Electronic Colloquium on Computational Complexity, Report No. 164 (2026)

Algorithmic List Decoding of Reed–Solomon Codes up to Capacity in the Low-Rate Regime

Joshua Brakensiek* Yeyuan Chen† Aaron Putterman‡ Zihan Zhang§
Kai Zhe Zheng¶

Abstract

We provide a deterministic polynomial-time list decoding algorithm for Reed–Solomon codes over prime fields that approaches list decoding capacity on every evaluation set in the low (constant) rate regime.

1 Introduction

In the field of coding theory, the most fundamental error-correcting code is the Reed–Solomon (RS) code [RS60] which has found numerous practical applications including in data storage [PS85, Pla97, PLS+09, GW16], wireless communication [WHPH87, Wic92, KV03, GKKG06], and cryptography [Sha79, MS81, CCD88, RB89, Ped91, FY92, JS06b, Gol07, SW08, BGW19, BBHR18, ACFY24, ACFY25]. Simply stated, a Reed–Solomon interprets a message to be transmitted as the coefficients of a low-degree univariate polynomial which is then evaluated at many evaluation points to introduce redundancy. More formally, given a finite field $\mathbb{F}_q$ and distinct evaluation points $\alpha_1, \dots, \alpha_n \in \mathbb{F}_q$, one can construct a Reed–Solomon code of block length n and rate $R := k/n$ as follows:

$$\mathsf{RS}_{n,k}(\alpha_1, \dots, \alpha_n) = \left\{(P(\alpha_1), \dots, P(\alpha_n)) \in \mathbb{F}_q^n : P \in \mathbb{F}_q[X] \text{ and } \deg P < k\right\}.$$

A defining characteristic of Reed–Solomon codes is that they are maximum distance separable (MDS). In other words, every nonzero codeword $\vec{c} \in \mathsf{RS}_{n,k}(\alpha_1, \dots, \alpha_n)$ has Hamming weight at least $n - k + 1$. Equivalently, the code has relative distance $\delta = \frac{n-k+1}{n}$, perfectly meeting the Singleton bound [Sin64]. From the perspective of error-correction, where we receive a message $\vec{y} \in \mathbb{F}_q^n$, Reed–Solomon codes have optimal *unique decoding*. Concretely, if there exists a codeword $\vec{c} \in \mathsf{RS}_{n,k}(\alpha_1, \dots, \alpha_n)$ with Hamming distance at most $\lfloor \frac{n-k}{2} \rfloor$ (approximately $(1 - R)/2$ relative distance) from $\vec{y}$, then $\vec{c}$ is unique and in fact can be found efficiently [Pet60, GZ61, Ber68, Mas69].

A much more ambitious quest by the coding theory community has been to understand the decodability of Reed–Solomon codes beyond the unique decoding regime, namely in the paradigm of *list decoding*, introduced independently by Elias [Eli57] and Wozencraft [Woz58]. In this setting, we seek to find a list

*University of California, Berkeley. josh.brakensiek@berkeley.edu Supported in part by a Simons Investigator award of Venkatesan Guruswami, and NSF awards CCF-2211972 and DMS-2503280.

†Department of EECS, University of Michigan, Ann Arbor. yeyuanch@umich.edu Supported in part by NSF award CCF-2236931.

‡Harvard University, Cambridge, Massachusetts. aputterman@g.harvard.edu Supported in part by a Jane Street Graduate Research Fellowship, the Simons Investigator awards of Madhu Sudan and Salil Vadhan, and AFOSR award FA9550-25-1-0112.

§Simons Institute for the Theory of Computing, Berkeley and Institute for Advanced Study, Princeton. zzhsdj@foxmail.com

¶Simons Institute for the Theory of Computing, Berkeley and Institute for Advanced Study, Princeton. kzzheng@mit.edu

ISSN 1433-8092

of codewords which are within some specified Hamming distance of our corrupted message. By relaxing this notion of recovery, a much larger fraction of errors can be recovered from [Sud97, GS99]. The concept of list decoding has proved to be remarkably versatile, finding applications across the whole spectrum of theoretical computer science such as pseudorandomness and randomness extraction [STV01, Tre01, GUV09], average-case complexity and hardness amplification [GL89, GRS00], cryptography and related combinatorial problems [SSW01, AGS03, JS06a, AGKS07], and interactive proof systems [BCI+23, BCH+26].

We say that a Reed–Solomon code is (ρ, L) list decodable if for every possible message $y \in \mathbb{F}_q^n$, there are at most L codewords which have relative Hamming distance at most ρ from y. Unlike unique decoding which fails to exist beyond a radius of $(1-R)/2$, list decoding is a meaningful notion up to a radius of $1-R$, known as *list-decoding capacity* [GRS12, Theorem 7.4.1] — beyond a radius of $1-R$, a list of size exponential in n is always required. More precisely, we say that our Reed-Solomon code attains list-decoding capacity if for any $\varepsilon > 0$, our code is $(1-R-\varepsilon, n^{O_\varepsilon(1)})$ list decodable. A parameter regime of particular interest in this paper is the *low-rate regime* (or *high-noise regime*) where rate $R = \Theta(\varepsilon)$ and (relative) radius $\rho = 1-\varepsilon$.

In general, it is not understood which Reed–Solomon codes achieve list-decoding capacity. For the past three decades, the benchmark for the Reed–Solomon and other classes of codes has been the *Johnson radius* of $1-\sqrt{R}$, for which the seminal works of Sudan [Sud97] and Guruswami and Sudan [GS99] gave polynomial-time list-decoding algorithms[1] for RS codes, culminating in efficient decoding up to the *Johnson radius* $1-\sqrt{R}$ up to lower-order terms. This falls short of the list-decoding capacity $1-R$, leaving a substantial gap between what is information-theoretically possible and what is known to be efficiently achievable for RS codes.

The Johnson radius has served as a hard barrier for progress in the field for good reason. Several works provided evidence of barriers to list decoding RS codes substantially beyond the Johnson radius [RR03, GR06, CW07, BKR10]. In particular, Ben-Sasson, Kopparty, and Radhakrishnan [BKR10] exhibited, over fields of bounded characteristic, RS codes with superpolynomially large lists at decoding radius approaching the Johnson bound in certain parameter regimes. Complementing these combinatorial obstructions, several works established computational hardness for RS decoding at substantially larger decoding radius [GV05, CW07, GGG18]. On the positive side, a recent line of work has shown that RS codes with random evaluation points achieve list-decoding capacity with the optimal list size [RW14, ST23, GLS+21, GST23, BGM24, GZ23, AGL24, AGG+25].[2] In fact, stronger results [BGM24, GZ23, AGL24, AGG+25] attaining the generalized Singleton bound of Shangguan and Tamo [ST23] are known for every fixed list size. In contrast, no explicit family of RS codes is known to achieve comparable combinatorial guarantees. Moreover, even for the RS codes with random evaluation points covered by these results, no efficient algorithm is known to attain their beyond-Johnson decoding radius.

In part due to the lack of progress on list-decoding of RS codes beyond the Johnson codes, many coding theorists considered variants of RS codes such as Parvaresh–Vardy codes [PV05], Folded Reed-Solomon Codes [GR08], and Univariate Multiplicity codes [KSY14]. In particular, foleded Reed–Solomon codes introduced by Gurusawmi and Rudra [GR08] are the first explicit family of codes that can be efficiently list decoded up to capacity, albeit with a field size $n^{\Theta(1/\varepsilon)}$ when ε-close to capacity.

A subsequent line of work further developed and strengthened the folded Reed–Solomon framework, improving various aspects of its list-decoding guarantees [GW13, Kop15, KRSW23, GHKS24, Sri25, CZ25, AHS26]. Related ideas have also led to several other explicit capacity-achieving algebraic codes such as [GR21, GX22].

Despite this remarkable recent progress, the following fundamental question, dating back to the work of

[1] Recent work of Chatterjee, Harsha, and Kumar [CHK25] gives a deterministic polynomial-time list decoding algorithm up to the Johnson radius.

[2] These are combinatorial results: they bound the number of codewords in a Hamming ball, without providing an efficient algorithm for finding them.

Guruswami and Sudan, remains open.

Can Reed–Solomon codes of rate R be efficiently list decoded beyond the Johnson radius $1 - \sqrt{R}$?

In fact, even an explicit Reed–Solomon code with combinatorial list-decoding bound beyond the Johnson radius remained unknown [3].

In this paper, we resolve this question in the affirmative for sufficiently low but constant rates over prime fields. Specifically, we give a polynomial-time algorithm that list decodes Reed–Solomon codes substantially beyond the Johnson radius (even achieving list decoding capacity!) in this regime. The result holds for arbitrary sets of distinct evaluation points, with no randomness assumption. A simplified statement of our main result is given in the following informal theorem.

Theorem 1.1 (Main Result, Informal Version in Low (Constant) Rate Regime)**.** *Fix a constant parameter $\theta \in (0, 1)$. Then there exists a constant $\varepsilon_0(\theta) > 0$ such that for every constant ε with $\varepsilon_0(\theta) > \varepsilon > 0$, any Reed–Solomon code with arbitrary distinct evaluation points over a prime field $\mathbb{F}_q$ of size $q = \Theta_{\theta,\varepsilon}(n)$ and rate*

$$R \leq (1 - \theta)\varepsilon$$

can be efficiently list decoded from up to $1 - \epsilon$ fraction of errors, with list size bounded by $n^{\mathrm{poly}_\theta(1/\epsilon)}$.

We remark that the Johnson radius only has rate $\Omega(\epsilon^2)$ with respect to the radius $1 - \epsilon$. Below, we discuss the origins of our new ideas and provide an overview of the techniques. See Theorem 4.1 for a more precise asymptotic statement. It is notable to mention that in the regime for which $n \gg \Theta_{\theta,\varepsilon}(n/k)$, we can actually select $q = n$. In particular, for small constant rates, our algorithm works for Reed–Solomon codes where the evaluation points are precisely $\mathbb{F}_q$ itself.

1.1 Origins of the Main Ideas and Technique Overview

In the list-decoding problem, we are given the evaluation points $(\alpha_i \in \mathbb{F}_q)_{i=1}^n$, the received word $(y_i \in \mathbb{F}_q)_{i=1}^n$, the target agreement A, and the code dimension k. The goal is to find all polynomials $P \in \mathbb{F}_q[X]_{\leq k-1}$ (i.e., with degree at most $k - 1$) such that $\sum_{i=1}^n [P(\alpha_i) = y_i] \geq A$ in polynomial time, where A should be as small as possible. Our algorithm still follows the high level template introduced by Sudan [Sud97] and Guruswami–Sudan [GS99] consisting of interpolation and root-finding. In our setting however, the object that we interpolate is significantly different. Indeed, rather than constructing an interpolating polynomial involving only the unknown message polynomial $P(X)$, we introduce several of its *Hasse derivatives* as additional formal variables.

Recall that the Hasse derivatives are the coefficients in the formal Taylor expansion

$$P(X + T) = \sum_{j \geq 0} P^{[j]}(X) T^j.$$

Crucially, however, the decoder is *not* given any derivative information about the unknown polynomial P. At each evaluation point α_i, the only information available to us is the ordinary received value y_i, and at an agreement position we know only that

$$P(\alpha_i) = y_i.$$

In particular, the values

$$P^{[1]}(\alpha_i), \ldots, P^{[d]}(\alpha_i)$$

[3] Very recently, we learned through personal communication of partial derandomization results attaining the generalized Singleton bound for a certain constant list size, albeit with a super-exponential alphabet size [CGG+26].

are completely unknown. A central challenge is therefore to impose interpolation constraints involving these higher-order derivatives using only the zeroth-order agreement condition $P(\alpha_i) = y_i$. The central idea is to construct a multivariate polynomial Q which vanishes when evaluated on the tuple $\big(P(X), P^{[1]}(X), \ldots, P^{[d]}(X)\big)$ for every polynomial P which has sufficiently high agreement with the received word. With this context, our algorithm consists of an interpolation procedure and a root-finding procedure.

