs in S. Let $B \subseteq \{0,1\}^{2k}$ denote the set of all balanced inputs (i.e. inputs of Hamming weight exactly k).

▶ **Lemma 6.1.** *Fix d, k as above. Then the set $B \subseteq \{0,1\}^{2k}$ of balanced inputs is useful.*

The proof of the above lemma will use Lucas' theorem, which we recall below.

▶ **Theorem 6.2** (Lucas' theorem). *Let p be any prime and $a, b \in \mathbb{N}$. Let $a_1, \ldots, a_\ell \in \{0, \ldots, p-1\}$ and $b_1, \ldots, b_\ell \in \{0, \ldots, p-1\}$ be the digits in the p-ary expansion of a and b, i.e., $a = \sum_{j \in [\ell]} a_j p^{j-1}$ and $b = \sum_{j \in [\ell]} b_j p^{j-1}$. Then, we have*

$$\binom{a}{b} \equiv \prod_{i \leq \ell} \binom{a_i}{b_i} \pmod p$$

where $\binom{a_i}{b_i}$ is defined to be 0 if $a_i < b_i$.

▶ **Corollary 6.3.** *For $i \in \{0, \ldots, d\}$, we have $\binom{d+k-i}{k-i} \not\equiv 0 \pmod p$ if and only if $i = 0$.*

Proof omitted from this version.

Proof of Lemma 6.1. Fix any $G \in \mathcal{F}(2k, d)$. Assume that

$$G(Y_1, \ldots, Y_{2k}) = \sum_{I \subseteq [2k] : |I| \leq d} \alpha_I Y^I$$

where Y^I denotes $\prod_{i \in I} Y_i$.

Let B' denote all those inputs in B where the last $k - d$ bits are set to 0. We will compute the sum of G on inputs from B'. But let us first consider a monomial Y^I and see what its sum over $y \in B'$ looks like. The monomial evaluates to 1 on $y \in B'$ if $y_i = 1$ for every $i \in I$, and evaluates to 0 otherwise. There are exactly $\binom{d+k-|I|}{k-|I|}$ choices of $y \in B'$ that satisfy $y_i = 1$ for every $i \in I$. Thus summing over $y \in B'$ we get $\sum_{y \in B'} y^I = \binom{d+k-|I|}{k-|I|}$. Summing over all monomials we get:

$$\begin{aligned} \sum_{y \in B'} G(y) &= \sum_{I \subseteq [2k] : |I| \leq d} \alpha_I \cdot \sum_{y \in B'} Y^I \\ &= \sum_{I \subseteq [2k] : |I| \leq d} \alpha_I \cdot \binom{d+k-|I|}{k-|I|} \end{aligned} \tag{2}$$

By Corollary 6.3, it follows that for $i \in \{0, \ldots, d\}$, we have

$$\binom{d+k-i}{k-i} \not\equiv 0 \pmod p$$

if and only if $i = 0$ and so $\sum_{y \in B'} G(y) = \binom{d+k}{k} \cdot \alpha_\emptyset$. Let $c = \binom{d+k}{k} \pmod p$. We have $c \in \mathbb{F}_p^* \subseteq \mathbb{F}^*$ and in particular c is invertible in $\mathbb{F}$, and $\sum_{y \in B'} G(y) = c \cdot \alpha_\emptyset = c \cdot G(0^{2k})$. Hence, we get $G(0^{2k}) = c^{-1} \cdot \sum_{y \in B'} G(y)$. Therefore, $G(0^{2k})$ is determined by the restriction of G to B' and hence also by its restriction to B. ◀

We now show that $\mathcal{F}(n, d)$ has a $(1/(4 \cdot \binom{2k}{k}), \binom{2k}{k})$-local decoder.

The decoder. We now give the formal description of the decoder. Let the decoder be given oracle access to f with the promise that f is $1/(4 \cdot \binom{2k}{k})$-close to some $F \in \mathcal{F}(n, d)$. Let the input to the decoder be $x \in \{0,1\}^n$. The problem is to find $F(x)$.

We describe the decoder below:

Decoder $D_k^f(x)$.

- Partition $[n]$ into $2k$ parts by choosing a *uniformly* random map $h : [n] \to [2k]$. I.e. each $h(j)$ is chosen i.u.a.r. from $[2k]$.
- For $i \in [2k]$ and $j \in [n]$ such that $h(j) = i$, identify X_j with $Y_i \oplus x_j$.
- Let $g(Y_1, \ldots, Y_{2k})$ and $G(Y_1, \ldots, Y_{2k})$ be the restrictions of f and F respectively. Assuming $g|_B = G|_B$, query g at all inputs in B and decode $G(0^{2k})$ from $G|_B$. Output the value decoded.

The main theorem of this section is the following. Note that this implies Theorem 3.3.

