

Explicit Reed-Solomon Subspace Designs: Optimal List and Line Decoding

Omar Alrabiah* Rohan Goyal† Venkatesan Guruswami‡

Abstract

For any desired rate $R \in (0, 1)$, and gap to list-decoding capacity $\epsilon > 0$, we give an explicit $n^{O_\epsilon(1)}$ time construction of Reed-Solomon codes over extension fields of size polynomial in the block length n that can be list-decoded from a fraction $1 - R - \epsilon$ of errors with the optimal list-size of $O(\frac{1-R}{\epsilon})$. We also give a simpler, strongly explicit construction of evaluation points for which the associated Reed-Solomon code can be list-decoded from $1 - R - \epsilon$ errors with list size $(1/\epsilon)^{O(1/\epsilon)}$. We provide a deterministic list-decoding algorithm for both constructions running in time $O_\epsilon(1) \text{poly}(n)$.

In fact, our Reed-Solomon codes are near-optimal subspace-design codes over a subfield. Via known results, this directly implies the combinatorial list-decoding guarantee, as well as near-optimal list-recovery bounds. We also present a new reduction showing that the subspace-design guarantee also implies line decoding over the extension field itself, which then yields proximity gaps and mutual correlated agreement up to proximity parameter approaching the optimal $1 - R$ radius.

Compared to the recent algorithm of Brakensiek, Chen, Putterman, Zhang, and Zheng (2026), which list decodes every Reed-Solomon code over prime fields up to capacity, our approach achieves constant (in fact, near-optimal) list size rather than a polynomially growing one, at the cost of carefully choosing the code's evaluation points. In a companion paper, we show that these respective restrictions are unavoidable.

*Department of Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, CA 94709, USA. Email: oarabiah@berkeley.edu. Research supported in part by a Saudi Arabian Cultural Mission (SACM) Scholarship, ONR grant N00014-24-1-2491, and V. Guruswami's Simons Investigator Award.

†Massachusetts Institute of Technology, Cambridge, MA, USA. Email: rohan_g@mit.edu. Research supported by Yael Tauman Kalai's NSF Award 6956054.

‡Department of Electrical Engineering and Computer Sciences, Department of Mathematics, and the Simons Institute for the Theory of Computing, UC Berkeley, Berkeley, CA 94709, USA. Email: venkatg@berkeley.edu. Research supported in part by a Simons Investigator Award, a UC Noyce initiative award, DOD Advanced Research Projects Agency grant HR0011262E031, and NSF grant 2211972.

Contents

1	Introduction	1
1.1	Our results	4
1.2	Technical overview	7
1.3	Organization	10
1.4	AI Disclosure	10
2	Preliminaries	11
2.1	Subspace-design codes and decoding consequences	11
2.2	Polynomial modules and degree	14
2.3	Construction of Field Extensions and Deterministic Factorization	15
2.4	Independent conjugates and coordinate movement	15
3	Main theorem and combinatorial consequences	16
3.1	List recovery	17
3.2	Curve decoding	18
3.3	Line decoding	18
3.4	Proximity gap, correlated agreement, and mutual correlated agreement	18
4	Reed-Solomon Subspace Designs	19
4.1	Assumptions and the subspace-design theorem	19
4.2	From effective evaluation points to subspace designs	20
4.3	Proof of the transfer theorem	22
5	Efficient List Decoding	23
5.1	The two-step list-decoding theorems	24
5.2	Finding a low-dimensional $\mathbb{K}(X)$ -space (Theorem 5.2)	25
6	Pruning low-dimensional spaces of Reed-Solomon codes	26
6.1	Pruning for $\mathbb{F}$ -subspaces of Reed-Solomon codes (Theorem 6.1)	28
6.1.1	Interpolate a new relation or branch	29
6.1.2	Bounding the Output List-Size	31
6.2	Pruning $\mathbb{K}(X)$ -subspaces (Theorem 5.3)	33
6.2.1	Finding a new relation	36
6.2.2	From interpolation failure to branching	37
6.2.3	Bounding the list and the running time	40
7	Effective Evaluation Points and Explicit Instantiation	40
8	Improved line decoding and proximity gaps	43
8.1	Improved combinatorial consequences	43
8.2	Proof of the extension-field line-decoding theorem	44
9	Acknowledgements	47
	References	47
A	Computing a common zero	54

1 Introduction

Reed-Solomon (RS) codes [RS60] are among the most fundamental and widely deployed error-correcting codes. At their core, they encode messages by evaluating low-degree polynomials over a finite field. Specifically, given a finite field $\mathbb{F}$ and n distinct evaluation points $(\alpha_1, \dots, \alpha_n)$ in $\mathbb{F}$, the Reed-Solomon code of dimension k is defined as

$$\text{RS}_{\mathbb{F}}((\alpha_1, \dots, \alpha_n), k) := \{(f(\alpha_1), \dots, f(\alpha_n)) : f \in \mathbb{F}[X], \deg f < k\} \subseteq \mathbb{F}^n,$$

and has rate $R = k/n$. Since any two distinct polynomials of degree less than k agree at at most $k - 1$ points, the code has minimum distance $n - k + 1$, attaining the Singleton bound. Thus, Reed-Solomon codes are canonical examples of maximum distance separable (MDS) codes. Beyond their optimal distance properties, Reed-Solomon codes also admit efficient decoding algorithms. Classical decoders due to Peterson, Berlekamp-Massey and Berlekamp-Welch [Pet60, Ber68, Mas69, WB86] can correct any fraction of errors approaching $(1 - R)/2$, which is the most that *unique* decoding can achieve at rate R .

Beyond half the minimum distance, a decoder can no longer always identify the transmitted codeword, but it can output the *list* of all codewords within the given radius. This notion, known as *list decoding* [Eli57, Woz58], is useful when the output list remains small. By the Johnson bound [Joh62], *every* code of relative distance $1 - R$ has polynomially bounded list sizes up to $1 - \sqrt{R}$, which exceeds $(1 - R)/2$ for every rate $R \in (0, 1)$ (the $1 - \sqrt{R}$ threshold is therefore referred to as the *Johnson radius*). Just as in the unique-decoding setting, Reed-Solomon codes admit efficient decoding algorithms for list decoding. In particular, the algorithms of Sudan [Sud97] and Guruswami and Sudan [GS99] showed that Reed-Solomon codes can be efficiently list-decoded all the way up to the Johnson radius.

List-decoding capacity. While this result remained the state of the art for explicit Reed-Solomon codes till very recently, the Johnson radius is far from the ultimate limit of list decoding. For every rate $R \in (0, 1)$ and $\varepsilon > 0$, there are codes of rate R that are list-decodable from a fraction $1 - R - \varepsilon$ of errors with list-size bounded by $O(1/\varepsilon)$, whereas list decoding beyond radius $1 - R$ requires exponentially large lists [ZP82, Eli91]. *Variants* of Reed-Solomon codes were shown to achieve list-decoding capacity explicitly and efficiently: building on Parvaresh and Vardy [PV05], Guruswami and Rudra [GR08] showed that *folded* Reed-Solomon codes can be efficiently list-decoded from a fraction $1 - R - \varepsilon$ of errors, and the same was later shown for univariate *multiplicity codes* [GW13, Kop15]. While the original list-size guarantee grew as $n^{O(1/\varepsilon)}$, constant list sizes (depending only on ε) were then obtained first for subcodes of these codes [Gur11, DL12, GW13] and then for the codes themselves [KRSW23, Tam24, Sri25], further improved to $O(1/\varepsilon^2)$ by [Sri25], and culminating in the breakthrough work of [CZ25] who showed the matching optimal $O(1/\varepsilon)$ bound [CZ25] for these codes. In fact, the results of [Sri25, CZ25] held for a broader class of codes called *subspace-design* codes, which are a class of codes typically over a vector-valued alphabet.

None of these codes is a Reed-Solomon code, however. A folded Reed-Solomon code bundles evaluations at several related points into a single symbol over a larger alphabet, whereas a multiplicity code bundles each evaluation of a polynomial with the evaluations of its derivatives. It is exactly this extra structure that the decoders exploit. Given the centrality of Reed-Solomon codes, this has led to the following prominent question:

Can Reed-Solomon codes themselves, for a suitable choice of evaluation points, achieve list-decoding capacity?

Generic Reed–Solomon codes and list-decoding capacity. To address this question, a large body of work focused on whether there even *exist* RS codes (even if they are not explicitly constructed) that can achieve list-decoding capacity, or at least beat the Johnson radius, in terms of combinatorially having small lists (even if not accompanied by an efficient list-decoding algorithm). The focus was on “generic” Reed–Solomon codes whose evaluation points are sampled randomly from a large enough underlying field. After a sequence of works establishing list-decodability of random Reed–Solomon codes beyond the Johnson radius in various regimes [RW14, ST23, GLS⁺24, FKS22, GST23], a remarkable work of Brakensiek, Gopi and Makam [BGM24] showed that generic Reed–Solomon codes achieve list-decoding capacity. In fact, they showed a more fine-grained statement. Let us say a code is (δ, L) -list-decodable if every Hamming ball of relative radius δ contains at most L codewords. Shangguan and Tamo [ST23] showed that a (δ, L) -list-decodable code of rate R satisfies the *generalized Singleton bound* (GSB)

$$\delta \leq \frac{L}{L+1} (1 - R),$$

which interpolates between the unique-decoding radius (for $L = 1$) and capacity (as $L \rightarrow \infty$). [ST23] also conjectured that generic Reed–Solomon codes attain the GSB for every fixed L , meaning that they can be list-decoded up to radius $\frac{L}{L+1}(1 - R)$ with list-size L and proved tightness for $L = 2, 3$. This conjecture was established in [BGM24], using the theory of *higher-order MDS codes* [BGM22, Rot22] and the GM-MDS theorem [Lov21, YH19]. The field size in the result of [BGM24] grows exponentially in the block length n , and this is in fact necessary for codes meeting the GSB exactly [BDG24]. However, allowing a small gap with respect to the GSB limit, one can get much smaller alphabet sizes: Guo and Zhang [GZ23] showed that randomly punctured Reed–Solomon codes over fields of quadratic size are $(\frac{L}{L+1}(1 - R) - \varepsilon, L)$ -list-decodable for every fixed L and $\varepsilon > 0$, and subsequent work [AGL24, AGG⁺25] reduced the field size to linear in n .

Despite this progress, the existence of good evaluation points is known only through the probabilistic method. A uniformly random set works with high probability, but no deterministic construction was known, nor any efficient way to certify that a given set works. In particular, an explicit construction of Reed–Solomon evaluation points approaching the generalized Singleton bound (GSB) or having constant list-sizes approaching capacity has remained open.¹ As a spoiler, this is the question that the present work resolves. Our codes are constructed over extension fields and have a stronger subspace-design property that yields list-decodability approaching the GSB as a corollary. Further, we also give an efficient algorithm to perform the guaranteed list decoding.

Subspace-Design Codes. Subspace-designs were first introduced by Guruswami and Xing [GX13] as a way to precode Reed–Solomon codes given by evaluating over a subfield in order to make them list-decodable with small lists, and they were first explicitly constructed by Guruswami and Kopparty [GK16] using folded Reed–Solomon and Univariate Multiplicity codes.

In a recent breakthrough [CZ25], it was demonstrated that any code with a near-optimal subspace-design property is optimally list-decodable in the strong sense of approaching the generalized Singleton bound. Subspace-designs have also formed the backbone of many breakthroughs in coding theory recently, such as [Sri25, AHS26, GG26, GG25, BCDZ26a, BCDZ26b, GGSW26, GGH26, GRSZ26, CDG26, Liu26]. They have also found further applications in derandomization, leading to the breakthrough of [CGG⁺26] showing that bipartite matching is in NC.

Informally, [GX13] defined a collection $\{H_1, \dots, H_n\}$ of linear subspaces in $\mathbb{K}^k$ to be a subspace-design if any low-dimensional subspace $A \subseteq \mathbb{K}^k$, the average intersection $\mathbb{E}_i \dim(H_i \cap A)$ is much

¹The earlier mentioned variants of RS codes, like folded Reed–Solomon and univariate multiplicity codes, do approach the GSB, as shown in [CZ25], by virtue of their having the subspace design property [GK16].

smaller than $\dim(A)$. For a $\mathbb{K}$ -additive code $C \subseteq (\mathbb{K}^s)^n$, we say that it is a subspace design code if the collection $\{C_i\}_{i \in [n]}$, where $C_i = \{x \in C \mid x_i = 0\}$, forms a subspace design (see [Definition 2.1](#)). We will use the subspace-design-code terminology coined by [\[GG26\]](#), but an earlier definition was made by Chen and Zhang in [\[CZ25\]](#) who called it a subspace-designable code.

[\[BCDZ26b\]](#) showed that in fact, subspace-design codes have all *local-coordinatewise-linear* (LCL) properties of random linear codes. LCL properties are a broad class of properties defined by [\[LMS25\]](#) to capture list-decoding and list-recovery, and further shown to capture line and curve-decoding by [\[GG26, GGSW26\]](#). Thus, this leads to a natural question in our quest for finding explicit Reed-Solomon codes approaching the generalized Singleton bound: can we construct subspace-design Reed-Solomon codes?

This question as is seems ill-posed as we require the alphabets to be low-dimensional vectors to get the subspace-design property. We can thus modify the question to ask:

Can explicit Reed-Solomon codes over extension-fields have the subspace-design property when viewed as an additive code over a subfield?

This is the key question we tackle in this paper and answer it in the affirmative. This is the first construction of a near-optimal subspace-design code that is also linear over its alphabet. The previous near-optimal constructions with $\tau(r) \leq R + \varepsilon$ are due to [\[GK16\]](#) and [\[GGH26\]](#) who gave the first constant alphabet construction of subspace-design codes building on the ideas of [\[AEL95, ST25, JS26\]](#) using the AEL transform.

Decoding every Reed-Solomon code over prime fields. Our work comes on the heels of a very recent breakthrough by Brakensiek, Chen, Putterman, Zhang and Zheng [\[BCP⁺26\]](#), who gave an efficient algorithm that list-decodes *every* Reed-Solomon code of rate R over a prime field up to capacity. For every prime $p \geq O_\varepsilon(n)$ and every set of n evaluation points in $\mathbb{F}_p$, the associated Reed-Solomon code can be list-decoded from a fraction $1 - R - \varepsilon$ of errors in polynomial time, with list-size $n^{O_\varepsilon(1)}$. (See also the recent exposition by [\[HKS26\]](#).) Thus, over prime fields, *all* Reed-Solomon codes, and not just generic ones, achieve list-decoding capacity! These codes, however, do not approach the GSB and in fact list-size grows with n ; we give a detailed comparison between our results and [\[BCP⁺26\]](#) after [Theorem 1.2](#).

Proximity gaps. In addition to its intrinsic interest in coding theory and pseudorandomness, the question of list decoding Reed-Solomon codes up to capacity has gained recent prominence due to its connection with practical proof systems through the problem of *proximity-gaps*.

A code $C \subseteq \mathbb{F}^n$ has a proximity gap at proximity parameter δ if for every affine line $\{u + \alpha v : \alpha \in \mathbb{F}\}$ in $\mathbb{F}^n$, either *all* of its points are δ -close to C , or only a negligible fraction are [\[AHIV17, BKS18\]](#). This dichotomy underlies the soundness of interactive oracle proofs [\[RRR21, BSCS16\]](#) and the succinct efficient arguments systems built on them, such as FRI, SNARKs, STARKs and their descendants [\[BBHR18, BSBHR19, ACFY26, ACFY25\]](#): the prover claims that several words are close to a Reed-Solomon code, and the verifier tests a random linear combination. Larger proximity parameter δ and better error guarantees lead to shorter proofs.

[\[BSCI⁺23\]](#) proved proximity gaps for Reed-Solomon codes up to the Johnson radius and conjectured that they extend up to $1 - R$. This challenge is the basis of Ethereum’s proximity prize [\[Eth25\]](#), and has attracted intense recent attention [\[DG26, CS26, BSCH⁺26, GG26, BCGM26, Jer26\]](#). Our work constructs Reed-Solomon codes which are subspace-designs over a subfield.

This automatically bestows upon them a “line decoding” guarantee that implies proximity gaps (and stronger correlated agreement guarantees) up to proximity parameter approaching $1 - R$ [\[GG26\]](#)

for affine lines over a subfield of the code. In [Section 8](#), we extend [\[GG26\]](#)'s techniques to also apply over extension fields even when the subspace-design guarantee is only over a subfield. This lets us get the first explicit construction of codes over an alphabet Σ with proximity gaps with error $\frac{\text{poly}(1/\varepsilon)n}{|\Sigma|}$.

1.1 Our results

We now state our results more precisely. As mentioned, our list-decodability and proximity gap guarantees both stem from a stronger subspace-design property of our Reed-Solomon construction, which we now describe.

Our codes are constructed over an extension field $\mathbb{F}/\mathbb{K}$, and therefore their symbols can be viewed as vectors over $\mathbb{K}$. Slightly more formally, we say a $\mathbb{K}$ -linear code $C \subseteq (\mathbb{K}^s)^n$ of rate R is a τ -subspace-design code for $\tau : \mathbb{N} \mapsto \mathbb{R}$, if for all $\mathbb{K}$ -subspaces A of C of dimension at most r , if we consider the subspace A_i of codewords in A which are 0 in the i 'th coordinate, i.e., $A_i = \{a \in A \mid a_i = 0\}$, then on average, for a random $i \in [n]$, $\dim A_i \leq \tau(r) \dim A$. See [Definition 2.1](#) for a formal definition.

A subspace-design code is near optimal up to dimension r if $\tau(r) \leq R + \varepsilon$. That is, the dimension shrinks by a factor of about R (which is best possible) when a random codeword position is set to 0. When $r = 1$, this property is the same as the code's relative distance being at least $1 - R - \varepsilon$, and thus the code being near-MDS. The subspace-design guarantee is thus a higher-dimensional generalization of this near-MDS property.² And just as the MDS property implies optimal unique decoding, due to [\[CZ25\]](#), the subspace-design property implies optimal list-decoding approaching the GSB (for list-sizes up to r).

Theorem 1.1 (Explicit RS codes that are near-optimal subspace-design codes). *For every integer $r \geq 2$, every $0 < \varepsilon < 1$, all sufficiently large n , and every finite field $\mathbb{K}$ with $|\mathbb{K}| \geq n$, one can deterministically construct in time $|\mathbb{K}|^{O_{r,\varepsilon}(1)}$ an extension $\mathbb{F}/\mathbb{K}$ of degree $O_{r,\varepsilon}(1)$ and evaluation points $A = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}^n$ with the following properties. For every $1 \leq k \leq n$, the Reed-Solomon code $C = \text{RS}_{\mathbb{F}}(A, k)$, of rate $R = k/n$, is a τ -subspace-design code over every subfield $\mathbb{K}_0 \subseteq \mathbb{K}$, where*

$$\tau(d) = \min\{1, R + \varepsilon/4\} \quad (1 \leq d \leq r), \quad \tau(d) = 1 \quad (d > r).$$

Moreover, for every integer $L \geq 1$, C is (δ_L, L) -average-radius list-decodable, where

$$\delta_L = \frac{L}{L+1}(1 - R - \varepsilon).$$

Given any received word, the lists within relative distances δ_L and $1 - R - \varepsilon$ can be found deterministically in time $O_{r,\varepsilon}(1) \text{poly}(n, \log |\mathbb{F}|)$, with sizes at most L and $O((1 - R)/\varepsilon)$ respectively.

We in fact show that the subspace-design property holds over $\mathbb{K}$ and prove that the subspace-design property transfers to subfields generically. This is closely related to a question raised by [\[GGH26\]](#) in their paper about the behavior of local properties under field extensions.

We next state our main result for list-decoding more explicitly. The combinatorial guarantee of approaching the GSB follows, via [\[CZ25\]](#), from the subspace-design property above; we additionally develop an efficient list-decoding algorithm.

²Since $\dim A_i \geq \dim A - s$, one needs s growing with r in order for the subspace design property to hold up to dimension r ; in particular for RS codes, one has to work over extension fields.

Theorem 1.2 (Explicit RS codes approaching GSB and efficiently list-decodable up to capacity (informal)). *Fix $\varepsilon \in (0, 1)$. For all sufficiently large n , and an arbitrary field $\mathbb{K}$ of size at least n , one can deterministically construct in time $|\mathbb{K}|^{O_\varepsilon(1)}$, an extension field $\mathbb{F}$ over $\mathbb{K}$ with $[\mathbb{F} : \mathbb{K}] = O_\varepsilon(1)$ and evaluation points $A = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}^n$, such that for every $1 \leq k \leq n$ the Reed-Solomon code $\text{RS}_{\mathbb{F}}(A, k)$ of rate $R = k/n$ has the following properties:*

- (a) (Generalized Singleton bound.) *For every list-size L , it is $(\frac{L}{L+1}(1 - R - \varepsilon), L)$ -list-decodable.*
- (b) (Efficient list decoding up to capacity.) *There is a deterministic algorithm that, given $y \in \mathbb{F}^n$, outputs all Reed-Solomon codewords within relative distance $1 - R - \varepsilon$ of y in time $O_\varepsilon(1) \text{poly}(n, \log |\mathbb{K}|)$.*

Additionally, we give a weaker construction achieving $O_\varepsilon(1)$ list-size instead of the near-optimal $O(1/\varepsilon)$ that is simpler and strongly explicit.

Theorem 1.3 (Strongly explicit list-decodable Reed-Solomon codes). *Fix $\varepsilon \in (0, 1)$. For all sufficiently large n , and an arbitrary field $\mathbb{K}$ of size at least n , there is a strongly explicit construction of an extension field $\mathbb{F}$ over $\mathbb{K}$ with $[\mathbb{F} : \mathbb{K}] = O_\varepsilon(1)$ and evaluation points $A = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}^n$, such that for every $1 \leq k \leq n$ the Reed-Solomon code $\text{RS}_{\mathbb{F}}(A, k)$ of rate $R = k/n$ is*

$$(1 - R - \varepsilon, (1/\varepsilon)^{O(1/\varepsilon)})$$

list-decodable, and the list can be found deterministically in time $O_\varepsilon(1) \text{poly}(n, \log |\mathbb{F}|)$. Given an index $i \in [n]$, α_i can be computed in time $\text{poly}(\log |\mathbb{F}|)$.

In fact, the algorithm used for both constructions is the same. A simpler version of the same idea also lets us get a more efficient deterministic list-decoding algorithm for folded Reed-Solomon codes that was previously known. We present this algorithm and the corresponding result in [Theorem 6.1](#) and [Corollary 6.2](#), using it as a warm-up to our main list-decoding algorithm.

Comparison with [BCP⁺26]. Both [Theorem 1.2](#) and [BCP⁺26] give efficient list decoding of Reed-Solomon codes up to capacity, but the results apply under incomparable settings.

- **Underlying field.** The algorithm of [BCP⁺26] applies to prime fields, whereas our codes are constructed over extension fields. The base field $\mathbb{K}$ that is extended can be arbitrary, and in particular we can construct RS codes over fields of characteristic 2, perhaps the most traditional use case of RS codes in practice. In terms of field size, we need fields of size $n^{O_\varepsilon(1)}$, whereas the result of [BCP⁺26] applies to fields of size $O_\varepsilon(n)$.³
- **List size.** We achieve constant, in fact optimal, list size that approaches the generalized Singleton bound. In contrast, the list size in [BCP⁺26] is a polynomial in n with exponent growing doubly exponentially in $1/\varepsilon$.⁴
- **Evaluation points.** The result of [BCP⁺26] works for *arbitrary* evaluation points over a prime field. In contrast, our evaluation points are carefully chosen.

We now elaborate on the respective restrictions within which our result and that of [BCP⁺26] operate, and why they are unavoidable.

³This can be improved to any prime at least n , see [Jer26, HKS26].

⁴If the goal is just to decode past the Johnson radius, the list-size can be improved to $O(n)$ [Gab26, DKT26].

Necessity to choose field or evaluation points carefully. Our list decoding result (Theorem 1.2) above *cannot* be established for arbitrary evaluation points. Likewise, the restriction to prime (or large characteristic) fields is necessary for the result in [BCP⁺26] which applies to arbitrary evaluation points. Known limitations of list decodability of RS codes over fields of small characteristic show that the list-size can grow linearly in the block length at the Johnson radius [JH01]. In fact, just past the Johnson radius, the list-size can grow faster than any fixed polynomial, for small enough (but constant) rates [BSKR10]. See [GR06, CW07] for related limitations.

In an upcoming companion paper, we in fact show that the list-size can grow super-polynomially just beyond the Johnson radius [AGG26]. (This was also concurrently shown in [Zhu26, DKTZ26].) So without restrictions on the field (as in [BCP⁺26]) or the evaluation points (as in Theorem 1.2), the Johnson radius is the limit of list decoding Reed-Solomon codes with polynomial-sized lists.

Necessity of growing lists for the prime field result of [BCP⁺26]. In the same paper [AGG26], we also show that over prime fields, list decoding up to capacity is in general *not* possible with constant-size lists: in particular for any $\varepsilon > 0$, there are certain full-length Reed-Solomon codes of constant rate $R = R(\varepsilon)$ over prime fields, for which some Hamming ball of relative radius bounded away from capacity ($1 - R - \Omega(\varepsilon^2)$) contains $n^{1/\varepsilon}$ codewords.⁵ Consequently, a result like [BCP⁺26], which is oblivious to the evaluation points, cannot establish a list-size bound independent of the block length.

Implication for proximity gaps/correlated agreement. Finally, we state our result for proximity gaps of Reed-Solomon codes. Towards this, we simply leverage the subspace-design property proved for our Reed-Solomon codes together with the line decodability (and more generally, low-degree curve decodability) of subspace-design codes established in [GG26], and its downstream implication of proximity gaps and mutual correlated agreement up to capacity.

Informally, (ℓ, δ, a, b) -curve decoding provides the guarantee that amongst any a codewords that are δ -close to points on a degree ℓ curve, there are b that are themselves on a degree ℓ curve. As is, this approach will give a result for lines parameterized by the field $\mathbb{K}$ and not the extension field $\mathbb{F}$. We present these weaker consequences from [GG26] in Section 3.

However, we also show that one can lift the guarantee to one for lines parameterized by $\mathbb{F}$ with a modest loss in parameters. Here we state one variant for line decoding; see Section 8 for more details and precise implications for proximity gaps and (mutual) correlated agreement (the formal definitions of these notions appear in Subsection 2.1). As far as we are aware, these are the first explicit constructions of codes with proximity gaps with soundness $\frac{\text{poly}(1/\varepsilon)}{\gamma|\Sigma|}$ at distance $\delta = 1 - R - \varepsilon$ and proximity loss γ . In particular, previous subspace-design constructions are not linear over the alphabet, and thus have soundness worse than $\frac{\text{poly}(1/\varepsilon)}{|\mathbb{K}|}$ but the alphabet is $\mathbb{K}^s$ for a large constant s , typically at least $1/\varepsilon^2$.