1. Interpolation Step. Find a *non-zero* polynomial $Q \in \mathbb{F}_q[X, Y_0, Y_1, \ldots, Y_d]$ with $(1, k-1, \ldots, k-d-1)$-degree less than mA, where d represents the highest-order Hasse derivative we consider, and m is our target root multiplicity. In other words, for every monomial $X^a Y_0^{j_0} Y_1^{j_1} Y_2^{j_2} \cdots Y_d^{j_d}$ appearing in Q, we have that $a + (k-1)j_0 + (k-2)j_1 + \cdots + (k-d-1)j_d < mA$. This property ensures that for any polynomial $P \in \mathbb{F}_q[X]$ of degree less than k, we have that $Q(X, P(X), P^{[1]}(X), \ldots, P^{[d]}(X))$ is a univariate polynomial of degree less than mA. We also require that Q satisfies the following additional constraint: If $P \in \mathbb{F}_q[X]_{\leq k-1}$ satisfies $P(\alpha_i) = y_i$, then $(X - \alpha_i)^m$ is a factor of $Q(X, P(X), \ldots, P^{[d]}(X))$. The degree restrictions on Q are chosen so that, for every polynomial $P \in \mathbb{F}_q[X]_{\leq k-1}$,

$$\deg_X Q\big(X, P(X), P^{[1]}(X), \ldots, P^{[d]}(X)\big) < mA.$$

   Consequently, if P agrees with the received word in at least A positions, then the multiplicity condition above gives at least mA roots of this polynomial counted with multiplicity, which guarantees

$$Q\big(X, P(X), P^{[1]}(X), \ldots, P^{[d]}(X)\big) \equiv 0. \tag{1}$$

   We call this space of the polynomials satisfying such degree restrictions our interpolation space and henceforth denote it by $\mathcal{Q}$. For technical reasons, it will also be helpful to isolate the allowed monomials involving only $Y_2, \ldots, Y_d$ and denote them by $\mathcal{B}$.

2. Root-finding Step. Having found Q as above, the guarantee in (1) reduces our list-decoding problem to a root-finding one. Namely, we must find all low degree $P \in \mathbb{F}_q[X]_{\leq k-1}$ which satisfy (1). Fortunately, this can be done by using the Univariate Multiplicity decoder of Kopparty [Kop15] essentially as a blackbox. Altogether, we are guaranteed to find a list containing every polynomial P with agreement at least A. Since this procedure may also find spurious polynomials, we check that for each such recovered P whether it is indeed a solution with at least A agreements with the received word.

Finding the Prescribed Q. Given the above outline, one can see that the main difficulty is therefore the interpolation step: how can we construct a nonzero Q satisfying the required multiplicity condition without knowing the candidate polynomial P?

Fix one evaluation point α_i and received value $y_i \in \mathbb{F}_q$. When $P \in \mathbb{F}_q[X]$ satisfies $P(\alpha_i) = y_i$, we require $(X - \alpha_i)^m \mid Q(X, P(X), \ldots, P^{[d]}(X))$, which is equivalent to

$$Q\left(\alpha_i + T, P(\alpha_i + T), P^{[1]}(\alpha_i + T), \ldots, P^{[d]}(\alpha_i + T)\right) \equiv 0 \mod T^m. \tag{2}$$

Notice that there is already an issue here. We do not know any of the values

$$P(\alpha_i + T), P^{[1]}(\alpha_i + T), \ldots, P^{[d]}(\alpha_i + T).$$

A naive approach is to introduce formal variables $Y_0, \ldots, Y_d$ for the respective derivatives above, and simply require

$$Q\left(\alpha_i + T, Y_0, \ldots, Y_d\right) \equiv 0 \pmod{T^m}, \tag{3}$$

over every formal choice of $Y_0, \ldots, Y_d$ where the congruence is in $\mathbb{F}_q[T, Y_0, \ldots, Y_d]$. Then, to find our desired interpolant Q, we set up a homogeneous linear system in variables corresponding to the monomials in $\mathcal{Q}$ with one equation per constraint imposed. Unfortunately, this approach leads to more constraints than monomials and thus has no guarantee of a non-zero solution.

Reducing the Interpolation Constraints. Observe that such a requirement is far stronger than necessary, however. Indeed, we only need the condition to hold for tuples that can arise from a polynomial passing through (α_i, y_i). The key observation is that these tuples are not actually arbitrary. By the backward Taylor identity, they satisfy the relation

$$Y_0 = y_i + \sum_{j=1}^{d} (-1)^{j+1} T^j Y_j + TE. \tag{4}$$

where E is a remainder term divisible by T^d.

Thus, our first improvement over the naive method above is to no longer allow Y_0 to vary independently of $Y_1, \ldots, Y_d$. Instead, we substitute Y_0 according to the Taylor relation (4), while treating $Y_1, \ldots, Y_d$ and E as free formal variables.

After this substitution, we expand

$$Q\left(\alpha_i + T,\ y_i + \sum_{j=1}^{d} (-1)^{j+1} T^j Y_j + TE,\ Y_1, \ldots, Y_d\right) = \sum_{b,\vec{e}} q_{b,\vec{e}}(T) E^b Y^{\vec{e}}.$$

we impose the constraints

$$q_{b,\vec{e}}(T) \equiv 0 \pmod{T^{m-db}} \tag{5}$$

for all b such that $bd < m$. Note the reason we have different moduli per exponent b is that the term E^b supposedly contributes a factor of T^{db} already, while we need the overall term $q_{b,\vec{e}}(T) E^b Y^{\vec{e}}$ to vanish modulo T^m.

These constraints are imposed at all received points α_i, for $i \in [n]$. The substitution performed is what allows us to only impose the vanishing condition at a more restricted class of structured points consistent with genuine successive derivatives.

Finally, to carefully bound the number of linearly independent constraints imposed in this way, we view the local constraints at each received point as a linear map on the interpolation space and upper bound its rank. We do this by finding a large dimensional subspace in the kernel of this map and then performing a careful counting argument involving weighted lattice-points. We remark that there is a more direct way to count the number of linearly independent constraints, but we include the current approach as it reveals non-trivial structure about the kernel of the constraint map and could potentially be useful towards extending our results to higher rates or improving other constant dependencies.

Comparison with Guruswami–Sudan [GS99]. The Guruswami–Sudan algorithm [GS99] focuses on (2) in the case $d = 0$, but they cannot express $P(\alpha_i + T)$, either. [GS99] instead imposes a stronger condition that

$$\text{for all } a, b \geq 0 \text{ with } a + b < m, \text{ the coefficient of } T^a Y^b \text{ in } Q(\alpha_i + T, y_i + Y) \text{ must be zero.} \tag{6}$$

We can prove that (6) implies the target (2) when $d = 0$ so it suffices to ensure that (6) holds for all $i \in [n]$. Indeed these coefficients can be expressed as a known linear combination of $(c_f)_{f \in Q}$. For each $i \in [n]$, (6) introduces $\binom{m+1}{2}$ linear constraints, so there are $M = \binom{m+1}{2} n$ constraints in total. To ensure such a solution exists, it suffices to set parameters so that $|Q| > M$. A suitable calculation shows this is only possible when $A > \sqrt{nk}$. Importantly, using (6) as a proxy for (2) incurs a substantial inefficiency in the derivation of Q. As such, the Guruswami–Sudan algorithm is unable to go beyond the Johnson bound.

Acknowledgment and Statement on AI Usage

Zihan Zhang and Kai Zhe Zheng thank the Simons Institute for the Theory of Computing for its hospitality and support. This work was carried out during their Research Fellowships at the Simons Institute as part of

the program on Pseudorandomness and High-Dimensional Expansion. Kai Zhe Zheng is also grateful to Scott Duke Kominers and Justin Thaler for introducing him to better.codes.

The authors were initially inspired by a submission to the crowd-sourced Proximity Prize effort on better.codes, where a proof was published by user nasqret that a specific blocklength 2^{18}, rate $1/2$ Reed–Solomon code established a combinatorial list size bound up to radius

$$\frac{76790}{2^{18}} \approx 0.2929306,$$

slightly beyond the Johnson radius $1 - 1/\sqrt{2}$. This submission was, to the best of our knowledge, the first meaningful step beyond the Johnson radius for a specific evaluation domain, and a similar form of the $d = 1$ version of the techniques in this paper appeared there.[4] After processing the techniques therein, the authors aimed to uncover the core mathematical novelty responsible for the improvement and subsequently extended the techniques to the general setting developed here. They relied on AI interaction, specifically GPT-5.6 Sol, to help produce the rest of the results of this paper. All mathematical statements, proofs, and conclusions were subsequently developed and independently verified by the authors, who take full responsibility for the contents of this paper.

2 Preliminaries

Unless otherwise stated, we let q denote a sufficiently large prime.

Hasse derivatives. Given a univariate polynomial $P = \sum_{i=0}^{k-1} a_i X^i \in \mathbb{F}_q[X]$, we define its ℓ-th Hasse derivative (e.g., [DKSS13]) to be

$$P^{[\ell]}(X) := \sum_{i=\ell}^{k-1} \binom{i}{\ell} a_i X^{i-\ell}.$$

Crucially, the Hasse derivatives satisfy the identity that

$$P(X+T) = \sum_{\ell=0}^{k-1} P^{[\ell]}(X) T^\ell. \tag{7}$$

Substituting $X' = X + T$ and $T' = -T$ into the above formula gives us the Möbius inversion

$$P(X) = \sum_{\ell=0}^{k-1} P^{[\ell]}(X+T)(-T)^\ell, \tag{8}$$

which we make key use of in our interpolation.

Weighted Degrees. Given a multivariate polynomial $Q \in \mathbb{F}_q[X_1, \ldots, X_d]$, we define the weighted degree of Q with respect to weights $w_1, \ldots, w_d \in \mathbb{Z}_{\geq 0}$ to be the maximum value of $w_1 i_1 + \cdots + w_d i_d$ among all monomials $X_1^{i_1} \cdots X_d^{i_d}$ appearing in Q. We denote this quantity by $\deg_{w_1,\ldots,w_d} Q$. In the special case where $w_i = 1$ and $w_j = 0$ for all $j \in [d] \setminus \{i\}$, we denote this weighted degree more succinctly by $\deg_{X_i} Q$.

[4] See the submission here: https://github.com/proximity-prize/proximity-prize/pull/122.

Solving Polynomial Differential Equations. A crucial ingredient in our list-decoding algorithm is finding all univariate polynomials $P \in \mathbb{F}_q[X]$ with $\deg P \le k-1$ which satisfy the identity

$$Q(X, P(X), P^{[1]}(X), \ldots, P^{[d]}(X)) \equiv 0 \tag{9}$$

for some (fixed) multivariable polynomial $Q \in \mathbb{F}_q[X, Y_0, Y_1, \ldots, Y_d]$. By a result of Kopparty [Kop15] (see also [KT22]), as long as our prime field q is larger than some suitable weighted degrees of Q, we can find a list of all such P efficiently. We state this result as follows.

