List-decoding algorithms for lifted codes

Alan Guo ∗ Swastik Kopparty †

November 26, 2014

Abstract

Lifted Reed-Solomon codes are a natural affine-invariant family of error-correcting codes which generalize Reed-Muller codes. They were known to have efficient local-testing and local-decoding algorithms (comparable to the known algorithms for Reed-Muller codes), but with significantly better rate. We give efficient algorithms for list-decoding and local list-decoding of lifted codes. Our algorithms are based on a new technical lemma, which says that codewords of lifted codes are low degree polynomials when viewed as univariate polynomials over a big field (even though they may be very high degree when viewed as multivariate polynomials over a small field).

1 Introduction

By virtue of their many powerful applications in complexity theory, there has been much interest in the study of error-correcting codes which support “local” operations. The operations of interest include local decoding, local testing, local correcting, and local list-decoding. Error correcting codes equipped with such local algorithms have been useful, for example, in proof-checking, private information retrieval, and hardness amplification.

The canonical example of a code which supports all the above local operations is the Reed-Muller code, which is a code based on evaluations of low-degree polynomials. Reed-Muller codes have nontrivial local algorithms across a wide range of parameters. In this paper, we will be interested in the constant rate regime. For a long time, Reed-Muller codes were the only known codes in this regime supporting nontrivial locality. Concretely, for every constant integer m and every constant $R < \frac{1}{m}$, there are Reed-Muller codes of arbitrarily large length n, rate R, constant relative distance δ, which are locally decodable/testable/correctable from $(\frac{1}{2} - \epsilon) \cdot \delta$ fraction fraction errors using $O(n^{1/m})$ queries. In particular, no nontrivial locality was known for Reed-Muller codes (or any other codes, until recently) with rate $R > 1/2$.

In the last few years, new families of codes were found which had interesting local algorithms in the high rate regime (i.e., with rate $R$ near 1). These codes include multiplicity codes [KSY11, Kop12], lifted codes [GKS13, Guo13], expander codes [HOW13] and tensor codes [Vid10]. Of these, lifted codes are the only ones that are both locally decodable and locally testable. This paper gives new and improved decoding and testing algorithms for lifted codes.

∗CSAIL, Massachusetts Institute of Technology, 32 Vassar Street, Cambridge, MA, USA. aguo@mit.edu. Research supported in part by NSF grants CCF-0829672, CCF-1065125, and CCF-6922462, and an NSF Graduate Research Fellowship

†Department of Mathematics & Department of Computer Science, Rutgers University. swastik.kopparty@rutgers.edu. Research supported in part by a Sloan Fellowship and NSF CCF-1253886.
1.1 Lifted Codes and our Main Result

Lifted codes are a natural family of algebraic, affine-invariant codes which generalize Reed-Muller codes. We give a brief introduction to these codes now. Let $q$ be prime power, let $d < q$ and let $m > 1$ be an integer. Define alphabet $\Sigma = \mathbb{F}_q$. We define the lifted code $C = C(q, d, m)$ to be a subset of $\Sigma^{\mathbb{F}_q^m}$, the space of functions from $\mathbb{F}_q^m$ to $\mathbb{F}_q$. A function $f : \mathbb{F}_q^m \rightarrow \mathbb{F}_q$ is in $C$ if for every line $L \subseteq \mathbb{F}_q^m$, the restriction of $f$ to $L$ is a univariate polynomial of degree at most $d$. Note that if $f$ is the evaluation table of an $m$-variate polynomial of degree $\leq d$, then $f$ is automatically in $C$. The surprising (and useful) fact is that if $d$ is large and $\mathbb{F}_q$ has small characteristic, then $C$ has significantly more functions. This leads to its improved rate relative to the corresponding Reed-Muller code, which only contains the evaluation tables of low degree polynomials.

Our main result is an algorithm for list-decoding and local list-decoding of lifted codes. We show that lifted codes of distance $\delta$ can be efficiently list-decoded and locally list-decoded (in sublinear-time) upto their “Johnson radius” $(1 - \sqrt{1 - \delta})$. Combined with the local testability of lifted codes, this also implies that lifted codes can be locally tested in the high-error regime, upto the Johnson radius.

It is well known that Reed-Muller codes can be list decoded and locally list-decoded upto the Johnson radius [PW04, STV99]. Our result shows that a lifted code, which is a natural algebraic supercode of Reed-Muller codes, despite having a vastly greater rate than the corresponding Reed-Muller code, loses absolutely nothing in terms of any (local) algorithmic decoding / testing properties.

In the appendix, we also prove two other results as part of the basic toolkit for working with lifted codes.

- **Explicit interpolating sets**: For a lifted code $C$, we give a strongly explicit subset $S$ of $\mathbb{F}_q^m$ such that for every $g : S \rightarrow \mathbb{F}_q$, there is a unique lifted codeword $f : \mathbb{F}_q^m \rightarrow \mathbb{F}_q$ from $C$ with $f|_S = g$. The main interest in explicit interpolating sets for us is that it allows us to convert the sublinear-time local correction algorithm for lifted codes into a sublinear-time local decoding algorithm for lifted codes (earlier the known sublinear-time local correction, only implied low-query-complexity local decoding, without any associated sublinear-time local decoding algorithm).