Theorem 1.4 (Explicit RS codes line-decodable up to capacity). *Fix $0 < R < 1$ and $0 < \varepsilon < 1$. For all sufficiently large n and every finite field $\mathbb{K}$ with $|\mathbb{K}| \geq n$, there is an explicit construction of an extension field $\mathbb{F}$ with $[\mathbb{F} : \mathbb{K}] = O_\varepsilon(1)$ along with explicit evaluation points $\alpha_1, \dots, \alpha_n \in \mathbb{F}$ in time $|\mathbb{K}|^{O_\varepsilon(1)}$ such that for all $t \geq 1$, $\text{RS}((\alpha_1, \dots, \alpha_n), Rn)$ is*

$$\left(1 - R - \varepsilon, O\left(\frac{t}{\varepsilon} + \varepsilon^{-3}\right), t\right)\text{-}\mathbb{F}\text{-line-decodable.}$$

⁵This also refutes a conjecture in the S-two white paper [CGH⁺26] which said that every prime-field RS code can be list decoding up to capacity with list-size independent of the block length.

Corollary 1.5 (Improved proximity gaps and correlated agreement). *Let $\mathbb{F}$, $\alpha_1, \dots, \alpha_n$ and ε be as in Theorem 1.4. Then, over $\mathbb{F}$, the code $\text{RS}((\alpha_1, \dots, \alpha_n), Rn)$ has:*

(i) *For every real $m > 1$,*

$$\left(1, 1 - R - \varepsilon, O\left(\frac{m/\varepsilon + \varepsilon^{-3}}{|\mathbb{F}|}\right), \frac{1}{m}\right)$$

correlated agreement and proximity gap,

(ii) *and*

$$\left(1, 1 - R - \varepsilon, O\left(\frac{n/\varepsilon + \varepsilon^{-3}}{|\mathbb{F}|}\right)\right)$$

mutual correlated agreement.

1.2 Technical overview

Reed-Solomon codes over extension fields that are subspace designs. Interestingly, our starting point is the subfield-evaluation construction from the work of Guruswami and Xing [GX13, GX22] which also introduced subspace-designs.

In their proof that folded Reed-Solomon codes yield subspace-designs, the central observation that Guruswami and Kopparty [GK16] used is that when polynomials $f_1(X), \dots, f_r(X)$ are linearly independent, the Wronskian matrix given by $W(f_1, \dots, f_r)$ given by $\{f_i(X\gamma^{j-1})\}_{i,j \in [r]}$ is full rank where γ is any fixed element of sufficiently high order. They then used failure of the subspace-design property for folded Reed-Solomon codes to show that $\det W(f_1, \dots, f_r)$ must have too many roots for a non-zero polynomial.

The natural analog of the above Wronskian for extension field Reed-Solomon codes is the Moore/Galois determinant. In particular, if $\mathbb{K} \subseteq \mathbb{F}$ is a subfield and $f_1, \dots, f_r \in \mathbb{F}[X]$ are linearly independent over $\mathbb{K}(X)$, then the matrix $G(f_1, \dots, f_r) = \{\sigma(f_i(X))\}_{i \in [r], \sigma \in \text{Gal}(\mathbb{F}/\mathbb{K})}$ is full rank. Here, $\text{Gal}(\mathbb{F}/\mathbb{K})$ is the group of all automorphisms on $\mathbb{F}$ fixing $\mathbb{K}$, and we let σ act on $\mathbb{F}[X]$ coefficient-wise fixing the variable X .

The first issue with direct root counting methods is that if we select a full rank submatrix M of G corresponding to automorphisms $\omega_1, \dots, \omega_r$, then it is possible that $\omega_i(\alpha) \neq \alpha$ for an evaluation point α . Thus α may not correspond to a root of $\det M$ at all.

In [GX13], the evaluation points were all chosen to belong to the subfield $\mathbb{K}$ itself. This runs into the secondary issue that the independence assumption for the Galois matrix to be full rank is over $\mathbb{K}[X]$. Thus, the space could be rank 1 while being all of $\mathbb{K}[X]$. In particular, each coordinate α_i is now not a root directly of $\det M$ but of the determinant of MT_i where T_i is a low-rank matrix with coefficients in $\mathbb{K}[X]$, which introduces the possibility that α_i is a root of $\det T_i$. Thus, we could face the issue that each α_i is simply a root of the determinant of the T_i chosen for that coordinate.

In order to circumvent the second issue, it would be convenient to take all α_i to be of sufficiently high extension degree so that T_i cannot introduce new zeroes. This would work, but now we are back to the possibility that $\sigma(\alpha) \neq \alpha$.

We get around both issues by allowing α_i to be of high extension degree but picking them so that there is always a choice of $\mathcal{B} = \{\omega_1, \dots, \omega_r\} \subseteq \text{Gal}(\mathbb{F}/\mathbb{K})$ such that M given by picking rows corresponding to $\mathcal{B}$ is full rank, and moreover $|\{i \mid \omega(\alpha_i) \neq \alpha_i\}| < \varepsilon n$ for any $\omega \in \mathcal{B}$. A combination of these two ideas immediately results in a natural construction with $|\mathbb{F}| = n^{O_{r,\varepsilon}(1)}$. This is, however, unsatisfactory since the resulting field size is then exponential.

Thus, we observe that the determinants $\det T_i$ can be further restricted to a small class of polynomials given by the span of minors of a low-dimensional space, if we further assume that we are interested only in low-dimensional $\mathbb{K}$ -spaces and not low-dimensional $\mathbb{K}(X)$ spaces. For a formalization of this idea, see [Definition 4.1](#) for *effective evaluation points* and the corresponding proof of the subspace-design property. We note that the final proof, once we capture both of these ideas in our definition of effective evaluation points, is rather clean and short. We then present a simple construction of such points in [Section 7](#).

We additionally give a simpler construction of evaluation points, that does not have the full property of controlling how many roots they contribute to $\det T_i$, but with the points still of sufficiently high degree, and show that this suffices for list-decoding with size $O_\varepsilon(1)$. The advantage of this construction is that it is strongly-explicit, i.e., every evaluation point can be output in time $\text{poly}(\log |\mathbb{F}|)$.

The broad construction idea works as follows. We pick pairwise non-conjugate $\alpha_1, \dots, \alpha_n$ as coming from extension fields of $\mathbb{K}$ in $g = O((\log r)/\varepsilon)$ different directions, i.e., we pick co-prime integers $d_1, \dots, d_g$ and pick roughly n/g evaluation points from each of $\mathbb{K}^{d_1}, \dots, \mathbb{K}^{d_g}$. This lets us naturally pick the automorphisms as $\sigma_j : x \mapsto x^{|\mathbb{K}|^{\left(\prod_{i=1}^g d_i\right)/d_j}}$. They generate the Galois group since the d_i are co-prime, and there is a choice of $\mathcal{B}$ as before such that each $\omega \in \mathcal{B}$ is supported on at most $\log r$ of these σ_i , giving us the required guarantee that $|\{i \mid \omega(\alpha_i) \neq \alpha_i\}| \leq n(\log r)/g$ (see [Lemma 2.20](#)).

For the strongly explicit construction, it suffices to pick d_i to be distinct sufficiently large primes co-prime to $|\mathbb{K}|$. For the full subspace-design construction, the construction begins with the same idea, but we need to ensure that the determinants $\det T_i$ belong to a small space, and they do not automatically vanish for our choice of α_i . This requires a brute force search in each coordinate for a specific evaluation point that is not a root of any bounded-degree sparse polynomial, which takes time $|\mathbb{K}|^{O_{r,\varepsilon}(1)}$.

List-decoding Algorithm, List decoding algorithms for subspace-design codes have proceeded along the following general blueprint:

Given a $\mathbb{K}$ -subspace design code $C \subseteq (\mathbb{K}^s)^n$ and a received word $y \in (\mathbb{K}^s)^n$, do the following two steps:

- Find a low-dimensional subspace $A \subseteq C$ such that all close-by codewords to y in C also lie in A .
- Find all close-by codewords in the low-dimensional space.

The first step is usually a code-specific procedure, and we do not know how to implement it generically for all subspace-design codes. In particular, [\[GW13\]](#) showed that the first step can indeed be implemented for the folded Reed-Solomon and univariate Multiplicity codes. This argument can be viewed as an algorithmization of the Wronskian proof used by [\[GK16\]](#) to show that in fact folded Reed-Solomon codes and Univariate Multiplicity codes form subspace-designs.

The second step is known to work generically for all linear codes. In particular, [\[KRSW23, Tam24\]](#) showed that given any d -dimensional affine space $A \subset C$ with distance δ , and $y \in \Sigma^n$, we have that $|\{a \in A \mid \Delta(a, y) < (\delta - \varepsilon)\}| \leq \varepsilon^{-d}$. They also provided a randomized *pruning* procedure to implement this step.

In a recent work, [\[AHS26\]](#) showed how to improve the randomized algorithm if C in fact forms a near optimal-subspace design code, resulting in lists of size $O(\frac{d}{\varepsilon} + 1)$. They also showed that for general codes, they can implement [\[KRSW23, Tam24\]](#)'s procedure deterministically but in time

$(1/\varepsilon)^{\exp(d)} \cdot \text{poly}(n)$.

For us, we follow a variant of this blueprint. In particular, our codes are linear over an extension field $\mathbb{F}$ and thus $\mathbb{K}^s = \mathbb{F}$. Even though we know the combinatorial fact that there is a low-dimensional $\mathbb{K}$ space containing all close by codewords, we do not know a direct algorithmic way of finding such a space. This is primarily since our proof of subspace-design first proved a variant of the subspace-design property over spaces of bounded $\mathbb{K}(X)$ -dimension, and then applied the *transfer* result [Theorem 4.4](#). There is no natural algorithmization of the latter, thus we focus on the first step and in fact do recover a bounded $\mathbb{K}(X)$ dimension-space.

Now, the goal has shifted from pruning a bounded $\mathbb{K}$ -dimension space to pruning a bounded $\mathbb{K}(X)$ dimension space. This cannot be done generically for all choices of evaluation points, as all of $\mathbb{K}[X]$ has dimension 1. Even with stronger assumptions, we do not see a natural way of adapting [\[KRSW23, Tam24, AHS26\]](#) to this setting. We therefore present a variant.

We first revisit the procedure in the aforementioned papers. The idea is to prune by randomly sampling a coordinate $i \in [n]$ and restricting A to a subspace A_i , and repeating until the resulting space has a unique element. This process is then shown to find any given codeword with constant probability. We view this procedure broadly as *repeatedly imposing linear constraints on A until exactly one codeword remains*. This process can only find a single codeword at a time, so we can perform a branching process where we sample a few coordinates at each step.

We implement a version of this algorithm via the following process:

Repeatedly impose *low-degree* constraints on A until exactly one codeword remains.

As a warmup we implement this procedure to improve the deterministic list-decoding of low-dimensional subspaces of Reed-Solomon codes. Specifically, this yields a more efficient (in the sense of improved dependence on ε , as [\[GHKS24, AHS26\]](#) both give near-linear time decoders in terms of n) deterministic list-decoder of folded Reed-Solomon codes than was known via [\[AHS26\]](#), while almost matching the best list-size known for this task due to [\[KRSW23, Tam24\]](#).

Moreover, we show that the above can be adapted to bounded $\mathbb{K}(X)$ -dimensional spaces as long as all evaluation points α_i are pairwise non-conjugate and have comparable and sufficiently large (but still bounded by $O_{d,\varepsilon}(1)$) extension-degrees.

Line decoding and proximity gaps. As discussed, our codes are linear over an extension field $\mathbb{F}$ but the subspace-design property is only when viewed as an additive code over a subfield $\mathbb{K}$. The results of [\[GG26\]](#) thus imply line-decoding and proximity-gaps results, but only over $\mathbb{K}$ -affine lines, and a naive lift to $\mathbb{F}$ by partitioning into $\mathbb{K}$ lines is rather lossy. We nevertheless present this black-box argument in [Section 3](#).

In [Section 8](#), we show that the methods of [\[GG26\]](#), which were inspired by the pruning step we described for list-decoding as in [\[AHS26\]](#), are nevertheless useful. In particular, the sampling-based pruning method, that proceeds via sampling one coordinate at a time and pruning the space to agree with the received word on that coordinate till we are left with a zero-dimensional space, is also meaningful when we stop the pruning at an earlier point. In fact, it can achieve a better success probability than what can be proven when we keep pruning all the way down to a 0-dimensional space.

In particular, for near-optimal subspace-design codes, the works [\[AHS26, GG26\]](#) showed that starting from an d -dimensional linear space A , there is an algorithm that samples a set $S \subseteq [n]$ such that $\{a \in A \mid a_S = 0\} = \{0\}$, and for any $T \subseteq [n]$ with $|T| \geq (R + \varepsilon)n$ has the property that $S \subseteq T$ with probability at least $\frac{\varepsilon}{d+\varepsilon}$. [\[GG26\]](#) then uses various sets T as the agreement positions of

different codewords, i.e., $T_\alpha = \{i \mid c_{\alpha,i} = u_i + \alpha v_i\}$, to conclude that many must have agreed on S . This implies that for many α , there is a common set $S \subseteq T_\alpha$ and $\{a \in A \mid a_S = 0\} = \{0\}$. This implies that all such c_α must themselves have been collinear.

We use the same pruning method but show the following slightly more general conclusion: for any D -dimensional space A and integer d , there is an algorithm that samples a set $S \subseteq [n]$ such that $\dim\{a \in A \mid a_S = 0\} < d$ and for any T as above, $\Pr[S \subseteq T] \geq \frac{\varepsilon}{D/d + \varepsilon}$.

Now we need to choose what space A to apply this algorithm on, and what choice of d to use. To mimic [GG26] where $D = O(1/\varepsilon)$ and they stop at $d = 1$, we use $d = \Omega(\varepsilon D)$.

We show that if C is an $\mathbb{F}$ -linear code which is a near-optimal subspace-design over a subfield $\mathbb{K}$, then given any $c_{\alpha_1}, \dots, c_{\alpha_a}$ such that $\Delta(c_{\alpha_i}, u + \alpha_i v) < 1 - R - \varepsilon$ for all $i \in [a]$, we can find a $\mathbb{K}$ -linear space $V \subseteq \mathbb{F}$ of $O_{\varepsilon,a}(1)$ dimension such that if

$$V_1 = \text{span}_{\mathbb{K}}\{\alpha_i V\}_{i \in [a]} \quad V_2 = \text{span}_{\mathbb{K}}\{\alpha_i \alpha_j V\}_{i,j \in [a]}.$$

Then we have $\dim V_2 \leq 2 \dim V$ (this is for our special choice of V).

Using a dimension counting argument, this lets us upper bound the dimension of $W = \text{span}_{\mathbb{K}}\{c_{\alpha_i} V_1\}_{\alpha_i}$ by $O((\dim V)/\varepsilon)$. This W plays the role of the space A described above.

Now, we use $D = \dim W$ and $d = \dim V$ and sample the set S as before. We then observe that if $\alpha_1, \alpha_2, \alpha_3$ are such that $S \subseteq T_{\alpha_1}, T_{\alpha_2}, T_{\alpha_3}$, then for $z = (\alpha_3 - \alpha_2)c_{\alpha_1} + (\alpha_1 - \alpha_3)c_{\alpha_2} + (\alpha_2 - \alpha_1)c_{\alpha_3}$, we must have $z|_S = 0$. If $z \neq 0$, then $Vz \subseteq W$ but also $Vz|_S = 0$. In particular, $d > \dim Vz = \dim V = d$ which is a contradiction. By the sampling bound, some S is contained in T_α for $\Omega(\varepsilon^2 a)$ of the assigned codewords, which are then guaranteed to lie on a common codeword line.

1.3 Organization

We begin with the necessary definitions in Section 2. Section 3 proves Theorem 1.1 and gives its combinatorial consequences. The subspace-design property (Theorem 4.2) is established in Section 4. We then turn to the list-decoding algorithm which is presented in Section 5, with the crucial pruning step established in Section 6. Section 7 constructs evaluation points satisfying both sets of hypotheses. Finally, the improved line-decoding and proximity-gap results are established in Section 8. The algorithmic list-decoding result uses a root-finding step that follows from standard reductions combined with the recent [Ope26] deterministic polynomial factorization algorithm, and this reduction is presented in Appendix A.

1.4 AI Disclosure

Our key results, and the paper writing were substantially aided by the use of ChatGPT 5.6 Sol and ChatGPT 6 Astra.

The question of constructing explicit list-decodable Reed-Solomon near the generalized Singleton bound via extension field subspace-designs has been a central goal for the authors for the past year. Consequentially, we queried ChatGPT 5.6 Sol to construct explicit near generalized Singleton-bound codes, and it found one of the central ideas in the paper of replacing Guruswami-Xing's subfield evaluation and Frobenius automorphism method with a small family of automorphisms generating the Galois group and fixing all but ε -fraction of evaluation points. The initial construction by GPT only worked for list-decoding and had $O(nr)$ extension degree for each evaluation points leading to a construction growing as $\exp(n^{O(\log(1/\varepsilon)/\varepsilon)})$.

GPT 5.6 Sol over many rounds of prompting was able to extend this to an algorithm for list-decoding as well. The authors on understanding the ideas knew that it should extend to subspace-designs due to the known connection with Galois/Moore determinants and this was then achieved over many rounds of discussion and prompting with GPT 5.6.

The authors then worked extensively with ChatGPT 6 Astra and simplified, reworked, and developed the techniques to further improve the results, and lead to clearer exposition. Definitions such as effectiveness and generalizations such as to deterministic subspace-decoders for standard Reed-Solomon codes and faster deterministic decoders of folded Reed-Solomon codes were then discovered by the human authors once the algorithmic aspects of GPT’s construction were cleared up, and this was then further used to simplify our main algorithm and results. Similar processes happened for the constructions of our choice of evaluation points.

The improvements for line-decoding and proximity-gaps in section 8 were found completely by GPT 6 Astra on being asked to extend [GG26]’s results when only subspace-design guarantees over a subfield are provided in a way independent of the extension-field’s extension order.

AI tools also assisted with the exposition, under detailed guidance of the authors. The authors assume responsibility for the correctness of the proofs and the quality of exposition.

2 Preliminaries

We write $[n] = \{1, \dots, n\}$ and $\mathbb{N} = \{1, 2, \dots\}$. All logarithms are to base two unless a base is displayed. For a finite field $\mathbb{K}$, a $\mathbb{K}$ -*additive code* is a $\mathbb{K}$ -linear subspace $C \subseteq \Sigma^n$ with alphabet $\Sigma = \mathbb{K}^s$ for some $s \geq 1$. In this paper, we typically use $\Sigma = \mathbb{F}$, where $\mathbb{F}$ is a finite extension of $\mathbb{K}$. The rate of C is $\log_{|\Sigma|} |C|/n$, and the relative Hamming distance between $x, y \in \Sigma^n$ is $\Delta(x, y) = \frac{|\{i \in [n] \mid x_i \neq y_i\}|}{n}$.

For distinct $\alpha_1, \dots, \alpha_n \in \mathbb{F}$ and $1 \leq k \leq n$, the Reed–Solomon code is

$$\text{RS}_{\mathbb{F}}((\alpha_1, \dots, \alpha_n), k) = \{(f(\alpha_1), \dots, f(\alpha_n)) : f \in \mathbb{F}[X]_{<k}\}.$$

It has dimension k over $\mathbb{F}$, rate k/n , and distance $1 - (k - 1)/n$.

2.1 Subspace-design codes and decoding consequences

Guruswami and Xing [GX13] introduced subspace designs to restrict the message spaces of certain Reed-Solomon and algebraic-geometric codes and obtain small lists in list decoding. Informally, a subspace design is a family of subspaces such that every low-dimensional space meets only a few members nontrivially. Their constructions were random; Guruswami and Kopparty [GK16] gave explicit constructions from folded Reed–Solomon codes. Constant-alphabet subspace-design codes were constructed in [GGH26] using the Alon–Edmonds–Luby transformation [AEL95].

We follow the terminology for subspace-design codes in [GG26]. A similar definition first appears in [CZ25] under the name *subspace-designable codes*.

Definition 2.1 (Subspace-design codes). For a nondecreasing function $\tau : \mathbb{Z}_{\geq 1} \rightarrow [0, 1]$, a $\mathbb{K}$ -additive code $\mathcal{C} \subseteq \Sigma^n$ is a τ -*subspace-design code* over $\mathbb{K}$ if for every integer $r \geq 1$ and every $\mathbb{K}$ -linear subspace $\mathcal{A} \subseteq \mathcal{C}$ of dimension at most r ,

$$\frac{\sum_{i=1}^n \dim_{\mathbb{K}} \mathcal{A}_i}{n} \leq \dim_{\mathbb{K}}(\mathcal{A}) \cdot \tau(r)$$

where $\mathcal{A}_i = \{a \in \mathcal{A} \mid a_i = 0\}$.

For Reed–Solomon codes, evaluation identifies $\mathcal{A}$ with a $\mathbb{K}$ -subspace $U \subseteq \mathbb{F}[X]_{<k}$ and $\mathcal{A}_i$ with $U_i = \{f \in U : f(\alpha_i) = 0\}$.

Definition 2.2 (List decoding). Let $\delta < 1$ and let $L \geq 1$ be an integer. A code $C \subseteq \Sigma^n$ is (δ, L) -list-decodable if, for every $y \in \Sigma^n$,

$$|\{c \in C : \Delta(c, y) \leq \delta\}| \leq L.$$

Definition 2.3 (Average-radius list decoding). Let $\delta < 1$ and let $L \geq 1$ be an integer. A code $C \subseteq \Sigma^n$ is (δ, L) -average-radius list-decodable if, for every $y \in \Sigma^n$ and every set of $L + 1$ distinct codewords $c_1, \dots, c_{L+1}$,

$$\frac{\sum_{i=1}^{L+1} \Delta(c_i, y)}{L + 1} > \delta.$$

The following list-decoding statement is due to [CZ25] and the list-recovery statement is due to [BCDZ26a]. We use the subspace-design formulation of their results as used by [GGH26], and presented for list-recovery in [GG25].

Theorem 2.4 (Average-radius list decoding [CZ25]). *Every τ -subspace-design code over $\mathbb{K}$ is (δ, L) -average-radius list-decodable, and hence list-decodable, for every integer $L \geq 1$ and $\delta < \frac{L}{L+1}(1 - \tau(L))$.*

Corollary 2.5 (Generalized Singleton bound for all list sizes). *Let $0 < \varepsilon < 1$ and $r = \lfloor 1/\varepsilon \rfloor$. Every τ -subspace-design code over $\mathbb{K}$ is $\left(\frac{L}{L+1}(1 - \tau(r) - \varepsilon), L\right)$ -average-radius list-decodable, and hence list-decodable, for every integer $L \geq 1$.*

Proof. For $L \leq r$, the claim follows from Theorem 2.4 and the monotonicity of τ . For $L > r$, fix a received word y and order any $L + 1$ distinct codewords by increasing distance to y . Their average distance is at least the average of the first $r + 1$, which, by Theorem 2.4, is at least

$$\frac{r}{r+1}(1 - \tau(r)) \geq 1 - \tau(r) - \frac{1}{r+1} > 1 - \tau(r) - \varepsilon.$$

The claim follows. $\square$

Definition 2.6 (List recovery). A code $C \subseteq \Sigma^n$ is (δ, ℓ, L) -list-recoverable if, for every $S_1, \dots, S_n \subseteq \Sigma$ with $|S_i| \leq \ell$,

$$|\{c \in C : |\{i \in [n] : c_i \notin S_i\}| \leq \delta n\}| \leq L.$$

Theorem 2.7 (List recovery [BCDZ26a]). *For every integer $\ell \geq 1$ and $\eta > 0$, set $r = \lceil \ell/\eta \rceil$. Every τ -subspace-design code over $\mathbb{K}$ is*

$$\left(\delta, \ell, \left(\frac{\ell}{\tau(r) + \eta}\right)^{(\tau(r) + \eta)/\eta}\right)$$

list-recoverable for every $\delta < 1 - \tau(r) - \eta$.

Curve-decoding was first defined in [GG26] as a new notion capturing proximity gaps, correlated agreement, and mutual correlated agreement. The line-decoding case was independently defined by Haböck and is also used in [CGH⁺26].

Definition 2.8 (Curve decodability). A $\mathbb{K}$ -additive code $C \subseteq \Sigma^n$ is (ℓ, δ, a, b) - $\mathbb{K}$ -curve-decodable if, for every $u_0, u_1, \dots, u_\ell \in \Sigma^n$ and every function $f : \mathbb{K} \rightarrow C$, whenever the set

$$A = \left\{ \alpha \in \mathbb{K} : \Delta \left(\sum_{j=0}^{\ell} u_j \alpha^j, f(\alpha) \right) \leq \delta \right\}$$

has at least a elements, there exist $c_0, c_1, \dots, c_\ell \in C$ such that

$$\left| \left\{ \alpha \in A : f(\alpha) = \sum_{j=0}^{\ell} c_j \alpha^j \right\} \right| \geq b.$$

When $\ell = 1$, we call this *line decodability*.

Theorem 2.9 (Curve decoding from subspace designs [GG26]). Let $C \subseteq \Sigma^n$ be a $\mathbb{K}$ -additive τ -subspace-design code over $\mathbb{K}$. For positive integers r, ℓ, a and $\eta \geq (\ell + 1)/r$, the code C is

$$\left(\ell, 1 - \tau(r) - \eta, a, \frac{\eta a}{r + \eta} \right)\text{-}\mathbb{K}\text{-curve-decodable}.$$

Proximity gaps were studied in [RVW13, BKS18, BSCI⁺23], with correlated agreement formalized in [BSCI⁺23]. We use the curve formulations of [GG26].

Definition 2.10 (Proximity gaps (PG)). A $\mathbb{K}$ -additive code $C \subseteq \Sigma^n$ has an $(\ell, \delta, \varepsilon, \gamma)$ proximity gap if for all $u_0, u_1, \dots, u_\ell \in \Sigma^n$, and for all $\delta' \leq \delta$,

$$\mathbb{P}_{\alpha \in \mathbb{K}} \left(\Delta \left(\sum_{j=0}^{\ell} u_j \alpha^j, C \right) \leq \delta' \right) > \varepsilon \implies \forall \alpha \in \mathbb{K}, \Delta \left(\sum_{j=0}^{\ell} u_j \alpha^j, C \right) \leq \frac{\delta'}{1 - \gamma}.$$

When $\gamma = 0$, we say that C has an $(\ell, \delta, \varepsilon)$ proximity gap.