Theorem 2.1 (Theorem 4.3 [Kop15], restated)**.** *Assume $q \ge k > d$ with q prime. Assume nonzero $Q \in \mathbb{F}_q[X, Y_0, \ldots, Y_d]$ satisfies $\deg_{Y_i} Q < q$ for all $i \in \{0, 1, \ldots, d\}$ and that*

$$\deg_{1, k-1, k-2, \ldots, k-d-1} Q < q^2,$$

then one can find, in $q^{O(d+1)}$ time, all $P \in \mathbb{F}_q[X]$ of degree at most $k-1$ satisfying (9). Furthermore, the number of such P found is at most q^{4d+6}.

3 Novel Interpolation via Hidden Derivatives

Our goal in this section is to interpolate an *explainer polynomial* $Q \in \mathbb{F}_q[X, Y_0, Y_1, \ldots, Y_d]$ which at a high level has the following two properties

- Q is "low-degree" under some suitable notions of low-degree.
- For any degree at most $k-1$ polynomial P with desired agreement with the received word $\vec{y} = (y_1, \ldots, y_n) \in \mathbb{F}_q^n$, P satisfies

$$Q(X, P(X), P^{[1]}(X), \ldots, P^{[d]}(X)) \equiv 0.$$

By computing Q satisfying these two properties, we can later efficiently find all low-degree polynomials sufficiently correlated with the received word by finding all P satisfying the relation in the second item.

List of Parameters. Fix a target list decoding radius $1-\varepsilon$ and a slack $\theta \in (0, 1)$ as in Theorem 1.1. We set

$$A = \lceil \varepsilon n \rceil, \qquad d = \left\lceil \varepsilon^{-3/\theta} \right\rceil, \qquad m = d^3, \tag{10}$$

$$B = \left\lceil \frac{mA}{k-1} \right\rceil. \tag{11}$$

We additionally make the rate assumption that

$$R = \frac{k}{n} \le (1-\theta)\varepsilon \tag{12}$$

Intuitively, A represents the minimum agreement for our decoder, d is the maximum order of a Hasse derivative considered by Q, m is the multiplicity we impose on each received point, and B is the maximum degree of Q with respect to the inputs $Y_0, \ldots, Y_d$.

Local Constraints at Each Received Point (α, y). We require Q to satisfy a series of local constraints at each received point (α, y). The purpose of these constraints is to ensure that if a candidate polynomial P satisfies $P(\alpha) = y$, then the specialization

$$Q\big(X, P(X), P^{[1]}(X), \ldots, P^{[d]}(X)\big)$$

has a zero of multiplicity at least m at $X = \alpha$. The constraint is motivated by trying to match the low degree terms in (13). In particular, setting $P(\alpha) = y$ there and looking at terms of T-degree at most d, we can write

$$P(\alpha + T) \equiv y + TP^{[1]}(\alpha + T) - T^2P^{[2]}(\alpha + T) + \cdots + (-1)^{d+1}T^dP^{[d]}(\alpha + T) \mod T^{d+1}. \tag{13}$$

Thus, although we do not know the derivatives of P at the time of interpolation, any polynomial passing through (α, y), along with its derivatives, must satisfy this formal relation.

We encode this relation by introducing formal variables $Y_1, \ldots, Y_d$ and a remainder variable E, and making the substitution

$$X = \alpha + T, \qquad Y_0 = y + TY_1 - T^2Y_2 + \cdots + (-1)^{d+1}T^dY_d + TE.$$

For a polynomial P satisfying $P(\alpha) = y$, after setting $Y_j = P^{[j]}(\alpha + T)$, the corresponding remainder satisfies

$$E \equiv 0 \mod T^d.$$

Hence, our explainer polynomial can encode this relation as follows. Make the substitution

$$\begin{aligned} Q^{(\alpha,y)}(T, E, Y_1, \ldots, Y_d) &:= Q\left(\alpha + T, y + \sum_{j=1}^{d}(-1)^{j+1}T^jY_j + TE, Y_1, \ldots, Y_d\right) \\ &= \sum_{b,\vec{e}} q_{b,\vec{e}}(T)E^bY_1^{e_1} \cdots Y_d^{e_d}, \end{aligned} \tag{14}$$

where the $q_{b,\vec{e}}(T) \in \mathbb{F}_q[T]$ are coefficients. We enforce the following local constraint for every (α, y) and nonnegative $(b, \vec{e}) := (b, e_1, e_2, \ldots, e_d) \in \mathbb{Z}^{d+1}$ such that $db < m$:

$$q_{b,\vec{e}}(T) \equiv 0 \mod T^{m-db} \tag{15}$$

Note that these are homogeneous linear constraints on the coefficients of Q.

Lemma 3.1. *Suppose $P(\alpha) = y$, and suppose Q satisfies (15) at (α, y). Then*

$$Q\Big(\alpha + T,\ P(\alpha + T),\ P^{[1]}(\alpha + T), \ldots, P^{[d]}(\alpha + T)\Big) \equiv 0 \mod T^m.$$

Proof. We can write

$$Q\left(\alpha + T,\ y + \sum_{j=1}^{d}(-1)^{j+1}T^jY_j + TE,\ Y_1, \ldots, Y_d\right) = \sum_{b,\vec{e}} q_{b,\vec{e}}(T)E^bY_1^{e_1} \cdots Y_d^{e_d}.$$

Now substitute

$$Y_j = P^{[j]}(\alpha + T)$$

for each $j \in [d]$ and

$$E = E_P(T) := \frac{P(\alpha + T) - y - \sum_{j=1}^{d}(-1)^{j+1}T^jP^{[j]}(\alpha + T)}{T}. \tag{16}$$

Under this substitution, the second input to Q now becomes $P(\alpha + T)$ and by (13), we have $E_P(T) \equiv 0 \mod T^d$. Hence $E_P(T)^b$ is divisible by T^{db}. If $db < m$, the local constraint imposed by (15) gives

$$q_{b,\vec{e}}(T) \equiv 0 \mod T^{m-db},$$

so $q_{b,\vec{e}}(T)E_P(T)^b$ is divisible by T^m. If $db \geq m$, then $E_P(T)^b$ is already divisible by T^m. Thus every summand in (14) is divisible by T^m, and therefore

$$Q\Big(\alpha + T,\ P(\alpha + T),\ P^{[1]}(\alpha + T), \ldots, P^{[d]}(\alpha + T)\Big) \equiv 0 \mod T^m. \qquad \square$$

3.1 Dimension of the Interpolation Space

Here we define the monomials that we use when interpolating Q and give a lower bound on their number. For

$$c = (c_2, \ldots, c_d) \in \mathbb{Z}_{\geq 0}^{d-1},$$

it will be helpful to define.

$$\omega(c) = \sum_{j=2}^{d} (j-1)c_j, \qquad |c| = \sum_{j=2}^{d} c_j, \qquad Y^c = \prod_{j=2}^{d} Y_j^{c_j}.$$

The function ω tracks a weighted-degree like quantity, assigning weight $j-1$ to the variable Y_j. To give some intuition, it will be used as follows. When writing out a Taylor expansion and replacing successive derivatives with $Y_1, \ldots, Y_d$, we get an expression that looks like $\sum_{j\geq 0}(-1)^j T^{j-1} Y_j$. Then, $\omega(c)$ gives the degree of T in any coefficient of the monomial Y^c when raising this expression to any power. To see this, observe that Y_j can only occur with T^{j-1}, hence a power $Y_j^{c_j}$ must be accompanied by a factor $T^{(j-1)c_j}$. On the other hand, $|c|$, is simply the unweighted degree of the monomial Y^c.

Now we set

$$W := \left\lfloor \frac{(1 + \theta/2)dm}{\log(ed)} \right\rfloor \tag{17}$$

Let

$$\mathcal{B} = \left\{ c \in \mathbb{Z}_{\geq 0}^{d-1} : \omega(c) \leq W, \quad |c| \leq \left\lceil \left(1 + \frac{3\theta}{4}\right) m \right\rceil \right\}.$$

and finally, let our space of interpolants $\mathcal{Q} \subseteq \mathbb{F}_q[X, Y_0, \ldots, Y_d]$ be the span of the monomials

$$\left\{ X^a Y_0^{b_0} Y_1^{b_1} Y^c : \begin{array}{l} a, b_0 \in \mathbb{Z}_{\geq 0}, \quad c \in \mathcal{B}, \quad 0 \leq b_1 \leq m, \\ b_0 + b_1 + |c| \leq B, \\ a + (k-1)(b_0 + b_1 + |c|) < mA \end{array} \right\}. \tag{18}$$

At a high level, requiring $c \in \mathcal{B}$ restricts both ordinary degree and ω-weighted degree of monomials in the variable $Y_2, \ldots, Y_d$, while a separate degree requirement is imposed on the variable Y_1 for our interpolant polynomials.

Lemma 3.2. *The interpolation space $\mathcal{Q}$ satisfies the following properties:*

- *Every $Q \in \mathcal{Q}$ satisfies*

$$\deg_{Y_i} Q \leq B,$$

  for each $i \in \{0, \ldots, d\}$.

- *For every $Q \in \mathcal{Q}$ and every $P \in \mathbb{F}_q[X]$ with* $\deg P \le k-1$,

$$\deg_X Q\big(X, P(X), P^{[1]}(X), \ldots, P^{[d]}(X)\big) < mA.$$

- *The dimension of $\mathcal{Q}$ satisfies*

$$\dim \mathcal{Q} > \frac{\theta^3}{384}|\mathcal{B}|(k-1)m^3.$$

Proof. The first item follows directly from the definition of $\mathcal{Q}$. Indeed, every monomial

$$X^a Y_0^{b_0} Y_1^{b_1} Y^c$$

in $\mathcal{Q}$ satisfies

$$b_0 + b_1 + |c| \le B.$$

Hence the exponent of each variable Y_i, for $i \in \{0, \ldots, d\}$, is at most B, and therefore

$$\deg_{Y_i} Q \le B$$

for every $i \in \{0, \ldots, d\}$.

The second item follows straightforwardly too. Fix $P \in \mathbb{F}_q[X]$ of degree at most $k-1$. Since $\deg P^{[j]} \le k-1$ for every j, one can check that

$$X^a P(X)^{b_0} \big(P^{[1]}(X)\big)^{b_1} \prod_{j=2}^{d} \big(P^{[j]}(X)\big)^{c_j},$$

has X-degree at most

$$a + (k-1)(b_0 + b_1 + |c|) < mA.$$

Hence, any $Q \in \mathcal{Q}$ has $\deg_X Q\big(X, P, P^{[1]}, \ldots, P^{[d]}\big) < mA$.

