Electronic Colloquium on Computational Complexity, Report No. 125 (2023)

Randomly punctured Reed–Solomon codes achieve list-decoding capacity over linear-sized fields

Omar Alrabiah[*] Venkatesan Guruswami[†] Ray Li[‡]

Abstract

Reed–Solomon codes are a classic family of error-correcting codes consisting of evaluations of low-degree polynomials over a finite field on some sequence of distinct field elements. They are widely known for their optimal unique-decoding capabilities, but their list-decoding capabilities are not fully understood. Given the prevalence of Reed-Solomon codes, a fundamental question in coding theory is determining if Reed–Solomon codes can optimally achieve list-decoding capacity.

A recent breakthrough by Brakensiek, Gopi, and Makam, established that Reed–Solomon codes are combinatorially list-decodable all the way to capacity. However, their results hold for randomly-punctured Reed–Solomon codes over an exponentially large field size $2^{O(n)}$, where n is the block length of the code. A natural question is whether Reed–Solomon codes can still achieve capacity over smaller fields. Recently, Guo and Zhang showed that Reed–Solomon codes are list-decodable to capacity with field size $O(n^2)$. We show that Reed–Solomon codes are list-decodable to capacity with linear field size $O(n)$, which is optimal up to the constant factor. We also give evidence that the ratio between the alphabet size q and code length n cannot be bounded by an absolute constant.

Our techniques also show that random linear codes are list-decodable up to (the alphabet-independent) capacity with optimal list-size $O(1/\varepsilon)$ and near-optimal alphabet size $2^{O(1/\varepsilon^2)}$, where ε is the gap to capacity. As far as we are aware, list-decoding up to capacity with optimal list-size $O(1/\varepsilon)$ was not known to be achievable with any linear code over a constant alphabet size (even non-constructively), and it was also not known to be achievable for random linear codes over any alphabet size.

Our proofs are based on the ideas of Guo and Zhang, and we additionally exploit symmetries of reduced intersection matrices. With our proof, which maintains a hypergraph perspective of the list-decoding problem, we include an alternate presentation of ideas from Brakensiek, Gopi, and Makam that more directly connects the list-decoding problem to the GM-MDS theorem via a hypergraph orientation theorem.

[*] Department of EECS, UC Berkeley, Berkeley, CA, 94709, USA. Email: oalrabiah@berkeley.edu. Research supported in part by a Saudi Arabian Cultural Mission (SACM) Scholarship, NSF CCF-2210823 and V. Guruswami's Simons Investigator Award.

[†] Departments of EECS and Mathematics, and the Simons Institute for the Theory of Computing, UC Berkeley, Berkeley, CA, 94709, USA. Email: venkatg@berkeley.edu. Research supported by a Simons Investigator Award and NSF grants CCF-2210823 and CCF-2228287.

[‡] Department of EECS, UC Berkeley, Berkeley, CA, 94709, USA. Email: rayyli@berkeley.edu. Research supported by the NSF Mathematical Sciences Postdoctoral Research Fellowships Program under Grant DMS-2203067, and a UC Berkeley Initiative for Computational Transformation award.

ISSN 1433-8092

Contents

1 Introduction

An *(error-correcting) code* is simply a set of strings (*codewords*). In this paper, all codes are *linear*, meaning our code $C \subset \mathbb{F}_q^n$ is a space of vectors over a finite field $\mathbb{F}_q$, for some prime power q. A *Reed–Solomon code* [RS60] is a linear code obtained by evaluating low-degree polynomials over $\mathbb{F}_q$. More formally,

$$\mathsf{RS}_{n,k}(\alpha_1, \ldots, \alpha_n) \stackrel{\text{def}}{=} \{(f(\alpha_1), \ldots, f(\alpha_n)) \in \mathbb{F}_q^n : f \in \mathbb{F}_q[X], \deg(f) < k\}. \tag{1}$$

The *rate* R of a code C is $R \stackrel{\text{def}}{=} \log_q |C|/n$, which, for a Reed–Solomon code, is k/n. Famously, Reed–Solomon codes are optimal for the *unique decoding problem* [RS60]: for any rate R Reed–Solomon code, for every received word $y \in \mathbb{F}_q^n$, there is at most one codeword within Hamming distance pn of y,[1] and this *error parameter* $p = \frac{1-R}{2}$ is optimal by the *Singleton bound* [Sin64].

In this paper, we study Reed–Solomon codes in the context of *list-decoding*, a generalization of unique-decoding that was introduced by Elias and Wozencraft [Eli57, Woz58]. Formally, a code $C \subset \mathbb{F}_q^n$ is (p, L)*-list-decodable* if, for every received word $y \in \mathbb{F}_q^n$, there are at most L codewords of C within Hamming distance pn of y.