Definition 2.11 (Correlated agreement (CA)). A $\mathbb{K}$ -additive code $C \subseteq \Sigma^n$ has $(\ell, \delta, \varepsilon, \gamma)$ correlated agreement if for all $u_0, u_1, \dots, u_\ell \in \Sigma^n$ and $\delta' \leq \delta$

$$\mathbb{P}_{\alpha \in \mathbb{K}} \left(\Delta \left(\sum_{j=0}^{\ell} u_j \alpha^j, C \right) \leq \delta' \right) > \varepsilon \implies \exists c_0, c_1, \dots, c_\ell \in C : \left| \left\{ i : \bigvee_{j=0}^{\ell} c_{j,i} \neq u_{j,i} \right\} \right| \leq \frac{\delta' n}{1 - \gamma}.$$

The parameter γ is often called the *proximity loss* [BSCH⁺26].

Definition 2.12 (Mutual correlated agreement (MCA) [ACFY26]). A $\mathbb{K}$ -additive code $C \subseteq \Sigma^n$ has $(\ell, \delta, \varepsilon)$ mutual correlated agreement if, for every $u_0, \dots, u_\ell \in \Sigma^n$ and $0 \leq \delta' \leq \delta$,

$$\begin{aligned} & \Pr_{\alpha \in \mathbb{K}} \left[\exists S \subseteq [n], c \in C, j \in \{0, \dots, \ell\} : \right. \\ & \quad \left| S \right| \geq (1 - \delta')n, \quad c|_S = \left(\sum_{h=0}^{\ell} \alpha^h u_h \right) \Big|_S, \\ & \quad \left. c'|_S \neq u_j|_S \text{ for every } c' \in C \right] \leq \varepsilon. \end{aligned}$$

Correlated agreement implies proximity gaps with the same parameters; mutual correlated agreement implies both with $\gamma = 0$.

Theorem 2.13 (Consequences of curve decodability [GG26, Theorem 3.6 and Corollary 3.9]). *Let ℓ, L, a, b be positive integers with $b > \ell$, and assume $|\mathbb{K}| > (L+1)^2\ell$. If a $\mathbb{K}$ -additive code $C \subseteq \Sigma^n$ is*

$$\left(\delta\left(1 + \frac{\ell}{b-\ell}\right), L\right) \text{ list decodable} \quad \text{and} \quad (\ell, \delta, a, b)\text{-}\mathbb{K}\text{-curve-decodable}$$

then C has:

1. *for any integer $t \geq 1$, $(\ell, \delta, (t-1)L + a, t)$ - $\mathbb{K}$ -curve decodability,*
2. *for any $m > 1$, $(\ell, \delta, \frac{\ell m L + a}{|\mathbb{K}|}, \frac{1}{m})$ correlated agreement and proximity gap,*
3. *$(\ell, \delta, \frac{\ell n L + a}{|\mathbb{K}|})$ mutual correlated agreement.*

For [Item 2](#), take $t = \lceil \ell m \rceil$ in the curve-decoding amplification: $(t-1)L + a \leq \ell m L + a$ and $\ell/t \leq 1/m$.

2.2 Polynomial modules and degree

Theorem 2.14 (Cauchy–Binet). *Let $\mathbb{E}$ be a field and $1 \leq t \leq e$. For $B \in \mathbb{E}^{t \times e}$ and $D \in \mathbb{E}^{e \times t}$,*

$$\det(BD) = \sum_{\substack{S \subseteq [e] \\ |S|=t}} \det B_{[t], S} \det D_{S, [t]}.$$

The submatrices use the indicated rows and columns in increasing order.

For a polynomial vector $a = (a^{(1)}, \dots, a^{(m)})$, write $\deg a = \max_{1 \leq j \leq m} \deg a^{(j)}$. We use the conventions $\deg 0 = -\infty$ and $\max \emptyset = -\infty$. For a field E and an integer r , write $E[X]_{<r} = \{f \in E[X] : \deg f < r\}$; in particular, $E[X]_{<r} = \{0\}$ when $r \leq 0$.

A $\mathbb{K}[X]$ -module $M \subseteq \mathbb{K}[X]^m$ is an additive subgroup closed under multiplication by polynomials in $\mathbb{K}[X]$. Its *rank* is the dimension of $\text{span}_{\mathbb{K}(X)} M$ over $\mathbb{K}(X)$.

Theorem 2.15 (Reduced bases [Ale05]). *Let $\mathbb{K}$ be a field and let $M \subseteq \mathbb{K}[X]^m$ be a $\mathbb{K}[X]$ -module of rank d . Then M has a basis $a_1(X), \dots, a_d(X)$ over $\mathbb{K}[X]$ such that*

$$M = \left\{ \sum_{i=1}^d c_i(X) a_i(X) \mid c_i(X) \in \mathbb{K}[X] \right\}$$

and, for every $c_1(X), \dots, c_d(X) \in \mathbb{K}[X]$,

$$\deg \left(\sum_{i=1}^d c_i(X) a_i(X) \right) = \max_{1 \leq i \leq d} (\deg a_i + \deg c_i). \quad (2.1)$$

We call such a basis reduced. Given s polynomial generators of M of degree at most D , a reduced basis can be computed deterministically using $\text{poly}(m, s, D+1)$ arithmetic operations over $\mathbb{K}$.

Corollary 2.16. *Let $\mathbb{K} \subseteq \mathbb{F}$ be finite fields, let $\mathbb{L} = \mathbb{K}(X)$, and let $A \subseteq \mathbb{F}(X)$ be an $\mathbb{L}$ -linear space of dimension d . Set*

$$\mathcal{M} = A \cap \mathbb{F}[X], \quad \mathcal{M}_k = A \cap \mathbb{F}[X]_{<k} \quad (k \in \mathbb{Z}_{\geq 0}).$$

Then $\mathcal{M}$ has rank d over $\mathbb{K}[X]$ and has a reduced $\mathbb{K}[X]$ -basis $a_1(X), \dots, a_d(X)$ with respect to ordinary polynomial degree. In particular, for every integer $k \geq 0$,

$$\mathcal{M}_k = \left\{ \sum_{i=1}^d a_i(X) c_i(X) \mid c_i(X) \in \mathbb{K}[X]_{<k-\deg a_i} \right\}.$$

2.3 Construction of Field Extensions and Deterministic Factorization

Theorem 2.17 (OpenAI [Ope26, Theorem 1.1]). *Given a prime p and a nonzero polynomial $f \in \mathbb{F}_p[X]$ of degree d , there is a deterministic algorithm that computes the leading coefficient of f , its distinct monic irreducible factors, and their multiplicities in time $\text{poly}(d, \log p)$.*

Berlekamp [Ber70] gives a deterministic polynomial-time reduction from factorization over an arbitrary finite field to root finding over its prime subfield. Thus Theorem 2.17 also yields deterministic factorization over non-prime finite fields.

Corollary 2.18 (Factorization over finite fields). *Given a finite field $\mathbb{F}$ and a nonzero polynomial $f \in \mathbb{F}[X]$ of degree d , there is a deterministic algorithm that computes its polynomial factorization into irreducibles in time $\text{poly}(d, \log |\mathbb{F}|)$.*

Shoup [Sho90] gave a deterministic polynomial-time reduction from constructing irreducible polynomials over a finite field to factoring polynomials over its prime subfield. Combining this reduction with Theorem 2.17 gives the following immediate corollary.

Corollary 2.19 (Construction of field extensions). *Given a finite field $\mathbb{K}$ and an integer $d \geq 1$, one can deterministically construct an irreducible polynomial $f \in \mathbb{K}[X]$ of degree d and the extension $\mathbb{K}[X]/(f)$ in time $\text{poly}(d, \log |\mathbb{K}|)$.*

We note that the above results were known to work in run time $\text{poly}(\text{char}(\mathbb{F}), \log |\mathbb{F}|, d)$ earlier and the characteristic requirement is now removed due to OpenAI's result.

2.4 Independent conjugates and coordinate movement

For a field extension $\mathbb{F}/\mathbb{K}$, we extend $\sigma \in \text{Gal}(\mathbb{F}/\mathbb{K})$ to $\mathbb{F}(X)$ by applying it to coefficients and fixing X .

Lemma 2.20 (A downward-closed set of independent conjugates). *Let $\mathbb{K} \subseteq \mathbb{F}$ be finite fields, let $\sigma_1, \dots, \sigma_g$ generate $G = \text{Gal}(\mathbb{F}/\mathbb{K})$, and let $u_1, \dots, u_d \in \mathbb{F}[X]$ be linearly independent over $\mathbb{K}(X)$, where $g, d \geq 1$. For $\omega \in \mathbb{Z}_{\geq 0}^g$ write $\sigma^\omega = \sigma_1^{\omega_1} \dots \sigma_g^{\omega_g}$. There is a set $\mathcal{B} \subseteq \mathbb{Z}_{\geq 0}^g$ of size d , downward closed under the coordinatewise order, with the following properties:*

(i) *The polynomial*

$$P(X) = \det(\sigma^\omega(u_a)(X))_{\omega \in \mathcal{B}, 1 \leq a \leq d}$$

is nonzero.

(ii) For any distinct points $\alpha_1, \dots, \alpha_n \in \mathbb{F}$, the following bound holds:

$$\sum_{i=1}^n |\{\omega \in \mathcal{B} : \sigma^\omega(\alpha_i) \neq \alpha_i\}| \leq d \log_2 d \max_{\sigma \in \{\sigma_1, \dots, \sigma_g\}} |\{\alpha_i : \sigma(\alpha_i) \neq \alpha_i\}|. \quad (2.2)$$

Such a set $\mathcal{B}$ can be computed deterministically in polynomial time.

Proof. Put $E = \mathbb{F}(X)$. The matrix $M = (\sigma(u_a))_{\sigma \in G, a \in [d]}$ has rank d over E . Indeed, otherwise choose a nonzero kernel vector c of minimal support and normalize $c_j = 1$. For every $\tau \in G$, reindexing rows gives $\tau(c) \in \ker M$. Minimality forces $c - \tau(c) = 0$, since its j -th coordinate is zero. Thus $c \in \mathbb{K}(X)^d$, and the identity row contradicts the independence of the u_a .

Write $r_\omega = (\sigma^\omega(u_1), \dots, \sigma^\omega(u_d))$. Process the indices in graded lexicographic order, retaining r_ω whenever it is independent over E of the previously retained rows. Let $\mathcal{B}$ be the retained indices. The family $(r_\omega)_\omega$ contains every row of M , so $|\mathcal{B}| = d$ and the retained determinant is nonzero. If $\omega \notin \mathcal{B}$, then

$$r_\omega = \sum_{\nu \prec \omega} c_\nu r_\nu \implies r_{\omega+e_j} = \sum_{\nu \prec \omega} \sigma_j(c_\nu) r_{\nu+e_j}.$$

The generators commute, and the order is preserved by addition. Hence $\omega + e_j \notin \mathcal{B}$, proving downward closure.

To compute $\mathcal{B}$, start with the index 0 and process indices in this order, adding unseen successors $\omega + e_1, \dots, \omega + e_g$ only when ω is already in $\mathcal{B}$. Independence is tested when each index is processed. Downward closure ensures that every element of $\mathcal{B}$ is reached. Since at most $\text{poly}(g, d)$ indices are checked in the greedy process, the algorithm runs in polynomial time.

Finally, let $\omega \in \mathcal{B}$ have support S . Observe that

$$|\{i \in [n] : \sigma^\omega(\alpha_i) \neq \alpha_i\}| \leq |S| \max_{j \in [g]} |\{i \in [n] : \sigma_j(\alpha_i) \neq \alpha_i\}|.$$

By downward closedness, $d = |\mathcal{B}| \geq 2^{|S|}$. Thus $|S| \leq \log d$, and summing over $\mathcal{B}$ proves the result. $\square$

3 Main theorem and combinatorial consequences

We prove [Theorem 1.1](#) by combining the subspace-design bound proved in [Theorem 4.2](#), the list-decoding algorithm as in [Theorem 5.1](#), and the construction of evaluation points from [Theorem 7.2](#). We use the list-decoding algorithm to find an $O_\varepsilon(1)$ sized-list containing all nearby codewords, and the subspace-design property gives the requisite near-optimal list-sizes, and we just check all elements in the list outputted by our algorithm.

Theorem 1.1 (Explicit RS codes that are near-optimal subspace-design codes). *For every integer $r \geq 2$, every $0 < \varepsilon < 1$, all sufficiently large n , and every finite field $\mathbb{K}$ with $|\mathbb{K}| \geq n$, one can deterministically construct in time $|\mathbb{K}|^{O_{r,\varepsilon}(1)}$ an extension $\mathbb{F}/\mathbb{K}$ of degree $O_{r,\varepsilon}(1)$ and evaluation points $A = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}^n$ with the following properties. For every $1 \leq k \leq n$, the Reed–Solomon code $C = \text{RS}_{\mathbb{F}}(A, k)$, of rate $R = k/n$, is a τ -subspace-design code over every subfield $\mathbb{K}_0 \subseteq \mathbb{K}$, where*

$$\tau(d) = \min\{1, R + \varepsilon/4\} \quad (1 \leq d \leq r), \quad \tau(d) = 1 \quad (d > r).$$

Moreover, for every integer $L \geq 1$, C is (δ_L, L) -average-radius list-decodable, where

$$\delta_L = \frac{L}{L+1}(1 - R - \varepsilon).$$

Given any received word, the lists within relative distances δ_L and $1 - R - \varepsilon$ can be found deterministically in time $O_{r,\varepsilon}(1) \text{poly}(n, \log |\mathbb{F}|)$, with sizes at most L and $O((1 - R)/\varepsilon)$ respectively.

Proof. We may assume $r \geq \lceil 4/\varepsilon \rceil$. Let $\mathbb{K}'/\mathbb{K}$ be a quadratic extension, so $|\mathbb{K}'| > nr$ for sufficiently large n . Apply [Theorem 7.2](#) over $\mathbb{K}'$ with $d = \lceil C(\varepsilon/2) \rceil$ and sufficiently large $g = O_{r,\varepsilon}(1)$, so that the points satisfy the hypotheses of [Theorem 5.1](#) with gap $\varepsilon/2$ and

$$\frac{\lceil n/g \rceil}{n} \log_2 r \leq \varepsilon/4.$$

The construction takes time $|\mathbb{K}|^{O_{r,\varepsilon}(1)}$ and gives $[\mathbb{F} : \mathbb{K}] = O_{r,\varepsilon}(1)$. Thus, [Theorem 4.2](#) gives the claimed subspace-design bound over $\mathbb{K}'$, which also holds over every $\mathbb{K}_0 \subseteq \mathbb{K}$ by [Lemma 4.5](#).

Write $R = k/n$. Since $\lfloor 2/\varepsilon \rfloor \leq r$ and $\tau(\lfloor 2/\varepsilon \rfloor) \leq R + \varepsilon/4$, [Corollary 2.5](#) with gap $\varepsilon/2$ gives (δ_L, L) -average-radius list decodability for every integer $L \geq 1$.

Now, run [Theorem 5.1](#) with gap $\varepsilon/2$ and retain the codewords within distance δ_L or $1 - R - \varepsilon$, respectively. For the latter, let $L_0 = \lceil 2(1 - R)/\varepsilon \rceil \leq r$. Since $\tau(L_0) \leq R + \varepsilon/4$ and

$$\frac{L_0}{L_0 + 1}(1 - R - \varepsilon/4) > 1 - R - \varepsilon,$$

[Theorem 2.4](#) gives a list of size at most L_0 . The initial list has size $(1/\varepsilon)^{O(1/\varepsilon)}$, and the enumeration and filtering run deterministically in time $O_{r,\varepsilon}(1) \text{poly}(n, \log |\mathbb{F}|)$. $\square$

Remark 3.1. We expect our algorithmic techniques to extend naturally to list-recovery and curve-decoding. We restrict ourselves here to the combinatorial consequences for these tasks.

We now apply the results in [Section 2](#) to the codes of [Theorem 1.1](#) to obtain various combinatorial consequences. Throughout, n is sufficiently large in the fixed parameters and $\mathbb{K}$ is a finite field with $|\mathbb{K}| \geq n$.

3.1 List recovery

Theorem 3.2 (Capacity list recovery). *Fix $0 < R < 1$, $0 < \varepsilon < 1 - R$, and an integer $\ell \geq 2$, and let*

$$r = \left\lceil \frac{2\ell}{\varepsilon} \right\rceil, \quad L = \left\lceil \left(\frac{\ell}{R + \varepsilon/2} \right)^{3+2R/\varepsilon} \right\rceil.$$

Let $\mathbb{F}, A$ be supplied by [Theorem 1.1](#) with parameters r, ε , and put $C = \text{RS}_{\mathbb{F}}(A, \lfloor Rn \rfloor)$. Then C is $(1 - R - \varepsilon, \ell, L)$ -list-recoverable.

Proof. [Theorem 1.1](#) gives $\tau(r) \leq R + \varepsilon/4$. Apply [Theorem 2.7](#) to C with $\eta = \varepsilon/2$ and design bound $R + \varepsilon/4$. At radius $1 - R - \varepsilon$, the list size is at most

$$\left(\frac{\ell}{R + 3\varepsilon/4} \right)^{2R/\varepsilon + 3/2} \leq \left(\frac{\ell}{R + \varepsilon/2} \right)^{2R/\varepsilon + 3} \leq L. \quad \square$$

3.2 Curve decoding

Theorem 3.3 (Base-field capacity curve decoding). *Fix $0 < R < 1$, $0 < \varepsilon < 1 - R$, and an integer $\ell \geq 1$. Set*

$$L = \left\lceil \frac{8}{\varepsilon} \right\rceil, \quad b_0 = \ell + \left\lceil \frac{8(\ell + 1)}{\varepsilon} \right\rceil, \quad r = \left\lceil \frac{8(\ell + 1)}{\varepsilon} \right\rceil, \quad a_0 = \left\lceil \frac{2b_0(r + \varepsilon/2)}{\varepsilon} \right\rceil.$$

Let $\mathbb{F}, A$ be supplied by Theorem 1.1 with parameters r, ε , and put $C = \text{RS}_{\mathbb{F}}(A, \lfloor Rn \rfloor)$. Then C is

$$(\ell, 1 - R - \varepsilon, a_0, b_0)\text{-}\mathbb{K}\text{-curve-decodable}.$$

Proof. Theorem 1.1 gives $\tau(d) \leq R + \varepsilon/4$ for $d \leq r$. Theorem 2.9, applied over $\mathbb{K}$ with $\eta = \varepsilon/2$, gives the displayed (a_0, b_0) statement: our choices ensure $\eta \geq (\ell + 1)/r$ and $\eta a_0/(r + \eta) \geq b_0$. $\square$

Theorem 3.4 (Extension-field capacity curve decoding). *Under the hypotheses and parameter choices of Theorem 3.3, assume also $|\mathbb{F}| > (L + 1)^2 \ell$, as holds for sufficiently large n , and put*

$$a_{\mathbb{F}} = \left\lceil \frac{|\mathbb{F}|}{|\mathbb{K}|} a_0 \right\rceil.$$

Then, for every integer $t \geq 1$, the code C is

$$(\ell, 1 - R - \varepsilon, (t - 1)L + a_{\mathbb{F}}, t)\text{-}\mathbb{F}\text{-curve-decodable}.$$

Proof. Partition $\mathbb{F}$ into affine lines over $\mathbb{K}$. Among $a_{\mathbb{F}}$ good parameters, some line contains at least a_0 . Applying Theorem 3.3 after translation gives $(\ell, 1 - R - \varepsilon, a_{\mathbb{F}}, b_0)\text{-}\mathbb{F}\text{-curve}$ decodability. The result follows from Theorem 2.13, using the $(1 - R - \varepsilon/2, L)\text{-list-decoding}$ bound from Theorem 2.4. $\square$

3.3 Line decoding

Theorem 3.5 (Capacity line decoding). *Fix $0 < R < 1$ and $0 < \varepsilon < 1 - R$. Set*

$$L = \left\lceil \frac{8}{\varepsilon} \right\rceil, \quad r = \left\lceil \frac{16}{\varepsilon} \right\rceil, \quad b_0 = 1 + \left\lceil \frac{16}{\varepsilon} \right\rceil, \quad a_0 = \left\lceil \frac{2b_0(r + \varepsilon/2)}{\varepsilon} \right\rceil.$$

Let $\mathbb{F}, A$ be supplied by Theorem 1.1 with parameters r, ε , and put $C = \text{RS}_{\mathbb{F}}(A, \lfloor Rn \rfloor)$. Then, for every integer $t \geq 1$, C is

$$(1, 1 - R - \varepsilon, (t - 1)L + \lceil |\mathbb{F}| a_0 / |\mathbb{K}| \rceil, t)\text{-}\mathbb{F}\text{-curve-decodable}.$$

Proof. This is Theorem 3.4 with $\ell = 1$. $\square$

3.4 Proximity gap, correlated agreement, and mutual correlated agreement

Let $C, L, a_{\mathbb{F}}$ be as in Theorem 3.4, and put

$$\delta = 1 - R - \varepsilon.$$

We now apply Theorem 2.13 over $\mathbb{F}$ to the list-decoding and curve-decoding guarantees proved above.

Corollary 3.6 (Capacity correlated agreement and proximity gap). *For every real $m > 1$, C has*

$$\left(\ell, \delta, \frac{\ell m L + a_{\mathbb{F}}}{|\mathbb{F}|}, \frac{1}{m} \right)$$

correlated agreement and proximity gap over $\mathbb{F}$.

Corollary 3.7 (Capacity mutual correlated agreement). *C has*

$$\left(\ell, \delta, \frac{\ell n L + a_{\mathbb{F}}}{|\mathbb{F}|} \right)$$

mutual correlated agreement over $\mathbb{F}$.

Remark 3.8. The affine-line reduction introduces a factor $|\mathbb{F}|/|\mathbb{K}|$ in the curve-decoding threshold. For $\ell = 1$, [Section 8](#) gives a better result under a stronger subspace-design assumption. By assuming the same design bound through a larger dimension, we obtain a line-decoding threshold without the $|\mathbb{F}|/|\mathbb{K}|$ factor.

4 Reed-Solomon Subspace Designs

In this section, we detail sufficient properties of evaluation points so that the resulting Reed-Solomon code forms a subspace-design code over a subfield. We prove that these properties suffice, and an explicit construction matching these properties is given in [Section 7](#).

4.1 Assumptions and the subspace-design theorem

Definition 4.1 (Properties required of the evaluation points). Let $\mathbb{K} \subseteq \mathbb{F}$ be finite fields, fix integers $n, g \geq 1$, $0 \leq b \leq n$, and $r \geq 2$.

A set of n distinct points $A = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}^n$ is $(n, g, b, r)_{\mathbb{F}, \mathbb{K}}$ -effective if:

- (i) There are automorphisms $\sigma_1, \dots, \sigma_g$ generating $\text{Gal}(\mathbb{F}/\mathbb{K})$ such that

$$|\{i \in [n] : \sigma_j(\alpha_i) \neq \alpha_i\}| \leq b \quad (j \in [g]).$$

These automorphisms commute, since the Galois group of a finite-field extension is cyclic and generated by the Frobenius automorphism $x \mapsto x^{|\mathbb{K}|}$.

- (ii) For every nonzero $\mathbb{K}$ -subspace $\mathcal{Q} \subseteq \mathbb{K}[X]_{<nr}$ of dimension at most 2^r , then

$$\sum_{i=1}^n \max_{0 \neq q \in \mathcal{Q}} \text{mult}_{\alpha_i} q \leq \max_{0 \neq q \in \mathcal{Q}} \deg q. \quad (4.1)$$

Here $\text{mult}_{\alpha} q$ is the multiplicity of α as a root of q .

Theorem 4.2 (Subspace-design bound). *Let $\mathbb{K} \subseteq \mathbb{F}$ be finite fields, and let $A = (\alpha_1, \dots, \alpha_n)$ be $(n, g, b, r)_{\mathbb{F}, \mathbb{K}}$ -effective. For every $1 \leq k \leq n$, the code $C = \text{RS}_{\mathbb{F}}(A, k)$ is a τ -subspace-design code over $\mathbb{K}$, with*

$$\tau(d) = \min \left\{ 1, \frac{k-1}{n} + \frac{b}{n} \log_2 d \right\} \quad (1 \leq d \leq r), \quad (4.2)$$

with $\tau(d) = 1$ for $d > r$.

The proof here is inspired by Guruswami and Kopparty's Wronskian proof of the subspace-design bound for folded Reed-Solomon codes [GK16]. It is adapted to suit working over extension field Reed-Solomon codes and automorphisms using Moore/Galois determinants as in the work of Guruswami and Xing. The key innovation here in comparison to [GX13] is that we allow a small family of automorphisms generating the Galois group each fixing most evaluation points. In comparison, Guruswami and Xing used the Frobenius automorphism to generate the group. Their argument requires this automorphism to fix all evaluation points, so the evaluation points lie in the subfield $\mathbb{K}$. This variation allows us to have the second property, as we can now consider evaluation points to all be non-conjugate and of high degree.

4.2 From effective evaluation points to subspace designs

Our claimed subspace-design property is a property of spaces with bounded $\mathbb{K}$ -dimension. We first prove a bound in terms of $\mathbb{K}(X)$ -ranks and then use a transfer theorem to convert it into a bound on $\mathbb{K}$ -dimensions. Moreover, we show that the same *transfer theorem* (Theorem 4.4) transfers the subspace-design property to subfields of $\mathbb{K}$.