It remains to lower bound $\dim \mathcal{Q}$. First, by the definition of A in (10) and the fact that $k \le (1-\theta)\varepsilon$, we have

$$\frac{A}{k-1} \ge \frac{\epsilon n}{(1-\theta)\epsilon n} = \frac{1}{1-\theta} \ge 1 + \theta.$$

On the other hand, for every $c \in \mathcal{B}$, the definition of $\mathcal{B}$ gives

$$|c| \le \left\lfloor \left(1 + \frac{3\theta}{4}\right) m \right\rfloor \le \left(1 + \frac{3\theta}{4}\right) m.$$

Consequently,

$$m\frac{A}{k-1} - |c| \ge \frac{\theta m}{4}. \tag{19}$$

Now fix $c \in \mathcal{B}$ and an integer

$$0 \le b_1 \le \left\lfloor \frac{\theta m}{4} \right\rfloor.$$

For every pair $(a, b_0) \in \mathbb{Z}^2_{\ge 0}$ satisfying

$$a + (k-1)b_0 < mA - (k-1)(|c| + b_1),$$

observe that the corresponding monomial $X^a Y_0^{b_0} Y_1^{b_1} Y^c$ belongs to Q. Indeed, the weighted-degree condition holds by construction, and since $a \geq 0$,

$$b_0 + b_1 + |c| < \frac{mA}{k-1} \leq \left\lceil \frac{mA}{k-1} \right\rceil = B.$$

The number of such pairs (a, b_0) is at least

$$\frac{\big(mA - (k-1)(|c| + b_1)\big)^2}{2(k-1)}.$$

Using (19), this is at least

$$\frac{k-1}{2}\left(\frac{\theta m}{4} - b_1\right)^2.$$

Therefore,

$$\begin{aligned}
\dim Q &\geq \frac{k-1}{2}|\mathcal{B}| \sum_{b_1=0}^{\lfloor \theta m/4 \rfloor} \left(\frac{\theta m}{4} - b_1\right)^2 \\
&\geq \frac{k-1}{2}|\mathcal{B}| \int_0^{\theta m/4} \left(\frac{\theta m}{4} - x\right)^2 dx \\
&= \frac{\theta^3}{384}|\mathcal{B}|(k-1)m^3.
\end{aligned}$$

This proves the final item. □

3.2 Upper Bounding The Number of Constraints

At every received point (α, y), we impose the divisibility constraints (15). It remains to bound how many independent linear conditions these constraints impose. For this rank calculation only, it is convenient to write

$$Y_0 = y + TU.$$

For $z \geq 0$, define

$$\Lambda(z) = \left|\{c \in \mathbb{Z}_{\geq 0}^{d-1} : \omega(c) \leq z\}\right|.$$

3.2.1 A Bound on $\Lambda(z)$'s Growth

As a first, modular step, in this section we will prove the following bound on how quickly $\Lambda(z)$ grows.

Lemma 3.3. *We have*

$$\Lambda(z) \leq \frac{\left(z + \binom{d}{2}\right)^{d-1}}{((d-1)!)^2},$$

Further, if $d \geq \lceil \varepsilon^{-3/\theta} \rceil$ *and*

$$\varepsilon < \left(\frac{\theta^3(1-\theta)}{768}\right)^{\frac{5+\theta}{1-\theta}} \tag{20}$$

then

$$\Lambda(W + m) \leq |\mathcal{B}| d^{2/(2+\theta)+\theta(1-\theta)/18}.$$

Before proving this, we require some elementary facts which govern the volume of these scaled lattices. We start by recalling a folklore, basic fact about the volume of the unit simplex:

Proposition 3.4. *Consider the $d-1$ dimensional simplex*

$$\mathcal{S} = \left\{ x \in \mathbb{R}_{\geq 0}^{d-1} : \sum_i x_i \leq 1 \right\}.$$

Then, $\mathrm{Vol}(\mathcal{S}) = \frac{1}{(d-1)!}$.

With this bound on the volume of the unit simplex, we can easily extend this to a formula for the volume of the "scaled" simplex:

Corollary 3.5. *Let $z \in \mathbb{R}_{\geq 0}$. Then, for the scaled simplex*

$$\mathcal{S}_z = \left\{ x \in \mathbb{R}_{\geq 0}^{d-1} : \sum_i i \cdot x_i \leq z \right\},$$

we have that

$$\mathrm{Vol}(\mathcal{S}_z) = \frac{z^{d-1}}{((d-1)!)^2}.$$

Proof. We start by assuming that $z = 1$. In this case, we consider the body

$$\left\{ x \in \mathbb{R}_{\geq 0}^{d-1} : \sum_i i \cdot x_i \leq 1 \right\}.$$

Now, we perform a change of variables with $y_i = i \cdot x_i$. In the y-variables, the volume of the body

$$\left\{ y \in \mathbb{R}_{\geq 0}^{d-1} : \sum_i y_i \leq 1 \right\}$$

is exactly $\frac{1}{(d-1)!}$ by Proposition 3.4. Translating back to the x variables, we lose a factor of exactly $(d-1)!$ in the volume. This is because a linear map multiplies the volume by a factor of exactly its determinant (which in this case is exactly $(d-1)!$). Thus, $\mathrm{Vol}(\mathcal{S}_1) = \frac{1}{((d-1)!)^2}$, and uniformly scaling the coordinates by z, and thus the volume by a factor of z^{d-1} yields the desired corollary. □

Our goal is to use these simplices to provide a tighter bound on the size of $\Lambda(z)$. To do this, we now compare the volume of $\mathcal{S}_z$ with $\Lambda(z)$. We consider any $u \in \mathbb{Z}_{\geq 0}^{d-1}$ which satisfies $\sum_i i \cdot u_i \leq z$. For any such point, we then consider its corresponding "half-open unit cube," obtained as

$$u + [0,1)^{d-1}.$$

We have:

Proposition 3.6. *Consider any $u \in \mathbb{Z}_{\geq 0}^{d-1}$ which satisfies $\sum_i i \cdot u_i \leq z$. Then, the unit cubes $u + [0,1)^{d-1}$ are pairwise disjoint and moreover*

$$\mathrm{Vol}\left(\bigcup_{u \in \mathbb{Z}_{\geq 0}^{d-1} : \sum_i i \cdot u_i \leq z} u + [0,1)^{d-1} \right) = \Lambda(z).$$

Proof. Pairwise disjointness follows from the fact that each unit cube is anchored at a distinct integral lattice point. The volume follows from the fact that

$$\mathrm{Vol}\left(\bigcup_{u\in\mathbb{Z}_{\geq 0}^{d-1}:\sum_i i\cdot u_i\leq z} u+[0,1)^{d-1}\right)=\sum_{u\in\mathbb{Z}_{\geq 0}^{d-1}:\sum_i i\cdot u_i\leq z} 1=\Lambda(z).$$

□

We now prove the following containment property, which will allow us to directly translate our volume estimates for the scaled simplex to bounds on $\Lambda(z)$:

Proposition 3.7. *For z an integer, we have that*

$$\mathcal{S}_z\subseteq\bigcup_{\sum_i iu_i\leq z}\left(u+[0,1)^{d-1}\right)\subseteq\mathcal{S}_{z+d(d-1)/2}.$$

Proof. The first containment follows because if $x\in\mathcal{S}_z$, then $u_i=\lfloor x_i\rfloor$ satisfies

$$\sum_{i=1}^{d-1} i\cdot u_i\leq\sum_{i=1}^{d-1} i\cdot x_i\leq z,$$

and $x\in u+[0,1)^{d-1}$. The second containment follows because for any $x\in u+[0,1)^{d-1}$ and $\sum_i iu_i\leq z$, then

$$\sum_{i=1}^{d-1} ix_i<\sum_{i=1}^{d-1} i(u_i+1)\leq z+\sum_{i=1}^{d-1} i\leq z+d(d-1)/2.$$

Thus, any such $u+[0,1)^{d-1}$ is contained in $\mathcal{S}_{z+d(d-1)/2}$. □

By using Corollary 3.5, we then see that:

Corollary 3.8. *Let $z\in\mathbb{R}_{\geq 0}$. Then,*

$$\frac{z^{d-1}}{((d-1)!)^2}\leq\Lambda(z)\leq\frac{(z+d(d-1)/2)^{d-1}}{((d-1)!)^2}.$$

This estimate of $\Lambda(z)$ already proves the first point of Lemma 3.3.

Before the proof of the second item of Lemma 3.3, we introduce one key lemma that we will take advantage of, which relates the expectation of a distribution over a simplex to its centroid:

Lemma 3.9. *Let $S=\mathrm{conv}\{v_0,\ldots,v_{d-1}\}$ be a $d-1$-dimensional simplex, and let X be uniformly distributed on S. Then*

$$\mathbb{E}[X]=\frac{1}{d}\sum_{i=0}^{d-1} v_i.$$

In particular, the expectation of X is the centroid of S.

We provide a self-contained proof of this lemma in the appendix, as it is a folklore result with a simple proof.

Proof of the second item of Lemma 3.3. Recall that we wish to show that $\Lambda(W+m) \le |\mathcal{B}| d^{2/(2+\theta)+\theta(1-\theta)/18}$. $\mathcal{B}$ is already defined with respect to W, indeed,

$$\mathcal{B} = \left\{ c \in \mathbb{Z}_{\ge 0}^{d-1} : \omega(c) \le W, \quad |c| \le \left\lceil \left(1 + \frac{3\theta}{4}\right) m \right\rceil \right\}.$$

If we remove the second constraint from the definition of $\mathcal{B}$ (i.e., the restriction on $|c|$), our desired inequality is not hard to show, as it boils down to comparing the number of points with $\omega(c) \le W$ to the number of points with $\omega(c) \le W + m$. This could be directly bounded by (3.8). Unfortunately, $\mathcal{B}$ is also defined with the second constraint on $|c|$. Our goal going forward is thus to show that this constraint *does not* decrease the number of points in $\mathcal{B}$ by too much.

Next, recall that $W := \left\lfloor \frac{(1+\theta/2)dm}{\log(ed)} \right\rfloor$, that $m = d^3$, and that $d := \lceil \varepsilon^{-3/\theta} \rceil$. Since $e^x \ge x + 1$ for all $x \in \mathbb{R}$, we have that $d \ge \log(d) + 1 = \log(ed)$. Therefore, $W \ge \lfloor (1 + \theta/2)m \rfloor \ge 1$. Now, our intermediate goal is to show that *most points* in $\mathcal{S}_W$ also satisfy the additional constraint $|c| \le \left\lceil \left(1 + \frac{3\theta}{4}\right) m \right\rceil$. To do this, we sample a random point $X = (X_1, \dots X_{d-1})$ from the simplex $\mathcal{S}_W$, and will bound its expected magnitude. Crucially, the simplex $\mathcal{S}_W$ has as its vertices

$$0, \quad W \cdot e_1, \quad \frac{W}{2} \cdot e_2, \quad \dots, \quad \frac{W}{d-1} \cdot e_{d-1},$$

and the expectation of a uniform point in a simplex is its centroid (as per Lemma 3.9). Thus, we see that

$$\mathbb{E}\left[\sum_{i=1}^{d-1} X_i\right] = \frac{W}{d} \cdot \sum_{i=1}^{d-1} \frac{1}{i} \le \frac{W \log(ed)}{d} \le (1 + \theta/2) \cdot m,$$

by our choice of W. Now, we can apply a simple Markov bound to see that

$$\Pr\left[\sum_{i=1}^{d-1} X_i > \left\lceil \left(1 + \frac{3\theta}{4}\right) m \right\rceil\right] \le \frac{(1 + \theta/2)}{(1 + 3\theta/4)}.$$

Equivalently, this means that

$$\mathrm{Vol}\left(\left\{ X \in \mathcal{S}_W : |X| \le \left\lceil \left(1 + \frac{3\theta}{4}\right) m \right\rceil \right\}\right) \ge \left(1 - \frac{(1 + \theta/2)}{(1 + 3\theta/4)}\right) \cdot \mathrm{Vol}(\mathcal{S}_W) = \frac{\theta}{4 + 3\theta} \cdot \frac{W^{d-1}}{((d-1)!)^2},$$

where the last equality uses Corollary 3.5.