It is well known that the *list-decoding capacity*, namely the largest fraction of errors that can be list-decoded with small lists, is $1 - R$ [GRS22, Theorem 7.4.1]. Specifically, for $p = 1 - R - \varepsilon$, there are (infinite families of) rate R codes that are (p, L) list-decodable for a list-size L as small as $O(1/\varepsilon)$. On the other hand, for $p = 1 - R + \varepsilon$, if a rate R code is (p, L) list decodable, the list size L must be exponential in the code length n. Informally, a code that is list-decodable up to radius $p = 1 - R - \varepsilon$ with list size $O_\varepsilon(1)$, or even list size $n^{O_\varepsilon(1)}$ where n is the code length, is said to *achieve (list-decoding) capacity*.

The list-decodability of Reed–Solomon codes is important for several reasons. Reed–Solomon codes are the most fundamental algebraic error-correcting code. In fact, all of the prior explicit constructions of codes achieving list-decoding capacity are algebraic constructions that generalize Reed–Solomon codes, for example, Folded Reed–Solomon codes [GR08, KRZSW18], Multiplicity codes [GW13, Kop15, KRZSW18], and algebraic-geometric codes [GX13]. Thus, it is natural to wonder whether and when Reed–Solomon codes themselves achieve list-decoding capacity. Additionally, all Reed–Solomon codes are optimally *unique-decodable*, so (equivalently) they are optimally list-decodable $L = 1$, making them a natural candidate for codes achieving list-decoding capacity. Further, capacity-achieving Reed–Solomon codes would potentially offer advantages over existing explicit capacity-achieving codes, such as simplicity and potentially smaller alphabet sizes (which we achieve in this work). Lastly, list-decoding of Reed–Solomon codes has found several applications in complexity theory and pseudorandomness [CPS99, STV01, LP20].

For all these reasons, the list-decodability of Reed–Solomon codes is well-studied. As rate R Reed–Solomon codes are uniquely decodable up to the optimal radius $\frac{1-R}{2}$ given by the Singleton Bound, the Johnson-bound [Joh62] automatically implies that Reed–Solomon codes are (p, L)-list-decodable for error parameter $p = 1 - \sqrt{R} - \varepsilon$ and list size $L = O(1/\varepsilon)$. Guruswami and Sudan [GS99] showed how to *efficiently* list-decode Reed–Solomon codes up to the Johnson radius $1 - \sqrt{R}$. For a long time, this remained the best list-decodability result (even non-constructively) for Reed–Solomon codes.

Since then, several results suggested Reed–Solomon codes could *not* be list-decoded up to capacity, and in fact, not much beyond the Johnson radius $1 - \sqrt{R}$. Guruswami and Rudra [GR06] showed that, for a generalization of list-decoding called *list-recovery*, Reed–Solomon codes are not list-recoverable beyond the (list-recovery) Johnson bound in some parameter settings. Cheng and Wan [CW07] showed that efficient list-decoding of Reed–Solomon codes beyond the Johnson radius in certain parameter settings implies fast algorithms for the discrete logarithm problem. Ben-Sasson, Kopparty, and Radhakrishnan [BKR10] showed that full-length Reed–Solomon codes ($q = n$) are not list-decodable much beyond the Johnson bound in some parameter settings.

Since then, an exciting line of work [RW14, ST20, GLS$^+$22, FKS22, GST22, BGM23, GZ23] has shown the existence of Reed–Solomon codes that could in fact be list-decoded beyond the Johnson radius. These works all consider *combinatorial* list-decodability of *randomly punctured* Reed–Solomon codes. By combinatorial list-decodability, we mean that the code is proved to be list-decodable without providing an algorithm to

[1] The Hamming distance between two codewords is the number of coordinates on which they differ.

efficiently decode the list of nearby codewords. By randomly punctured Reed–Solomon code, we mean a code $\mathsf{RS}_{n,k}(\alpha_1, \ldots, \alpha_n)$ where $(\alpha_1, \ldots, \alpha_n)$ are chosen uniformly over all n-tuples of pairwise distinct elements of $\mathbb{F}_q$. Several of these works [RW14, FKS22, GST22] proved more general list-decoding results about randomly puncturing any code with good unique-decoding properties, not just Reed–Solomon codes.

In this line of work, a recent breakthrough of Brakensiek, Gopi, and Makam [BGM23] showed, using notions of "higher-order MDS codes" [BGM22, Rot22], that Reed–Solomon codes can actually be list-decoded up to capacity. In fact, they show, more strongly, that Reed–Solomon codes can be list-decoded with list size L with radius $p = \frac{L}{L+1}(1-R)$, exactly meeting the *generalized Singleton bound* [ST20], resolving a conjecture of Shangguan and Tamo [ST20]. However, their results require randomly puncturing Reed–Solomon codes over an exponentially large field size $2^{O(n)}$, where n is the block length of the code.