- **Simple local decoding upto half the minimum distance**: We note that there is a simple algorithm for local decoding of lifted codes upto half the minimum distance. This is a direct translation of the elegant lines-weight local decoding algorithm for matching-vector codes [BET10] to the Reed-Muller code / lifted codes setting.

1.2 Methods

We first talk about the (global) list-decoding algorithm. The main technical lemma underlying this algorithm says that codewords of lifted codes are low-degree in a certain sense.

The codewords of a lifted code are in general very high degree as $m$-variate polynomials over $\mathbb{F}_q$. There is a description of these codes in terms of spanning monomials [GKS13], but it is not even clear from this description that lifted codes have good distance. The handle that we get on lifted codes arises by considering the big field $\mathbb{F}_{q^m}$, and letting $\phi$ be an $\mathbb{F}_q$-linear isomorphism between $\mathbb{F}_{q^m}$

---

1 Technically we are talking about lifted Reed-Solomon codes, but for brevity we refer to them as lifted codes.

2 To locally list-decode all the way upto the Johnson bound, one actually needs a variant of [STV99] given in [BK09].
and $F_{q^m}$. Given a function $f : F_{q^m} \rightarrow F_q$, we can consider the composed function $f \circ \phi : F_{q^m} \rightarrow F_q$, and view it as a function from $F_{q^m} \rightarrow F_{q^m}$. Our technical lemma says that this function $f \circ \phi$ is low-degree as a univariate polynomial over $F_{q^m}$ (irrespective of the choice of the map $\phi$).

Through this lemma, we reduce the problem of list-decoding lifted codes over the small field $F_q$ to the problem of list-decoding univariate polynomials (i.e., Reed-Solomon codes) over the large field $F_{q^m}$. This latter problem can be solved using the Guruswami-Sudan algorithm [GS99].

Our local list-decoding algorithm uses the above list-decoding algorithm. Following [AS03, STV99, BK09], local list-decoding of $m$-variate Reed-Muller codes over $F_q$ reduces to (global) list-decoding of $t$-variate Reed-Muller codes over $F_q$ (for some $t < m$). For the list-decoding radius to approach the Johnson radius, one needs $t \geq 2$. This is where the above list-decoding algorithm gets used.

**Organization of this paper** Section 2 introduces notation and preliminary definitions and facts to be used in later proofs. Section 3 proves our main technical result, that lifted RS codes over domain $F_{q^m}$ are low degree when viewed as univariate polynomials over $F_{q^m}$, as well as the consequence for global list decoding. Section 4 presents and analyzes the local list decoding algorithm for lifted RS codes, along with the consequence for local testability. Appendix A describes the explicit interpolating sets for arbitrary lifted affine-invariant codes. Appendix B presents and analyzes the local correction algorithm up to half the minimum distance.

## 2 Preliminaries

### 2.1 Notation

For a positive integer $n$, we use $[n]$ to denote the set $\{1, \ldots, n\}$. For sets $A$ and $B$, we use $\{A \rightarrow B\}$ to denote the set of functions mapping $A$ to $B$.

For a prime power $q$, $F_q$ is the finite field of size $q$. We think of a code $C \subseteq \{F_Q^n \rightarrow F_q\}$ as a family of functions $f : F_Q^n \rightarrow F_q$, where $F_Q$ is an extension field of $F_q$, but each codeword is a vector of evaluations $(f(x))_{x \in F_Q}$ assuming some canonical ordering of elements in $F_Q$; we abuse notation and say $f \in C$ to mean $(f(x))_{x \in F_Q} \in C$.

If $f : F_q^m \rightarrow F_q$ and line $\ell$ is a line in $F_q$, this formally means $\ell$ is specified by some $a, b \in F_q$ and the restriction of $f$ to $\ell$, denoted by $f|_{\ell}$, means the function $t \mapsto f(a + bt)$. Similarly, if $P$ is a plane, then it is specified by some $a, b, c \in F_q$ and the restriction of $f$ to $P$, denoted by $f|_P$, means the function $(t, u) \mapsto f(a + bt + cu)$.

### 2.2 Interpolating sets and decoding

**Definition 2.1 (Interpolating set).** A set $S \subseteq F_Q^n$ is an interpolating set for $C$ if for every $\hat{f} : S \rightarrow F_q$ there exists a unique $f \in C$ such that $f|_S = \hat{f}$.

Note that if $S$ is an interpolating set for $C$, then $|C| = q^{|S|}$.