Now, we claim that

$$\left\{ X \in \mathcal{S}_W : |X| \le \left\lceil \left(1 + \frac{3\theta}{4}\right) m \right\rceil \right\} \subseteq \bigcup_{u \in \mathcal{B}} \left(u + [0,1)^{d-1}\right).$$

This is because if we consider any point x in the first set, if we apply the coordinate-wise floor $\lfloor x_i \rfloor$, the resulting point u now must be in the set $\mathcal{B}$. Adding the half-open unit cube to every point $u \in \mathcal{B}$ thus captures all points whose coordinate-wise floor would be in $\mathcal{B}$. Importantly, by plugging in our volume bounds, we know that

$$\begin{aligned} |\mathcal{B}| = \mathrm{Vol}\left(\bigcup_{u \in \mathcal{B}} \left(u + [0,1)^{d-1}\right)\right) &\ge \mathrm{Vol}\left(\left\{ X \in \mathcal{S}_W : |X| \le \left\lceil \left(1 + \frac{3\theta}{4}\right) m \right\rceil \right\}\right) \\ &\ge \frac{\theta}{4 + 3\theta} \cdot \frac{W^{d-1}}{((d-1)!)^2}. \end{aligned}$$

At the same time, (3.8) implies that

$$\Lambda(W+m) \le \frac{\left(W+m+\binom{d}{2}\right)^{d-1}}{((d-1)!)^2}.$$

Dividing these two estimates then implies that

$$\frac{\Lambda(W+m)}{|\mathcal{B}|} \le \frac{4+3\theta}{\theta} \cdot \left(1+\frac{m+\binom{d}{2}}{W}\right)^{d-1}.$$

This is exactly the quantity we seek to bound for Lemma 3.3, and we now simply plug in the relationships between m, d, W, θ to obtain our desired result.

Since $\log(1+x) \le x$, $m = d^3$, and

$$W = \left\lfloor \frac{(1+\theta/2)d^4}{\log(ed)} \right\rfloor,$$

we have

$$\begin{aligned}\log \frac{\Lambda(W+m)}{|\mathcal{B}|} &\le \log \frac{4+3\theta}{\theta} + \frac{(d-1)\left(m+\binom{d}{2}\right)}{W} \\ &= \frac{2}{2+\theta}\log d + O_\theta(1).\end{aligned}$$

Here the final equality also absorbs the floor in the definition of W. Since $d = \lceil \varepsilon^{-3/\theta} \rceil$ and (20) holds, we have that the last expression is at most

$$\left(\frac{2}{2+\theta} + \frac{\theta(1-\theta)}{18}\right)\log d.$$

Exponentiating gives

$$\Lambda(W+m) \le |\mathcal{B}| d^{2/(2+\theta)+\theta(1-\theta)/18},$$

as desired. □

3.2.2 Bounding the Rank

With this bound on the growth of $\Lambda(z)$, we can now proceed to bound the rank of all the imposed constraints. Throughout this section, we fix one pair (α, y) where α is an evaluation point and y is a received value. We will bound the number of linearly independent constraints imposed across all of the coefficient constraints arising from (15) applied at (α, y).

To this end, we will think of the constraints as a linear map from the space of interpolation monomials Q, into some suitable $\mathbb{F}_q$ vector space. Specifically, let

$$\mathcal{I} = \left\{(i,b,e_1,c) \in \mathbb{Z}_{\ge 0}^{d+2} : \begin{array}{llll} i,b,e_1 \in \mathbb{Z}_{\ge 0}, & c \in \mathbb{Z}_{\ge 0}^{d-1}, & & \\ i+db < m, & e_1 \le 2m-1, & \omega(c) \le W+m-1 & \end{array}\right\},$$

denote the set of possible exponent vectors indexing the monomials over $T, E, Y_1, \ldots, Y_d$ that are required to vanish by (15). Then, we let $\Phi_{\alpha,y} : Q \to \mathbb{F}_q^{\mathcal{I}}$, where $\Phi_{\alpha,y}(Q)_{(i,b,e_1,c)}$ is the coefficient of $T^i E^b Y_1^{e_1} Y^c$ in $Q\left(\alpha+T,\ y+\sum_{j=1}^{d}(-1)^{j+1}T^j Y_j + TE,\ Y_1, \ldots, Y_d\right)$, i.e. the substitution in (14) when defining the constraints in (15). In this language, the constraints of (15) imposed by (α, y) are exactly equivalent to requiring $\Phi_{\alpha,y}$

to vanish. Hence, the goal of this subsection is to bound the rank of $\Phi_{\alpha,y}$. Multiplying this bound by the number of received points then gives a bound on the overall number of linearly independent constraints (15) we impose.

To bound $\operatorname{rank}(\Phi)$, it will be helpful to first perform one intermediate substitution towards (14), and instead bound the rank of the map *after* this substitution. Let $\Psi_{\alpha,y}$ be the map which takes $Q \in \mathcal{Q}$ as input, substitutes

$$X = \alpha + T, \qquad Y_0 = y + TU,$$

and reduces modulo T^m. Notice that every monomial appearing lies in

$$\mathcal{V} = \operatorname{span}\left\{T^r U^a Y_1^b Y^c : 0 \le r < m,\ 0 \le a \le r,\ 0 \le b \le m,\ \omega(c) \le W + r\right\}.$$

Indeed, every factor of U arising from $(y + TU)^{b_0}$ is accompanied by a factor of T, while the conditions $b_1 \le m$ and $\omega(c) \le W$ already hold for every monomial of $\mathcal{Q}$. Hence $\Psi_{\alpha,y}$ maps from $\mathcal{Q} \to \mathcal{V}$.

By performing this substitution and modular reduction, we can instead focus on bounding the "remaining" transformation needed for $\Phi_{\alpha,y}$, which we define by $\Gamma : \mathcal{V} \to \mathbb{F}_q^{\mathcal{I}}$. For $F \in \mathcal{V}$, rewrite F in the variables $T, E, Y_1, \ldots, Y_d$ using

$$E = U - \sum_{j=1}^{d} (-1)^{j+1} T^{j-1} Y_j. \tag{21}$$

Then, for every $(i, b, e_1, c) \in \mathcal{I}$, define $\Gamma(F)_{(i,b,e_1,c)}$ to be the coefficient of $T^i E^b Y_1^{e_1} Y^c$ in the resulting polynomial. Thus, $\Gamma(F)$ records exactly the coefficients required to vanish by the local constraints.

One can check that the map of interest $\Phi_{\alpha,y}$ factors as $\Phi_{\alpha,y} = \Gamma \circ \Psi_{\alpha,y}$. Therefore,

$$\operatorname{rank} \Phi_{\alpha,y} \le \operatorname{rank} \Gamma,$$

and it suffices to upper bound $\operatorname{rank} \Gamma$.

Our strategy is to find a large number of linearly independent vectors in $\ker \Gamma$, and to this end we start by observing a simple divisibility condition which ensures membership in $\ker \Gamma$.

Lemma 3.10. *Suppose $F \in \mathcal{V}$ is divisible by $T^r E^h$ for some $r, h \ge 0$ satisfying $r + dh \ge m$. Then $F \in \ker \Gamma$.*

Proof. After rewriting F in the variables $T, E, Y_1, \ldots, Y_d$, every monomial containing E^b has $b \ge h$ and is divisible by T^r. Hence

$$r + db \ge r + dh \ge m.$$

Hence after rewriting F, none of these monomials are recorded, and $\Gamma(F) = 0$. □

For $0 \le r < m$, define

$$h_r := \left\lceil \frac{m - r}{d} \right\rceil,$$

so that h_r is the smallest integer satisfying $r + dh_r \ge m$. Using Lemma 3.10, we can show that Γ has a large kernel by finding many multiples of $T^r E^h$ in $\mathcal{V}$, for suitable r, h, that are linearly independent.

Lemma 3.11. *We have*

$$\dim \ker \Gamma \ge \sum_{r=0}^{m-1} \max(0, (r - h_r + 1)(m - h_r + 1)\Lambda(W + r)). \tag{22}$$

Proof. For each $0 \le r < m$, define

$$\mathcal{K}_r = \operatorname{span}\left\{T^r E^{h_r} U^a Y_1^b Y^c : 0 \le a \le r - h_r,\ 0 \le b \le m - h_r,\ c \in \mathbb{Z}_{\ge 0}^{d-1},\ \omega(c) \le W + r\right\}.$$

Here, $\mathcal{K}_r$ is a subspace of $\mathbb{F}_q[T, U, Y_1, \ldots, Y_d]/(T^m)$ and we define $\mathcal{K}_r = 0$ if there are no monomials. We first verify that $\mathcal{K}_r \subseteq \mathcal{V}$ as this may not be true apriori. Fix a monomial $T^r E^{h_r} g \in \mathcal{K}_r$. We verify that it indeed lies in $\mathcal{V}$. Expand E using (21) and consider a monomial appearing. Let e_j be the exponent of Y_j in it for $j \ge 2$ and set $M = \sum_{j=2}^{d} (j-1)e_j$. It suffices to show this monomial is in $\mathcal{V}$. Suppose that, in expanding E^{h_r}, we select e_j copies of Y_j for each $j \ge 2$, and let

$$M = \sum_{j=2}^{d} (j-1)e_j.$$

These choices contribute a factor of T^M and a monomial in $Y_2, \ldots, Y_d$ of ω-weight M. Since the monomial Y^c coming from g satisfies $\omega(c) \le W + r$, the resulting monomial has T-degree $r + M$ and ω-weight at most $W + r + M$. Its U-degree is at most

$$(r - h_r) + h_r = r \le r + M,$$

and its Y_1-degree is at most

$$(m - h_r) + h_r = m.$$

Thus, after reducing modulo T^m, every resulting monomial lies in $\mathcal{V}$.

Additionally, $r + dh_r \ge m$, so Lemma 3.10 implies $\mathcal{K}_r \subseteq \ker \Gamma$ and it is straightforward to compute that the dimension of $\mathcal{K}_r$ is

$$\dim \mathcal{K}_r = \max(0, (r - h_r + 1)(m - h_r + 1)\Lambda(W + r)). \tag{23}$$

Finally, we check that the spaces $\mathcal{K}_0, \ldots, \mathcal{K}_{m-1}$ are linearly independent. To this end, suppose

$$\sum_{r=0}^{m-1} F_r = 0, \qquad F_r \in \mathcal{K}_r,$$

and let $r^\star$ be the smallest index for which $F_{r^\star} \ne 0$. Since

$$E \equiv U - Y_1 \pmod{T},$$

the coefficient of $T^{r^\star}$ in $F_{r^\star}$ is

$$(U - Y_1)^{h_{r^\star}} g_{r^\star} \ne 0.$$

Every F_r with $r > r^\star$ is divisible by $T^{r^\star+1}$, so none of them can cancel out this coefficient, contradicting $\sum_r F_r = 0$. Hence the spaces $\mathcal{K}_r$ are linearly independent. Summing (23) proves the lemma. □

We are now ready to upper bound the rank of Γ.