A natural question is how small we can take the field size in a capacity-achieving Reed–Solomon code. Brakensiek, Dhar, and Gopi [BDG22, Corollary 1.7, Theorem 1.8] showed that the exponential field size in [BGM23] is indeed necessary to *exactly* achieve the generalized Singleton bound for $L = 2$ for rates bounded away from 0, but smaller field sizes remained possible if one allowed a small ε slack in the parameters. Recently, an exciting work of Guo and Zhang [GZ23] showed that Reed–Solomon codes are list-decodable up to capacity, in fact up to (but not exactly at) the generalized Singleton bound, with alphabet size $O(n^2)$.

1.1 Our results

List-decoding Reed–Solomon codes. Building on Guo and Zhang's argument, we show that Reed–Solomon codes are list-decodable up to capacity and the generalized Singleton bound with linear alphabet size $O(n)$, which is evidently optimal up to the constant factor. Our main result is the following.

Theorem 1.1. *Let $\varepsilon \in (0, 1)$, $L \geq 2$ and q be a prime power such that $q \geq n + k \cdot 2^{10L/\varepsilon}$. Then with probability at least $1 - 2^{-Ln}$, a randomly punctured Reed–Solomon code of block length n and rate k/n over $\mathbb{F}_q$ is $(\frac{L}{L+1}(1 - R - \varepsilon), L)$ average-radius list-decodable.*

As in previous works like [BGM23, GZ23], Theorem 1.1 gives *average-radius list-decodability*, a stronger guarantee than list-decodability: for any distinct $L + 1$ codewords $c^{(1)}, \ldots, c^{(L+1)}$ and any vector $y \in \mathbb{F}_q^n$, the average Hamming distance from $c^{(1)}, \ldots, c^{(L+1)}$ to y is at least $\frac{L}{L+1}(1 - R - \varepsilon)$. Taking $L = O(1/\epsilon)$ in Theorem 1.1, it follows that Reed–Solomon codes achieve list-decoding capacity even over linear-sized alphabets.

Corollary 1.2. *Let $\varepsilon \in (0, 1)$ and q be a prime power such that $q \geq n + k \cdot 2^{O(1/\varepsilon^2)}$. Then with probability at least $1 - 2^{-\Omega(n/\varepsilon)}$, a randomly punctured Reed–Solomon code of block length n and rate k/n over $\mathbb{F}_q$ is $(1 - R - \varepsilon, O(\frac{1}{\varepsilon}))$ average-radius list-decodable.*

The alphabet size in [GZ23] is $2^{O(L^2/\varepsilon)}nk$. Our main contribution is improving their alphabet size from quadratic to linear. As a secondary improvement, we also bring down the constant factor from $2^{O(L^2/\varepsilon)}$ to $2^{O(L/\varepsilon)}$. We defer the proof overview of Theorem 1.1 to Section 3.1 after setting up the necessary notions in Section 2.

In our proof of Theorem 1.1, we maintain a hypergraph perspective of the list-decoding problem, which was introduced in [GLS+22]. Section 2.2 elaborates on the advantages of this perspective, which include (i) more conpact notations, definitions, and lemma statements, (ii) our improved constant factor of $2^{O(L/\varepsilon)}$, (iii) an improved alphabet size in our random linear codes result below (Theorem 1.3), and (iv) an alternate presentation of ideas from Brakensiek, Gopi, and Makam [BGM23] that more directly connects the list-decoding problem to the GM-MDS theorem [DSY14, Lov18, YH19] via a hypergraph orientation theorem (see Appendix A).

List-decoding random linear codes. A random linear code of rate R and length n over $\mathbb{F}_q$ is a random subspace of $\mathbb{F}_q^n$ of dimension Rn. List-decoding random linear codes is well-studied [ZP81, Eli91, GHSZ02, GHK11, Woo13, RW14, RW18, LW20, MRRZ+20, GLM+21, GM22, PP23] and is an important question for several reasons. First, finding explicit codes approaching list-decoding capacity is a major challenge, and random linear codes provide a stepping stone towards explicit codes: a classic result says that uniformly random codes achieve list-decoding capacity [Eli57, Woz58], and showing list-decodability of random linear codes can be viewed as a derandomization of the uniformly random construction. Mathematically, the list-decodability of random linear codes concerns a fundamental geometric question: to what extent do random subspaces over $\mathbb{F}_q$ behave like uniformly random sets? In coding theory, list-decodable random linear codes are useful building blocks in other coding theory constructions [GI01, HW18]. Lastly, the algorithmic question of decoding random linear codes is closely related to the Learning With Errors (LWE) problem in cryptography [Reg09] and Learning Parity with Noise (LPN) problem in learning theory [BKW03, FGKP06].

The list-decodability of random linear codes is more difficult to analyze than uniformly random codes, because codewords do not enjoy the same independence as in random codes. Thus the naive argument that shows that random linear codes achieve list-decoding capacity [ZP81] gives an exponentially worse list size of $q^{1/\varepsilon}$ than for random codes (ε is the gap to the "q-ary capacity", $R = 1 - H_q(p)$, where $H_q(x) \overset{\text{def}}{=} x\log_q(q-1) - x\log_q(x) - (1-x)\log_q(1-x)$ is the q-ary entropy function). Several works have sought to circumvent this difficulty [Eli91, GHSZ02, GHK11, Woo13, RW14, RW18, LW20, GLM$^+$21] improving the list-size bound to $O_q(1/\varepsilon)$, matching the list-size of uniformly random codes.