Proposition 4.3 (Rational ranks of the coordinate kernels). *Let $\mathbb{K} \subseteq \mathbb{F}$ be finite fields, let $1 \leq k \leq n$, and let $A = (\alpha_1, \dots, \alpha_n)$ be $(n, g, b, r)_{\mathbb{F}, \mathbb{K}}$ -effective. Let $\{0\} \neq W \subseteq \mathbb{F}[X]_{<k}$ with $\dim_{\mathbb{K}} W = e \leq r$, and $\dim_{\mathbb{K}(X)} \text{span}_{\mathbb{K}(X)}(W) = t \leq e$. Set $W_i = \{f \in W : f(\alpha_i) = 0\}$ and $t_i = \dim_{\mathbb{K}(X)} \text{span}_{\mathbb{K}(X)}(W_i)$. Then*

$$\sum_{i=1}^n t_i \leq t(k-1) + bt \log_2 t.$$

Proof. Put $\mathbb{L} = \mathbb{K}(X)$, and choose generators $\sigma_1, \dots, \sigma_g$ witnessing the effectiveness of A . Apply Corollary 2.16 to $\text{span}_{\mathbb{L}} W$ and choose its reduced polynomial basis $u_1, \dots, u_t$, and let

$$\nu = \sum_{a=1}^t \deg u_a, \quad A_W = t(k-1) - \nu.$$

Write a $\mathbb{K}$ -basis of W as

$$(w_1, \dots, w_e) = (u_1, \dots, u_t)C(X), \quad C(X) \in \mathbb{K}[X]^{t \times e}.$$

By (2.1), the degree of row a of C is at most $k-1 - \deg u_a$. Let $\mathcal{Q}_W$ be the $\mathbb{K}$ -span of the $t \times t$ minors of C . The matrix has rank t , so $\mathcal{Q}_W \neq 0$. Moreover,

$$\dim_{\mathbb{K}} \mathcal{Q}_W \leq \binom{e}{t} \leq 2^r, \quad \deg q \leq A_W \leq r(n-1) < rn \quad (q \in \mathcal{Q}_W).$$

Apply Lemma 2.20 to $u_1, \dots, u_t$ to obtain a row index set $\mathcal{B} \subseteq \mathbb{Z}_{\geq 0}^g$ of size t such that

$$P(X) = \det(\sigma^\omega(u_a)(X))_{\omega \in \mathcal{B}, a \in [t]} \neq 0.$$

Expanding the determinant gives $\deg P \leq \sum_a \deg u_a = \nu$.

Fix a coordinate i , and set

$$b_i = |\{\omega \in \mathcal{B} : \sigma^\omega(\alpha_i) \neq \alpha_i\}|.$$

Choose $h_1, \dots, h_{t_i} \in W_i$ independent over $\mathbb{L}$, and extend them, using elements of W , to an $\mathbb{L}$ -basis $h_1, \dots, h_t$ of $\text{span}_{\mathbb{L}} W$. There is a matrix $T_i \in \mathbb{K}^{e \times t}$ such that

$$(h_1, \dots, h_t) = (u_1, \dots, u_t)C(X)T_i.$$

By the Cauchy-Binet formula for determinant (Theorem 2.14),

$$q_i := \det(C(X)T_i) = \sum_{\substack{S \subseteq [e] \\ |S|=t}} \det C_{[t],S}(X) \det(T_i)_{S,[t]} \in \mathcal{Q}_W \setminus \{0\}.$$

Here the coefficients $\det(T_i)_{S,[t]}$ lie in $\mathbb{K}$, and $q_i \neq 0$ because the h_j are independent over $\mathbb{L}$. Since every Galois row fixes the entries of $C(X)T_i$, the determinant for this adapted basis is $q_i(X)P(X)$.

In every row whose automorphism fixes α_i , the first t_i entries vanish at α_i . Each determinant term uses a moving row in at most b_i of these columns. Hence

$$\begin{aligned} t_i - b_i &\leq \text{mult}_{\alpha_i}(q_i P) \\ &= \text{mult}_{\alpha_i} q_i + \text{mult}_{\alpha_i} P \\ &\leq \max_{0 \neq q \in \mathcal{Q}_W} \text{mult}_{\alpha_i} q + \text{mult}_{\alpha_i} P. \end{aligned}$$

Summing and using distinctness of the points, (4.1), and (2.2), we obtain

$$\begin{aligned} \sum_i t_i &\leq \sum_i b_i + \sum_i \text{mult}_{\alpha_i} P + \sum_i \max_{0 \neq q \in \mathcal{Q}_W} \text{mult}_{\alpha_i} q \\ &\leq bt \log_2 t + \deg P + A_W \\ &\leq bt \log_2 t + \nu + A_W = t(k-1) + bt \log_2 t. \end{aligned} \quad \square$$

Theorem 4.4 (From extension-field ranks to dimensions). *Let $\mathbb{E}_0 \subseteq \mathbb{E}_1$ be fields, let C be an $\mathbb{E}_0$ -subspace of an $\mathbb{E}_1$ -vector space, and let $H_1, \dots, H_n$ be $\mathbb{E}_0$ -subspaces of C . Let $r \geq 1$ and $\tau \geq 0$. Suppose every nonzero $\mathbb{E}_0$ -subspace $W \subseteq C$ with $\dim_{\mathbb{E}_0} W \leq r$ satisfies*

$$\sum_{i=1}^n \dim_{\mathbb{E}_1} \text{span}_{\mathbb{E}_1}(W \cap H_i) \leq n\tau \dim_{\mathbb{E}_1} \text{span}_{\mathbb{E}_1} W. \quad (4.3)$$

Then every $\mathbb{E}_0$ -subspace $U \subseteq C$ of dimension $1 \leq d \leq r$ satisfies

$$\sum_{i=1}^n \dim_{\mathbb{E}_0}(U \cap H_i) \leq n\tau d.$$

We prove Theorem 4.4 in Subsection 4.3 by breaking any $\mathbb{E}_0$ -linear space into a chain of spaces on which the hypothesis applies and then taking a nonnegative linear combination of the resulting inequalities. Before proving the result, we use it to obtain both the Reed-Solomon subspace-design guarantee of Theorem 4.2 and additionally prove that the same transfers to subfields of $\mathbb{K}$ as below:

Lemma 4.5 (Subspace-design transfers to subfields). *Let $\mathbb{K}_0 \subseteq \mathbb{K}$ be finite fields. If a $\mathbb{K}$ -additive code $C \subseteq \Sigma^n$ is a τ -subspace design over $\mathbb{K}$, then it is also a τ -subspace design over $\mathbb{K}_0$.*

Proof of Theorem 4.2 and Lemma 4.5. Both statements follow from Theorem 4.4. For Theorem 4.2, let $\mathbb{E}_1 = \mathbb{K}(X)$, and $\mathbb{E}_0 = \mathbb{K}$. For Lemma 4.5, let $\mathbb{E}_1 = \mathbb{K}$ and $\mathbb{E}_0 = \mathbb{K}_0$.

Observe that the hypothesis in the first case is true by Proposition 4.3, and the conclusion is exactly the guarantee of Theorem 4.2.

Similarly, if C is a τ -subspace design over $\mathbb{K}$, then the hypothesis of Theorem 4.4 holds with $\mathbb{K} = \mathbb{E}_1$, and the conclusion is exactly that C is also a τ subspace-design over $\mathbb{K}_0$. □

4.3 Proof of the transfer theorem

Lemma 4.6 (Dimension comparison). *Let $\mathbb{E}_0 \subseteq \mathbb{E}_1$ be fields, and let U be a finite-dimensional $\mathbb{E}_0$ -subspace of an $\mathbb{E}_1$ -vector space. For $S \subseteq U$, write $\rho(S) = \dim_{\mathbb{E}_1} \text{span}_{\mathbb{E}_1} S$. There are spaces*

$$0 = U_0 \subsetneq U_1 \subsetneq \cdots \subsetneq U_h = U, \quad U_i = U \cap \text{span}_{\mathbb{E}_1} U_i \quad \forall i \in \{0, 1, \dots, h\}$$

and nonnegative real numbers $a_1, \dots, a_h$ such that, for every $\mathbb{E}_0$ -subspace $S \subseteq U$,

$$\dim_{\mathbb{E}_0} S \leq \sum_{\ell=1}^h a_\ell \rho(S \cap U_\ell), \quad \sum_{\ell=1}^h a_\ell \rho(U_\ell) = \dim_{\mathbb{E}_0} U.$$

Proof. If $U = 0$, take $h = 0$. Otherwise, starting with $U_0 = 0$, choose a $\mathbb{E}_0$ -subspace $U_{\ell-1} \subsetneq U_\ell \subseteq U$ maximizing

$$\lambda_\ell = \frac{\dim_{\mathbb{E}_0} U_\ell - \dim_{\mathbb{E}_0} U_{\ell-1}}{\rho(U_\ell) - \rho(U_{\ell-1})}.$$

Inductively, $U_{\ell-1} = U \cap \text{span}_{\mathbb{E}_1} U_{\ell-1}$, so every denominator is positive. A maximizer exists because there are only finitely many dimension–rank pairs. Replacing U_ℓ by $U \cap \text{span}_{\mathbb{E}_1} U_\ell$ preserves its rank and would increase its dimension unless the two spaces coincide. Thus the required intersection property persists, and strict dimension growth gives a sequence of spaces ending at U .

The slope from $U_{\ell-1}$ to $U_{\ell+1}$ is a weighted average of λ_ℓ and $\lambda_{\ell+1}$. Maximality of λ_ℓ therefore gives $\lambda_{\ell+1} \leq \lambda_\ell$. Set $\lambda_{h+1} = 0$ and $a_\ell = \lambda_\ell - \lambda_{\ell+1} \geq 0$.

For a $\mathbb{E}_0$ -subspace $S \subseteq U$, put $S_\ell = S \cap U_\ell$. Maximality applied to $S_\ell + U_{\ell-1}$ and the dimension formulas over $\mathbb{E}_0$ and $\mathbb{E}_1$ give

$$\begin{aligned} \dim_{\mathbb{E}_0} S_\ell - \dim_{\mathbb{E}_0} S_{\ell-1} &\leq \lambda_\ell (\rho(S_\ell + U_{\ell-1}) - \rho(U_{\ell-1})) \\ &\leq \lambda_\ell (\rho(S_\ell) - \rho(S_{\ell-1})). \end{aligned}$$

Here the second inequality uses $\text{span}_{\mathbb{E}_1} S_{\ell-1} \subseteq \text{span}_{\mathbb{E}_1} S_\ell \cap \text{span}_{\mathbb{E}_1} U_{\ell-1}$; if $S_\ell \subseteq U_{\ell-1}$, both increments are zero.

Summing these inequalities and then summing by parts yields

$$\begin{aligned} \dim_{\mathbb{E}_0} S &\leq \sum_{\ell=1}^h \lambda_\ell (\rho(S_\ell) - \rho(S_{\ell-1})) \\ &= \sum_{\ell=1}^h a_\ell \rho(S \cap U_\ell). \end{aligned}$$

For $S = U$, every increment inequality is an equality by the definition of λ_ℓ , giving the required normalization. $\square$

Proof of Theorem 4.4. Fix a $\mathbb{E}_0$ -subspace $U \subseteq C$ of dimension $1 \leq d \leq r$. Write $\rho(S) = \dim_{\mathbb{E}_1} \text{span}_{\mathbb{E}_1} S$. Apply Lemma 4.6 with scalar field $\mathbb{E}_0$ and rank field $\mathbb{E}_1$, obtaining spaces U_ℓ and weights a_ℓ . Each U_ℓ has $\mathbb{E}_0$ -dimension at most r , so (4.3) gives

$$\begin{aligned} \sum_i \dim_{\mathbb{E}_0}(U \cap H_i) &\leq \sum_\ell a_\ell \sum_i \rho(U_\ell \cap H_i) \\ &\leq n\tau \sum_\ell a_\ell \rho(U_\ell) = n\tau d. \end{aligned}$$

The equality is the normalization in Lemma 4.6. $\square$

5 Efficient List Decoding

As discussed in [Subsection 1.2](#), our efficient decoder follows the following blueprint, given our constructed Reed-Solomon code $C \subseteq \mathbb{F}^n$ with $\mathbb{K} \subseteq \mathbb{F}$.

- Finds a low-dimensional $\mathbb{K}(X)$ space of $\mathbb{F}(X)$ containing all close by message polynomials. ([Theorem 5.2](#))
- *Prunes* a low-dimensional $\mathbb{K}(X)$ space via low-degree constraints on the space. ([Theorem 5.3](#))

We first state our combined result and the required properties of the evaluation points.

Theorem 5.1 (List decoding near capacity). *For every $0 < \varepsilon < 1/2$, there exists a constant*

$$C(\varepsilon) = \exp(O(\varepsilon^{-3} \log(1/\varepsilon)))$$

with the following property. Let $\mathbb{K} \subseteq \mathbb{F}$ be finite fields, and let $\alpha_1, \dots, \alpha_n \in \mathbb{F}$, where n is sufficiently large in terms of ε . Write $\deg_{\mathbb{K}} \alpha = [\mathbb{K}(\alpha) : \mathbb{K}]$. Suppose that the evaluation points satisfy the following conditions:

1. *The points $\alpha_1, \dots, \alpha_n$ are pairwise nonconjugate over $\mathbb{K}$.*
2. *There is a positive integer*

$$g \geq \frac{12 \log_2(1/\varepsilon)}{\varepsilon}$$

and a generating set $\sigma_1, \dots, \sigma_g$ of $\text{Gal}(\mathbb{F}/\mathbb{K})$ such that, for every $1 \leq \ell \leq g$,

$$|\{j \in \{1, \dots, n\} : \sigma_\ell(\alpha_j) \neq \alpha_j\}| \leq \left\lceil \frac{n}{g} \right\rceil.$$

3. *For every $1 \leq i, j \leq n$,*

$$C(\varepsilon) \leq \deg_{\mathbb{K}} \alpha_i \leq 2 \deg_{\mathbb{K}} \alpha_j.$$

Then, for every integer $1 \leq k \leq n$, the Reed-Solomon code $\text{RS}_{\mathbb{F}}((\alpha_1, \dots, \alpha_n), k)$ evaluated at $\alpha_1, \dots, \alpha_n$ admits a deterministic list-decoding algorithm with the following guarantee. Given the evaluation points, the generators, and a received word $y = (y_1, \dots, y_n) \in \mathbb{F}^n$, the algorithm outputs all $f \in \mathbb{F}[X]_{<k}$ satisfying

$$|\{j \in \{1, \dots, n\} : f(\alpha_j) \neq y_j\}| \leq n - k - \varepsilon n.$$

The output list has size at most

$$(1/\varepsilon)^{O(1/\varepsilon)}.$$

*and the algorithm runs in time $O_{g,\varepsilon}(1) \text{poly}(n, \log |\mathbb{F}|)$.*⁶

Combining [Theorem 5.1](#) with [Theorem 7.1](#) gives the following as a corollary.

Theorem 1.3 (Strongly explicit list-decodable Reed-Solomon codes). *Fix $\varepsilon \in (0, 1)$. For all sufficiently large n , and an arbitrary field $\mathbb{K}$ of size at least n , there is a strongly explicit construction of an extension field $\mathbb{F}$ over $\mathbb{K}$ with $[\mathbb{F} : \mathbb{K}] = O_\varepsilon(1)$ and evaluation points $A = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}^n$, such that for every $1 \leq k \leq n$ the Reed-Solomon code $\text{RS}_{\mathbb{F}}(A, k)$ of rate $R = k/n$ is*

$$(1 - R - \varepsilon, (1/\varepsilon)^{O(1/\varepsilon)})$$

list-decodable, and the list can be found deterministically in time $O_\varepsilon(1) \text{poly}(n, \log |\mathbb{F}|)$. Given an index $i \in [n]$, α_i can be computed in time $\text{poly}(\log |\mathbb{F}|)$.

⁶We use the deterministic finite-field factorization algorithm of [Corollary 2.18](#).

Proof. We can apply [Theorem 7.1](#) with $g = \lceil 12 \log_2(1/\varepsilon)/\varepsilon \rceil$ and $d = \lceil C(\varepsilon) \rceil$, and the result follows using [Theorem 5.1](#) with gap ε . $\square$

5.1 The two-step list-decoding theorems

The two theorems below give the two steps of the list-decoding algorithm. Interestingly, the two assumptions of the choice of automorphisms σ_i , and the requirement of sufficiently high degree of α_i are used in the two steps separately and do not interact with each other.

Theorem 5.2 (Finding a low-dimensional $\mathbb{K}(X)$ -space). *Fix $0 < \varepsilon < 1/2$. Let $\mathbb{K} \subseteq \mathbb{F}$ be finite fields, and consider a Reed–Solomon code of length n and dimension k over $\mathbb{F}$, where n is sufficiently large in terms of ε , with distinct evaluation points $\alpha_1, \dots, \alpha_n \in \mathbb{F}$. Suppose $\text{Gal}(\mathbb{F}/\mathbb{K})$ has $g \geq 1$ generators $\sigma_1, \dots, \sigma_g$ with*

$$g \geq \frac{12 \log_2(1/\varepsilon)}{\varepsilon},$$

such that each σ_ℓ , we have that $|\{\alpha_i \mid \sigma_\ell(\alpha_i) \neq \alpha_i\}| \leq \lceil n/g \rceil$.

Given the evaluation points, these generators, and any received word $y = (y_1, \dots, y_n) \in \mathbb{F}^n$, one can compute in time $\text{poly}(n, g, \log |\mathbb{F}|)$ a $\mathbb{K}(X)$ -basis consisting of polynomials in $\mathbb{F}[X]_{<k}$ of a space $A \subseteq \mathbb{F}(X)$ with

$$\dim_{\mathbb{K}(X)} A \leq \frac{2}{\varepsilon}$$

that contains every polynomial $f \in \mathbb{F}[X]$ of degree less than k satisfying

$$|\{j \in \{1, \dots, n\} : f(\alpha_j) = y_j\}| \geq k + \varepsilon n.$$

Theorem 5.3 (Pruning a low-dimensional $\mathbb{K}(X)$ -space). *For every integer $d \geq 1$ and every $0 < \varepsilon < 1/2$, there is a constant*

$$C(d, \varepsilon) = \exp(O(d^3 \log(1/\varepsilon)))$$

with the following property. Let $\mathbb{K} \subseteq \mathbb{F}$ be finite fields, and consider a Reed–Solomon code of length n and dimension k over $\mathbb{F}$, where n is sufficiently large in terms of d and ε . Suppose its evaluation points $\alpha_1, \dots, \alpha_n \in \mathbb{F}$ are pairwise nonconjugate over $\mathbb{K}$ and satisfy

$$C(d, \varepsilon) \leq \deg_{\mathbb{K}} \alpha_i \leq 2 \deg_{\mathbb{K}} \alpha_j \quad (1 \leq i, j \leq n),$$

where $\deg_{\mathbb{K}} \alpha = [\mathbb{K}(\alpha) : \mathbb{K}]$.

Given the evaluation points, a received word $y = (y_1, \dots, y_n) \in \mathbb{F}^n$, and a $\mathbb{K}(X)$ -linear space $A \subseteq \mathbb{F}(X)$ of dimension at most d , given by a basis over $\mathbb{K}(X)$ consisting of polynomials in $\mathbb{F}[X]_{<k}$, one can output all polynomials $f \in A \cap \mathbb{F}[X]$ of degree less than k satisfying

$$|\{j \in \{1, \dots, n\} : f(\alpha_j) = y_j\}| \geq k + \varepsilon n.$$

The list has size at most

$$(O(1/\varepsilon))^d,$$

and can be computed in time $O_{d,\varepsilon}(1) \text{poly}(n, \log |\mathbb{F}|)$.

Proof of Theorem 5.1 assuming Theorem 5.2, Theorem 5.3. Apply [Theorem 5.2](#), followed by [Theorem 5.3](#) with $d = \lceil 2/\varepsilon \rceil$ and $C(\varepsilon) = C(d, \varepsilon)$. The bounds in [Theorem 5.3](#) give the claimed degree threshold and list size. $\square$

We now present the proof of [Theorem 5.2](#), and defer [Theorem 5.3](#) to the next section.

5.2 Finding a low-dimensional $\mathbb{K}(X)$ -space (Theorem 5.2)

To prove Theorem 5.2, we first prove a lemma showing that we can repeatedly refine any given $\mathbb{K}(X)$ -space containing close by codewords to be of lower dimension while continuing to contain all close by codewords until we get to dimension $\leq 2/\varepsilon$.

Lemma 5.4 (Dimension reduction). *Fix $0 < \varepsilon < 1/2$. Let $\mathbb{K} \subseteq \mathbb{F}$ be finite fields and let $\alpha_1, \dots, \alpha_n \in \mathbb{F}$ be distinct, where n is sufficiently large in terms of ε . Suppose $\sigma_1, \dots, \sigma_g$ generate $\text{Gal}(\mathbb{F}/\mathbb{K})$, where $g \geq 12 \log_2(1/\varepsilon)/\varepsilon$, and for each σ_ℓ , we have $|\{\alpha_i \mid \sigma_\ell(\alpha_i) \neq \alpha_i\}| \leq \lceil n/g \rceil$.*

Given these points and generators, a received word $y \in \mathbb{F}^n$, an integer $1 \leq k \leq n$, and a $\mathbb{K}(X)$ -linear space $A \subseteq \mathbb{F}(X)$ of dimension greater than $2/\varepsilon$, given by a basis over $\mathbb{K}(X)$ consisting of polynomials in $\mathbb{F}[X]_{<k}$, one can compute in time $\text{poly}(n, g, \log |\mathbb{F}|)$ a strict subspace $A' \subsetneq A$ containing every $f \in A \cap \mathbb{F}[X]_{<k}$ with at least $k + \varepsilon n$ agreements with y . Moreover, A' has a basis of polynomials of degree less than k .

Proof. Put $\mathbb{L} = \mathbb{K}(X)$ and $d = \lceil 2/\varepsilon \rceil$. Let $V = A \cap \mathbb{F}[X]_{<k}$ and choose $u_1, \dots, u_d$ from the given basis of A . Lemma 2.20 gives automorphisms $\tau_1, \dots, \tau_d \in \text{Gal}(\mathbb{F}/\mathbb{K})$ such that

$$\det(\tau_i(u_a))_{1 \leq i, a \leq d} \neq 0.$$

By the support bound in the proof of Lemma 2.20, if

$$S_i = \{j : \tau_i(\alpha_j) \neq \alpha_j\}, \quad s_i = |S_i|,$$

we have:

$$s_i \leq \left\lceil \frac{n}{g} \right\rceil \log_2 d \leq \frac{2n \log_2 d}{12 \log_2(1/\varepsilon)/\varepsilon} \leq \frac{\varepsilon n}{2}$$

for sufficiently large n . Let $h = \lceil \varepsilon n \rceil$ and define

$$E_i(X) = \prod_{j \in S_i} (X - \alpha_j).$$

Find polynomials $Q_i = E_i R_i$ of degree at most h , imposing the constraints

$$\sum_{i=1}^d Q_i(\alpha_j) \tau_i(y_j) = 0 \quad (1 \leq j \leq n). \quad (5.1)$$

These are n homogeneous linear equations over $\mathbb{F}$. Since $s_i \leq \varepsilon n/2 \leq h$, the number of unknown coefficients in the R_i is

$$\sum_{i=1}^d (h - s_i + 1) \geq d \left(\frac{\varepsilon n}{2} + 1 \right) > n.$$

Thus a nonzero tuple $(Q_1, \dots, Q_d)$ can be found by Gaussian elimination in time $\text{poly}(n, \log |\mathbb{F}|)$.

Define the $\mathbb{L}$ -linear map

$$\Psi : \mathbb{F}(X) \longrightarrow \mathbb{F}(X), \quad \Psi(f) = \sum_{i=1}^d Q_i \tau_i(f).$$

Suppose $f \in V$ agrees with y at position j . If $j \notin S_i$, then

$$\tau_i(f)(\alpha_j) = \tau_i(f(\alpha_j)) = \tau_i(y_j).$$

If $j \in S_i$, the factor E_i gives $Q_i(\alpha_j) = 0$. (5.1) therefore implies $\Psi(f)(\alpha_j) = 0$ in either case, after summing over i . If f has at least $k + \varepsilon n$ agreements, then $\Psi(f)$ has at least $k + h > \deg \Psi(f)$ distinct roots. Hence, $\Psi(f) = 0$.

Since the vectors $(\tau_i(u_1), \dots, \tau_i(u_d))$, $i \in [d]$, are linearly independent over $\mathbb{F}(X)$ and the Q_i are not all zero, Ψ is nonzero on some $u_a \in A$. We may therefore return

$$A' = \text{span}_{\mathbb{L}}\{f \in V : \Psi(f) = 0\} \subseteq A \cap \ker \Psi \subsetneq A.$$

This space contains every required polynomial and has a basis in $\mathbb{F}[X]_{<k}$. This can also be computed in time $\text{poly}(n, \log |\mathbb{F}|)$ as desired. $\square$

Proof of Theorem 5.2. Start with $A = \mathbb{F}(X)$, using a $\mathbb{K}$ -basis of $\mathbb{F}$ as a constant basis over $\mathbb{K}(X)$. Its dimension is $[\mathbb{F} : \mathbb{K}]$. While $\dim_{\mathbb{K}(X)} A > 2/\varepsilon$, apply Lemma 5.4 and replace A by A' . Every iteration strictly decreases the dimension and preserves all polynomials in the output list. Hence at most $[\mathbb{F} : \mathbb{K}]$ iterations produce the required space and the algorithm runs in time $\text{poly}(n, g, \log |\mathbb{F}|)$. $\square$

6 Pruning low-dimensional spaces of Reed-Solomon codes

Our goal is to prove the following theorem.

Theorem 5.3 (Pruning a low-dimensional $\mathbb{K}(X)$ -space). *For every integer $d \geq 1$ and every $0 < \varepsilon < 1/2$, there is a constant*

$$C(d, \varepsilon) = \exp(O(d^3 \log(1/\varepsilon)))$$

with the following property. Let $\mathbb{K} \subseteq \mathbb{F}$ be finite fields, and consider a Reed-Solomon code of length n and dimension k over $\mathbb{F}$, where n is sufficiently large in terms of d and ε . Suppose its evaluation points $\alpha_1, \dots, \alpha_n \in \mathbb{F}$ are pairwise nonconjugate over $\mathbb{K}$ and satisfy

$$C(d, \varepsilon) \leq \deg_{\mathbb{K}} \alpha_i \leq 2 \deg_{\mathbb{K}} \alpha_j \quad (1 \leq i, j \leq n),$$

where $\deg_{\mathbb{K}} \alpha = [\mathbb{K}(\alpha) : \mathbb{K}]$.

Given the evaluation points, a received word $y = (y_1, \dots, y_n) \in \mathbb{F}^n$, and a $\mathbb{K}(X)$ -linear space $A \subseteq \mathbb{F}(X)$ of dimension at most d , given by a basis over $\mathbb{K}(X)$ consisting of polynomials in $\mathbb{F}[X]_{<k}$, one can output all polynomials $f \in A \cap \mathbb{F}[X]$ of degree less than k satisfying

$$|\{j \in \{1, \dots, n\} : f(\alpha_j) = y_j\}| \geq k + \varepsilon n.$$

The list has size at most

$$(O(1/\varepsilon))^d,$$

and can be computed in time $O_{d,\varepsilon}(1) \text{poly}(n, \log |\mathbb{F}|)$.