**Definition 2.2 (Local decoding).** Let $\Sigma$ be an alphabet and let $C : \Sigma^k \rightarrow \Sigma^n$ be an encoding map. A $(\rho, l)$-local decoding algorithm for $C$ is a randomized algorithm $D : [k] \rightarrow \Sigma$ with oracle access to an input word $r \in \Sigma^n$ and satisfies the following:
1. If there is a message \( m \in \Sigma^k \) such that \( \delta(C(m), r) \leq \rho \), then for every input \( i \in [k] \), we have \( \Pr[D^r(i) = m_i] \geq \frac{2}{3} \).

2. On every input \( i \in [k] \), \( D^r(i) \) always makes at most \( l \) queries to \( r \).

We call \( \rho \) the fraction of errors decodable, or the decoding radius, and we call \( l \) the query complexity.

**Definition 2.3** (Local correction). Let \( C \subseteq \Sigma^n \) be a code. A \((\rho, l)\)-local correction algorithm for \( C \) is a randomized algorithm \( C : [n] \to \Sigma \) with oracle access to an input word \( r \in \Sigma^n \) and satisfies the following:

1. If there is a codeword \( c \in \Sigma^n \) such that \( \delta(c, r) \leq \rho \), then for every input \( i \in [n] \), we have \( \Pr[C^r(i) = c_i] \geq \frac{2}{3} \).

2. On every input \( i \in [n] \), \( C^r(i) \) always makes at most \( l \) queries to \( r \).

As before, \( \rho \) is the decoding radius and \( l \) is the query complexity.

The definition and construction of interpolating sets is motivated by the fact that if we have an explicit interpolating set for a code \( C \), then we have an explicit systematic encoding for \( C \), which allows us to easily transform a local correction algorithm into a local decoding algorithm.

**Definition 2.4** (List decoding). Let \( C \subseteq \Sigma^n \) be a code. A \((\rho, L)\)-list decoding algorithm for \( C \) is an algorithm which takes as input a received word \( r \in \Sigma^n \) and outputs a list \( \mathcal{L} \subseteq \Sigma^n \) of size \( |\mathcal{L}| \leq L \) containing all \( c \in C \) such that \( \delta(c, r) \leq \rho \). The parameter \( \rho \) is the list-decoding radius and \( L \) is the list size.

**Definition 2.5** (Local list decoding). Let \( C \subseteq \Sigma^n \) be a code. A \((\rho, L, l)\)-local list decoding algorithm for \( C \) is a randomized algorithm \( A \) with oracle access to an input word \( r \in \Sigma^n \) and outputs a collection of randomized oracles \( A_1, \ldots, A_L \) with oracle access to \( r \) satisfying the following:

1. With high probability, it holds that for every \( c \in C \) such that \( \delta(c, r) \leq \rho \), there exists a \( j \in [L] \) such that for every \( i \in [n] \), \( \Pr[A^r_j(i) = c_i] \geq \frac{2}{3} \).

2. \( A \) makes at most \( l \) queries to \( r \), and on any input \( i \in [n] \) and for every \( j \in [L] \), \( A^r_j \) makes at most \( l \) queries to \( r \).

As before, \( \rho \) is the list decoding radius, \( L \) is the list size, and \( l \) is the query complexity.

### 2.3 Affine-invariant codes

**Definition 2.6** (Affine-invariant code). A code \( C \subseteq \{F_Q^n \to F_q \} \) is affine-invariant if for every \( f \in C \) and affine permutation \( A : F_Q^n \to F_Q^n \), the function \( x \mapsto f(A(x)) \) is in \( C \).

**Definition 2.7** (Degree set). For a function \( f : F_Q \to F_q \), written as \( f = \sum_{d=0}^{Q-1} f_d X^d \), its support is \( \text{supp}(f) := \{d \in \{0, \ldots, Q-1\} \mid f_d \neq 0\} \). If \( C \subseteq \{F_Q \to F_q \} \) is an affine-invariant code, then its degree set \( \deg(C) \) is

\[
\deg(C) := \bigcup_{f \in C} \text{supp}(f).
\]

**Proposition 2.8** ([BGM+11]). If \( C \subseteq \{F_q^m \to F_q \} \) is an affine-invariant code, then \( \dim_{F_q}(C) = |\deg(C)| \).

In particular, if \( S \) is an interpolating set for an affine-invariant code \( C \subseteq \{F_q^m \to F_q \} \), then \( |S| = |\deg(C)| \). Proposition 2.8 will be used in Appendix A.
2.4 Lifted codes

**Definition 2.9** (Lift). Let $\mathcal{C} \subseteq \{F_q \rightarrow F_q\}$ be an affine-invariant code. For integer $m \geq 2$, the $m$-th dimensional lift of $\mathcal{C}$, $\text{Lift}_m(\mathcal{C})$, is the code

$$\text{Lift}_m(\mathcal{C}) := \{f : F_q^m \rightarrow F_q \mid f|_{\ell} \in \mathcal{C} \text{ for every } \ell \in F_q^m\}$$