However, these results are more relevant for smaller alphabet sizes q, and approaching the alphabet-independent capacity of $p = 1 - R$ is less understood. In this setting, uniformly random codes are, with high probability, list-decodable to capacity with optimal alphabet size $2^{O(1/\varepsilon)}$ [2] and optimal list size $O(1/\varepsilon)$.[3] However, it was not known whether random linear codes (or, in general, more structured codes) could achieve similar parameters. In particular, both of the following questions were open (as far as we are aware).

- Are rate R random linear codes $(1 - R - \varepsilon, O(1/\varepsilon))$-list-decodable with high probability? Previously, this was not known for *any* alphabet size q, even alphabet size growing with the length of the code. Previously, the best list size for random linear codes list-decodable to radius $p = 1 - R - \varepsilon$ was at least $2^{\Omega(1/\varepsilon)}$ [GHK11, RW18].[4]
- Do there exist *any* linear codes (even non-constructively) over constant-sized (independent of n) alphabets that are $(1 - R - \varepsilon, O(1/\varepsilon))$-list-decodable?

Using the same framework as the proof of Theorem 1.3, we answer both questions affirmatively. We show that, with high probability, random linear codes approach the generalized Singleton bound, and thus capacity, with alphabet size close to the optimal.

Theorem 1.3. *For all $L \geq 1, \varepsilon \in (0,1)$, a random linear code over alphabet size $q \geq 2^{10L/\varepsilon}$ and n sufficiently large is with high probability $(\frac{L}{L+1}(1 - R - \varepsilon), L)$-average-radius-list-decodable.*

By taking $L = O(1/\varepsilon)$, we see that random linear codes achieve capacity with optimal list size $O(1/\varepsilon)$ and near optimal alphabet size $2^{O(1/\varepsilon^2)}$.

Corollary 1.4. *For all $\varepsilon > 0$, a random linear code over alphabet size $q \geq 2^{O(1/\varepsilon^2)}$ and n sufficiently large is with high probability $(1 - R - \varepsilon, O(1/\varepsilon))$-average-radius-list-decodable.*

The techniques developed in this work for the proof of Theorem 1.1 are important for obtaining the strong alphabet size guarantees of Theorem 1.3. One could also have adapted the proof of Guo and Zhang, but doing so in the same natural way would only yield an alphabet size of $O(n)$ (see Section 4.4 for discussions). Further, our use of the hypergraph machinery, which gives a secondary improvement over [GZ23] in constant factor in the alphabet size in Corollary 1.2, gives the primary improvement in the alphabet size in Corollary 1.4 from $2^{O(1/\varepsilon^3)}$ to $2^{O(1/\varepsilon^2)}$.

As the proof of Theorem 1.3 is very similar to the proof of Theorem 1.1, we focus most of the paper on Theorem 1.1 for brevity and clarity of presentation in Section 2 and Section 3. In Section 4, we show how the definitions and proof can be modified to work for random linear codes.

[2] This follows from the list-decoding capacity theorem [Eli57, Woz58]. Over q-ary alphabets, the list-decoding capacity is given by $p = H_q^{-1}(1 - R)$, which is larger than $1 - R - \varepsilon$ when $q \geq 2^{\Omega(1/\varepsilon)}$.

[3] For codes over smaller alphabets, the list size $O(1/\varepsilon)$, where ε is the gap to capacity, is believed to be optimal, but a proof is only known for large radius [GV10]. However, for approaching the alphabet independent capacity, the list size $O(1/\varepsilon)$ *is* known to be optimal by the generalized Singleton bound [ST20].

[4] [GHK11] appears to give a list-size bound of $O(q^{O_R(1)}/\varepsilon)$, and [RW18] appears to give a list size bound that is at least $q^{\log^2(1/\varepsilon)}$, and we need $q \geq 2^{\Omega(1/\varepsilon)}$

Alphabet size lower bounds. Above, we saw that random linear codes achieve list-decoding capacity with optimal list-size and near-optimal alphabet size. A natural question, asked by Guo and Zhang, is how large the alphabet size needs to be for capacity-achieving Reed–Solomon codes. We showed that $q \geq n \cdot 2^{O(1/\varepsilon^2)}$ suffices, and by the list-decoding capacity theorem [Eli57, Woz58], we cannot have better than an exponential-type dependence on $1/\varepsilon$ for subconstant $\varepsilon < O(1/\log n)$.