We first prove the following warmup result.

Theorem 6.1. *Given $\varepsilon > 0$, a finite field $\mathbb{F}$, distinct evaluation points $\alpha_1, \dots, \alpha_n \in \mathbb{F}$, $y \in \mathbb{F}^n$, and an $\mathbb{F}$ -linear space $A \subseteq \mathbb{F}[X]_{<k}$ with $\dim A = d$, given by a basis, we give a deterministic algorithm running in*

$$(O(1/\varepsilon))^{O(d)} \cdot \tilde{O}(n \text{poly}(1/\varepsilon, d, \log |\mathbb{F}|))$$

which outputs

$$\mathcal{L} := \{f \in A : |\{i : f(\alpha_i) = y_i\}| \geq k + \varepsilon n\}, \quad |\mathcal{L}| \leq (O(1/\varepsilon))^d.$$

The key ingredients in proving [Theorem 6.1](#), are then used in the proof of [Theorem 5.3](#) with various modifications.

[KRSW23, Tam24] already give a randomized algorithm for a more general version of [Theorem 6.1](#); we do it deterministically. The previous best deterministic algorithm is due to [AHS26] (but it is also more general), getting parameter dependence $(2/\varepsilon)^{O(2^d)} \tilde{O}(n \text{ poly}(\log |\mathbb{F}|))$ in comparison to our $(O(1/\varepsilon))^{O(d)} \tilde{O}(n \text{ poly}(\log |\mathbb{F}|))$.

Proof differences between [Theorem 6.1](#) and [Theorem 5.3](#). The broad idea is similar, essentially replacing $\mathbb{F}$ with $\mathbb{K}(X)$. The resulting issue is that coefficients have degree. We will always control this degree using Nullstellensatz and linear algebra!

We will use high extension degree to show that any nonzero $\mathbb{K}[X]$ polynomial with degree $< \varepsilon C(d, \varepsilon) n$ has at most εn roots from our evaluation points. Comparable degrees are for simplicity in an internal interpolation step.

Proof method (describing only [Theorem 6.1](#)). Choose a basis $f_1, \dots, f_d$ of the initial linear space

$$A = \left\{ \sum_i c_i f_i : c_i \in \mathbb{F} \right\}.$$

Starting with $R = \{0\}$, we maintain a relation space

$$R \subseteq \mathbb{F}[Z_1, \dots, Z_d]_{\leq t}, \quad t \sim d/\varepsilon.$$

If $1 \in R$, we stop. Thus $1 \notin R$ whenever we continue. We try to find $Q \in \mathbb{F}[Z_1, \dots, Z_d]_{\leq t} \setminus R$ such that if

$$\sum_i c_i f_i \in \mathcal{L}, \quad R(c_1, \dots, c_d) = 0,$$

then $Q(c_1, \dots, c_d) = 0$.

There are three failure cases. The first is $R + \mathbb{F} = \mathbb{F}[Z_1, \dots, Z_d]_{\leq t}$. The second is that at least k evaluation coordinates are determined: after relabeling them,

$$\sum_{i=1}^d c_i f_i(\alpha_j) = y_j \quad \text{for each } j \in [k] \text{ and all } (c_1, \dots, c_d) \in Z(R).$$

We can then interpolate the unique polynomial from the k determined coordinates and check whether it belongs to $\text{Enc}(Z(R)) \cap \mathcal{L}$. In the third case, there exist

$$\ell \in \mathbb{F}[Z_1, \dots, Z_d]_{=1} \setminus R, \quad r \in \mathbb{F}[Z_1, \dots, Z_d]_{\leq t-1} \setminus R,$$

such that $\ell r \in R$. We branch into $R \cup \{\ell\}$ and $R \cup \{r\}$.

Naively, this gives an output-size bound of $\exp(\dim \mathbb{F}[Z_1, \dots, Z_d]_{\leq t})$. The first observation is that adding ℓ removes at least one remaining variable, which strengthens the bound to

$$(\dim \mathbb{F}[Z_1, \dots, Z_d]_{\leq t})^d.$$

The second observation is that the failure condition is stronger: there are savings in $R \cup \{r\}$ as well. This leads to

$$2^d \cdot \dim \mathbb{F}[Z_1, \dots, Z_d]_{\leq t}.$$

As a corollary of our subspace-decoding of Reed-Solomon codes, we also improve the best known deterministic list-decoding algorithm run-time for folded Reed-Solomon codes.

Corollary 6.2 (Deterministic FRS decoding). *Let $0 < \varepsilon < 1 - R$, put $m = \lceil 2/\varepsilon \rceil$ and $s = m^2$, and consider the s -folded Reed-Solomon code. There is a deterministic algorithm that runs in time*

$$(2/\varepsilon)^{O(1/\varepsilon)} \cdot \tilde{O}(n \text{ polylog}(q))$$

and, given a received word in $(\mathbb{F}_q^s)^n$, outputs all messages agreeing with it on at least $(R + \varepsilon)n$ coordinates.

Proof. We use the standard linear-interpolation stage for FRS codes, with its fast implementation in [GHKS24]. For parameters $m \leq s$, this stage computes a linear space of message polynomials of dimension at most m containing every message within relative distance

$$1 - R - \varepsilon \leq \frac{m}{m+1} \left(1 - \frac{sR}{s-m+1} \right)$$

of the received word due to the choice of m, s . Thus this linear space contains the entire desired folded list and has dimension $d = O(1/\varepsilon)$.

Unfold the received word into $N = sn$ scalar coordinates. Every desired message agrees with it on at least

$$s(R + \varepsilon)n = k + \varepsilon N$$

scalar coordinates. Apply Theorem 6.1 to this unfolded word within the computed linear space and compute all close by codewords. Finally, retain only the returned messages agreeing on at least $(R + \varepsilon)n$ coordinates of the original folded Reed-Solomon code. $\square$

6.1 Pruning for $\mathbb{F}$ -subspaces of Reed-Solomon codes (Theorem 6.1)

We now follow the proof strategy described above. But, we first set up notation.

Fix a basis $f_1, \dots, f_d$ of $A \subseteq \mathbb{F}[X]_{<k}$ and distinct evaluation points $\alpha_1, \dots, \alpha_n \in \mathbb{F}$. We maintain an $\mathbb{F}$ -linear relation space $R \subseteq \mathbb{F}[Z_1, \dots, Z_d]_{\leq t}$, initially $R = \{0\}$.

For $0 \leq s \leq t$, let $M_s = \{Z^a : \text{wt}(a) \leq s\}$ and $S_s = \text{span}(M_s)$, where $\text{wt}(a) = a_1 + \dots + a_d$ is the total degree.

Definition 6.3. We say that $R \subseteq S_t$ is *ready* if

$$(R \cap S_{t-1})Z_a \subseteq R \quad \text{for every } a \in [d].$$

Write $\text{ready}(R) = R'$ for the space obtained by closing the space of relations R under the above operation. All the added relations in R' also vanish on $Z(R)$, and hence $Z(R) = Z(R')$.

Notation. The relation space R is stored as a matrix in $\mathbb{F}^{\dim R \times M_t}$, where the M_t monomials are ordered by decreasing total degree, with arbitrary tie-breaking. The pivot monomials of R are the $\dim R$ monomials with a 1 entry in the corresponding relation and no entry in any preceding monomial. The remaining monomials are $P_t = M_t \setminus \{\text{pivots}\}$. For $0 \leq s \leq t$, write $P_s = P_t \cap M_s$ and $W_s = \text{span}(P_s)$. Row reduction gives $S_s = (R \cap S_s) + W_s$, with $(R \cap S_s) \cap W_s = \{0\}$.

Definition 6.4. Let ℓ_j be the relation

$$\ell_j = \left(\sum_{i=1}^d f_i(\alpha_j) Z_i \right) - y_j.$$

A coordinate $j \in [n]$ is determined if $\ell_j \in R$. Let $J = \{j \in [n] : \ell_j \notin R\}$ be the undetermined coordinates.

Notation. $Z(R)$ denotes the common zeros of R in $\mathbb{F}^d$, and $\text{Enc}(c_1, \dots, c_d) = \sum_i c_i f_i$.

Claim 6.5. *For every ready R ,*

$$\frac{d|P_t|}{d+t} \geq |P_t| - |P_{t-1}|.$$

Proof. For every $s \leq t$, the number of remaining monomials of degree at most s is

$$|P_s| = \dim S_s - \dim(R \cap S_s).$$

These counts do not depend on how ties are broken. For this proof only, break ties lexicographically.

If a pivot monomial has degree less than t , multiplying its corresponding relation by any Z_j gives another relation in R , since R is *ready*. Its pivot is the original pivot multiplied by Z_j . Thus every divisor of a remaining monomial is also remaining.

For $2 \leq s \leq t$, let N_s be the number of triples $(Z^{\bar{a}}, Z_i, b_i)$ such that $Z^{\bar{a}} \in P_s \setminus P_{s-1}$, $Z_i^{b_i} \mid Z^{\bar{a}}$ and $1 \leq b_i \leq a_i$. Note that $N_s = s(|P_s| - |P_{s-1}|)$.

We can also count N_s starting from $Z^{\bar{a}}/Z_i \in P_{s-1} \setminus P_{s-2}$ and then choosing Z_i, b_i . Observe that there are at most $s + d - 1$ choices for the pair Z_i, b_i given $Z^{\bar{a}}/Z_i$. Hence

$$s(|P_s| - |P_{s-1}|) \leq (s + d - 1)(|P_{s-1}| - |P_{s-2}|).$$

At $s = 1$, the same count gives $|P_1| - |P_0| \leq d|P_0|$. It follows that

$$\frac{|P_s| - |P_{s-1}|}{\binom{d+s-1}{s}} \geq \frac{|P_t| - |P_{t-1}|}{\binom{d+t-1}{t}} \quad (1 \leq s \leq t),$$

and $|P_0|$ is at least the expression on the right. Therefore

$$\begin{aligned} |P_t| &= |P_0| + \sum_{s=1}^t (|P_s| - |P_{s-1}|) \\ &\geq \frac{|P_t| - |P_{t-1}|}{\binom{d+t-1}{t}} \left(\sum_{s=0}^t \binom{d+s-1}{d-1} \right) = \frac{|P_t| - |P_{t-1}|}{\binom{d+t-1}{t}} \binom{d+t}{d} \\ &= \frac{d+t}{d} (|P_t| - |P_{t-1}|). \end{aligned} \quad \square$$

6.1.1 Interpolate a new relation or branch

The goal is now to interpolate a new relation $Q \notin R$.

Lemma 6.6. *Let $t = \lfloor 2d/\varepsilon \rfloor$, and let $R \subseteq S_t$ be ready. Then one of the following holds:*

- We find $Q \in S_t \setminus R$ such that $Q(c_1, \dots, c_d) = 0$ whenever $(c_1, \dots, c_d) \in Z(R)$ and $\text{Enc}(c_1, \dots, c_d) \in \mathcal{L}$.
- $\mathbb{F} + R = S_t$.
- There are at least k determined coordinates.
- There exists $j \in J$ such that

$$\dim(\ell_j W_{t-1} \cap R) \geq \frac{\varepsilon |P_t|}{2}.$$

Proof. First check the two terminal cases. If $\mathbb{F} + R = S_t$, the linear relations fix every coordinate of a possible common zero. Solve these relations and test the resulting tuple against R and the agreement requirement. If at least k coordinates are determined, interpolate the unique polynomial of degree less than k agreeing there, then check whether it belongs to $\text{Enc}(Z(R)) \cap \mathcal{L}$. Either case returns at most one message polynomial.

Otherwise, let $D = \lfloor \varepsilon n \rfloor + 1$ and interpolate a nonzero polynomial

$$E(X, Z_1, \dots, Z_d) = \sum_{m \in P_t} E_m(X) m,$$

such that $\deg_X E_m < D$ for all $m \in P_t$, with constraints

$$\text{for every } j \in J, \quad E(\alpha_j, \vec{Z}) \in (\ell_j W_{t-1} + R) \cap W_t.$$

The polynomials E_m have $D|P_t|$ unknown field coefficients. The membership conditions give a homogeneous linear system with

$$|J| |P_t| - \sum_{j \in J} \dim((\ell_j W_{t-1} + R) \cap W_t)$$

equations. For $j \in J$, the polynomial ℓ_j is nonzero. Multiplication by ℓ_j is therefore injective, and reduction along R into W_t gives

$$\dim((\ell_j W_{t-1} + R) \cap W_t) = \dim W_{t-1} - \dim(\ell_j W_{t-1} \cap R).$$

Thus the number of equations is at most

$$\begin{aligned} n(|P_t| - |P_{t-1}|) + \sum_{j \in J} \dim(\ell_j W_{t-1} \cap R) \\ \leq n \frac{d}{d+t} |P_t| + \sum_{j \in J} \dim(\ell_j W_{t-1} \cap R). \end{aligned}$$

Our choice $t = \lfloor 2d/\varepsilon \rfloor$ gives $d/(d+t) < \varepsilon/2$.

Case 1.

$$\frac{\varepsilon n}{2} |P_t| > \sum_{j \in J} \dim(\ell_j W_{t-1} \cap R).$$

The number of equations is strictly less than $\varepsilon n |P_t|$, whereas the number of unknown field coefficients is

$$D|P_t| > \varepsilon n |P_t|.$$

We can therefore find a nonzero E by solving this homogeneous linear system. If $(c_1, \dots, c_d) \in Z(R)$ satisfies

$$\sum_i c_i f_i(\alpha_j) = y_j \quad \text{and} \quad \ell_j \notin R,$$

then $E(\alpha_j, c_1, \dots, c_d) = 0$. Since fewer than k coordinates are determined, either

$$\begin{aligned} & |\{j \in [n] : \ell_j \in R\} \cup \{j \in [n] : E(\alpha_j, c_1, \dots, c_d) = 0\}| \\ & \leq |\{j \in [n] : \ell_j \in R\}| + (D-1) \\ & \leq k + D - 2 < k + \varepsilon n, \end{aligned}$$

or $E(X, c_1, \dots, c_d) = 0$ identically. Therefore, for every $f \in \text{Enc}(Z(R)) \cap \mathcal{L}$, $E(X, c_1, \dots, c_d) = 0$. Write

$$E = \sum_{j=0}^{D-1} r_j X^j.$$

Every coefficient r_j vanishes at every $c \in Z(R)$ with $\text{Enc}(c) \in \mathcal{L}$. Moreover, $r_j \in W_t$, and at least one r_j is nonzero because $E \neq 0$. Since $W_t \cap R = \{0\}$, any such nonzero coefficient is a new relation outside R . Replace R with

$$\text{ready}(\text{span}(R \cup \{r_0, \dots, r_{D-1}\})).$$

Case 2. The Case 1 inequality fails. Since $|J| \leq n$, averaging gives $j \in J$ such that

$$\dim(\ell_j W_{t-1} \cap R) \geq \frac{\varepsilon |P_t|}{2}.$$

Branch into

$$\text{ready}(\text{span}(R \cup \{\ell_j\})).$$

and

$$\text{ready}(\text{span}(R \cup \{p \in W_{t-1} : p\ell_j \in R\})).$$

These branches cover every $c \in Z(R)$. If $\ell_j(c) = 0$, c lies in the first branch. Otherwise, each relation $p\ell_j \in R$ forces $p(c) = 0$, so it lies in the second branch. Taking spans and making the spaces *ready* preserves these conclusions. $\square$

6.1.2 Bounding the Output List-Size

Claim 6.7. *There are $|P_1| - 1$ remaining variables, and*

$$|\mathcal{L} \cap \text{Enc}(Z(R))| \leq 2^{|P_1|-1} |P_t|.$$

Proof. Since $1 \notin R$, the constant monomial is remaining. Row reduction of the affine linear relations expresses the other pivot monomials of degree one in terms of the remaining variables. Thus their number is $|P_1| - 1$.

We induct on $|P_t|$. The terminal cases return at most one message polynomial, which satisfies the claimed bound. In particular, if $|P_t| = 1$, then $R + \mathbb{F} = S_t$, so this is a terminal case. If a new relation is found, adding it strictly decreases $|P_t|$ and cannot increase the number of remaining variables. The induction hypothesis then gives the bound.

Otherwise, use the two branches from Case 2. The first adds the independent affine linear relation $\ell_j \notin R$, so it removes at least one remaining variable unless it contains 1 and is discarded. Once made *ready*, it contains $R + \ell_j W_{t-1}$. Its remaining dimension is therefore at most

$$|P_t| - |P_{t-1}| + \dim(\ell_j W_{t-1} \cap R).$$

The space $\{p \in W_{t-1} : p\ell_j \in R\}$ has dimension $\dim(\ell_j W_{t-1} \cap R)$, because multiplication by ℓ_j is injective. It meets R only in zero. Thus the second branch has remaining dimension at most

$$|P_t| - \dim(\ell_j W_{t-1} \cap R).$$

Both dimensions are strictly less than $|P_t|$: the first branch adds $\ell_j \notin R$, and the second adds a space of positive dimension disjoint from R .

By the branch coverage proved above and the induction hypothesis, the total list size is at most

$$\begin{aligned}
& 2^{|P_1|-2}(|P_t| - |P_{t-1}| + \dim(\ell_j W_{t-1} \cap R)) \\
& + 2^{|P_1|-1}(|P_t| - \dim(\ell_j W_{t-1} \cap R)) \\
& = 2^{|P_1|-1}|P_t| \\
& + 2^{|P_1|-2}(|P_t| - |P_{t-1}| - \dim(\ell_j W_{t-1} \cap R)) \\
& \leq 2^{|P_1|-1}|P_t|,
\end{aligned}$$

since Claim 6.5 and the choice of t give

$$|P_t| - |P_{t-1}| \leq \frac{d}{d+t}|P_t| < \frac{\varepsilon}{2}|P_t| \leq \dim(\ell_j W_{t-1} \cap R). \quad \square$$

Initially $R = \{0\}$ and $P_t = M_t$, so the list size is at most

$$2^d|M_t| = 2^d \binom{d+t}{d} \leq (O(1/\varepsilon))^d.$$

Running time. Observe that checking whether $j \in J$ by row reduction takes $\text{poly}(|M_t|, \log |\mathbb{F}|)$ time, given the values $f_i(\alpha_j)$. Thus computing J takes $n \text{ poly}(|M_t|, \log |\mathbb{F}|)$ time. Similarly, for each $j \in J$, we can compute $\dim(\ell_j W_{t-1} \cap R)$ and a basis for $\{p \in W_{t-1} : p\ell_j \in R\}$ in $\text{poly}(|M_t|, \log |\mathbb{F}|)$ time. Replacing R with $\text{ready}(R)$ has the same cost. The basis evaluations, terminal interpolation, and agreement checks take $\tilde{O}(n) \text{ poly}(|M_t|, \log |\mathbb{F}|)$ time using fast univariate evaluation and interpolation.

It remains to interpolate E within this bound when

$$\frac{\varepsilon n}{2}|P_t| > \sum_{j \in J} \dim(\ell_j W_{t-1} \cap R).$$

We use the following specialization of the fast interpolation theorem.

Theorem 6.8 (Fast interpolation [JNSV17, Theorem 1.5]). *Let $\mathbb{F}$ be a finite field, let $u \geq 1$ and $v \geq 0$ be integers, and let $\beta_1, \dots, \beta_v \in \mathbb{F}$ and $a_1, \dots, a_v \in \mathbb{F}^u$. The points β_h need not be distinct. There is a deterministic algorithm that finds a nonzero tuple $(q_1, \dots, q_u) \in \mathbb{F}[X]^u$ satisfying*

$$\sum_{i=1}^u a_{h,i} q_i(\beta_h) = 0 \quad (h \in [v])$$

and minimizing $\max_i \deg q_i$ among all such tuples, using $\tilde{O}(u^2(v+u))$ field operations.

For each $j \in J$, row reduction writes the membership condition

$$E(\alpha_j, \vec{Z}) \in (\ell_j W_{t-1} + R) \cap W_t$$

as $c_j \leq |P_t|$ equations

$$\sum_{m \in P_t} a_{j,h,m} E_m(\alpha_j) = 0 \quad (h \in [c_j]),$$

with coefficients $a_{j,h,m} \in \mathbb{F}$. Apply Theorem 6.8 with $u = |P_t|$, $v = \sum_{j \in J} c_j \leq n|P_t|$, and one copy of α_j for each of its c_j equations. Case 1 of Lemma 6.6 gives a nonzero solution with $\deg E_m < D$, so

the minimum-degree solution has this property. The cost is $\tilde{O}(n|P_t|^3)$ field operations. Extracting the coefficient relations and adjoining them to R also takes $n \text{ poly}(|M_t|, \log |\mathbb{F}|)$ time.

The induction in Claim 6.7 also bounds the number of terminal nodes by $2^d |M_t|$, whether or not they return a message polynomial; children containing 1 are discarded at their parent. Every path has at most $|M_t|$ strict enlargements of R , so there are at most $2^d |M_t|^2$ steps. Since $|M_t| = (O(1/\varepsilon))^d$, the total running time is

$$(O(1/\varepsilon))^{O(d)} \cdot \tilde{O}(n \text{ poly}(1/\varepsilon, d, \log |\mathbb{F}|)).$$

This proves Theorem 6.1.

6.2 Pruning $\mathbb{K}(X)$ -subspaces (Theorem 5.3)

We follow the proof of Theorem 6.1, now over $\mathbb{L} = \mathbb{K}(X)$. The coefficients $c_i(X)$ representing a message polynomial lie in $\mathbb{K}[X]$, and the relations have coefficients in $\mathbb{L}$. We still try to add a new relation, and branch when interpolation fails. The additional work is to control coefficient degrees and to pass from relations at an evaluation point to relations over $\mathbb{L}$.

A space of rational dimension zero contains only the zero polynomial. We use d for the actual rational dimension in this proof; applying the bound with this dimension implies the bound in the theorem.

In the previous subsection, choosing a basis gave constant coefficients $c_i \in \mathbb{F}$. Here we use a reduced polynomial basis to obtain coefficients $c_i(X) \in \mathbb{K}[X]$ with controlled degrees. To do this, first work in the polynomial module

$$\mathcal{A} = A \cap \mathbb{F}[X].$$

Lemma 6.9 (Reduced polynomial basis (Corollary 2.16)). *The module $\mathcal{A}$ has a $\mathbb{K}[X]$ -basis $f_1, \dots, f_d$ such that*

$$\deg \left(\sum_{i=1}^d c_i(X) f_i(X) \right) = \max_{i \in [d]} (\deg c_i + \deg f_i).$$

In particular,

$$\mathcal{A} \cap \mathbb{F}[X]_{<k} = \left\{ \sum_{i=1}^d c_i(X) f_i(X) : c_i \in \mathbb{K}[X]_{<k - \deg f_i} \right\}.$$

Additionally, given a basis $g_1, \dots, g_d$ of A , $f_1, \dots, f_d$ can be computed in $\text{poly}(d, \max \deg g_i, n, \log |\mathbb{F}|)$ time.

Proof. The basis properties follow from Corollary 2.16. For computation, put $D = \max_i \deg g_i$ and identify $\mathbb{F}[X]$ with $\mathbb{K}[X]^{\mathbb{F}:\mathbb{K}}$. Linear algebra over $\mathbb{K}(X)$, followed by comparison of coefficients, computes a $\mathbb{K}$ -basis of $A \cap \mathbb{F}[X]_{\leq D}$. The predictable-degree property implies that every element of a reduced module basis has degree at most D , since the g_i span A over $\mathbb{K}(X)$. Thus the computed polynomials generate $\mathcal{A}$ over $\mathbb{K}[X]$, and we apply Theorem 2.15. $\square$

Notation. For $c = (c_1, \dots, c_d) \in \mathbb{L}^d$, write $\text{Enc}(c) = \sum_i c_i f_i$. If $\text{Enc}(c) \in \mathbb{F}[X]_{<k}$, then $c_i \in \mathbb{K}[X]_{<k - \deg f_i}$; in particular, every c_i has degree less than k . Set

$$t = \lfloor 16d/\varepsilon \rfloor, \quad M_s = \{Z^a : \text{wt}(a) \leq s\}, \quad S_s = \text{span}_{\mathbb{L}}(M_s).$$

We maintain an $\mathbb{L}$ -linear relation space $R \subseteq S_t$, starting with $R = \{0\}$, and discard a branch if $1 \in R$. Here $Z(R)$ denotes the common zeros of R in $\mathbb{L}^d$. For a polynomial tuple $c(X)$, membership in $Z(R)$ means that each relation vanishes identically after substituting $Z_i = c_i(X)$.

The definition of *ready* is unchanged, with spans now taken over $\mathbb{L}$. We use the same monomial order, pivot monomials, and remaining monomials P_s , and put $W_s = \text{span}_{\mathbb{L}}(P_s)$. In particular,

$$S_s = (R \cap S_s) + W_s, \quad (R \cap S_s) \cap W_s = \{0\}.$$

[Claim 6.5](#) used only row reduction and closure under multiplication. Its proof therefore applies over $\mathbb{L}$ and gives

$$|P_t| - |P_{t-1}| \leq \frac{d}{d+t} |P_t| < \frac{\varepsilon}{16} |P_t|. \quad (6.1)$$

Definition 6.10. Store R using a row-echelon basis over $\mathbb{L}$, with denominators cleared in each row. Thus its stored matrix has entries in $\mathbb{K}[X]$, and its pivot entries need not be 1. Write $\deg R$ for the largest X -degree of the entries in this representation, with $\deg\{0\} = 0$. We say that R is *H-prepared* if it is *ready* and has such a representation of degree at most H .

Preparation plays the role of making the relation space *ready* in the previous subsection, but also controls the degrees needed for evaluation. It only needs to preserve the polynomial tuples under consideration, rather than every zero in $\mathbb{L}^d$.

Lemma 6.11 (Preparation). *Let $B(d, t) = |M_t|t$. Given $R \subseteq S_t$, we can compute $R' \supseteq R$ with the following properties:*

- R' is $B(d, t)k$ -prepared.
- It preserves the polynomial zeros under consideration:

$$Z(R') \cap (\mathbb{K}[X]_{<k})^d = Z(R) \cap (\mathbb{K}[X]_{<k})^d.$$

- The computation takes time $\text{poly}(n, \deg R, |M_t|, \log |\mathbb{F}|)$.