Let $\text{RS}(q,d)$ be the Reed-Solomon code of degree $d$ over $F_q$,

$$\text{RS}(q,d) := \{f : F_q \rightarrow F_q \mid \deg(f) \leq d\}.$$  

**Definition 2.10** (Lifted Reed-Solomon code). The $m$-variate lifted Reed-Solomon code of degree $d$ over $F_q$ is the code

$$\text{LiftedRS}(q,d,m) := \text{Lift}_m(\text{RS}(q,d)).$$

For positive integers $d,e$, we say $e$ is the $p$-shadow of $d$, or $e \leq_p d$, if $d$ dominates $e$ digit-wise in base $p$: in other words, if $d = \sum_{i \geq 0} d(i)p^i$ and $e = \sum_{i \geq 0} e(i)p^i$ are the $p$-ary representations, then $e(i) \leq d(i)$ for all $i \geq 0$. A vector $(e_1, \ldots, e_m)$ lies in the $p$-shadow of $d$ if for every $(f_1, \ldots, f_m)$ such that $f_i \leq_p e_i$ for $i \in [m]$, it holds that $\sum_{i=1}^m f_i \leq d$. The following fact motivates these definitions.

**Proposition 2.11** (Lucas’ theorem). Let $e_1, \ldots, e_m$ be positive integers and $d = e_1 + \cdots + e_m$ and let $p$ be a prime. The multinomial coefficient $\binom{d}{e_1, \ldots, e_m} = \frac{d!}{e_1! \cdots e_m!}$ is nonzero modulo $p$ only if $(e_1, \ldots, e_m) \leq_p d$.

For positive integers $a$ and $b$, we define the mod-star operator by

$$a (\text{mod}^* b) = \begin{cases} a \quad &a \leq b \\ a \mod b &a > b \end{cases}$$

motivated by the fact that $X^d$ defines the same function as $X^d (\text{mod}^* q - 1)$ over $F_q$.

**Proposition 2.12** ([GKS13]). The lifted Reed-Solomon code $\text{LiftedRS}(q,d,m)$ is spanned by monomials $\prod_{i=1}^m X_i^{d_i}$ such that for every $e_i \leq_p d_i$, $i \in [m]$, we have $\sum_{i=1}^m e_i (\text{mod}^* q - 1) \leq d$.

2.5 Finite field isomorphisms

Let $\text{Tr} : F_q^m \rightarrow F_q$ be the $F_q$-linear trace map $z \mapsto \sum_{i=0}^{q-1} z^i$. Let $\alpha_1, \ldots, \alpha_m \in F_q^m$ be linearly independent over $F_q$ and let $\phi : F_q^m \rightarrow F_q^m$ be the map $z \mapsto (\text{Tr}(\alpha_1 z), \ldots, \text{Tr}(\alpha_m z))$. Since $\text{Tr}$ is $F_q$-linear, $\phi$ is an $F_q$-linear map and in fact it is an isomorphism. Observe that $\phi$ induces a $F_q$-linear isomorphism $\phi^* : \{F_q^m \rightarrow F_q\} \rightarrow \{F_q^m \rightarrow F_q\}$ defined by $\phi^*(f)(x) = f(\phi(x))$ for all $x \in F_q^m$.

3 Global list decoding

In this section, we present an efficient global list decoding algorithm for $\text{LiftedRS}(q,d,m)$. Define $\alpha_1, \ldots, \alpha_m \in F_q^m$, $\phi$, and $\phi^*$ as in Section 2.5. Our main result states that $\text{LiftedRS}(q,d,m) \subseteq \{F_q^m \rightarrow F_q\}$ is isomorphic to a subcode of $\text{RS}(q^m, (d + m)q^{m-1}) \subseteq \{F_q^m \rightarrow F_q\}$. In particular, one can simply list decode $\text{LiftedRS}(q,d,m)$ by list decoding $\text{RS}(q^m, (d + m)q)$ up to the Johnson radius. We will use this algorithm for $m = 2$ as a subroutine in our local list decoding algorithm in Section 4.
Theorem 3.1. If \( f \in \text{LiftedRS}(q,d,m) \), then \( \deg(\phi^*(f)) \leq (d+m)q^{m-1} \).

Proof. By linearity, it suffices to prove this for a monomial \( f(X_1, \ldots, X_m) = \prod_{i=1}^{m} X_i^{d_i} \). We have

\[
\phi^*(f)(Z) = \sum_{(e_1, \ldots, e_m) \leq \rho d_1} \cdots \sum_{(e_m, \ldots, e_m) \leq \rho d_m} (\cdots) Z^\sum_{i,j} e_{ij} q^{m-i},
\]

so it suffices to show that \( \sum_{j=1}^{m} \sum_{i=1}^{m} e_{ij} q^{m-j} \mod q^m - 1 \leq (d+m)q^{m-1} \). By Proposition 2.12, for every \( e_i \leq d_i \), \( i \in [m] \), we have \( \sum_{i=1}^{m} e_i (\mod^* q - 1) \leq d \). Therefore, there is some integer \( 0 \leq k < m \) such that \( \sum_{i=1}^{m} e_i \in [kq, k(q - 1) + d] \). Thus,