For approaching capacity with constant ε, Ben-Sasson, Kopparty, and Radhakrishnan [BKR10] showed that, for any $c \geq 1$, there exist full-length Reed–Solomon codes that are not list-decodable much beyond the Johnson bound with list-sizes $O(n^c)$. Thus in order to achieve list-decoding capacity, one needs $q > n$ in some cases. However, while full-length Reed–Solomon codes could not achieve capacity, perhaps it was possible that Reed–Solomon codes over field size, say $q = 2n$ or even $q = (1+\gamma)n$, could achieve capacity in all parameter settings. We observe that, as a corollary of [BKR10], such a strong guarantee is not possible. We show that, for any $c > 1$, there exist a constant rate $R = R(c) > 0$ and infinitely many field sizes q such that all Reed–Solomon codes of length $n \geq q/c$ and rate R over $\mathbb{F}_q$ are not list-decodable to capacity $1 - R$ with list size n^c. The proof is in Appendix B.

Proposition 1.5. *Let $\delta = 2^{-b}$ for some positive integer $b \geq 3$. There exists infinitely many q such that any Reed–Solomon code of length $n \geq 4\delta^{0.99}q$ and rate δ is not $(1-2\delta, n^{\Omega(\log(1/\delta))})$-list-decodable.*

2 Preliminaries

2.1 Basic notation

For positive integers t, let $[t]$ denote the set $\{1, 2, \ldots, t\}$. The *Hamming distance* $d(x, y)$ between two vectors $x, y \in \mathbb{F}_q^n$ is the number of indices i where $x_i \neq y_i$. For a finite field $\mathbb{F}_q$, we follow the standard notation that $\mathbb{F}_q[X_1, \ldots, X_n]$ denotes the ring of multivariate polynomials with variables $X_1, \ldots, X_n$ over $\mathbb{F}_q$, and $\mathbb{F}_q(X_1, \ldots, X_n)$ denotes the field of fractions of the polynomial ring $\mathbb{F}_q[X_1, \ldots, X_n]$. By abuse of notation, we let $X_{\leq i}$ or $X_{[i]}$ to denote the sequence $X_1, \ldots, X_i$, and we let, for example, $X_{\leq i} = \alpha_{\leq i}$ to denote $X_1 = \alpha_1, X_2 = \alpha_2, \ldots, X_i = \alpha_i$. Given a matrix M over the field of fractions $\mathbb{F}_q(X_1, \ldots, X_n)$ and field elements $\alpha_1, \ldots, \alpha_i \in \mathbb{F}_q$, let $M(X_{\leq i} = \alpha_{\leq i})$ denote the matrix over $\mathbb{F}_q(X_{i+1}, X_{i+2}, \ldots, X_n)$ obtained by setting $X_{\leq i} = \alpha_{\leq i}$ in M.

2.2 Hypergraphs and connectivity

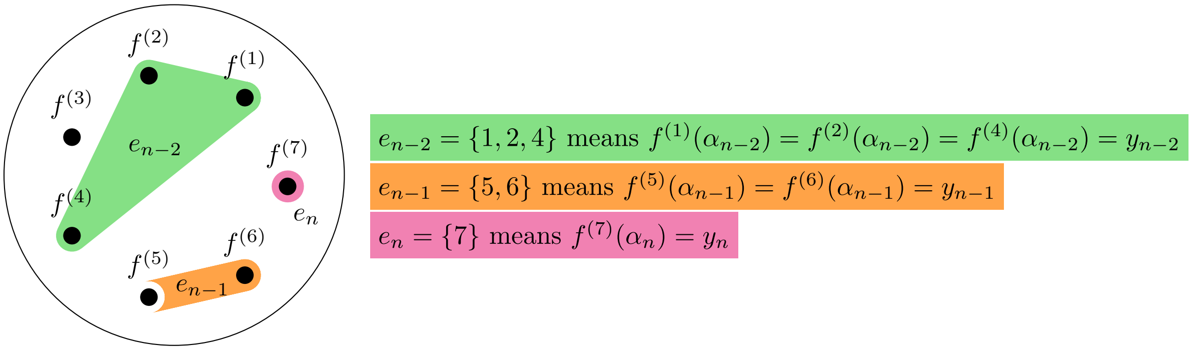
In this work, we maintain a hypergraph perspective of the list-decoding problem, which was introduced in [GLS+22]. We describe a bad list-decoding instance with a hypergraph where the $L+1$ bad codewords identify the vertices and the n evaluation points identify the hyperedges (Definition 2.1). While prior works described a bad list-decoding instance by $L+1$ sets indicating the agreements of the codewords with the received word, this hypergraph perspective gives us several advantages:

1. The constraints imposed by a bad list-decoding configuration yield a hypergraph that is *weakly-partition-connected.* This is a natural notion of hypergraph connectivity, which is well-studied in combinatorics [FKK03b, FKK03a, Kir03] and optimization [JMS03, FK09, Fra11, CX18], and which generalizes a well-known notion (k-partition-connectivity) for graphs [NW61, Tut61].[5] This connection allows us to have more compact notation, definitions, and lemma statements.