Proof. For $c \in Z(R) \cap (\mathbb{K}[X]_{<k})^d$ and $r \in R$,

$$\sum_{m \in M_t} r_m(X) m(c) = 0, \quad \deg m(c) < tk.$$

Let

$$U = \text{span}_{\mathbb{L}} \{u \in (\mathbb{K}[X]_{<tk})^{M_t} : u \cdot r = 0 \text{ for every } r \in R\}, \quad R' = U^\perp.$$

Then $R \subseteq R'$. For every such c , the vector $(m(c))_{m \in M_t}$ lies in U , so $c \in Z(R')$. The reverse inclusion follows from $R \subseteq R'$, proving preservation of the stated polynomial zeros. Computing the bounded-degree solutions by coefficient comparison and selecting an $\mathbb{L}$ -basis takes $\text{poly}(\log |\mathbb{K}|, tk, \deg R, |M_t|)$ time.

Write this basis as the rows of a matrix V with $D = \dim_{\mathbb{L}} U$ rows and entries of degree less than tk . In the fixed monomial order, choose independent columns greedily from right to left. Let $\mathcal{I}$ be their indices and put $\mathcal{J} = M_t \setminus \mathcal{I}$. For each $j \in \mathcal{J}$, solve

$$V_{\mathcal{I}} f_j = -V_j, \quad f_j \in \mathbb{L}^{\mathcal{I}}.$$

The vector with coordinate 1 at j , coordinates as f_j on $\mathcal{I}$, and zero elsewhere lies in $U^\perp$. Every nonzero coordinate of f_j is to the right of j , so these vectors, ordered by j , form a row-echelon basis of R' . Multiplying each row by $\det V_{\mathcal{I}}$ converts entries from $\mathbb{K}(X)$ elements to $\mathbb{K}[X]$. By Cramer's rule, the resulting entries have degree at most $Dtk \leq |M_t|tk$, giving the required degree bound.

If R' is not ready, adjoin a product $Z_i r \notin R'$ of a stored row $r \in R' \cap S_{t-1}$ and repeat the construction. This preserves the stated polynomial zeros and strictly increases the dimension. The added row has X -degree at most $B(d, t)k$, so each repetition takes $\text{poly}(|M_t|, tk, \log |\mathbb{F}|)$ time. After at most $|M_t|$ repetitions we obtain a $B(d, t)k$ -prepared space in $\text{poly}(n, \deg R, |M_t|, \log |\mathbb{F}|)$ time. $\square$

Notation. Write

$$\delta_j = [\mathbb{K}(\alpha_j) : \mathbb{K}], \quad \delta_{\min} = \min_j \delta_j, \quad \delta_{\max} = \max_j \delta_j.$$

The hypotheses give $\delta_{\min} \geq C(d, \varepsilon)$ and $\delta_{\max} \leq 2\delta_{\min}$.

Proposition 6.12. *For every nonzero $g \in \mathbb{K}[X]$,*

$$\sum_{j: g(\alpha_j)=0} \delta_j \leq \deg g.$$

Proof. If $g(\alpha_j) = 0$, the minimal polynomial of α_j over $\mathbb{K}$ divides g . Non-conjugacy means that these minimal polynomials are distinct irreducible polynomials. Their product therefore divides g , giving the inequality. $\square$

Notation. Let $R(\alpha_j)$ be the $\mathbb{K}(\alpha_j)$ -span of the evaluations of the stored polynomial rows of R , and let $W_s(\alpha_j) = \text{span}_{\mathbb{K}(\alpha_j)}(P_s)$. Let $\mathcal{B}$ be the set of coordinates at which some pivot entry of the stored matrix vanishes. Coordinates outside $\mathcal{B}$ are called regular.

Claim 6.13. *If the stored matrix has degree at most H , then*

$$|\mathcal{B}| \leq \frac{|M_t|H}{\delta_{\min}}.$$

At every regular coordinate, the pivot monomials are unchanged, and $W_s(\alpha_j)$ is a complement of $R(\alpha_j) \cap \mathbb{K}(\alpha_j)[\vec{Z}]_{\leq s}$.

Proof. The product of the pivot entries is a nonzero polynomial of degree at most $(\dim R)H \leq |M_t|H$. It vanishes at every coordinate in $\mathcal{B}$, so Proposition 6.12 gives the bound. At any other coordinate, the evaluated matrix is still in row-echelon form with the same pivots. The complement statement then follows by row reduction, just as before. $\square$

For the estimates below, choose $C(d, \varepsilon)$ large enough that

$$C(d, \varepsilon) \geq \frac{16|M_t|^2 t}{\varepsilon}. \tag{6.2}$$

At a prepared node, $H \leq B(d, t)k$, so $k \leq n$ gives $|\mathcal{B}| \leq \varepsilon n/16$.

The single equation ℓ_j from the previous subsection is replaced by a space of affine linear equations. This is because $c_i(\alpha_j) \in \mathbb{K}(\alpha_j)$, whereas $f_i(\alpha_j)$ and y_j lie in $\mathbb{F}$.

Definition 6.14. Expand the equation

$$\sum_{i=1}^d f_i(\alpha_j) Z_i - y_j = 0$$

in a basis of $\mathbb{F}$ over $\mathbb{K}(\alpha_j)$, and let L_j be the $\mathbb{K}(\alpha_j)$ -span of the resulting affine linear forms. Thus for $z \in \mathbb{K}(\alpha_j)^d$, all forms in L_j vanish at z if and only if $\sum_i f_i(\alpha_j) z_i = y_j$. We say that j is determined if $L_j \subseteq R(\alpha_j)$, and put $J = \{j : L_j \not\subseteq R(\alpha_j)\}$.

A polynomial tuple $c \in Z(R)$ specializes to a zero of $R(\alpha_j)$, so $\text{Enc}(c)$ agrees with y at each determined coordinate. Thus k determined coordinates leave at most one message polynomial of degree less than k .

6.2.1 Finding a new relation

We now repeat the interpolation argument of [Lemma 6.6](#). At a regular undetermined coordinate, choose any $\ell_j \in L_j \setminus R(\alpha_j)$ and put

$$a_j = \dim_{\mathbb{K}(\alpha_j)}(\ell_j W_{t-1}(\alpha_j) \cap R(\alpha_j)).$$

The interpolation count uses them in exactly the same way as before.

Lemma 6.15 (Interpolation). *Suppose R is $B(d, t)k$ -prepared and [\(6.2\)](#) holds. Then one of the following holds:*

- We find $E \in S_t \setminus R$ vanishing identically at every tuple $c(X) \in Z(R)$ with $\text{Enc}(c) \in \mathcal{L}$.
- $\mathbb{L} + R = S_t$.
- There are at least k determined coordinates.
- We have

$$\sum_{j \in J \setminus \mathcal{B}} a_j \geq \frac{\varepsilon n}{8} |P_t|.$$

Proof. The two terminal cases are handled as in [Lemma 6.6](#). If $\mathbb{L} + R = S_t$, solve for the unique possible tuple over $\mathbb{L}$, then check whether it encodes a nearby message polynomial. If at least k coordinates are determined, interpolate and test the unique possible message. Henceforth assume neither case holds.

Set

$$D = \left\lfloor \frac{\delta_{\min} \varepsilon n}{2} \right\rfloor + 1$$

and seek a nonzero polynomial

$$E(X, \vec{Z}) = \sum_{m \in P_t} E_m(X) m, \quad E_m \in \mathbb{K}[X]_{< D}.$$

At every $j \in J \setminus \mathcal{B}$, impose

$$E(\alpha_j, \vec{Z}) \in (\ell_j W_{t-1}(\alpha_j) + R(\alpha_j)) \cap W_t(\alpha_j).$$

At every $j \in J \cap \mathcal{B}$, impose the stronger condition $E(\alpha_j, \vec{Z}) = 0$. This avoids needing a complement at the exceptional coordinates.

There are $D|P_t|$ unknown coefficients over $\mathbb{K}$. At a regular coordinate, multiplication by the nonzero form ℓ_j is injective. As in [Lemma 6.6](#), the membership condition imposes $|P_t| - |P_{t-1}| + a_j$ equations over $\mathbb{K}(\alpha_j)$, hence at most $\delta_j(|P_t| - |P_{t-1}| + a_j)$ equations over $\mathbb{K}$. At an exceptional coordinate, there are at most $\delta_j|P_t|$ equations over $\mathbb{K}$. The total is therefore at most

$$\delta_{\max} \left(n(|P_t| - |P_{t-1}|) + \sum_{j \in J \setminus \mathcal{B}} a_j + |\mathcal{B}| |P_t| \right).$$

If the last outcome of the lemma does not hold, this is strictly less than

$$2\delta_{\min} \left(\frac{\varepsilon n}{16} + \frac{\varepsilon n}{8} + \frac{\varepsilon n}{16} \right) |P_t| = \frac{\delta_{\min} \varepsilon n}{2} |P_t| < D|P_t|.$$

Thus the homogeneous linear system has a nonzero solution.

For $c(X) \in Z(R)$ with $\text{Enc}(c) \in \mathcal{L}$, consider $E(X, c_1(X), \dots, c_d(X)) \in \mathbb{K}[X]$. Its degree is at most

$$D - 1 + t(k - 1) \leq \frac{\delta_{\min} \varepsilon n}{2} + tn \leq \frac{3\delta_{\min} \varepsilon n}{4}.$$

At every agreement coordinate in J , this polynomial vanishes: at a regular coordinate, both ℓ_j and $R(\alpha_j)$ vanish at $c(\alpha_j)$; at an exceptional coordinate, we imposed $E(\alpha_j, \vec{Z}) = 0$. Fewer than k coordinates are determined, so there are more than εn such roots. [Proposition 6.12](#) forces $E(X, c_1(X), \dots, c_d(X)) = 0$ identically.

Finally, E is a nonzero element of W_t , so $E \notin R$. We add E and prepare the enlarged space using [Lemma 6.11](#). Here we add E itself as a relation over $\mathbb{L}$. Since the c_i depend on X , its individual coefficients in X need not vanish after substitution, as they did in the previous subsection. $\square$

6.2.2 From interpolation failure to branching

We now use the last outcome of [Lemma 6.15](#) to find either a new relation or a branching form over $\mathbb{L}$. First, we present two lemmas that will be useful in the proof of our key theorem.

Lemma 6.16 (Effective Nullstellensatz [[Kol88](#), Corollary 1.9]). *Let E be a field, let $a, D \geq 1$, and let $f_1, \dots, f_s \in E[Y_1, \dots, Y_a]$ satisfy $\deg f_i \leq D$. If the f_i have no common zero over $\bar{E}$, then there are $g_1, \dots, g_s \in E[Y_1, \dots, Y_a]$ such that*

$$\sum_{i=1}^s g_i f_i = 1, \quad \deg(g_i f_i) \leq 2(3D)^a \quad (1 \leq i \leq s).$$

Lemma 6.17 (Computing a common zero). *Let $\mathbb{K}$ be a finite field and $\mathbb{L} = \mathbb{K}(X)$. Given $f_1, \dots, f_s \in \mathbb{K}[X, Y_1, \dots, Y_a]$ with $\deg_Y f_j \leq D$ and $\deg_X f_j \leq H$, one can deterministically decide whether they have a common zero over $\bar{\mathbb{L}}$ and, if so, compute a finite extension $E/\mathbb{L}$, given by defining polynomials and an $\mathbb{L}$ -basis, and a common zero $y \in E^a$, in time*

$$O_{s,D,a}(1) \text{poly}(H, \log |\mathbb{K}|).$$

We prove [Lemma 6.17](#) in [Appendix A](#).

Theorem 6.18 (Finding a relation or a branching form). *Suppose $R \subsetneq S_t$ is $B(d, t)k$ -prepared and*

$$\sum_{j \in J \setminus \mathcal{B}} a_j \geq \frac{\varepsilon n}{8} |P_t|.$$

Then we can compute one of the following:

- A relation $Q \in S_t \setminus R$ vanishing on $Z(R)$.
- An affine linear form $\ell \in S_1 \setminus R$, over $\mathbb{L}$, such that

$$\dim_{\mathbb{L}}(\ell W_{t-1} \cap R) \geq \frac{\varepsilon}{16} |P_t|.$$

Proof. Put $b = |P_t|$, $s = |P_{t-1}|$, and $r = \lceil \varepsilon b / 16 \rceil$. First we show that, over some finite extension $E/\mathbb{L}$, there is a nonzero $\ell \in \text{span}_E P_1$ for which

$$\dim_E \{p \in \text{span}_E P_{t-1} : \ell p \in \text{span}_E R\} \geq r. \quad (6.3)$$

Let

$$T = \{j \in J \setminus \mathcal{B} : a_j \geq r\}.$$

For $j \in (J \setminus \mathcal{B}) \setminus T$, we have $a_j \leq r - 1 < \varepsilon b/16$, and always $a_j \leq b$. Hence

$$\frac{\varepsilon nb}{8} \leq \sum_{j \in J \setminus \mathcal{B}} a_j \leq \frac{\varepsilon nb}{16} + b|T|, \quad |T| \geq \frac{\varepsilon n}{16}.$$

In particular $r \leq s$.

Without loss of generality, $P_1 = \{1, Z_1, \dots, Z_a\}$. For $Y = (Y_0, \dots, Y_a) \in \mathbb{L}^{a+1}$, let $A(Y) \in \mathbb{L}^{b \times s}$ be the matrix, in the bases P_{t-1} and P_t , of the linear map

$$A_Y : W_{t-1} \longrightarrow W_t, \quad A_Y(p) = \left(Y_0 + \sum_{i=1}^a Y_i Z_i \right) p \pmod{R}.$$

Here the remainder modulo R is taken in W_t , so the entries of $A(Y)$ are linear in Y . Let $\Pi(X)$ be the product of the pivot entries of the stored row-echelon matrix for R . Cramer's rule shows that multiplying $A(Y)$ by $\Pi(X)$ clears its denominators. The entries of $\Pi(X)A(Y)$ lie in $\mathbb{K}[X, Y]$, are linear in Y , and have X -degree at most

$$|M_t|B(d, t)k = |M_t|^2 tk.$$

At a regular coordinate j , $\Pi(\alpha_j) \neq 0$, so evaluating $A(Y)$ at $X = \alpha_j$ gives the multiplication matrix modulo $R(\alpha_j)$. Moreover, $R(\alpha_j)$ is ready, so replacing ℓ_j by its nonzero remainder in $W_1(\alpha_j)$ does not change a_j .

First assume that there is no nonzero ℓ satisfying (6.3) over any finite extension of $\mathbb{L}$. A nonzero coefficient vector Y gives such a form exactly when $A(Y)$ has rank at most $s - r$, or equivalently, when all $(s - r + 1) \times (s - r + 1)$ minors of $\Pi(X)A(Y)$ vanish. These minors therefore have no common zero in $\overline{\mathbb{L}}^{a+1} \setminus \{0\}$. Their Y -degree is at most $|M_t|$ and their X -degree is at most $|M_t|^2 B(d, t)k$.

For each $i \in \{0, \dots, a\}$, set $Y_i = 1$ and denote the resulting nonzero minors by $F_{\lambda, i}$. They have no common zero, so Lemma 6.16 gives polynomials $G_{\lambda, i}$ over $\mathbb{L}$ such that

$$\sum_{\lambda} G_{\lambda, i} F_{\lambda, i} = 1, \quad \deg_Y(G_{\lambda, i} F_{\lambda, i}) \leq 2(3|M_t|)^d.$$

Equating coefficients of monomials in Y gives a linear system for the coefficients of the $G_{\lambda, i}$ over $\mathbb{L}$, with at most

$$V = \binom{d + 2(3|M_t|)^d}{d}$$

rows and entries in $\mathbb{K}[X]$ of degree at most $|M_t|^2 B(d, t)k$. As in the proof of Lemma 6.11, Cramer's rule gives a solution with a common nonzero denominator $D_i \in \mathbb{K}[X]$ of degree at most $V|M_t|^2 B(d, t)k$. Clearing denominators gives $\tilde{G}_{\lambda, i} \in \mathbb{K}[X, Y]$ such that

$$D_i(X) = \sum_{\lambda} \tilde{G}_{\lambda, i}(X, Y) F_{\lambda, i}(X, Y).$$

Consequently $D = \prod_{i=0}^a D_i$ is nonzero and

$$\deg D = \sum_{i=0}^a \deg D_i \leq (d + 1)V|M_t|^2 B(d, t)k.$$

For every $j \in T$, the remainder of ℓ_j is nonzero and its multiplication matrix has kernel dimension at least r . All these minors therefore vanish at that coefficient vector. Rescaling a nonzero coefficient Y_i to 1 gives $D_i(\alpha_j) = 0$, and hence $D(\alpha_j) = 0$.

It suffices to take

$$C(d, \varepsilon) \geq \frac{32|M_t|B(d, t)}{\varepsilon} (1 + (d+1)|M_t|V). \quad (6.4)$$

Indeed, [Proposition 6.12](#) and $k \leq n$ would then imply

$$|T| \leq \frac{\deg D}{\delta_{\min}} \leq \frac{\varepsilon n}{32}.$$

This contradicts $|T| \geq \varepsilon n/16$. Thus the minor equations with $Y_i = 1$ have a common zero for some i . Apply [Lemma 6.17](#) to these systems to compute a finite extension $E/\mathbb{L}$ and $y = (y_0, \dots, y_a) \in E^{a+1}$ satisfying the minor equations, with $y_i = 1$ for some i . Set

$$\ell = y_0 + \sum_{j=1}^a y_j Z_j.$$

Then $\ell \neq 0$ and satisfies [\(6.3\)](#).

Let $e_1, \dots, e_m$ be the computed $\mathbb{L}$ -basis of E . By row reduction, compute an E -basis $p_1, \dots, p_v$ of the kernel in [\(6.3\)](#). Write

$$\ell = \sum_{u=1}^m e_u \ell_u, \quad p_i = \sum_{u=1}^m e_u p_{i,u}, \quad \ell_u \in W_1, \quad p_{i,u} \in W_{t-1},$$

and put $U = \text{span}_{\mathbb{L}}\{p_{i,u} : i \in [v], u \in [m]\}$. Since $\text{span}_E U$ contains $p_1, \dots, p_v$,

$$\dim_{\mathbb{L}} U = \dim_E \text{span}_E U \geq v \geq r.$$

For every $c \in Z(R)$, we have $\ell(c)p_i(c) = 0$ for all i . If $\ell(c) = 0$, then $\ell_u(c) = 0$ for every u . Otherwise $p_{i,u}(c) = 0$ for every i, u . Thus every product $\ell_u p_{i,w}$ vanishes on $Z(R)$.

Check by row reduction whether any such product lies outside R ; if so, return it as Q . Otherwise $\ell_u U \subseteq R$ for every u . Choose a nonzero $\ell_u \in W_1$. Then $\ell_u \notin R$, and multiplication by ℓ_u is injective, so

$$\dim_{\mathbb{L}}(\ell_u W_{t-1} \cap R) \geq \dim_{\mathbb{L}}(\ell_u U) = \dim_{\mathbb{L}} U \geq r \geq \frac{\varepsilon}{16}|P_t|.$$

Thus, we can use ℓ_u for branching.

The main cost of the algorithm is computing ℓ . Each system has at most d variables and $4^{|M_t|}$ equations, of Y -degree at most $|M_t|$ and X -degree at most $|M_t|^2 B(d, t)k$. By [Lemma 6.17](#), we can compute ℓ deterministically in time

$$O_{d,\varepsilon}(1) \text{poly}(B(d, t)k, \log |\mathbb{F}|).$$

Thus, the algorithm can be implemented in time $O_{d,\varepsilon}(1) \text{poly}(n, \log |\mathbb{F}|)$. $\square$

We can now branch into

$$R \cup \{\ell_u\} \quad \text{and} \quad R \cup \{p \in W_{t-1} : p\ell_u \in R\}.$$

We then prepare the resulting spaces and repeat until a single possible message polynomial remains to be checked, as before.

6.2.3 Bounding the list and the running time

The induction in Claim 6.7 applies, since at every branching step,

$$|P_t| - |P_{t-1}| < \frac{\varepsilon}{16} |P_t| \leq \dim_{\mathbb{L}}(\ell W_{t-1} \cap R).$$

Preparation only enlarges the relation spaces and preserves the polynomial tuples under consideration. Thus

$$|\mathcal{L} \cap \text{Enc}(Z(R))| \leq 2^{|P_1|-1} |P_t|.$$

Initially this gives a list of size at most

$$2^d |M_t| = 2^d \binom{d+t}{d} \leq (O(1/\varepsilon))^d.$$

Note that the choice in (6.4) satisfies $C(d, \varepsilon) = \exp(O(d^3 \log(1/\varepsilon)))$. The dimension of R strictly increases along each branch, so the binary tree has depth at most $|M_t|$ and $O_{d,\varepsilon}(1)$ nodes. The main cost at each node is finding a common zero as in Theorem 6.18. The remaining operations take time $O_{d,\varepsilon}(1) \text{poly}(n, \log |\mathbb{F}|)$. Thus the total running time is

$$O_{d,\varepsilon}(1) \text{poly}(n, \log |\mathbb{F}|).$$

This proves Theorem 5.3.

7 Effective Evaluation Points and Explicit Instantiation

We begin with a simpler construction that can in fact be made strongly-explicit. This construction does not achieve effectiveness Definition 4.1, but is still sufficient for our list-decoding application with list sizes $(1/\varepsilon)^{O(1/\varepsilon)}$ at a gap of ε to capacity.

Theorem 7.1 (Strongly explicit evaluation points). *For any fixed integers $g, d \in \mathbb{N}$, for any finite field $\mathbb{K}$, there exists a deterministic algorithm such that on input $\mathbb{K}, g, d$, it runs in time $\text{poly}(g, d, \log |\mathbb{K}|)$ and outputs irreducible polynomials $P_1(X), \dots, P_g(X) \in \mathbb{K}[X]$ such that:*

- $d \leq \deg P_i(X) \leq 2 \deg P_j(X) \leq \text{poly}(d, g)$;
- $\{\deg P_i\}_{i \in [g]} \cup \{\text{char}(\mathbb{K})\}$ is a set of $g+1$ distinct primes.

Moreover, if n is an integer with $\lceil n/g \rceil \leq |\mathbb{K}|$, and $\beta_1, \dots, \beta_g$ are the roots of $P_1, \dots, P_g(X)$, then if we let $\mathbb{F} = \mathbb{K}(\beta_1, \dots, \beta_g)$, and let $S_1, \dots, S_g$ be arbitrary sets of size $\lceil n/g \rceil$ of $\beta_1 + \mathbb{K}, \dots, \beta_g + \mathbb{K}$ respectively, and let $\{\alpha_1, \dots, \alpha_n\}$ be an arbitrary subset of $S_1 \cup \dots \cup S_g$, and let $\sigma_i : x \mapsto x^{|\mathbb{K}| \frac{\prod_{j=1}^g \deg P_j}{\deg P_i}}$,

- $[\mathbb{F} : \mathbb{K}] \leq (d+g)^{O(g)}$.
- $d \leq [\mathbb{K}(\alpha_i) : \mathbb{K}] \leq 2[\mathbb{K}(\alpha_j) : \mathbb{K}]$ for all $i \neq j$
- α_i and α_j are non-conjugate.
- $\sigma_1, \dots, \sigma_g$ generate $\text{Gal}(\mathbb{F}/\mathbb{K})$, and

$$|\{i \in [n] : \sigma_j(\alpha_i) \neq \alpha_i\}| \leq \lceil n/g \rceil \quad \forall j \in [g]$$

In particular, given an index $i \in [n]$, α_i can be outputted in time $\text{poly}(\log |\mathbb{F}|)$.

Proof. We first find via brute force primes $p_1, \dots, p_g$ such that $\text{char}(\mathbb{K}) \neq p_i$ and $d \leq p_i \leq 2p_j$ for all $i \neq j$. Note that such primes exist by the prime number theorem at scale $d + \text{poly}(g)$. This takes time $\text{poly}(d, g)$.

Now, we can construct $P_1, \dots, P_g$ with $\deg P_i = p_i$ via [Corollary 2.19](#) in time $\text{poly}(\log |\mathbb{K}|, d, g)$.

Now, it suffices to show that the resulting $\alpha_1, \dots, \alpha_n$ satisfy the requisite properties. Observe that $[\mathbb{F} : \mathbb{K}] = \prod_{j=1}^g \deg P_j \leq \text{poly}(d, g)^g \leq (d + g)^{O(g)}$.

$[\mathbb{K}(\beta_i + c_i) : \mathbb{K}] = [\mathbb{K}(\beta_i) : \mathbb{K}] = p_i$ for any $c_i \in \mathbb{K}$. Thus, the second result follows. Similarly, σ_i fixes $(S_1 \cup \dots \cup S_g) \setminus S_i$ and $|S_i| \leq \lceil n/g \rceil$. $\sigma_1, \dots, \sigma_g$ clearly generate $\text{Gal}(\mathbb{F}/\mathbb{K})$ since $\sigma : x \mapsto x^{|\mathbb{K}|}$ is in the group generated by $\sigma_1, \dots, \sigma_g$ since $\gcd\left(\frac{\prod_{j=1}^g p_j}{p_1}, \dots, \frac{\prod_{j=1}^g p_j}{p_g}\right) = 1$.

Now, it suffices to prove that all pairs α_i, α_j are non-conjugate. Write $\text{Tr} = \text{Tr}_{\mathbb{F}/\mathbb{K}}$. Observe that if α_i and α_j are conjugate then they must have the same extension degree. Thus, let $\alpha_1 = \beta_i + c_1$ and $\alpha_2 = \beta_i + c_2$. Observe that if α_1, α_2 are conjugate then $\text{Tr}(\alpha_1) = \text{Tr}(\alpha_2) \implies \text{Tr}(\beta_i) + \text{Tr}(c_1) = \text{Tr}(\beta_i) + \text{Tr}(c_2)$. But, $\text{Tr}(c_1) - \text{Tr}(c_2) = (c_1 - c_2) \prod_{i=1}^g p_i \neq 0$ since $\text{char}(\mathbb{K}) \nmid p_i$ for any $i \in [g]$. Thus, the result follows. $\square$

Now, we construct the fields and effective evaluation points to prove the full subspace-design condition.

We remind ourselves that the conditions we want are as in [Definition 4.1](#).