▶ **Theorem 6.4.** *Let $\mathbb{F}$ be a field of characteristic p. For integer $d \geq 0$, let k be the smallest power of p greater than d. Then the decoder D_k is a $(1/(4 \cdot \binom{2k}{k}), \binom{2k}{k})$-local decoder for $\mathcal{F}(n, d; \mathbb{F})$.*

Proof omitted from this version.

References

1 Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron. Testing reed-muller codes. *IEEE Trans. Information Theory*, 51(11):4032–4039, 2005. doi:10.1109/TIT.2005.856958.

2 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998. doi:10.1145/278298.278306.

3 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998. doi:10.1145/273865.273901.

4 Donald Beaver and Joan Feigenbaum. Hiding instances in multioracle queries. In C. Choffrut and T. Lengauer, editors, *Proceedings of the 7th Annual Symposium on Theoretical Aspects of Computer Science*, pages 37–48, Rouen, France, 22–24 February 1990. Springer. Lecture Notes in Computer Science, Volume 415.

5 Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 1–10. ACM, 1988. doi:10.1145/62212.62213.

6 Eli Ben-Sasson, Prahladh Harsha, and Sofya Raskhodnikova. Some 3cnf properties are hard to test. *SIAM J. Comput.*, 35(1):1–21, 2005. doi:10.1137/S0097539704445445.

7 Arnab Bhattacharyya, Swastik Kopparty, Grant Schoenebeck, Madhu Sudan, and David Zuckerman. Optimal testing of reed-muller codes. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 488–497. IEEE Computer Society, 2010. doi:10.1109/FOCS.2010.54.

8 Manuel Blum, Ashok K. Chandra, and Mark N. Wegman. Equivalence of free boolean graphs can be decided probabilistically in polynomial time. *Inf. Process. Lett.*, 10(2):80–82, 1980. doi:10.1016/S0020-0190(80)90078-2.

9 Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47(3):549–595, 1993.

10 Richard A. DeMillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Inf. Process. Lett.*, 7(4):193–195, 1978. doi:10.1016/0020-0190(78)90067-4.

11 Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007. doi:10.1145/1236457.1236459.

12 Oded Goldreich and Madhu Sudan. Locally testable codes and pcps of almost-linear length. *J. ACM*, 53(4):558–655, 2006. doi:10.1145/1162349.1162351.

13 Charanjit S. Jutla, Anindya C. Patthak, Atri Rudra, and David Zuckerman. Testing low-degree polynomials over prime fields. *Random Struct. Algorithms*, 35(2):163–193, 2009. doi:10.1002/rsa.20262.

14 Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In F. Frances Yao and Eugene M. Luks, editors, *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 80–86. ACM, 2000. doi:10.1145/335305.335315.

15 Tali Kaufman and Dana Ron. Testing polynomials over general fields. *SIAM J. Comput.*, 36(3):779–802, 2006. doi:10.1137/S0097539704445615.

16 Tali Kaufman and Madhu Sudan. Algebraic property testing: the role of invariance. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 403–412. ACM, 2008. doi:10.1145/1374376.1374434.

17 John Y. Kim and Swastik Kopparty. Decoding reed-muller codes over product sets. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPIcs*, pages 11:1–11:28. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPIcs.CCC.2016.11.

18 Richard Lipton. New directions in testing. In *Distributed Computing and Cryptography*, volume 2 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 191–202. AMS, 1991.

19 Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992. doi:10.1145/146585.146605.

20 D. E. Muller. Application of Boolean algebra to switching circuit design and to error detection. *IEEE Transactions on Computers*, 3:6–12, 1954.

21 Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987. doi:10.1007/BF02579206.

22 Yury Polyanskiy. Hypercontractivity of spherical averages in Hamming space. *CoRR*, abs/1309.3014, 2013.

23 Alexander A. Razborov. On the method of approximations. In David S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 167–176. ACM, 1989. doi:10.1145/73007.73023.

24 Irving S. Reed. A class of multiple-error-correcting codes and the decoding scheme. *IEEE Transactions on Information Theory*, 4:38–49, 1954.

25 Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, April 1996.

26 Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980. doi:10.1145/322217.322225.

27 Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979. doi:10.1145/359168.359176.

28 Adi Shamir. Ip=pspace. In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I*, pages 11–15. IEEE Computer Society, 1990. doi:10.1109/FSCS.1990.89519.

29 Srikanth Srinivasan and Madhu Sudan. Local decoding and testing of polynomials over grids. *CoRR*, abs/1709.06036, 2017. arXiv:1709.06036.

30 Richard Zippel. Probabilistic algorithms for sparse polynomials. In Edward W. Ng, editor, *Symbolic and Algebraic Computation, EUROSAM '79, An International Symposiumon Symbolic and Algebraic Computation, Marseille, France, June 1979, Proceedings*, volume 72 of *Lecture Notes in Computer Science*, pages 216–226. Springer, 1979. doi:10.1007/3-540-09519-5_73.