2. Because we work with weakly-partition-connected hypergraphs, we save a factor of L in Lemma 2.13 compared to the analogous lemma in [GZ23]. This allows us to improve the constant factor in alphabet size for Reed–Solomon codes from $2^{O(L^2/\varepsilon)}$ in [GZ23] to $2^{O(L/\varepsilon)}$ in Theorem 1.1.

3. For similar reasons, for random linear codes, the hypergraph perspective saves a factor of L in the alphabet size exponent, improving from $2^{O(L^2/\varepsilon)}$ to $2^{O(L/\varepsilon)}$ in Theorem 1.3.

[5] The notion of weakly-partition-connected sits between two other well-studied notions: *k-partition-connected* implies k-weakly-partition-connected implies *k-edge-connected* [Kir03]. Each of these three notions generalizes an analogous notion on graphs. On graphs, k-partition-connected and k-weakly-partition-connected are equivalent.



$e_{n-2} = \{1, 2, 4\}$ means $f^{(1)}(\alpha_{n-2}) = f^{(2)}(\alpha_{n-2}) = f^{(4)}(\alpha_{n-2}) = y_{n-2}$

$e_{n-1} = \{5, 6\}$ means $f^{(5)}(\alpha_{n-1}) = f^{(6)}(\alpha_{n-1}) = y_{n-1}$

$e_n = \{7\}$ means $f^{(7)}(\alpha_n) = y_n$

Figure 1: Example edges from an agreement hypergraph $\mathcal{H} = ([7], (e_1, \ldots, e_n))$ (Definition 2.1) arising from a bad list-decoding configuration with polynomials $f^{(1)}, \ldots, f^{(7)} \in \mathbb{F}_q[X]$, received word $y \in \mathbb{F}_q^n$, and evaluation points $\alpha_1, \ldots, \alpha_n$.

4. With the hypergraph perspective, we can give a new presentation of the results in [BGM23] and more directly connect the list-decoding problem to the GM-MDS theorem [DSY14, Lov18, YH19], as the heavy-lifting in the combinatorics is done using known results on hypergraph orientations. This is done in Appendix A.

A hypergraph $\mathcal{H} = (V, \mathcal{E})$ is given by a set of vertices V and a set $\mathcal{E}$ of *(hyper)edges*, which are subsets of the vertices V. In this work, all hypergraphs have *labeled* edges, meaning we enumerate our edges e_i by distinct indices i from some set, typically $[n]$, in which case we may also think of $\mathcal{E}$ as a tuple $(e_1, \ldots, e_n)$. Throughout this paper, the vertex set V is typically $[t]$ for some positive integer t. The *weight* of a hyperedge e is $\mathrm{wt}(e) \overset{\text{def}}{=} \max(0, |e| - 1)$, and the *weight* of a set of hyperedges $\mathcal{E}$ is simply $\mathrm{wt}(\mathcal{E}) \overset{\text{def}}{=} \sum_{e \in \mathcal{E}} \mathrm{wt}(e)$.

All hypergraphs that we will consider in this work are *agreement hypergraphs* for a bad list-decoding configuration. See Figure 1 for an illustration.

Definition 2.1 (Agreement Hypergraph)**.** Given vectors $y, c^{(1)}, \ldots, c^{(t)} \in \mathbb{F}_q^n$, the *agreement hypergraph* has a vertex set $[t]$ and a tuple of n hyperedges $(e_1, \ldots, e_n)$ where $e_i \overset{\text{def}}{=} \{j \in [t] : c_i^j = y_i\}$.

A key property of hypergraphs that we are concerned with is weak-partition-connectivity.

Definition 2.2 (Weak Partition Connectivity)**.** A hypergraph $\mathcal{H} = ([t], \mathcal{E})$ is *k-weakly-partition-connected* if, for every partition $\mathcal{P}$ of the set of vertices $[t]$,

$$\sum_{e \in \mathcal{E}} \max\{|\mathcal{P}(e)| - 1, 0\} \geq k(|\mathcal{P}| - 1) \tag{2}$$

where $|\mathcal{P}|$ is the number of parts of the partition, and $|\mathcal{P}(e)|$ is the number of parts of the partition that edge e intersects.

To give some intuition for weak partition connectivity, we state two of its combinatorial implications. First, if a graph is k-weakly-partition-connected, then it is *k-edge-connected* [Kir03], which, by the Hypergraph Menger's (Max-Flow-Min-Cut) theorem [Kir03, Theorem 1.11], equivalently means that every pair of vertices has k edge-disjoint (hyper)paths between them.[6] Second, suppose we replace every hyperedge e with an arbitrary spanning tree of its vertices (which we effectively do in Definition 2.5). The resulting (non-hyper)graph is *k-partition-connected*,[7] which, by the Nash-Williams-Tutte Tree-Packing theorem [NW61, Tut61], equivalently means there are k edge-disjoint spanning trees (this connection was used in [GLS+22]).

The key reason we consider weak-partition-connectivity is that a bad list-decoding configuration yields a k-weakly-partition-connected agreement hypergraph.