\[
kq^m \leq q^{m-1} \sum_{i=1}^{m} e_{i1} + \sum_{j=2}^{m} \sum_{i=1}^{m} e_{ij} q^{m-j}
\]

\[
\leq q^{m-1} \sum_{i=1}^{m} e_{i1} + q^{m-2} \sum_{j=2}^{m} \sum_{i=1}^{m} e_{ij}
\]

\[
\leq (k(q - 1) + d)q^{m-1} + mq^{m-1}
\]

\[
= k(q^m - 1) + (d + m - k)q^{m-1} + k
\]

\[
\leq k(q^m - 1) + (d + m)q^{m-1}
\]

and hence \( \sum_{j=1}^{m} \sum_{i=1}^{m} e_{ij} q^{m-j} \mod^* q^m - 1 \leq (d+m)q^{m-1} \).

Corollary 3.2. There is a polynomial time global list decoding algorithm for \( \text{LiftedRS}(q,d,m) \) that decodes up to \( 1 - \sqrt{\frac{d+m}{q}} \) fraction errors. In particular, if \( m = O(1) \) and \( d = (1-\delta)q \), then \( \delta(\text{LiftedRS}(q,d,m)) = \delta - o(1) \) and the list decoding algorithm decodes up to \( 1 - \sqrt{1-\delta} - o(1) \) fraction errors as \( q \rightarrow \infty \).

Proof. Given \( r : \mathbb{F}_q^m \rightarrow \mathbb{F}_q \), convert it to \( r' = \phi^*(r) \), and then run the Guruswami-Sudan list decoder for \( \text{RS} := \text{RS}(q^m, (d+m)q^{m-1}, m) \) on \( r' \) to obtain a list \( \mathcal{L} \) with the guarantee that any \( f \in \text{RS} \) with \( \delta(r', f) \leq 1 - \sqrt{\frac{d+m}{q}} \) lies in \( \mathcal{L} \). We require that any \( f \in \text{LiftedRS}(q,d,m) \) satisfying \( \delta(r', f) \leq 1 - \sqrt{\frac{d+m}{q}} \) lies in \( \mathcal{L} \), and this follows immediately from Theorem 3.1.

4 Local list decoding

In this section, we present a local list decoding algorithm for \( \text{LiftedRS}(q,d,m) \), where \( d = (1-\delta)q \) which decodes up to radius \( 1 - \sqrt{1-\delta} - \epsilon \) for any constant \( \epsilon > 0 \), with list size \( \text{poly}(\frac{1}{\epsilon}) \) and query complexity \( q^3 \).

Local list decoder: Oracle access to received word \( r : \mathbb{F}_q^m \rightarrow \mathbb{F}_q \).

1. Pick a random line \( \ell \) in \( \mathbb{F}_q^m \).

2. Run Reed-Solomon list decoder (e.g. Guruswami-Sudan) on \( r|_\ell \) from \( 1 - \sqrt{1-\delta} - \frac{\epsilon}{2} \) fraction errors to get list \( g_1, \ldots, g_L : \mathbb{F}_q \rightarrow \mathbb{F}_q \) of Reed-Solomon codewords.
3. For each $i \in [L]$, output $\text{Correct}(A_{\ell,g_i})$

where $\text{Correct}$ is a local correction algorithm for the lifted codes for $0.1\delta$ fraction errors, and $A$ is an oracle which takes as advice a line and a univariate polynomial and simulates oracle access to a function which is supposed to be $\ll 0.1\delta$ close to a lifted RS codeword.

**Oracle $A_{\ell,g}(x)$:**

1. If $\ell$ contains $x$, i.e. $\ell = \{a + bt \mid t \in \mathbb{F}_q\}$ for some $a, b \in \mathbb{F}^m_q$ and $x = at + b$, then output $g(t)$.
2. Otherwise, let $P$ be the plane containing $\ell$ and $x$, parametrized by $\{a + bt + (x-a)u \mid t, u \in \mathbb{F}_q\}$.
   
   (a) Use the global list decoder for bivariate lifted RS code given above to list decode $r|_P$ from $1 - \sqrt{1-\delta - \frac{2}{q}}$ fraction errors and obtain a list $L$.
   
   (b) If there exists a unique $h \in L$ such that $h|_\ell = g$, output $h(0,1)$, otherwise fail.

**Analysis:** To show that this works, we just have to show that, with high probability over the choice of $\ell$, for every lifted RS codeword $f$ such that $\delta(r,f) \leq 1 - \sqrt{1-\delta - \epsilon}$, there is $i \in [L]$ such that $\text{Correct}(A_{\ell,g_i}) = f$, i.e. $\delta(A_{\ell,g_i}, f) \leq 0.1\delta$.

We will proceed in two steps:

1. First, we show that with high probability over $\ell$, there is some $i \in [L]$ such that $f|_\ell = g_i$.
2. Next, we show that $\delta(A_{\ell,f|_\ell}, f) \leq 0.1\delta$.