Lemma 3.12 (Local rank)**.** *We have*

$$\operatorname{rank} \Gamma < |\mathcal{B}| m^3 d^{-\frac{2\theta}{5+\theta}}. \tag{24}$$

Proof. For each fixed T-degree r, we can directly count that there are $(r+1)(m+1)\Lambda(W+r)$-many monomials in $\mathcal{V}$, which means

$$\dim \mathcal{V} = \sum_{r=0}^{m-1} (r+1)(m+1)\Lambda(W+r).$$

By Lemma 3.11 and rank-nullity,

$$\operatorname{rank}(\Phi) \leq \sum_{r=0}^{m-1} \Lambda(W+r) B_r,$$

where

$$B_r := (r+1)(m+1) - \max(0, (r-h_r+1)(m-h_r+1)).$$

Observe that by definition of h_r, we have

$$B_r \leq h_r(r+m+2) \qquad \text{and} \qquad h_r \leq \frac{m-r}{d} + 1.$$

Summing over $0 \leq r < m$ gives

$$\begin{aligned}\sum_{r=0}^{m-1} B_r &\leq \frac{m^3-m}{6d} + \frac{(m+2)m(m+1)}{2d} + \frac{m(m-1)}{2} + m(m+2) \\ &= \frac{d^2(d+1)(2d+1)(d^2-d+1)(2d^2-d+5)}{6} \\ &\leq d^8 = \frac{m^3}{d},\end{aligned}$$

where the last inequality holds for $d \geq 3$. Finally, by Lemma 3.3,

$$\Lambda(W+r) \leq \Lambda(W+m) \leq |\mathcal{B}| d^{\frac{2}{2+\theta} + \frac{\theta(1-\theta)}{18}}.$$

Moreover,

$$\frac{2}{2+\theta} + \frac{\theta(1-\theta)}{18} < \frac{5-\theta}{5+\theta}.$$

Hence

$$\Lambda(W+r) < |\mathcal{B}| d^{\frac{5-\theta}{5+\theta}},$$

and therefore

$$\operatorname{rank} \Gamma < |\mathcal{B}| d^{\frac{5-\theta}{5+\theta}} \sum_{r=0}^{m-1} B_r < |\mathcal{B}| m^3 d^{-\frac{2\theta}{5+\theta}}.$$

□

3.3 The Interpolation Lemma

We now complete the interpolation, which achieves the main goal of this section.

Proposition 3.13 (Interpolation)**.** *Let degree k and blocklength n be such that $k/n \leq (1-\theta)\varepsilon$, let $A = \lceil \varepsilon n \rceil$, and let $\mathcal{Q} \subseteq \mathbb{F}_q[X, Y_0, \ldots, Y_d]$ be the interpolation space described in (18). Assume further that (20) holds. Then, for any received word $\vec{y} = (y_1, \ldots, y_n) \in \mathbb{F}_q^n$, there exists a nonzero $Q \in \mathcal{Q}$ such that any polynomial P of degree at most $k-1$ agreeing with $\vec{y}$ in at least A positions satisfies*

$$Q\big(X, P(X), P^{[1]}(X), \ldots, P^{[d]}(X)\big) \equiv 0.$$

Proof. By Lemma 3.2 and the inequality on A in (10),

$$\dim \mathcal{Q} \geq \frac{\theta^3}{384}|\mathcal{B}|(k-1)m^3$$

On the other hand, Lemma 3.12 shows that the number of homogeneous linearly independent constraints imposed over all instances of (15) is at most $n|\mathcal{B}|m^3 d^{\frac{-2\theta}{5+\theta}}$. We first verify that $\dim \mathcal{Q} > n|\mathcal{B}|m^3 d^{\frac{-2\theta}{5+\theta}}$, which will imply that the linear system has a nonzero solution Q satisfying (15). To see this, note that by the inequality above, it is sufficient to show

$$\frac{\theta^3}{384}\frac{k-1}{n}d^{\frac{2\theta}{5+\theta}} > 1.$$

Since we set $k = \lfloor (1-\theta)\varepsilon n \rfloor$, we have

$$\frac{k-1}{n} \geq \frac{(1-\theta)\varepsilon}{2},$$

and by choice of d in (10), we have

$$d^{\frac{2\theta}{5+\theta}} \geq \varepsilon^{-\frac{6}{5+\theta}}.$$

Altogether, this gives

$$\frac{\theta^3}{384}\frac{k-1}{n}d^{\frac{2\theta}{5+\theta}} \geq \frac{\theta^3(1-\theta)}{768}\varepsilon^{-\frac{(1-\theta)}{(5+\theta)}},$$

which is greater than 1 for all sufficiently small ε—precisely when (20) holds.

Fix this Q. We show that it satisfies the assertion of the lemma. Let $P \in \mathbb{F}_q[X]$ be a polynomial of degree at most $k-1$. Then, at every agreement evaluation point α between P and $\vec{y}$, Lemma 3.1 gives

$$Q\Big(\alpha + T, P(\alpha+T), P^{[1]}(\alpha+T), \ldots, P^{[d]}(\alpha+T)\Big) \equiv 0 \mod T^m.$$

Hence the specialization

$$Q\big(X, P(X), P^{[1]}(X), \ldots, P^{[d]}(X)\big)$$

has at least A distinct roots, each of multiplicity at least m. It therefore has at least mA roots counted with multiplicity. By the second item of Lemma 3.2, its degree is strictly smaller than mA, so the polynomial $Q\big(X, P(X), P^{[1]}(X), \ldots, P^{[d]}(X)\big)$ is identically zero. □

4 Algorithm and Analysis: Proof of Theorem 1.1

The goal of this section is to show how the interpolation techniques of Section 3 can be leveraged to prove Theorem 1.1. We begin by presenting our novel list-decoding algorithm.

4.1 The List-decoding Algorithm

We present our list-decoding algorithm in Algorithm 1. At a high-level, interpolation equations in Section 3 are used to compute an interpolation polynomial $Q \in \mathbb{F}_q[X, Y_0, Y_1, \ldots, Y_d]$ of suitable degree. Then, the algorithm of Kopparty [Kop15] is invoked (as stated in Theorem 2.1) to find the list of potential message polynomials P.

Algorithm 1: List-decoding of Reed–Solomon Codes

Input: Parameters $n, k, q, \theta, \varepsilon$. Evaluation points $(\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_q^n$, Received message $(y_1, \ldots, y_n) \in \mathbb{F}_q^n$.

Output: A list $\mathcal{L}$ of polynomials $P \in \mathbb{F}_q[X]$ of degree at most $k-1$.

1 Set $A = \lceil \varepsilon n \rceil$, $d = \lceil \varepsilon^{-3/\theta} \rceil$, $m = d^3$,$B = \lceil mA/(k-1) \rceil$, $W = \lfloor (1+\theta/2)dm/\log(ed) \rfloor$.

2 Define the monomial space $\mathcal{Q}$ as in (18). Namely,

$$\mathcal{Q} = \operatorname{span}\left\{ X^a Y_0^{b_0} Y_1^{b_1} Y^c \; : \; \begin{array}{l} a, b_0, b_1, c_2, \ldots, c_d \in \mathbb{Z}_{\geq 0}, \\ \sum_{j=2}^{d} c_j \leq \lceil (1+3\theta/4)m \rceil, \quad b_1 \leq m, \\ b_0 + b_1 + |c| \leq B, \quad \sum_{j=2}^{d} (j-1)c_j \leq W \\ a + (k-1)(b_0 + b_1 + |c|) < mA \end{array} \right\}.$$

3 Invoke Proposition 3.13 to find some nonzero $Q \in \mathcal{Q}$ satisfying (15) at (α_i, y_i) for all $i \in [n]$. That is, for all $i \in [n]$ enforce that the polynomial

$$Q\left(\alpha_i + T, y - \sum_{j=1}^{d} (-T)^j Y_j + T^{d+1} E, Y_1, \ldots, Y_d \right) \tag{25}$$

is divisible by T^m.

4 Invoke Theorem 2.1 to find a list $\mathcal{L}$ of polynomials $P \in \mathbb{F}_q[X]$ of degree at most $k-1$ for which

$$Q(X, P(X), P^{[1]}(X), \ldots, P^{[d]}(X)) \equiv 0.$$

5 Remove all $P \in \mathcal{L}$ for which $(P(\alpha_1), \ldots, P(\alpha_n))$ has agreement less than A with $(y_1, \ldots, y_n)$.

6 **return** $\mathcal{L}$

4.2 Analysis

We now state our formal list-decoding result.

Theorem 4.1 (Main Result)**.** *Fix parameters $n \geq k \geq 1$ and $\varepsilon, \theta \in (0,1)$ such that, $k > \lceil \varepsilon^{-3/\theta} \rceil$, $k/n \leq (1-\theta)\varepsilon$, and*

$$\varepsilon < \left(\frac{\theta^3 (1-\theta)}{768} \right)^{\frac{5+\theta}{1-\theta}}. \tag{26}$$

Further select a prime q such that $q \geq \max(n, 4\varepsilon^{1-9/\theta} n/k)$. Then, for any distinct evaluation points $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$ and any received word $(y_1, \ldots, y_n) \in \mathbb{F}_q^n$, then Algorithm 1 outputs, in $q^{O(\varepsilon^{-12/\theta})}$ time, a list of all univariate polynomials $P \in \mathbb{F}_q[X]$ of degree less than k such that the agreement between $(P(\alpha_1), \ldots, P(\alpha_n))$ and $(y_1, \ldots y_n)$ is at least εn. Furthermore, the lenght of this list is at most $q^{O(\varepsilon^{-3/\theta})}$.

Before we prove, Theorem 4.1, we show how Theorem 4.1 implies Theorem 1.1.

Proof of Theorem 1.1. For fixed $\theta \in (0,1)$ it is clear that (26) holds for all all sufficiently small $\varepsilon > 0$. Invoking Theorem 4.1, for any prime $q \geq \max(n, 4\varepsilon^{1-9/\theta} n/k) = \Theta_{\theta,\varepsilon}(n)$ (which exists by Bertrand's postulate), we have that any Reed-Solomon code with n distinct evaluation points over $\mathbb{F}_q$ of rate at most $(1-\theta)\varepsilon$ can be decoded up to radius $(1-\varepsilon)n$ in time $q^{O(\varepsilon^{-12/\theta})}$ with a list size at most $q^{O(\varepsilon^{-9/\theta})}$. □

We now turn to proving Theorem 4.1.

Proof of Theorem 4.1. We first prove that Algorithm 1 is correct. For our parameters $k, n, \varepsilon, \theta$ satisfying $k/n \le (1-\theta)\varepsilon$ and (26), we have by Proposition 3.13 we are guaranteed to find a nonzero $Q \in \mathcal{Q}$ such that for all $b \ge 0$, the coefficient of E^b in

$$Q\left(\alpha_i + T, y - \sum_{j=1}^{d} (-T)^j Y_j + TE, Y_1, \ldots, Y_d\right)$$

is divisible by T^{m-db}. By substituting E with $T^d E$, we have that the coefficient of E^b in

$$Q\left(\alpha_i + T, y - \sum_{j=1}^{d} (-T)^j Y_j + T^{d+1} E, Y_1, \ldots, Y_d\right)$$

is divisible by T^m. Thus, (25) is indeed divisible by T^m.

To apply the root-finding algorithm in Theorem 2.1, we can see the choice of the parameter k is valid as $k > \lceil \varepsilon^{-3/\theta} \rceil = d$. We need to verify that the field size q is sufficiently large relative to Q. First, we bound $\deg_{Y_i} Q$ for all $i \in \{0, 1, \ldots, d\}$. Since for any $X^a Y_0^{b_0} Y_1^{b_1} Y^c \in \mathcal{Q}$ we have that $b_0 + b_1 + |c| \le B$, observe

$$\deg_{Y_i} Q \le B \le \left\lceil \frac{mA}{k-1} \right\rceil \le 2d^3 \varepsilon \frac{n}{k} < 4\varepsilon^{1-9/\theta} \frac{n}{k} \le q.$$

where the last line follows from the fact that $q \ge 4\varepsilon^{1-9/\theta} \frac{n}{k}$.