Theorem 7.2 (Explicit effective points). *For any fixed integers $r \geq 2, g \geq 2^r, d \geq 1$, and integer $n \geq g$, for any finite field $\mathbb{K}$ with $|\mathbb{K}| > nr$, there exists an explicit extension field $\mathbb{F}/\mathbb{K}$ with $[\mathbb{F} : \mathbb{K}] \leq (d + g)^{O(g)}$ and points $\alpha_1, \dots, \alpha_n$ with automorphisms $\sigma_1, \dots, \sigma_g$ such that:*

- $d \leq [\mathbb{K}(\alpha_i) : \mathbb{K}] \leq 2[\mathbb{K}(\alpha_j) : \mathbb{K}]$ for all $i \neq j$
- α_i and α_j are non-conjugate.
- $\sigma_1, \dots, \sigma_g$ generate $\text{Gal}(\mathbb{F}/\mathbb{K})$, and

$$|\{i \in [n] : \sigma_j(\alpha_i) \neq \alpha_i\}| \leq \lceil n/g \rceil \quad \forall j \in [g]$$

- For every non-zero $\mathbb{K}$ -subspace $\mathcal{Q} \subseteq \mathbb{K}[X]_{<nr}$ of dimension at most 2^r ,

$$\sum_{i=1}^n \max_{0 \neq q \in \mathcal{Q}} \text{mult}_{\alpha_i} q \leq \max_{0 \neq q \in \mathcal{Q}} \deg q$$

Moreover, such evaluation points can be found in time $|\mathbb{K}|^{O_{d,g}(1)}$.

Proof. We begin by showing a broad class of explicit constructions that satisfy the first three properties. Let $d_1, \dots, d_g$ be any set of co-prime integers such that $d \leq d_i \leq 2d_j$ for all pairs i, j .

Let β_i be such that $[\mathbb{K}(\beta_i) : \mathbb{K}] = d_i$. Let $S_i \subseteq \mathbb{K} \cdot \beta_i \setminus \{0\}$ such that $|S_i| = \lceil n/g \rceil$. This is possible since $|\mathbb{K}| > n$. Additionally, we pick β_i such that all points in S_i are non-conjugate. Let the set of evaluation points be any n -element subset of $S = S_1 \cup S_2 \cup \dots \cup S_g$.

Now, let $\mathbb{F} = \mathbb{K}(\beta_1, \dots, \beta_g) = \mathbb{K}^{d_1 \dots d_g}$. Additionally, let σ_i be the map $x \mapsto x^{|\mathbb{K}|^{d_1 \dots d_g/d_i}}$. Observe that σ_i fixes all of $S \setminus S_i$. Additionally, these σ_i always generate $\text{Gal}(\mathbb{F}/\mathbb{K})$ and thus, the first three constraints hold naturally.

Now, we construct a special class of S_i to satisfy the final constraint. In particular, we will construct a β_i of degree d_i over $\mathbb{K}$ for each $i \in [g]$ and then show that any resulting S_i will be sufficient.

Note, that if $d \geq nr$ then all points naturally satisfy $\text{mult}_{\alpha_i} q = 0$ since minimal polynomial of each α_i has degree larger than $\deg q$.⁷

We will choose $d_i = p_i q_i$ for distinct primes $p_1, q_1, \dots, p_g, q_g \in [P, 1.4P]$ for some $P \geq 2^{r+1}g, d$. Note that this is possible for sufficiently large $P = \text{poly}(d, g)$ due to the prime number theorem.

Let γ_i be an element of degree p_i over $\mathbb{K}$, and let δ_i be an element of degree q_i over $\mathbb{K}$. We will choose δ_i carefully so that $\beta_i = \gamma_i \delta_i$ suffices. Additionally, we assume that $\gamma_i^{p_i}, \delta_i^{q_i} \notin \mathbb{K}$. Note that this condition suffices to show that β_i and $c\beta_i$ are non-conjugate for any $c \neq 1$ since if σ is an automorphism such that $\sigma(\beta_i) = c\beta_i$, then $\sigma^{p_i}(\beta_i) = c^{p_i} \beta_i = \gamma_i \bar{\delta}_i$ for some conjugate $\bar{\delta}_i$ of δ_i . Thus, either δ_i has a conjugate such $\delta_i/\bar{\delta}_i \in \mathbb{K}$, but this is not true by the assumption on δ_i , or $\sigma(\delta_i) = \delta_i$. Similarly, $\sigma(\gamma_i) = \gamma_i$. Thus, $c = 1$!

Thus, let $\mathcal{Q} \subseteq \mathbb{K}[X]_{<nr}$ be a $\mathbb{K}$ -subspace of dimension $h \leq 2^r$. Let $q_1, \dots, q_h$ be monic polynomials forming a basis of $\mathcal{Q}$ with $\deg q_1 < \deg q_2 < \dots < \deg q_h$. Now, let

$$D_i(Y) = \det(\{q_j(\delta_i^{|\mathbb{K}|^{u-1}} Y)\}_{u,j \in [h]}) \in \mathbb{K}(\delta_i)[Y]$$

We show that there is a choice of δ_i such that $D_i(Y)$ is a non-zero polynomial, and in particular $\deg D_i(Y) \leq h \deg q_h$. Choosing such a δ_i is possible due to the following lemma:

Lemma 7.3. *Let $\mathbb{K}$ be a finite field and let $n, r \in \mathbb{N}$ satisfy $|\mathbb{K}| > nr > 2^r$, and let $q_i \geq 2 \cdot 4^r$ be a prime. Then there exists a δ_i with $[\mathbb{K}(\delta_i) : \mathbb{K}] = q_i$ such that for any $1 \leq h \leq 2^r$ and $0 \leq e_1 < \dots < e_h < nr$, we have that $\delta_i^{e_1}, \dots, \delta_i^{e_h}$ are linearly independent over $\mathbb{K}$.*

Such a δ_i can in fact be computed in $|\mathbb{K}|^{O(q_i)}$ time.

We prove this lemma at the end, but conclude using it. Observe that $D_i(Y)$ has leading coefficient $\det(\{\delta_i^{(|\mathbb{K}|^{u-1}) \deg q_j}\}_{u,j \in [h]}) = \det(\{(\delta_i^{\deg q_j})^{|\mathbb{K}|^{u-1}}\}_{u,j \in [h]})$ which is non-zero since the $\delta_i^{\deg q_j}$ are linearly-independent over $\mathbb{K}$, and thus, by Lemma 2.20, using the fact that $x \mapsto x^{|\mathbb{K}|}$ generates the automorphism group of $\mathbb{K}(\delta_i)$, we get the resulting determinant is also non-zero.

Moreover, if $(X - c_i \delta_i \gamma_i)^t \mid q$ for some $q \in \mathcal{Q}$, then we also get that all conjugates of $c_i \delta_i \gamma_i$ are also order t roots of q . But, observe that $c_i \delta_i^{|\mathbb{K}|^u} \gamma_i$ are all conjugates of $c_i \delta_i \gamma_i$ since p_i and q_i are co-prime.

Let P_i be the minimal polynomial of γ_i over $\mathbb{K}$. Since p_i and q_i are coprime, P_i remains irreducible over $\mathbb{K}(\delta_i)$. If $0 \neq q \in \mathcal{Q}$ has multiplicity t at $c \gamma_i \delta_i$, taking q as a basis element of $\mathcal{Q}$ shows that $(Y - c \gamma_i)^t \mid D_i(Y)$. Thus $P_i(Y/c)^t \mid D_i(Y)$. For distinct $c \in \mathbb{K}^*$, these scaled minimal polynomials are pairwise coprime, since the corresponding $c \gamma_i$ are non-conjugate. Consequently,

$$\sum_{i=1}^g \sum_{c \in \mathbb{K}^*} \max_{0 \neq q \in \mathcal{Q}} \text{mult}_{c \gamma_i \delta_i} q \leq \sum_{i=1}^g \frac{\deg D_i}{p_i} \leq \sum_{i=1}^g \frac{h \deg q_h}{p_i} \leq \deg q_h.$$

Thus, this construction works! In particular, it suffices to just show Lemma 7.3. □

Proof of Lemma 7.3. Observe that any nonzero polynomial of the form $\sum_{i=1}^s \eta_i X^{e_i}$ for $s \leq 2^r$ and distinct $0 \leq e_i < nr$ has at most $nr < |\mathbb{K}|$ roots. There are at most $\binom{nr}{\leq 2^r} |\mathbb{K}|^{2^r} \leq |\mathbb{K}|^{4^r}$ choices of such polynomials. Thus, there is a $\delta_i \in \mathbb{K}^{q_i} \setminus \mathbb{K}$ which is not a root of any such polynomial. Now, we can simply brute force check over all δ_i in the field and polynomial pairs resulting in a construction in time $|\mathbb{K}|^{O(q_i)}$ as required.

⁷This was the initial construction provided to us by GPT 5.6 Sol, and we since improved it to use degree $O_{d,g}(1)$ as in this proof.

For the construction above, we additionally discard γ_i with $\gamma_i^{p_i} \in \mathbb{K}$ and δ_i with $\delta_i^{q_i} \in \mathbb{K}$. These exclude at most $p_i|\mathbb{K}|$ and $q_i|\mathbb{K}|$ elements, respectively. Together with the fewer than $|\mathbb{K}|^{4r+1}$ roots of nonzero sparse polynomials excluded above, these leave valid choices. Exhaustive search for both elements takes $|\mathbb{K}|^{O(p_i+q_i)} = |\mathbb{K}|^{O_{d,g}(1)}$ time. $\square$

8 Improved line decoding and proximity gaps

The following theorem gives line decoding over $\mathbb{F}$ from a subspace-design bound over $\mathbb{K}$. We write (δ, a, b) - $\mathbb{F}$ -line-decodable for $(1, \delta, a, b)$ - $\mathbb{F}$ -curve-decodable, as in [Definition 2.8](#).

Theorem 8.1 (Line decoding over the extension field). *Let $\mathbb{K} \subseteq \mathbb{F}$ be finite fields, and let $C \subseteq \Sigma^n$ be an $\mathbb{F}$ -additive code, where Σ is a finite-dimensional $\mathbb{F}$ -vector space. Suppose C is a τ -subspace-design code over $\mathbb{K}$. For an integer $a \geq 1$ and $0 < \varepsilon < 1$, set*

$$D_a = \left\lceil \frac{2^{5a}}{\varepsilon} \right\rceil.$$

If $\tau(D_a) + \varepsilon \leq 1$, then C is

$$\left(1 - \tau(D_a) - \varepsilon, a, \frac{a\varepsilon^2}{4}\right)\text{-}\mathbb{F}\text{-line-decodable}.$$

We prove this theorem in [Subsection 8.2](#). We will only apply the theorem with alphabet $\Sigma = \mathbb{F}$, but state it in more generality.

8.1 Improved combinatorial consequences

We apply [Theorem 8.1](#) to the codes of [Theorem 1.1](#), using a design order that depends only on the gap to capacity. All line parameters and probabilities in this subsection are over $\mathbb{F}$.

Corollary 8.2 (Improved capacity line decoding). *Fix $0 < R < 1$ and $0 < \varepsilon < 1 - R$, and set*

$$\begin{aligned} \delta &= 1 - R - \varepsilon, & L &= \left\lceil \frac{8}{\varepsilon} \right\rceil, & b_0 &= 1 + \left\lceil \frac{2}{\varepsilon} \right\rceil, \\ a_0 &= \left\lceil \frac{16b_0}{\varepsilon^2} \right\rceil, & r &= \left\lceil \frac{2^{5a_0+1}}{\varepsilon} \right\rceil. \end{aligned}$$

For all sufficiently large n and every finite field $\mathbb{K}$ with $|\mathbb{K}| \geq n$, let $\mathbb{F}$ and A be supplied by [Theorem 1.1](#) with parameters r, ε , and put $C = \text{RS}_{\mathbb{F}}(A, \lfloor Rn \rfloor)$. Then C is $(1 - R - \varepsilon/2, L)$ -list-decodable and (δ, a_0, b_0) - $\mathbb{F}$ -line-decodable. Moreover, for every integer $t \geq 1$, it is

$$(\delta, (t-1)L + a_0, t)\text{-}\mathbb{F}\text{-line-decodable}.$$

Proof. The design bound from [Theorem 1.1](#) gives $\tau(d) \leq R + \varepsilon/4$ for $d \leq r$. Apply [Theorem 8.1](#) with gap $\varepsilon/2$ and $a = a_0$. Its required order is exactly r , and

$$1 - \tau(r) - \varepsilon/2 \geq 1 - R - 3\varepsilon/4 > \delta, \quad \frac{a_0\varepsilon^2}{16} \geq b_0.$$

This proves the first line-decoding guarantee. Since $L \leq r$ and

$$\frac{L}{L+1}(1 - R - \varepsilon/4) > 1 - R - \varepsilon/2,$$

Theorem 2.4 gives the asserted list bound, including the closed radius. Also,

$$\delta \left(1 + \frac{1}{b_0 - 1} \right) \leq \delta + \varepsilon/2 = 1 - R - \varepsilon/2.$$

For sufficiently large n , we have $|\mathbb{F}| \geq |\mathbb{K}| \geq n > (L+1)^2$. Thus Item 1 of Theorem 2.13, with scalar field $\mathbb{F}$ and $\ell = 1$, gives the amplified line-decoding guarantee. $\square$

Corollary 8.3 (Improved proximity gaps and correlated agreement). *Let C , $\mathbb{F}$, δ , L , and a_0 be as in Corollary 8.2. Then, over $\mathbb{F}$, the code C has:*

(i) For every real $m > 1$,

$$\left(1, \delta, \frac{mL + a_0}{|\mathbb{F}|}, \frac{1}{m} \right)$$

correlated agreement and proximity gap.

(ii)

$$\left(1, \delta, \frac{nL + a_0}{|\mathbb{F}|} \right)$$

mutual correlated agreement, and

$$\left(1, \delta, \frac{nL + a_0}{|\mathbb{F}|}, 0 \right)$$

correlated agreement and proximity gap.

Proof. The proof of Corollary 8.2 verifies the hypotheses of Theorem 2.13 with scalar field $\mathbb{F}$, $\ell = 1$, $a = a_0$, and $b = b_0$. Its Items 2 and 3 give the stated bounds. Mutual correlated agreement implies correlated agreement and proximity gap with $\gamma = 0$, as recorded in Subsection 2.1. $\square$

The parameters above satisfy

$$L, b_0 = O(\varepsilon^{-1}), \quad a_0 = O(\varepsilon^{-3}), \quad r = 2^{O(\varepsilon^{-3})}.$$

In particular, the design order and the required extension degree depend only on ε .

The probability thresholds in the two conclusions are, respectively,

$$O\left(\frac{m/\varepsilon + \varepsilon^{-3}}{|\mathbb{F}|}\right) \quad \text{and} \quad O\left(\frac{n/\varepsilon + \varepsilon^{-3}}{|\mathbb{F}|}\right).$$

Both estimates have denominator $|\mathbb{F}|$, with no factor $|\mathbb{F}|/|\mathbb{K}|$ in the line-decoding threshold a_0 .

8.2 Proof of the extension-field line-decoding theorem

Lemma 8.4 (Slow growth of monomial spaces). *Let $\mathbb{K} \subseteq \mathbb{F}$ be fields, and let $A \subseteq \mathbb{F}$ have $a \geq 2$ elements, including 0 and 1. For $j \geq 0$, let*

$$V_j = \text{span}_{\mathbb{K}}\{\alpha_1 \cdots \alpha_j : \alpha_1, \dots, \alpha_j \in A\}, \quad V_0 = \mathbb{K}.$$

There is an integer $j \geq 0$ such that

$$\dim_{\mathbb{K}} V_{j+2} \leq 2 \dim_{\mathbb{K}} V_j, \quad \dim_{\mathbb{K}} V_{j+2} \leq 32^{a-2}.$$

Proof. Since $1 \in A$, the spaces V_j are nested. Put $m = a - 2$. If $m = 0$, take $j = 0$. Otherwise, monomials in the m elements of $A \setminus \{0, 1\}$ of total degree at most h span V_h , so

$$\dim_{\mathbb{K}} V_h \leq \binom{h+m}{m}, \quad \dim_{\mathbb{K}} V_{11m} \leq \binom{12m}{m} \leq \left(\frac{12^{12}}{11^{11}} \right)^m < 32^m.$$

If $\dim_{\mathbb{K}} V_{j+2} > 2 \dim_{\mathbb{K}} V_j$ for every $j \in \{0, 2, \dots, 10m - 2\}$, then $\dim_{\mathbb{K}} V_{10m} > 2^{5m} = 32^m$, a contradiction. Any index where this fails gives both conclusions. $\square$

We use a dimension-threshold version of the weighted pruning method of [AHS26, GG25].

Lemma 8.5 (Pruning to a dimension threshold). *Let $\mathbb{K}$ be a finite field and $C \subseteq \Sigma^n$ a $\mathbb{K}$ -additive τ -subspace-design code over $\mathbb{K}$. Let $D, d \geq 1$ be integers, and let $W \subseteq C$ be a $\mathbb{K}$ -subspace of dimension $M \leq D$. Fix $\varepsilon > 0$, and set $\delta = 1 - \tau(D) - \varepsilon$. There is a randomized procedure producing $S \subseteq [n]$ such that*

$$\dim_{\mathbb{K}} W_S < d, \quad W_S = \{w \in W : w|_S = 0\},$$

and, for every $T \subseteq [n]$ with $|T| \geq (\tau(D) + \varepsilon)n$,

$$\Pr[S \subseteq T] \geq \frac{\varepsilon d}{\delta M + \varepsilon d}.$$

Pruning to a dimension threshold PruneThreshold

Require: $W, d, \delta, \varepsilon$ as in Lemma 8.5.

- 1: $U \leftarrow W; S \leftarrow \emptyset$.
- 2: **while** $\dim_{\mathbb{K}} U \geq d$ **do**
- 3: Compute $U_i = \{u \in U : u_i = 0\}$ for each $i \in [n]$.
- 4: $I \leftarrow \{i \in [n] : U_i \subsetneq U\}$.
- 5: Sample $i \in I$ with probability $\frac{\delta \dim_{\mathbb{K}} U_i + \varepsilon d}{\sum_{h \in I} (\delta \dim_{\mathbb{K}} U_h + \varepsilon d)}$.
- 6: $S \leftarrow S \cup \{i\}; U \leftarrow U_i$.
- 7: **return** S .

Proof. The invariant is $U = W_S$. At each iteration, the design bound and $\tau(D) < 1$ ensure that $I \neq \emptyset$, and the dimension strictly decreases. Thus the procedure terminates with $\dim_{\mathbb{K}} W_S < d$.

Fix T as in the statement. We prove the probability bound by induction on M . If $M < d$, the output is empty and the bound holds. Otherwise, write $W_i = \{w \in W : w_i = 0\}$ and let h be the fraction of coordinates with $W_i = W$. The design bound gives $h \leq \tau(D)$ and bounds the total sampling weight by

$$Z := \sum_{i: W_i \subsetneq W} (\delta \dim_{\mathbb{K}} W_i + \varepsilon d) \leq n(\delta M(\tau(D) - h) + \varepsilon d(1 - h)).$$

For each sampled $i \in T$, induction bounds the probability that all subsequent samples also lie in T by $\varepsilon d / (\delta \dim_{\mathbb{K}} W_i + \varepsilon d)$. These denominators cancel the sampling weights, giving

$$\begin{aligned} \Pr[S \subseteq T] &\geq \frac{\varepsilon d n(\tau(D) + \varepsilon - h)}{Z} \\ &\geq \frac{\varepsilon d}{\delta M + \varepsilon d}. \end{aligned}$$

For the last inequality, the difference between $(\tau(D) + \varepsilon - h)(\delta M + \varepsilon d)$ and $\delta M(\tau(D) - h) + \varepsilon d(1 - h)$ is $\varepsilon \delta(M - d) \geq 0$. $\square$

Proof of Theorem 8.1. Write $\delta = 1 - \tau(D_a) - \varepsilon$. Fix a received line $u_0 + \alpha u_1$ and a set $A \subseteq \mathbb{F}$ of a parameters, with assigned codewords $c_\alpha \in C$ satisfying $\Delta(u_0 + \alpha u_1, c_\alpha) \leq \delta$. The case $a = 1$ follows by taking a constant codeword line. For $a \geq 2$, an invertible affine reparameterization lets us assume $0, 1 \in A$. Such a reparameterization preserves codeword lines because C is $\mathbb{F}$ -additive.

Choose j and the spaces V_j, V_{j+1}, V_{j+2} from Lemma 8.4, and put

$$d = \dim_{\mathbb{K}} V_j, \quad W = \text{span}_{\mathbb{K}}\{vc_\alpha : v \in V_{j+1}, \alpha \in A\} \subseteq C, \quad M = \dim_{\mathbb{K}} W.$$

We first show that

$$M < D_a, \quad \varepsilon M \leq \dim_{\mathbb{K}} V_{j+1} + \dim_{\mathbb{K}} V_{j+2} \leq 4d. \quad (8.1)$$

To apply the design hypothesis without assuming $M \leq D_a$, choose $m = \min\{M, D_a\}$ independent elements $v_\ell c_{\alpha_\ell}$ from the generating set of W , and let U be their $\mathbb{K}$ -span. At coordinate i , let b_i count the selected generators whose assigned codeword c_{α_ℓ} disagrees with the received line there. Then $\sum_i b_i \leq \delta nm$. Every other generator has its i th entry in

$$V_{j+1}u_{0,i} + V_{j+2}u_{1,i},$$

since $\alpha_\ell V_{j+1} \subseteq V_{j+2}$. Writing $U_i = \{u \in U : u_i = 0\}$, we obtain

$$m - \dim_{\mathbb{K}} U_i \leq \dim_{\mathbb{K}} V_{j+1} + \dim_{\mathbb{K}} V_{j+2} + b_i.$$

Summing and using the design bound on U yields

$$\varepsilon m \leq \dim_{\mathbb{K}} V_{j+1} + \dim_{\mathbb{K}} V_{j+2} \leq 2 \cdot 32^{a-2} < \varepsilon D_a.$$

Thus $m < D_a$, so $m = M$. The slow-growth of dimensions from Lemma 8.4 now gives (8.1).

For each $\alpha \in A$, let $T_\alpha = \{i : c_{\alpha,i} = u_{0,i} + \alpha u_{1,i}\}$. These sets have size at least $(\tau(D_a) + \varepsilon)n$. Apply Lemma 8.5 to W with $D = D_a$ and threshold d . By (8.1),

$$\Pr[S \subseteq T_\alpha] \geq \frac{\varepsilon d}{\delta M + \varepsilon d} \geq \frac{\varepsilon^2}{4\delta + \varepsilon^2} \geq \frac{\varepsilon^2}{4},$$

where the last inequality uses $\delta \leq 1 - \varepsilon$. By linearity of expectation, some output S satisfies

$$\dim_{\mathbb{K}} W_S < d, \quad |B| \geq \frac{a\varepsilon^2}{4}, \quad B = \{\alpha \in A : S \subseteq T_\alpha\}.$$

If $|B| = 1$, a constant codeword line suffices. Otherwise, fix distinct $\beta, \gamma \in B$. For every $\alpha \in B$, set

$$z_\alpha = (\gamma - \beta)c_\alpha - (\gamma - \alpha)c_\beta - (\alpha - \beta)c_\gamma.$$

The three codewords agree with the received line on S , so $z_\alpha|_S = 0$. Also, $V_j z_\alpha \subseteq W$, since $V_j(A - A) \subseteq V_{j+1}$. If $z_\alpha \neq 0$, multiplication by z_α is injective on V_j , giving a d -dimensional $\mathbb{K}$ -subspace $V_j z_\alpha \subseteq W_S$, a contradiction. Hence every $z_\alpha = 0$, and

$$c_\alpha = c_\beta + \frac{\alpha - \beta}{\gamma - \beta}(c_\gamma - c_\beta) \quad (\alpha \in B).$$

This is a line with both coefficients in C , proving the required $\mathbb{F}$ -line-decoding guarantee. $\square$

9 Acknowledgements

We would like to thank Josh Brakensiek, Yeyuan Chen, Jun-Ting (Tim) Hsieh and Zihan Zhang for various helpful discussions on the problem of constructing explicit list-decodable Reed-Solomon codes and extension-field subspace-designs.

R.G. would also like to thank Prahladh Harsha, Mrinal Kumar, Yihang (Kimi) Sun, Mary Wootters, and Madhur Tulsiani for various discussions and comments on extension field subspace-designs and LCL properties. R.G. thanks Shivam Nadimpalli for raising the question about whether our constructions were strongly explicit.

Part of this work was done while R.G. was visiting the Simons Institute for the Theory of Computing, and he thanks the institute for its hospitality.