[6] In general the converse is not true.

[7] In (non-hyper)graphs, k-partition-connectivity and k-weak-partition-connectivity are equivalent.

Lemma 2.3 (Bad list gives k-weakly-partition-connected hypergraph. See also Lemma 7.4 of [GLS+22])**.** *Suppose that vectors $y, c^{(1)}, \dots, c^{(L+1)} \in \mathbb{F}_q^n$ are such that the average Hamming distance from y to $c^{(1)}, \dots, c^{(L+1)}$ is at most $\frac{L}{L+1}(n-k)$. That is, $\sum_{j=1}^{L+1} d(y, c^{(j)}) \le L(n-k)$. Then, for some subset $J \subseteq [L+1]$ with $|J| \ge 2$, the agreement hypergraph of $(y, c^{(j)} : j \in J)$ is k-weakly-partition-connected.*

Proof. Consider the agreement hypergraph $([L+1], \mathcal{E})$ of $y, (c^{(1)}, \dots, c^{(L+1)})$. The edge weight is

$$\sum_{e \in \mathcal{E}} \mathrm{wt}(e) \ge -n + \sum_{e \in \mathcal{E}} |e| = -n + \sum_{i=1}^{n} \sum_{j=1}^{L+1} \mathbf{1}[y_i = c_i^{(j)}] = -n + \sum_{j=1}^{L+1} (n - d(y, c^{(j)})) \ge Lk. \tag{3}$$

Let J be an inclusion-minimal subset $J \subseteq [L+1]$ with $|J| \ge 2$ such that $\sum_{e \in \mathcal{E}} \mathrm{wt}(e \cap J) \ge (|J|-1)k$. By (3), $J = [L+1]$ works so J exists (note that singleton subsets of $[L+1]$ satisfy equality in the preceding inequality). Let $\mathcal{H} = (J, \mathcal{E}_J)$ be the agreement hypergraph for vectors $(y, c^{(j)} : j \in J)$. Note that the edges of $\mathcal{E}_J$ are exactly $(e_i \cap J : e_i \in \mathcal{E})$. By minimality of J, for all $J' \subsetneq J$, we have $\sum_{e \in \mathcal{E}_J} \mathrm{wt}(e \cap J') \le (|J'|-1)k$. Now, consider a non-trivial partition $\mathcal{P} = P_1 \sqcup \cdots \sqcup P_p$ of J where $P_i \ne J$ for all $i \in [p]$ (as otherwise (2) trivially follows). We have

$$\begin{aligned}
\sum_{e \in \mathcal{E}_J} \max\{|\mathcal{P}(e)| - 1, 0\} &= \sum_{e \in \mathcal{E}_J, e \ne \varnothing} \left(-1 + \sum_{\ell=1}^{p} \mathbf{1}[|e \cap P_\ell| > 0] \right) \\
&= \sum_{e \in \mathcal{E}_J, e \ne \varnothing} \left((|e| - 1) - \sum_{\ell=1}^{p} (|e \cap P_\ell| - \mathbf{1}[|e \cap P_\ell| > 0]) \right) \\
&= \sum_{e \in \mathcal{E}_J, e \ne \varnothing} \left(\max(|e| - 1, 0) - \sum_{\ell=1}^{p} \max(|e \cap P_\ell| - 1, 0) \right) \\
&= \sum_{e \in \mathcal{E}_J} \mathrm{wt}(e) - \sum_{\ell=1}^{p} \sum_{e \in \mathcal{E}_J} \mathrm{wt}(e \cap P_\ell) \\
&\ge (|J| - 1)k - \sum_{\ell=1}^{p} (|P_\ell| - 1)k \\
&= (p-1)k = (|\mathcal{P}| - 1)k.
\end{aligned} \tag{4}$$

This holds for all partitions $\mathcal{P}$ of J, so $\mathcal{H}_J$ is k-weakly-partition-connected. □

Remark 2.4. The condition $|J| \ge 2$ is needed later so that the reduced intersection matrix (defined below) is not a 0×0 matrix, in which case the matrix does not help establish list-decodability.

2.3 Reduced intersection matrices: definition and example

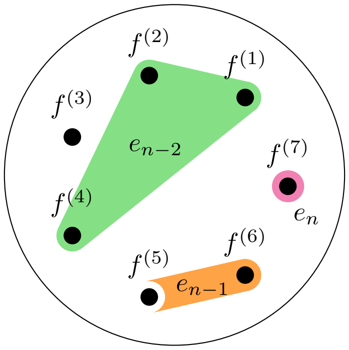
As in [GZ23], we work with the reduced intersection matrix, though our proof should work essentially the same with a different matrix called the (non-reduced) *intersection matrix*, which was considered in [ST20, GLS+22, BGM23].