Furthermore, since for any $X^a Y_0^{b_0} Y_1^{b_1} Y^c \in \mathcal{Q}$, we have that $a + (k-1)(b_0 + b_1 + |c|) < mA$, we have that

$$\deg_{1,k-1,k-2,\ldots,k-d-1} Q < mA < 2d^3 \varepsilon n \le 4\varepsilon^{1-9/\theta} \frac{n}{k} \cdot k \le q^2,$$

where the last inequality follows from the fact that $q \ge n \ge k$ and $q \ge 4\varepsilon^{1-9/\theta} \frac{n}{k}$.

To finish, we give a run-time analysis. The time needed by Theorem 2.1 is $q^{O(d+1)} = q^{O(\varepsilon^{-3/\theta})}$, so it suffices to bound the time it takes to compute Q.

First, we upper-bound the number of monomials in $\mathcal{Q}$. For any $X^a Y_0^{b_0} Y_1^{b_1} Y^c \in \mathcal{Q}$, we have that $\sum_{j=2}^{d} c_j \le W \le 2d^4 = O(\varepsilon^{-12/\theta})$, so there are at most $d^{O(\varepsilon^{-12/\theta})} \le q^{O(\varepsilon^{-12/\theta})}$ choices for c. Since $b_0 + b_1 \le B$, there are at most $B^2 \le q^2$ choices for b_0 and b_1. Finally, there are at most $mA \le q^2$ choice for a. Thus, $\mathcal{Q}$ has at most $q^4 \cdot q^{O(\varepsilon^{-12/\theta})} = q^{O(\varepsilon^{-12/\theta})}$ monomials. To count the number of constraints imposed, observe when writing out the expansion (25), each monomial of Q can become up to $(mA+1)\binom{B+d+1}{d+1} \le q^2 \cdot q^{2(d+1)}$ monomials. Thus, the total number of equations imposed is at most $n \cdot q^{2d+4} \cdot q^{O(\varepsilon^{-12/\theta})} = q^{O(\varepsilon^{-12/\theta})}$. It is clear that each constraint can be computed in $q^{O(\varepsilon^{-12/\theta})}$ time, so the total time it takes to set up the linear system defining Q can be done in $q^{O(\varepsilon^{-12/\theta})}$ time. Furthermore, it only takes $q^{O(\varepsilon^{-12/\theta})}$ to find a nonzero Q in the kernel of this linear system. Thus, altogether the runtime is $q^{O(\varepsilon^{-12/\theta})}$ with at most $q^{O(\varepsilon^{-3/\theta})}$ polynomials found. □

References

[ACFY24] Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. "STIR: Reed–Solomon Proximity Testing with Fewer Queries". In: *Advances in Cryptology – CRYPTO 2024*. Vol. 14929. Lecture Notes in Computer Science. Springer, 2024, pp. 380–413. DOI: 10.1007/978-3-031-68403-6_12.

[ACFY25] Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. "WHIR: Reed–Solomon Proximity Testing with Super-Fast Verification". In: *Advances in Cryptology – EUROCRYPT 2025*. Vol. 15604. Lecture Notes in Computer Science. Springer, 2025, pp. 214–243. DOI: 10.1007/978-3-031-91134-7_8.

[AGG+25] Omar Alrabiah, Zeyu Guo, Venkatesan Guruswami, Ray Li, and Zihan Zhang. "Random Reed–Solomon Codes Achieve List-Decoding Capacity with Linear-Sized Alphabets". In: *Advances in Combinatorics* 2025.8 (Oct. 2025), pp. 1–39. DOI: 10.19086/aic.2025.8.

[AGKS07] Noga Alon, Venkatesan Guruswami, Tali Kaufman, and Madhu Sudan. "Guessing Secrets Efficiently via List Decoding". In: *ACM Transactions on Algorithms* 3.4 (2007). DOI: 10.1145/1290672.1290679.

[AGL24] Omar Alrabiah, Venkatesan Guruswami, and Ray Li. "Randomly Punctured Reed–Solomon Codes Achieve List-Decoding Capacity over Linear-Sized Fields". In: *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*. 2024, pp. 1458–1469. DOI: 10.1145/3618260.3649664.

[AGS03] Adi Akavia, Shafi Goldwasser, and Shmuel Safra. "Proving Hard-Core Predicates Using List Decoding". In: *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2003, pp. 146–157. DOI: 10.1109/SFCS.2003.1238189.

[AHS26] Vikrant Ashvinkumar, Mursalin Habib, and Shashank Srivastava. "Algorithmic improvements to list decoding of folded reed-solomon codes". In: *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM. 2026, pp. 880–898.

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. "Fast Reed–Solomon Interactive Oracle Proofs of Proximity". In: *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*. Vol. 107. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018, 14:1–14:17. DOI: 10.4230/LIPIcs.ICALP.2018.14.

[BCH+26] Eli Ben-Sasson, Dan Carmon, Ulrich Haböck, Swastik Kopparty, and Shubhangi Saraf. "On Proximity Gaps of Reed–Solomon Codes". In: *Proceedings of the 58th Annual ACM Symposium on Theory of Computing (STOC)*. ACM, 2026, pp. 1157–1167. DOI: 10.1145/3798129.3800827.

[BCI+23] Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. "Proximity Gaps for Reed–Solomon Codes". In: *Journal of the ACM* 70.5 (2023). DOI: 10.1145/3614423.

[Ber68] Elwyn R. Berlekamp. *Algebraic Coding Theory*. New York: McGraw-Hill, 1968.

[BGM24] Joshua Brakensiek, Sivakanth Gopi, and Visu Makam. "Generic Reed–Solomon Codes Achieve List-Decoding Capacity". In: *SIAM Journal on Computing* 53.4 (2024), pp. 1395–1430. DOI: 10.1137/23M1598064.

[BGW19] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. "Completeness theorems for non-cryptographic fault-tolerant distributed computation". In: *Providing sound foundations for cryptography: on the work of Shafi Goldwasser and Silvio Micali*. 2019, pp. 351–371.

[BKR10] Eli Ben-Sasson, Swastik Kopparty, and Jaikumar Radhakrishnan. "Subspace Polynomials and Limits to List Decoding of Reed–Solomon Codes". In: *IEEE Transactions on Information Theory* 56.1 (2010), pp. 113–120. DOI: 10.1109/TIT.2009.2034780.

[CCD88] David Chaum, Claude Crépeau, and Ivan Damgard. "Multiparty unconditionally secure protocols". In: *Proceedings of the twentieth annual ACM symposium on Theory of computing*. 1988, pp. 11–19.

[CGG+26] Abhranil Chatterjee, Sumanta Ghosh, Zeyu Guo, Rohit Gurjar, Roshan Raj, and Zihan Zhang. *A Note on Explicit Constructions of List-Decodable Reed–Solomon Codes Beyond the Johnson Bound*. Personal communication. 2026.

[CHK25] Soham Chatterjee, Prahladh Harsha, and Mrinal Kumar. *Deterministic List Decoding of Reed–Solomon Codes*. Tech. rep. TR25-170. Revision 1, March 25, 2026. Electronic Colloquium on Computational Complexity, 2025.

[CW07] Qi Cheng and Daqing Wan. "On the List and Bounded Distance Decodability of Reed–Solomon Codes". In: *SIAM Journal on Computing* 37.1 (2007), pp. 195–209. DOI: 10.1137/S0097539705447335.

[CZ25] Yeyuan Chen and Zihan Zhang. "Explicit folded reed-solomon and multiplicity codes achieve relaxed generalized singleton bounds". In: *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*. 2025, pp. 1–12.

[DKSS13] Zeev Dvir, Swastik Kopparty, Shubhangi Saraf, and Madhu Sudan. "Extensions to the method of multiplicities, with applications to Kakeya sets and mergers". In: *SIAM Journal on Computing* 42.6 (2013), pp. 2305–2328.

[Eli57] Peter Elias. *List decoding for noisy channels*. Tech. rep. Research Laboratory of Electronics, Massachusetts Institute of Technology, 1957.

[FY92] Matthew Franklin and Moti Yung. "Communication complexity of secure computation". In: *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*. 1992, pp. 699–710.

[GGG18] Venkata Gandikota, Badih Ghazi, and Elena Grigorescu. "NP-Hardness of Reed–Solomon Decoding, and the Prouhet–Tarry–Escott Problem". In: *SIAM Journal on Computing* 47.4 (2018), pp. 1547–1584. DOI: 10.1137/16M110349X.

[GHKS24] Rohan Goyal, Prahladh Harsha, Mrinal Kumar, and Ashutosh Shankar. "Fast list decoding of univariate multiplicity and folded Reed-Solomon codes". In: *2024 IEEE 65th Annual Symposium on Foundations of Computer Science—FOCS 2024*. IEEE Computer Soc., Los Alamitos, CA, 2024, pp. 328–343. ISBN: 979-8-3315-1674-1. DOI: 10.1109/FOCS61266.2024.00028. URL: https://doi.org/10.1109/FOCS61266.2024.00028.

[GKKG06] Warren J Gross, Frank R Kschischang, Ralf Koetter, and P Glenn Gulak. "Applications of algebraic soft-decision decoding of Reed-Solomon codes". In: *IEEE transactions on communications* 54.7 (2006), pp. 1224–1234.

[GL89] Oded Goldreich and Leonid A. Levin. "A Hard-Core Predicate for All One-Way Functions". In: *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC)*. ACM, 1989, pp. 25–32. DOI: 10.1145/73007.73010.

[GLS+21] Zeyu Guo, Ray Li, Chong Shangguan, Itzhak Tamo, and Mary Wootters. "Improved List-Decodability of Reed–Solomon Codes via Tree Packings". In: *Proceedings of the 62nd IEEE Symposium on Foundations of Computer Science*. 2021, pp. 708–719. DOI: 10.1109/FOCS52979.2021.00074.

[Gol07] Ian Goldberg. "Improving the robustness of private information retrieval". In: *2007 IEEE Symposium on Security and Privacy (SP'07)*. IEEE. 2007, pp. 131–148.

[GR06] Venkatesan Guruswami and Atri Rudra. "Limits to List Decoding Reed–Solomon Codes". In: *IEEE Transactions on Information Theory* 52.8 (2006), pp. 3642–3649. DOI: 10.1109/TIT.2006.878164.

[GR08] Venkatesan Guruswami and Atri Rudra. "Explicit Codes Achieving List Decoding Capacity: Error-Correction with Optimal Redundancy". In: *IEEE Transactions on Information Theory* 54.1 (2008), pp. 135–150. DOI: 10.1109/TIT.2007.911222.

[GR21] Zeyu Guo and Noga Ron-Zewi. "Efficient List-Decoding with Constant Alphabet and List Sizes". In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 2021, pp. 1502–1515. DOI: 10.1145/3406325.3451046.

[GRS00] Oded Goldreich, Ronitt Rubinfeld, and Madhu Sudan. "Learning Polynomials with Queries: The Highly Noisy Case". In: *SIAM Journal on Discrete Mathematics* 13.4 (2000), pp. 535–570. DOI: 10.1137/S0895480198344540.

[GRS12] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. "Essential coding theory". In: *Draft available at http://www. cse. buffalo. edu/atri/courses/coding-theory/book* 2.1 (2012).

[GS99] Venkatesan Guruswami and Madhu Sudan. "Improved Decoding of Reed–Solomon and Algebraic-Geometric Codes". In: *IEEE Transactions on Information Theory* 45.6 (1999), pp. 1757–1767. DOI: 10.1109/18.782097.