References

- [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*, volume 15604 of *Lecture Notes in Computer Science*. Springer, 2025. [3](#)
- [ACFY26] Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. STIR: Reed–Solomon Proximity Testing with Fewer Queries. *Journal of the ACM*, 2026. Advance online publication, 16 October 2026. [3](#), [13](#)
- [AEL95] N. Alon, J. Edmonds, and M. Luby. Linear time erasure codes with nearly optimal recovery. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 512–519, 1995. [3](#), [11](#)
- [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. *Advances in Combinatorics*, October 2025. [2](#)
- [AGG26] Omar Alrabiah, Rohan Goyal, and Venkatesan Guruswami. Combinatorial limitations of list-decoding Reed-Solomon codes. Unpublished manuscript, 2026. [6](#)
- [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 (STOC)*, pages 1458–1469, 2024. [2](#)
- [AHIV17] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkatasubramanian. Ligerio: Lightweight sublinear arguments without a trusted setup. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS ’17*, page 2087–2104, New York, NY, USA, 2017. Association for Computing Machinery. [3](#)
- [AHS26] Vikrant Ashvinkumar, Mursalin Habib, and Shashank Srivastava. *Algorithmic Improvements to List Decoding of Folded Reed-Solomon Codes*, pages 880–898. Society for Industrial and Applied Mathematics, 2026. [2](#), [8](#), [9](#), [27](#), [45](#)

- [Ale05] Michael Alekhovich. Linear Diophantine equations over polynomials and soft decoding of Reed-Solomon codes. *IEEE Trans. Inform. Theory*, 51(7):2257–2265, 2005. (Preliminary version in *43rd FOCS*, 2002). [14](#)
- [BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed-Solomon interactive oracle proofs of proximity. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 14:1–14:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. [3](#)
- [BCDZ26a] Joshua Brakensiek, Yeyuan Chen, Manik Dhar, and Zihan Zhang. Combinatorial Bounds for List Recovery via Discrete Brascamp–Lieb Inequalities. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing*, pages 365–376. Association for Computing Machinery, 2026. [2](#), [12](#)
- [BCDZ26b] Joshua Brakensiek, Yeyuan Chen, Manik Dhar, and Zihan Zhang. From Random to Explicit via Subspace Designs With Applications to Local Properties and Matroids. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing*, pages 619–630. Association for Computing Machinery, 2026. [2](#), [3](#)
- [BCGM26] Sarah Bordage, Alessandro Chiesa, Ziyi Guan, and Ignacio Manzur. All Polynomial Generators Preserve Distance with Mutual Correlated Agreement. In *41st Computational Complexity Conference (CCC 2026)*, volume 383 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:18. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2026. [3](#)
- [BCP⁺26] Joshua Brakensiek, Yeyuan Chen, Aaron Putterman, Zihan Zhang, and Kai Zhe Zheng. Algorithmic list decoding of Reed–Solomon codes up to capacity, 2026. [3](#), [5](#), [6](#)
- [BDG24] Joshua Brakensiek, Manik Dhar, and Sivakanth Gopi. Improved field size bounds for higher order MDS codes. *IEEE Transactions on Information Theory*, 70(10):6950–6960, 2024. [2](#)
- [Ber68] Elwyn R. Berlekamp. *Algebraic Coding Theory*. McGraw-Hill, New York, 1968. [1](#)
- [Ber70] E. R. Berlekamp. Factoring polynomials over large finite fields. *Mathematics of Computation*, 24(111):713–735, 1970. [15](#), [55](#)
- [BGM22] Joshua Brakensiek, Sivakanth Gopi, and Visu Makam. Lower bounds for maximally recoverable tensor codes and higher order MDS codes. *IEEE Transactions on Information Theory*, 68(11):7125–7140, 2022. [2](#)
- [BGM24] Joshua Brakensiek, Sivakanth Gopi, and Visu Makam. Generic Reed-Solomon Codes Achieve List-Decoding Capacity. *SIAM Journal on Computing*, pages STOC23–118–STOC23–154, 2024. [2](#)
- [BKS18] Eli Ben-Sasson, Swastik Kopparty, and Shubhangi Saraf. Worst-case to average case reductions for the distance to a code. In Rocco A. Servedio, editor, *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*, volume 102 of *LIPIcs*, pages 24:1–24:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. [3](#), [13](#)

- [BSBHR19] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable zero knowledge with no trusted setup. In *Advances in Cryptology – CRYPTO 2019, Part III*, volume 11694 of *Lecture Notes in Computer Science*, pages 701–732. Springer, 2019. 3
- [BSCH⁺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*, pages 1157–1167. Association for Computing Machinery, 2026. 3, 13
- [BSCI⁺23] Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity gaps for Reed–Solomon codes. *J. ACM*, 70(5), October 2023. 3, 13
- [BSCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In *Proceedings, Part II, of the 14th International Conference on Theory of Cryptography - Volume 9986*, page 31–60, Berlin, Heidelberg, 2016. Springer-Verlag. 3
- [BSKR10] Eli Ben-Sasson, Swastik Kopparty, and Jaikumar Radhakrishnan. Subspace polynomials and limits to list decoding of Reed–Solomon codes. *IEEE Transactions on Information Theory*, 56(1):113–120, 2010. 6
- [CDG26] Gil Cohen, Dean Doron, and Noam Goldgraber. Algebraic-geometric Parvaresh–Vardy subspace designs and rank condensers. arXiv preprint arXiv:2609.32834, 2026. 2
- [CGG⁺26] Abhranil Chatterjee, Sumanta Ghosh, Rohit Gurjar, Roshan Raj, and Thomas Thierauf. Bipartite matching is in NC. https://image.informatik.htw-aalen.de/~thierauf/Papers/BPM_NC.pdf, 2026. Manuscript. 2
- [CGH⁺26] Dan Carmon, Lior Goldberg, Ulrich Haböck, Leonardo Lerer, Ilya Lesokhin, Shahar Papini, and Shahar Samocha. S-two whitepaper. Cryptology ePrint Archive, Paper 2026/532, 2026. 6, 12
- [CS26] Elizabeth Crites and Alistair Stewart. On Reed–Solomon Proximity Gaps Conjectures. In *Advances in Cryptology – CRYPTO 2026, Part IV*, volume 16803 of *Lecture Notes in Computer Science*, pages 128–156. Springer, 2026. 3
- [CW07] Qi Cheng and Daqing Wan. On the list and bounded distance decodability of Reed–Solomon codes. *SIAM Journal on Computing*, 37(1):195–209, 2007. 6
- [CZ25] Yeyuan Chen and Zihan Zhang. Explicit folded Reed–Solomon and multiplicity codes achieve relaxed generalized singleton bound. In Michal Koucký and Nikhil Bansal, editors, *Proc. 57th ACM Symp. on Theory of Computing (STOC)*, pages 1–12, 2025. 1, 2, 3, 4, 11, 12
- [DG26] Benjamin E. Diamond and Angus Gruen. On the Distribution of the Distances of Random Words. In *Advances in Cryptology – CRYPTO 2026, Part IV*, volume 16803 of *Lecture Notes in Computer Science*, pages 96–127. Springer, 2026. 3
- [DKT26] Quang Dao, Scott Duke Kominers, and Justin Thaler. Reed-solomon codes beyond johnson: Efficient decoding and smaller cryptographic proofs. Cryptology ePrint Archive, Paper 2026/2056, 2026. 5

- [DKTZ26] Quang Dao, Scott Duke Kominers, Justin Thaler, and Kai Zhe Zheng. Counterexamples to beyond-Johnson proximity gaps over binary fields. Technical Report TR26-237, Electronic Colloquium on Computational Complexity (ECCC), 2026. 6
- [DL12] Zeev Dvir and Shachar Lovett. Subspace evasive sets. In *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '12, page 351–358, New York, NY, USA, 2012. Association for Computing Machinery. 1
- [Eli57] Peter Elias. List decoding for noisy channels. Technical Report 335, Research Laboratory of Electronics, Massachusetts Institute of Technology, 1957. Reprinted from the 1957 IRE WESCON Convention Record, Part 2, pp. 94–104. 1
- [Eli91] Peter Elias. Error-correcting codes for list decoding. *IEEE Transactions on Information Theory*, 37(1):5–12, Jan 1991. 1
- [Eth25] Ethereum Foundation. The proximity prize – \$1m proximity gaps challenge, 2025. <https://proximityprize.org/>, Accessed: 2025-11-02. 3
- [FKS22] Asaf Ferber, Matthew Kwan, and Lisa Sauermann. List-decodability with large radius for Reed–Solomon codes. *IEEE Transactions on Information Theory*, 68(6):3823–3828, 2022. 2
- [Gab26] Ariel Gabizon. Linear list size bounds for Reed-Solomon beyond the Johnson radius. Cryptology ePrint Archive, Paper 2026/2048, 2026. 5
- [GG25] Rohan Goyal and Venkatesan Guruswami. Structure theorems (and fast algorithms) for list recovery of subspace-design codes, 2025. *arXiv preprint arXiv:2512.08017*. 2, 12, 45
- [GG26] Rohan Goyal and Venkatesan Guruswami. Optimal Proximity Gaps for Subspace-Design Codes and (Random) Reed-Solomon Codes. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing*, pages 1558–1568. Association for Computing Machinery, 2026. 2, 3, 4, 6, 9, 10, 11, 12, 13, 14
- [GGH26] Rohan Goyal, Venkatesan Guruswami, and Jun-Ting Hsieh. Explicit constant-alphabet subspace design codes, 2026. To appear in FOCS 2026. 2, 3, 4, 11, 12
- [GGSW26] Rohan Goyal, Venkatesan Guruswami, Yihang Sun, and Mary Wootters. Locality of curve-decoding and improved proximity gaps. *arXiv preprint arXiv:2607.08516*, 2026. 2, 3
- [GHKS24] Rohan Goyal, Prahladh Harsha, Mrinal Kumar, and Ashutosh Shankar. Fast list-decoding of univariate multiplicity and folded Reed-Solomon codes. In Santosh Vempala, editor, *Proc. 65th IEEE Symp. on Foundations of Comp. Science (FOCS)*, pages 328–343, 2024. 9, 28
- [GK16] Venkatesan Guruswami and Swastik Kopparty. Explicit subspace designs. *Combinatorica*, 36(2):161–185, 2016. 2, 3, 7, 8, 11, 20
- [GLS⁺24] Zeyu Guo, Ray Li, Chong Shangguan, Itzhak Tamo, and Mary Wootters. Improved list-decodability and list-recoverability of Reed–Solomon codes via tree packings. *SIAM Journal on Computing*, 53(2):389–430, 2024. 2

- [GR06] Venkatesan Guruswami and Atri Rudra. Limits to list decoding Reed–Solomon codes. *IEEE Transactions on Information Theory*, 52(8):3642–3649, 2006. [6](#)
- [GR08] Venkatesan Guruswami and Atri Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Trans. Inform. Theory*, 54(1):135–150, 2008. (Preliminary version in *38th STOC*, 2006). [1](#)
- [GRSZ26] Zeyu Guo, Roshan Raj, Chong Shangguan, and Zihan Zhang. Explicit rank extractors and subspace designs via function fields, with applications to strong blocking sets, April 2026. [2](#)
- [GS99] Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Trans. Inform. Theory*, 45(6):1757–1767, 1999. (Preliminary version in *39th FOCS*, 1998). [1](#)
- [GST23] Eitan Goldberg, Chong Shangguan, and Itzhak Tamo. List-decoding and list-recovery of Reed–Solomon codes beyond the Johnson radius for every rate. *IEEE Transactions on Information Theory*, 69(4):2261–2268, 2023. [2](#)
- [Gur11] Venkatesan Guruswami. Linear-algebraic list decoding of folded Reed–Solomon codes. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity (CCC)*, pages 77–85, 2011. [1](#)
- [GW13] Venkatesan Guruswami and Carol Wang. Linear-algebraic list decoding for variants of Reed-Solomon codes. *IEEE Trans. Inform. Theory*, 59(6):3257–3268, 2013. (Preliminary version in *26th IEEE Conference on Computational Complexity*, 2011 and *15th RANDOM*, 2011). [1](#), [8](#)
- [GX13] Venkatesan Guruswami and Chaoping Xing. List decoding Reed-Solomon, algebraic-geometric, and Gabidulin subcodes up to the Singleton bound. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC ’13, page 843–852, New York, NY, USA, 2013. Association for Computing Machinery. [2](#), [7](#), [11](#), [20](#)
- [GX22] Venkatesan Guruswami and Chaoping Xing. Optimal rate list decoding over bounded alphabets using algebraic-geometric codes. *J. ACM*, 69(2), January 2022. [7](#)
- [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 Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 164–176, 2023. [2](#)
- [HKS26] Prahladh Harsha, Mrinal Kumar, and Ramprasad Saptharishi. Reed-Solomon codes at capacity: Algorithmic list-decoding and proximity gaps. arXiv:2610.08610, 2026. [3](#), [5](#)
- [Jam14] Sebastian Jambor. Primary decomposition of zero-dimensional ideals over arbitrary fields, 2014. Preprint. [54](#), [55](#)
- [Jer26] Fernando Granha Jeronimo. Algorithmic list decoding at capacity and optimal proximity gaps for Reed–Solomon codes. arXiv preprint arXiv:2609.05870, 2026. [3](#), [5](#)
- [JH01] Jørn Justesen and Tom Høholdt. Bounds on list decoding of MDS codes. *IEEE Transactions on Information Theory*, 47(4):1604–1609, 2001. [6](#)

- [JNSV17] Claude-Pierre Jeannerod, Vincent Neiger, Éric Schost, and Gilles Villard. Computing minimal interpolation bases. *Journal of Symbolic Computation*, 83:272–314, 2017. [32](#)
- [Joh62] Selmer M. Johnson. A new upper bound for error-correcting codes. *IRE Transactions on Information Theory*, 8(3):203–207, 1962. [1](#)
- [JS26] Fernando Granha Jeronimo and Nikhil Shagrirhaya. Probabilistic guarantees to explicit constructions: Local properties of linear codes. In *Proc. 58th ACM Symp. on Theory of Computing (STOC)*, pages 806–813. Association for Computing Machinery, 2026. [3](#)
- [Kol88] János Kollár. Sharp effective Nullstellensatz. *Journal of the American Mathematical Society*, 1(4):963–975, 1988. [37](#)
- [Kop15] Swastik Kopparty. List-decoding multiplicity codes. *Theory of Computing*, 11(5):149–182, 2015. [1](#)
- [KRSW23] Swastik Kopparty, Noga Ron-Zewi, Shubhangi Saraf, and Mary Wootters. Improved list decoding of Folded Reed-Solomon and Multiplicity codes. *SIAM J. Comput.*, 52(3):794–840, 2023. (Preliminary version in *59th FOCS*, 2018). [1](#), [8](#), [9](#), [27](#)
- [Liu26] Shu Liu. Efficient construction of subspace design codes over constant alphabet. arXiv preprint arXiv:2610.19691, 2026. [2](#)
- [LMS25] Matan Levi, Jonathan Mosheiff, and Nikhil Shagrirhaya. Random Reed-Solomon codes and random linear codes are locally equivalent. In *2025 IEEE 66th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2097–2131, 2025. [3](#)
- [Lov21] Shachar Lovett. Sparse MDS Matrices over Small Fields: A Proof of the GM-MDS Conjecture. *SIAM Journal on Computing*, 50(4):1248–1262, 2021. [2](#)
- [Mas69] James L. Massey. Shift-register synthesis and BCH decoding. *IEEE Transactions on Information Theory*, 15(1):122–127, 1969. [1](#)
- [Ope26] OpenAI. Deterministic Polynomial Factorization over Prime Fields. OpenAI Math Release preprint [OAI:Deterministic-Polynomial-Factorization-over-Prime-Fields-October-4-2026](#), 2026. [10](#), [15](#)
- [Pet60] W. W. Peterson. Encoding and error-correction procedures for the Bose–Chaudhuri codes. *IRE Transactions on Information Theory*, 6(4):459–470, 1960. [1](#)
- [PV05] Farzad Parvaresh and Alexander Vardy. Correcting errors beyond the Guruswami–Sudan radius in polynomial time. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 285–294, 2005. [1](#)
- [Rot22] Ron M. Roth. Higher-order MDS codes. *IEEE Transactions on Information Theory*, 68(12):7798–7816, 2022. [2](#)
- [RRR21] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. *SIAM Journal on Computing*, 50(3):STOC16–255–STOC16–340, 2021. [3](#)
- [RS60] I. S. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960. [1](#)

- [RVW13] Guy N. Rothblum, Salil Vadhan, and Avi Wigderson. Interactive proofs of proximity: delegating computation in sublinear time. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, page 793–802, New York, NY, USA, 2013. Association for Computing Machinery. [13](#)
- [RW14] Atri Rudra and Mary Wootters. Every list-decodable code for high noise has abundant near-optimal rate puncturings. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '14, page 764–773, New York, NY, USA, 2014. Association for Computing Machinery. [2](#)
- [Sho90] Victor Shoup. New algorithms for finding irreducible polynomials over finite fields. *Mathematics of Computation*, 54(189):435–447, 1990. [15](#)
- [Sri25] Shashank Srivastava. Improved list size for folded Reed-Solomon codes. In Yossi Azar and Debmalya Panigrahi, editors, *Proc. 36th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 2040–2050, 2025. [1](#), [2](#)
- [ST23] Chong Shangguan and Itzhak Tamo. Generalized Singleton Bound and List-Decoding Reed-Solomon Codes Beyond the Johnson Radius. *SIAM Journal on Computing*, 52(3):684–717, 2023. [2](#)
- [ST25] Shashank Srivastava and Madhur Tulsiani. List decoding expander-based codes up to capacity in near-linear time. (manuscript), 2025. [3](#)
- [Sud97] Madhu Sudan. Decoding of Reed-Solomon codes beyond the error-correction bound. *J. Complexity*, 13(1):180–193, 1997. (Preliminary version in *37th FOCS*, 1996). [1](#)
- [Tam24] Itzhak Tamo. Tighter list-size bounds for list-decoding and recovery of folded Reed-Solomon and multiplicity codes. *IEEE Trans. Inform. Theory*, 70(12):8659–8668, 2024. [1](#), [8](#), [9](#), [27](#)
- [vzGK85] Joachim von zur Gathen and Erich Kaltofen. Factorization of multivariate polynomials over finite fields. *Mathematics of Computation*, 45(171):251–261, 1985. [54](#)
- [WB86] Lloyd R. Welch and Elwyn R. Berlekamp. Error correction for algebraic block codes. U.S. Patent 4,633,470, December 1986. [1](#)
- [Woz58] John M. Wozencraft. List decoding. *Quarterly Progress Report, Research Laboratory of Electronics, MIT*, 48:90–95, 1958. [1](#)
- [YH19] Hikmet Yildiz and Babak Hassibi. Optimum linear codes with support-constrained generator matrices over small fields. *IEEE Transactions on Information Theory*, 65(12):7868–7875, 2019. [2](#)
- [Zhu26] Minshen Zhu. Superpolynomial lists for reed-solomon codes over binary extension fields, 2026. [6](#)
- [ZP82] V. V. Zyablov and M. S. Pinsker. List cascade decoding. *Problems of Information Transmission*, 17(4):236–240, 1982. Translated from Problemy Peredachi Informatsii, 17(4):29–34, 1981. [1](#)

A Computing a common zero

Lemma 6.17 (Computing a common zero). *Let $\mathbb{K}$ be a finite field and $\mathbb{L} = \mathbb{K}(X)$. Given $f_1, \dots, f_s \in \mathbb{K}[X, Y_1, \dots, Y_a]$ with $\deg_Y f_j \leq D$ and $\deg_X f_j \leq H$, one can deterministically decide whether they have a common zero over $\overline{\mathbb{L}}$ and, if so, compute a finite extension $E/\mathbb{L}$, given by defining polynomials and an $\mathbb{L}$ -basis, and a common zero $y \in E^a$, in time*

$$O_{s,D,a}(1) \text{poly}(H, \log |\mathbb{K}|).$$

We reduce the proof to the following factorization lemma.

Lemma A.1 (Factorization over finite extensions). *Let $\mathbb{K}$ be a finite field and $\mathbb{L} = \mathbb{K}(X)$. Let $E/\mathbb{L}$ be an extension of degree m with a specified basis $1 = e_1, \dots, e_m$. Suppose we are given its multiplication table, consisting of rational functions $c_{ij\ell}(X) \in \mathbb{L}$ such that*

$$e_i e_j = \sum_{\ell=1}^m c_{ij\ell}(X) e_\ell.$$

We are also given a nonzero polynomial $g \in E[T]$ of degree at most d , with each coefficient expressed in this basis. Suppose all rational functions in these expressions and in the multiplication table have numerators and denominators of degree at most h .

Then one can deterministically compute the factorization $g(T) = c \prod_{\nu=1}^r g_\nu(T)^{\mu_\nu}$, where $c \in E$, the g_ν are distinct monic irreducible polynomials over E , and the μ_ν are their multiplicities, in time $O_{m,d}(1) \text{poly}(h, \log |\mathbb{K}|)$. The coefficients of each g_ν are returned in the given basis, with numerator and denominator degrees of their coordinates at most $O_{m,d}(1)h$.

We use the following two results.

Theorem A.2 (von zur Gathen–Kaltofen [vzGK85, Theorem 4.2(i)]). *Let $\mathbb{K}$ be a finite field. Factoring a polynomial in $\mathbb{K}[X, T]$ of total degree at most D reduces deterministically to factoring univariate polynomials of degree at most D over finite extensions of $\mathbb{K}$ of degree $\text{poly}(D)$. The reduction takes $\text{poly}(D, \log |\mathbb{K}|)$ time.*

Clearing denominators and applying [Corollary 2.18](#), we can therefore factor a degree- d polynomial over $\mathbb{K}(X)$ whose coefficient numerators and denominators have degree at most h deterministically in time $\text{poly}(d, h, \log |\mathbb{K}|)$.

Theorem A.3 (Consequence of Jambor [Jam14, Theorem 6.6 and Proposition 4.10]). *Let $\mathbb{K}$ be a finite field and $\mathbb{L} = \mathbb{K}(X)$. Let $E/\mathbb{L}$ have degree m and be given by a basis and multiplication table, and let $g \in E[T]$ be nonzero of degree at most d . Suppose the rational functions in these data have numerator and denominator degrees at most h .*

Computing the monic irreducible factors of g and their multiplicities reduces deterministically to $\text{poly}(m, d)$ factorizations of polynomials in $\mathbb{K}[X, T]$ with

$$\deg_X \leq O_{m,d}(1)h, \quad \deg_T \leq md.$$

The reduction takes $O_{m,d}(1) \text{poly}(h, \log |\mathbb{K}|)$ time outside these calls. The returned factor coefficients have coordinates with numerator and denominator degrees at most $O_{m,d}(1)h$ in the given basis.

Proof. Apply Jambor's algorithms to $E[T]/(g)$, whose dimension over $\mathbb{L}$ is at most md . In each resulting residue field, compute the minimal polynomial of the image of T over E . These are the irreducible factors of g , and repeated division gives their multiplicities. The algorithms use $\text{poly}(m, d)$ operations, including factorizations of polynomials over $\mathbb{L}$ of degree at most md and p -th root extractions, where $p = \text{char}(\mathbb{K})$. Clearing denominators turns the former into bivariate factorizations.

If $p \leq md$, Jambor's coefficientwise root computation [Jam14, Section 4.2] takes time polynomial in m, d , the current coefficient degrees, and $\log |\mathbb{K}|$, using parameter extensions of degree at most md . If $p > md$, no root extractions are needed, since the relevant polynomials have degree at most md .

For coprime $A, B \in \mathbb{K}[U]$, write $\text{ht}(A/B) = \max(\deg A, \deg B)$. An arithmetic operation increases the current height bound by at most a factor of two. Clearing denominators of a polynomial of degree at most md increases the parameter degree by at most a factor of $md + 1$, and its bivariate factors have no larger parameter degree. The root computations and changes of coordinates between the parameter fields also increase heights by at most $\text{poly}(m, d)$. Since there are $\text{poly}(m, d)$ operations and factorization calls, all intermediate and output heights are $O_{m,d}(1)h$. This gives the stated degrees for the bivariate polynomials, and the remaining computations take $O_{m,d}(1) \text{poly}(h, \log |\mathbb{K}|)$ time. $\square$

Proof of Lemma A.1. Apply Theorems A.2 and A.3 to reduce to univariate factorization over finite fields. Write $p = \text{char}(\mathbb{K})$. If $p \leq md$, Berlekamp's algorithm [Ber70] already gives the required bound: its running time is polynomial in p , the polynomial degree, and the logarithm of the field size. If $p > md$, use Corollary 2.18. In either case, each call takes $O_{m,d}(1) \text{poly}(h, \log |\mathbb{K}|)$ time, giving the claimed runtime and coefficient-degree bound. $\square$

Proof of Lemma 6.17. Put

$$Q = 2(3D)^a, \quad b = \binom{a + Q + D}{a}, \quad B = Db.$$

By Lemma 6.16, a system of polynomials of degree at most D in at most a variables has no common zero over the algebraic closure of its coefficient field exactly when

$$\sum_{j=1}^s G_j f_j = 1, \quad \deg G_j \leq Q$$

has a solution. Comparing coefficients gives a linear system with at most b rows and sb columns. Over $\mathbb{L}$, this decides whether the original system has a common zero in time $\text{poly}(s, b, H, \log |\mathbb{K}|)$. Henceforth assume it does.

Suppose we have chosen $\alpha_1, \dots, \alpha_{i-1}$ such that the remaining equations have a common zero over $\overline{E}$, where $E = \mathbb{L}(\alpha_1, \dots, \alpha_{i-1})$. After substituting these values, regard the f_j as polynomials in $T = Y_i$ and $Z = (Y_{i+1}, \dots, Y_a)$ over E . The identity

$$\sum_{j=1}^s G_j(T, Z) f_j(T, Z) = 1, \quad \deg_Z G_j \leq Q,$$

gives a linear system $M(T)c = e$ over $E(T)$, where e is the coefficient vector of 1. The matrix has at most b rows and sb columns, and its entries lie in $E[T]$ with degree at most D . For every algebraic

value α of T , the same degree bound shows that $M(\alpha)c = e$ is solvable exactly when the equations $f_j(\alpha, Z) = 0$ have no common zero.

If $M(T)c = e$ is solvable, Cramer's rule gives a common nonzero denominator $h \in E[T]$ of degree at most B . Clearing denominators gives

$$h(T) = \sum_{j=1}^s \tilde{G}_j(T, Z) f_j(T, Z), \quad \tilde{G}_j \in E[T, Z].$$

Thus the T -coordinate of every common zero is a root of h . Factor h using [Lemma A.1](#). For each irreducible factor p , put $E' = E[T]/(p)$ and $\alpha = T \bmod p$, and use the same linear-system test to check whether $f_j(\alpha, Z) = 0$ has a common zero over $\overline{E'}$. At least one factor passes this test, since the current system has a common zero. Choose such a factor and set $\alpha_i = \alpha$. This gives $[E' : E] = \deg p \leq B$.

Otherwise, put $r = \text{rank}_{E(T)} M(T)$. There is a nonzero $(r+1) \times (r+1)$ minor $h(T)$ of $[M(T) \mid e]$, again of degree at most B . Whenever $h(\alpha) \neq 0$, the augmented matrix has rank at least $r+1$, whereas $M(\alpha)$ has rank at most r . Thus $M(\alpha)c = e$ remains unsolvable, and the polynomial equations have a common zero. Choose α_i from $1, X, \dots, X^B$ with $h(\alpha_i) \neq 0$; these are $B+1$ distinct elements of $\mathbb{L}$. In this case the coefficient field does not change.

Repeating this procedure gives a common zero $(\alpha_1, \dots, \alpha_a)$ in an extension of degree at most B^a . The defining polynomials and the power bases of the successive extensions give the required representation and $\mathbb{L}$ -basis.

It remains to bound the running time using [Lemma A.1](#). At each step there is at most one factorization of a polynomial of degree at most B , followed by at most B linear-system tests. Let h_i bound the numerator and denominator degrees of the coordinates of the current coefficients, the chosen coordinates, and the entries of the multiplication table; initially $h_0 = H$. All matrix dimensions and extension degrees are $O_{s,D,a}(1)$. Cramer's rule and [Lemma A.1](#) bound the coefficient degrees after the linear-system computations and factorization by $O_{s,D,a}(1)h_i$. Reducing powers modulo the chosen factor constructs the new multiplication table and substitutes α_i into the remaining equations, with the same bound. Choosing α_i from $1, X, \dots, X^B$ contributes at most DB to these degrees. Consequently, all coefficient degrees are $O_{s,D,a}(1) \text{poly}(H)$. Since B depends only on D, a , each factorization takes $O_{D,a}(1) \text{poly}(h_i, \log |\mathbb{K}|)$ time. There are a steps, so the total running time is $O_{s,D,a}(1) \text{poly}(H, \log |\mathbb{K}|)$, as required. $\square$