For the first step, note that $f|_\ell \in \{g_1, \ldots, g_L\}$ if $\delta(f|_\ell, r|_\ell) \leq 1 - \sqrt{1-\delta - \frac{2}{q}}$. Note that $\delta(f|_\ell, r|_\ell)$ has mean $1 - \sqrt{1-\delta - \epsilon}$ with variance less than $\frac{1}{q}$ (by pairwise independence of points on a line), so by Chebyshev’s inequality the probability that $\delta(f|_\ell, r|_\ell) \leq 1 - \sqrt{1-\delta - \frac{2}{q}}$ is $1 - o(1)$.

For the second step, we want to show that $\Pr_{x \in \mathbb{F}^m_q}[A_{\ell,f|_\ell}(x) \neq f(x)] \leq 0.1\delta$. First consider the probability when we randomize $\ell$ as well. We get $A_{\ell,f|_\ell}(x) = f(x)$ as long as $f|_P \in L$ and no element $h \in L$ has $h|_\ell = f|_\ell$. With probability $1 - o(1)$, $\ell$ does no contain $x$, and conditioned on this, $P$ is a uniformly random plane. It samples the space $\mathbb{F}^m_q$ well, so with probability $1 - o(1)$ we have $\delta(f|_P, r|_P) \leq 1 - \sqrt{1-\delta - \frac{2}{q}}$ and hence $f|_P \in L$. For the probability that no two codewords in $L$ agree on $\ell$, view this as first choosing $P$, then choosing $\ell$ within $P$. The list size $|L|$ is a constant, polynomial in $1/\epsilon$. So we just need to bound the probability that two bivariate lifted RS codewords agree on a uniformly random line. This is at most $2/q$, since each line must divide the difference of two codewords, which has degree at most $2q$. Thus, with probability $1 - o(1)$, $f|_P$ is the unique codeword in $L$ which is consistent with $f|_\ell$ on $\ell$. Therefore,

$$\Pr_\ell[\delta(A_{\ell,f|_\ell}, f|_\ell) > 0.1\delta] = \Pr_\ell[\Pr_x[A_{\ell,f|_\ell}(x) \neq f(x)] > 0.1\delta]$$

$$\leq \frac{\Pr_{\ell,x}[A_{\ell,f|_\ell}(x) \neq f(x)]}{0.1\delta} = o(1).$$

As a corollary, we get the following testing algorithm.
**Theorem 4.1.** For any $\alpha < \beta < 1 - \sqrt{1 - \delta}$, there is an $O(q^4)$-query algorithm which, given oracle access to a function $r : \mathbb{F}_q^m \to \mathbb{F}_q$, distinguishes between the cases where $r$ is $\alpha$-close to LiftedRS$(d,m)$ and where $r$ is $\beta$-far.

**Proof.** Let $\rho = (\alpha + \beta)/2$ and let $\epsilon = (\beta - \alpha)/8$, so that $\alpha = \rho - 4\epsilon$ and $\beta = \rho + 4\epsilon$. Let $T$ be a local testing algorithm for LiftedRS$(q,d,m)$ with query complexity $q$, which distinguishes between codewords and words that are $\epsilon$-far from the code. The algorithm is to run the local list decoding algorithm on $r$ with error radius $\rho$ such that $\alpha < \rho < \beta$, to obtain a list of oracles $M_1, \ldots, M_L$. For each $M_i$, we use random sampling to estimate the distance between $r$ and the function computed by $M_i$ to within $\epsilon$ additive error, and keep only the ones with estimated distance less than $\rho + \epsilon$. Then, for each remaining $M_i$, we run $T$ on $M_i$. We accept if $T$ accepts some $M_i$, otherwise we reject.

If $r$ is $\alpha$-close to LiftedRS$(q,d,m)$, then it is $\alpha$-close to some codeword $f$, and by the guarantee of the local list decoding algorithm there is some $j \in [l]$ such that $M_j$ computes $f$. Moreover, this $M_j$ will not be pruned by our distance estimation. Since $f$ is a codeword, this $M_j$ will pass the testing algorithm and so our algorithm will accept.

Now suppose $r$ is $\beta$-far from LiftedRS$(q,d,m)$, and consider any oracle $M_i$ output by the local list decoding algorithm and pruned by our distance estimation. The estimated distance between $r$ and the function computed by $M_i$ is at most $\rho + \epsilon$, so the true distance is at most $\rho + 2\epsilon$. Since $r$ is $\beta$-far from any codeword, that means the function computed by $M_i$ is $(\beta - (\rho + 2\epsilon)) > \epsilon$-far from any codeword, and hence $T$ will reject $M_i$.