[GST23] Eitan Goldberg, Chong Shangguan, and Itzhak Tamo. "List-Decoding and List-Recovery of Reed–Solomon Codes Beyond the Johnson Radius for Every Rate". In: *IEEE Transactions on Information Theory* 69.4 (2023), pp. 2261–2268. DOI: 10.1109/TIT.2022.3222877.

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil P. Vadhan. "Unbalanced Expanders and Randomness Extractors from Parvaresh–Vardy Codes". In: *Journal of the ACM* 56.4 (2009), 20:1–20:34. DOI: 10.1145/1538902.1538904.

[GV05] Venkatesan Guruswami and Alexander Vardy. "Maximum-Likelihood Decoding of Reed–Solomon Codes is NP-Hard". In: *IEEE Transactions on Information Theory* 51.7 (2005), pp. 2249–2256. DOI: 10.1109/TIT.2005.850102.

[GW13] Venkatesan Guruswami and Carol Wang. "Linear-Algebraic List Decoding for Variants of Reed–Solomon Codes". In: *IEEE Transactions on Information Theory* 59.6 (2013), pp. 3257–3268. DOI: 10.1109/TIT.2013.2246816.

[GW16] Venkatesan Guruswami and Mary Wootters. "Repairing reed-solomon codes". In: *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*. 2016, pp. 216–226.

[GX22] Venkatesan Guruswami and Chaoping Xing. "Optimal Rate List Decoding over Bounded Alphabets Using Algebraic-Geometric Codes". In: *Journal of the ACM* 69.2 (2022), pp. 1–48. DOI: 10.1145/3506668.

[GZ23] Zeyu Guo and Zihan Zhang. "Randomly Punctured Reed–Solomon Codes Achieve the List Decoding Capacity over Polynomial-Size Alphabets". In: *Proceedings of the 64th IEEE Symposium on Foundations of Computer Science*. 2023, pp. 164–176. DOI: 10.1109/FOCS57990.2023.00019.

[GZ61] Daniel Gorenstein and Neal Zierler. "A class of error-correcting codes in pˆm symbols". In: *Journal of the Society for Industrial and Applied Mathematics* 9.2 (1961), pp. 207–214.

[JS06a] Ari Juels and Madhu Sudan. "A Fuzzy Vault Scheme". In: *Designs, Codes and Cryptography* 38.2 (2006), pp. 237–257. DOI: 10.1007/s10623-005-6343-z.

[JS06b] Ari Juels and Madhu Sudan. "A fuzzy vault scheme". In: *Designs, Codes and Cryptography* 38.2 (2006), pp. 237–257.

[Kop15] Swastik Kopparty. "List-Decoding Multiplicity Codes". In: *Theory of Computing* 11.5 (2015), pp. 149–182. DOI: 10.4086/toc.2015.v011a005.

[KRSW23] Swastik Kopparty, Noga Ron-Zewi, Shubhangi Saraf, and Mary Wootters. "Improved List Decoding of Folded Reed–Solomon and Multiplicity Codes". In: *SIAM Journal on Computing* 52.3 (2023), pp. 794–840. DOI: 10.1137/20M1370215.

[KSY14] Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. "High-rate codes with sublinear-time decoding". In: *J. ACM* 61.5 (2014), Art. 28, 20. ISSN: 0004-5411,1557-735X. DOI: 10.1145/2629416. URL: https://doi.org/10.1145/2629416.

[KT22] Itay Kalev and Amnon Ta-Shma. "Unbalanced expanders from multiplicity codes". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2022)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. 2022, pp. 12–1.

[KV03] Ralf Koetter and Alexander Vardy. "Algebraic soft-decision decoding of Reed-Solomon codes". In: *IEEE Transactions on Information Theory* 49.11 (2003), pp. 2809–2825.

[Mas69] James L. Massey. "Shift-Register Synthesis and BCH Decoding". In: *IEEE Transactions on Information Theory* 15.1 (1969), pp. 122–127. DOI: 10.1109/TIT.1969.1054260.

[MS81] Robert J. McEliece and Dilip V. Sarwate. "On sharing secrets and Reed-Solomon codes". In: *Communications of the ACM* 24.9 (1981), pp. 583–584.

[Ped91] Torben Pryds Pedersen. "A threshold cryptosystem without a trusted party". In: *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer. 1991, pp. 522–526.

[Pet60] Wesley Peterson. "Encoding and error-correction procedures for the Bose-Chaudhuri codes". In: *IRE Transactions on information theory* 6.4 (1960), pp. 459–470.

[Pla97] James S Plank. "A tutorial on Reed–Solomon coding for fault-tolerance in RAID-like systems". In: *Software: Practice and Experience* 27.9 (1997), pp. 995–1012.

[PLS+09] James S Plank, Jianqiang Luo, Catherine D Schuman, Lihao Xu, Zooko Wilcox-O'Hearn, et al. "A performance evaluation and examination of open-source erasure coding libraries for storage." In: *Fast*. Vol. 9. 2009, pp. 253–265.

[PS85] M Pursley and W Stark. "Performance of Reed-Solomon coded frequency-hop spread-spectrum communications in partial-band interference". In: *IEEE Transactions on Communications* 33.8 (1985), pp. 767–774.

[PV05] Farzad Parvaresh and Alexander Vardy. "Correcting errors beyond the Guruswami-Sudan radius in polynomial time". In: *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*. IEEE. 2005, pp. 285–294.

[RB89] Tal Rabin and Michael Ben-Or. "Verifiable secret sharing and multiparty protocols with honest majority". In: *Proceedings of the twenty-first annual ACM symposium on Theory of computing*. 1989, pp. 73–85.

[RR03] Gitit Ruckenstein and Ron M. Roth. "Bounds on the List-Decoding Radius of Reed–Solomon Codes". In: *SIAM Journal on Discrete Mathematics* 17.2 (2003), pp. 171–195. DOI: 10.1137/S0895480101395506.

[RS60] Irving S. Reed and Gustave Solomon. "Polynomial Codes over Certain Finite Fields". In: *Journal of the Society for Industrial and Applied Mathematics* 8.2 (1960), pp. 300–304. DOI: 10.1137/0108018.

[RW14] Atri Rudra and Mary Wootters. "Every List-Decodable Code for High Noise Has Abundant Near-Optimal Rate Puncturings". In: *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*. New York, NY, USA: Association for Computing Machinery, 2014, pp. 764–773. DOI: 10.1145/2591796.2591797.

[Sha79] Adi Shamir. “How to Share a Secret”. In: *Communications of the ACM* 22.11 (1979), pp. 612–613. DOI: 10.1145/359168.359176.

[Sin64] Richard Singleton. “Maximum distance q-nary codes”. In: *IEEE Transactions on Information Theory* 10.2 (1964), pp. 116–118.

[Sri25] Shashank Srivastava. “Improved list size for folded Reed-Solomon codes”. In: *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, Philadelphia, PA, 2025, pp. 2040–2050. ISBN: 978-1-61197-832-2. DOI: 10.1137/1.9781611978322.64. URL: https://doi.org/10.1137/1.9781611978322.64.

[SSW01] Alice Silverberg, Jessica Staddon, and Judy L. Walker. “Efficient Traitor Tracing Algorithms Using List Decoding”. In: *Advances in Cryptology – ASIACRYPT 2001*. Vol. 2248. Lecture Notes in Computer Science. Springer, 2001, pp. 175–192. DOI: 10.1007/3-540-45682-1_11.

[ST23] Chong Shangguan and Itzhak Tamo. “Generalized Singleton Bound and List-Decoding Reed–Solomon Codes Beyond the Johnson Radius”. In: *SIAM Journal on Computing* 52.3 (2023), pp. 684–717. DOI: 10.1137/20M138795X.

[STV01] Madhu Sudan, Luca Trevisan, and Salil Vadhan. “Pseudorandom Generators without the XOR Lemma”. In: *Journal of Computer and System Sciences* 62.2 (2001), pp. 236–266. DOI: 10.1006/jcss.2000.1730.

[Sud97] Madhu Sudan. “Decoding of Reed–Solomon Codes beyond the Error-Correction Bound”. In: *Proceedings of the 35th Annual Allerton Conference on Communication, Control, and Computing*. 1997, pp. 215–224.

[SW08] Hovav Shacham and Brent Waters. “Compact proofs of retrievability”. In: *International conference on the theory and application of cryptology and information security*. Springer. 2008, pp. 90–107.

[Tre01] Luca Trevisan. “Extractors and Pseudorandom Generators”. In: *Journal of the ACM* 48.4 (2001), pp. 860–879. DOI: 10.1145/502090.502099.

[WHPH87] Williamw Wu, David Haccoun, Robert Peile, and Yasuo Hirata. “Coding for satellite communication”. In: *IEEE Journal on Selected Areas in Communications* 5.4 (1987), pp. 724–748.

[Wic92] Stephen B Wicker. “Reed-Solomon error control coding for Rayleigh fading channels with feedback”. In: *IEEE transactions on Vehicular Technology* 41.2 (1992), pp. 124–133.

[Woz58] John M Wozencraft. “List decoding”. In: *Quarterly Progress Report* 48 (1958), pp. 90–95.

A A Proof of Lemma 3.9

We first recall the following lemma we seek to prove.

Lemma A.1. *Let $S = \mathrm{conv}\{v_0, \ldots, v_{d-1}\}$ be a $d-1$-dimensional simplex, and let X be uniformly distributed on S. Then*

$$\mathbb{E}[X] = \frac{1}{d}\sum_{i=0}^{d-1} v_i.$$

In particular, the expectation of X is the centroid of S.

Proof. We start by considering the standard simplex

$$\Delta_{d-1} = \left\{ (\lambda_0, \ldots, \lambda_{d-1}) \in \mathbb{R}^d_{\geq 0} : \sum_{i=0}^{d-1} \lambda_i = 1 \right\},$$

and define the affine map

$$F(\lambda_0, \ldots, \lambda_{d-1}) = \sum_{i=0}^{d-1} \lambda_i v_i.$$

Since the vertices $v_0, \ldots, v_{d-1}$ are affinely independent, F is an affine bijection from Δ_{d-1} onto S.

The Jacobian of F, restricted to the affine hull of Δ_{d-1}, is a fixed nonzero constant. Consequently, F multiplies the volume of every measurable subset of Δ_{d-1} by the same factor. Thus, if $Z = (Z_0, \ldots, Z_{d-1})$ is uniformly distributed on Δ_{d-1}, then $F(Z)$ is uniformly distributed on S. We may therefore write

$$X \overset{\mathrm{d}-1}{=} \sum_{i=0}^{d-1} Z_i v_i.$$

The uniform distribution on Δ_{d-1} is invariant under every permutation of its coordinates. Hence

$$\mathbb{E}[Z_0] = \mathbb{E}[Z_1] = \cdots = \mathbb{E}[Z_{d-1}].$$

On the other hand, $\sum_{i=0}^{d-1} Z_i = 1$ identically. Taking expectations gives

$$\sum_{i=0}^{d-1} \mathbb{E}[Z_i] = 1,$$

and therefore

$$\mathbb{E}[Z_i] = \frac{1}{d} \qquad \text{for every } i.$$

By linearity of expectation,

$$\mathbb{E}[X] = \sum_{i=0}^{d-1} \mathbb{E}[Z_i] v_i = \frac{1}{d} \sum_{i=0}^{d-1} v_i,$$

which is precisely the centroid of S. □

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il