All of the statements made above were deterministic, but the testing algorithm $T$ and distance estimation are randomized procedures. However, at a price of constant blowup in query complexity, we can make their failure probabilities arbitrarily small constants, so that by a union bound the distance estimations and tests run by $T$ simultaneously succeed with large constant probability. \qed

**References**


By linearity, it suffices to show that if so suppose $\text{Deg}(C)_{\overrightarrow{F}}$ suffices to show that $S$. Proof. The map $\phi$ induces a map $\phi^*: \{\mathbb{F}_q^m \rightarrow \mathbb{F}_q\} \rightarrow \{\mathbb{F}_q^m \rightarrow \mathbb{F}_q\}$ defined by $\phi^*(f) = f \circ \phi$. It suffices to show that $S$ is an interpolating set for $C' \triangleq \phi^*(C)$. Observe that $C'$ is affine-invariant over $\mathbb{F}_q^m$, and let $\text{Deg}(C') = \{i \mid \exists f \in C \ i \in \text{supp}(f)\}$. By Proposition 2.8, $\dim_{\mathbb{F}_q}(C') = |\text{Deg}(C')|$, so suppose $\text{Deg}(C') = \{i_1, \ldots, i_D\}$. Every $g \in C'$ is of the form $g(z) = \sum_{j=1}^D a_j z^{i_j}$, where $a_j \in \mathbb{F}_q^m$. By linearity, it suffices to show that if $g \in C'$ is nonzero, then $g(z) \neq 0$ for some $z \in S$. We have

$$\begin{bmatrix}
\omega^{i_1} & \omega^{i_2} & \cdots & \omega^{i_D} \\
\omega^{2i_1} & \omega^{2i_2} & \cdots & \omega^{2i_D} \\
\vdots & \vdots & \ddots & \vdots \\
\omega^{Di_1} & \omega^{Di_2} & \cdots & \omega^{Di_D}
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_2 \\
\vdots \\
a_D
\end{bmatrix}
= 
\begin{bmatrix}
g(\omega) \\
g(\omega^2) \\
\vdots \\
g(\omega^D)
\end{bmatrix}$$

and the leftmost matrix is invertible since it’s a generalized Vandermonde matrix. Therefore, if $g \neq 0$, then the right-hand side, which is simply the vector of evaluations of $g$ on $S$, is nonzero. \hfill \square
Local unique decoding upto half minimum distance

**Theorem B.1.** Let $C \subseteq \{\mathbb{F}_q \to \mathbb{F}_q\}$ be an affine-invariant code of distance $\delta$. For every positive integer $m \geq 2$ and for every $\epsilon > 0$, there exists a local correction algorithm for $\text{Lift}_m(C)$ with query complexity $O(q/\epsilon^2)$ that corrects up to $(\frac{1}{2} - \epsilon) \delta - \frac{1}{q}$ fraction errors.

**Proof.** Let $\text{Corr}_C$ be a correction algorithm for $C$, so that for every $f : \mathbb{F}_q \to \mathbb{F}_q$ that is $\delta/2$-close to some $g \in C$, $\text{Corr}_C(f) = g$. The following algorithm is a local correction algorithm that achieves the desired parameters.

**Local correction algorithm:** Oracle access to received word $r : \mathbb{F}_q^m \to \mathbb{F}_q$.

1. Let $c = \lceil \frac{4\ln 6}{e^2} \rceil$ and pick $a_1, \ldots, a_c \in \mathbb{F}_q^m$ independent and uniformly at random.

2. For each $i \in [c]$:
   
   (a) Set $r_i(t) := r(x + a_i t)$.
   
   (b) Compute $s_i := \text{Corr}_C(r_i)$ and $\delta_i := \delta(r_i, s_i)$.
   
   (c) Assign the value $s_i(0)$ a weight $W_i := \max \left(1 - \frac{\delta_i}{\delta/2}, 0\right)$.

3. Set $W := \sum_{i=1}^c W_i$. For every $\alpha \in \mathbb{F}_q$, let $w(\alpha) := \frac{1}{W} \sum_{i : s_i(0) = \alpha} W_i$. If there is an $\alpha \in \mathbb{F}_q$ with $w(\alpha) > \frac{1}{2}$, output $\alpha$, otherwise fail.

**Analysis:** Fix a received word $r : \mathbb{F}_q^m \to \mathbb{F}_q$ that is $(\tau - \frac{1}{q})$-close from a codeword $c \in \text{Lift}_m(C)$, where $\tau = (\frac{1}{2} - \epsilon) \delta$. The query complexity follows from the fact that the algorithm queries $O(1/\epsilon^2)$ lines, each consisting of $q$ points. Fix an input $x \in \mathbb{F}_q^m$. We wish to show that, with probability at least $2/3$, the algorithm outputs $c(x)$, i.e. $w(c(x)) > \frac{1}{2}$.

Consider all lines $\ell$ passing through $x$. For each such line $\ell$, define the following:

- $\tau_\ell := \delta(r|_\ell, c|_\ell)$
- $s_\ell := \text{Corr}_C(r|_\ell)$
- $\delta_\ell := \delta(r|_\ell, s_\ell)$
- $W_\ell := \max \left(1 - \frac{\delta_\ell}{\delta/2}, 0\right)$
- $X_\ell = \begin{cases} W_\ell & s_\ell = c|_\ell \\ 0 & s_\ell \neq c|_\ell. \end{cases}$

Let $p := \Pr[\ell | s_\ell = c|_\ell]$. Note that if $s_\ell = c_\ell$, then $\delta_\ell = \tau_\ell$, otherwise $\delta_\ell \geq \delta - \tau_\ell$. Hence, if $s_\ell = c_\ell$, then $W_\ell \geq 1 - \frac{\tau_\ell}{\delta/2}$, otherwise $W_\ell \leq \frac{\tau_\ell}{\delta/2} - 1$. 

\textcopyright 2023
Define
\[
\begin{align*}
\tau_{\text{good}} &= \mathbb{E}[\tau_{\ell} \mid s_{\ell} = c|]\ell \\
\tau_{\text{bad}} &= \mathbb{E}[\tau_{\ell} \mid s_{\ell} \neq c|]\ell \\
W_{\text{good}} &= \mathbb{E}[W_{\ell} \mid s_{\ell} = c|]\ell \\
W_{\text{bad}} &= \mathbb{E}[W_{\ell} \mid s_{\ell} \neq c|]\ell
\end{align*}
\]

Observe that
\[
\begin{align*}
\mathbb{E}[\tau_{\ell}] &\leq \frac{1 + (\tau - \frac{1}{q})(q - 1)}{q} \leq \tau \\
\mathbb{E}[X_{\ell}] &= p \cdot W_{\text{good}} \\
\mathbb{E}[W_{\ell}] &= p \cdot W_{\text{good}} + (1 - p) \cdot W_{\text{bad}}.
\end{align*}
\]

We claim that
\[
p \cdot W_{\text{good}} \geq (1 - p) \cdot W_{\text{bad}} + 2\epsilon. \quad (6)
\]
To see this, we start from
\[
\left(\frac{1}{2} - \epsilon\right) \delta = \tau \geq \mathbb{E}[\tau_{\ell}] = p \cdot \tau_{\text{good}} + (1 - p) \cdot \tau_{\text{bad}}.
\]
Dividing by \(\delta/2\) yields
\[
1 - 2\epsilon \geq p \cdot \frac{\tau_{\text{good}}}{\delta/2} + (1 - p) \cdot \frac{\tau_{\text{bad}}}{\delta/2}.
\]
Re-writing \(1 - 2\epsilon\) on the left-hand side as \(p + (1 - p) - 2\epsilon\) and re-arranging, we get
\[
p \cdot \left(1 - \frac{\tau_{\text{good}}}{\delta/2}\right) \geq (1 - p) \cdot \left(\frac{\tau_{\text{bad}}}{\delta/2} - 1\right) + 2\epsilon.
\]
The left-hand side is bounded from above by \(p \cdot W_{\text{good}}\) while the right-hand side is bounded from below by \((1 - p) \cdot W_{\text{bad}} + 2\epsilon\), hence (6) follows.

For each \(i \in [c]\), let \(\ell_{i}\) be the line \(\{x + a_{i}t \mid t \in \mathbb{F}_{q}\}\). Note that the \(X_{\ell}\) are defined such that line \(i\) contributes weight \(X_{\ell_{i}}/W\) to \(w(c(x))\), so it suffices to show that, with probability at least 2/3,
\[
\frac{\sum_{i=1}^{c} X_{\ell_{i}}}{\sum_{i=1}^{c} W_{\ell_{i}}} > \frac{1}{2}.
\]
Each \(X_{\ell}, W_{\ell} \in [0, 1]\), so by Hoeffding’s inequality,
\[
\begin{align*}
\Pr\left[\frac{1}{c} \sum_{i=1}^{c} X_{\ell_{i}} - \mathbb{E}[X_{\ell}] > \epsilon/2\right] &\leq \exp(-\epsilon^2c/4) \leq 1/6 \\
\Pr\left[\frac{1}{c} \sum_{i=1}^{c} W_{\ell_{i}} - \mathbb{E}[W_{\ell}] > \epsilon/2\right] &\leq \exp(-\epsilon^2c/4) \leq 1/6.
\end{align*}
\]
Therefore, by a union bound, with probability at least $2/3$ we have, after applying (6),

$$\frac{\sum_{i=1}^{c} X_i}{\sum_{i=1}^{c} W_i} \geq \frac{\mathbb{E}[X_\ell] - \alpha}{\mathbb{E}[W_\ell] + \epsilon/2}$$

$$\geq \frac{p \cdot W_{\text{good}} - \alpha}{p \cdot W_{\text{good}} + (1 - p) \cdot W_{\text{bad}} + \epsilon/2}$$

$$\geq \frac{(1 - p) \cdot W_{\text{bad}} + 3\epsilon/2}{2(1 - p) \cdot W_{\text{bad}} + 5\epsilon/2}$$

$$> \frac{1}{2}. \quad \Box$$