Definition 2.5 (Reduced intersection matrix)**.** The *reduced intersection matrix* $\mathsf{RIM}_{q,\mathcal{H}}$ associated with a prime power q and a hypergraph $\mathcal{H} = ([t], (e_1, \dots, e_n))$ is a $\mathrm{wt}(\mathcal{E}) \times (t-1)k$ matrix over the field of fractions $\mathbb{F}_q(X_1, \dots, X_n)$. For each hyperedge e_i with vertices $j_1 < j_2 < \cdots < j_{|e_i|}$, we add $\mathrm{wt}(e_i) = |e_i| - 1$ rows to $\mathsf{RIM}_{\mathcal{H}}$. For $u = 2, \dots, |e_i|$, we add a row $r_{i,u} = (r^{(1)}, \dots, r^{(t-1)})$ of length $(t-1)k$ defined as follows:

- If $j = j_1$, then $r^{(j)} = [1, X_i, X_i^2, \dots, X_i^{k-1}]$
- If $j = j_u$ and $j_u \ne t$, then $r^{(j)} = -[1, X_i, X_i^2, \dots, X_i^{k-1}]$
- Otherwise, $r^{(j)} = 0^k$.

We typically omit q and write $\mathsf{RIM}_{\mathcal{H}}$ as q is typically understood.

Example 2.6. Recall the example edges of the agreement hypergraph $\mathcal{H} = ([7], (e_1, \dots, e_n))$ in Figure 1.



The edges e_{n-2}, e_{n-1}, e_n from $\mathcal{H}$ contribute the following length $(t-1)k$ rows to its reduced intersection matrix:

$$\begin{bmatrix} V_{n-2} & -V_{n-2} & 0 & 0 & 0 & 0 \\ V_{n-2} & 0 & 0 & -V_{n-2} & 0 & 0 \\ 0 & 0 & 0 & 0 & V_{n-1} & -V_{n-1} \end{bmatrix} \tag{5}$$

Here $V_i = [1, X_i, X_i^2, \dots, X_i^{k-1}]$ is a "Vandermonde row", and 0 denotes the length-k vector $[0, 0, \dots, 0]$. Note that each edge e contributes $|e| - 1$ rows to the agreement matrix, and in particular e_n does not contribute any rows.

Reduced intersection matrices arise by encoding all agreements from a bad list-decoding configuration into linear constraints on the message symbols (the polynomial coefficients). These constraints are placed into one matrix that we call the reduced intersection matrix. The following lemma implies that, if every reduced intersection matrix arising from a possible bad list-decoding configuration has full column rank when $X_1 = \alpha_1, \dots, X_n = \alpha_n$, the corresponding Reed–Solomon code is list-decodable.

Lemma 2.7 (RIM of agreement hypergraphs are not full column rank)**.** *Let $\mathcal{H}$ be an agreement hypergraph for $(y, c^{(1)}, \dots, c^{(t)})$, where $c^{(j)} \in \mathbb{F}_q^n$ are codewords of $RS_{n,k}(\alpha_1, \dots, \alpha_n)$, not all equal to each other. Then the reduced intersection matrix $\mathsf{RIM}_{\mathcal{H}}(X_{[n]} = \alpha_{[n]})$ does not have full column rank.*

Proof. By definition,

$$\mathsf{RIM}_{\mathcal{H}}(X_{[n]} = \alpha_{[n]}) \cdot \begin{bmatrix} f^{(1)} - f^{(t)} \\ \vdots \\ f^{(t-1)} - f^{(t)} \end{bmatrix} = 0 \tag{6}$$

where $f^{(1)}, \dots, f^{(t)} \in \mathbb{F}_q^k$ are the vectors of coefficients of the polynomials that generate codewords $c^{(1)}, \dots, c^{(t)} \in \mathbb{F}_q^n$. Since these vectors are not all equal to each other, $\mathsf{RIM}_{\mathcal{H}}(X_{[n]} = \alpha_{[n]})$ does not have full column rank. □

Remark 2.8 (Symmetries of reduced intersection matrices)**.** From this definition, it should be clear that we can divide the variables $X_1, \dots, X_n$ into at most 2^L classes such that variables in the same class are *exchangeable* with respect to the reduced intersection matrix $\mathsf{RIM}_{\mathcal{H}}$: if e_i and $e_{i'}$ are the same hyperedge, then swapping X_i and $X_{i'}$ yields the same reduced intersection matrix (up to row permutations). This observation, which was alluded to in [GZ23], turns out to be crucial in our argument that allows us to improve the alphabet size in [GZ23] from quadratic to linear.

Remark 2.9. The pairwise distinctness requirement in the definition of average-radius-list-decodability (see Section 1.1) is nonetheless crucial in the proof of Theorem 1.1, despite the weaker requirement in Lemma 2.7. That is because we will eventually apply Lemma 2.7 on the subcollection of codewords given from Lemma 2.3, which can potentially be arbitrary. The guarantee that this subcollection of codewords is not all equal to each other would then follow from pairwise distinctness of the codewords in the original list.

2.4 Reduced intersection matrices: full column rank

The following theorem shows that reduced intersection matrices of k-weakly-partition-connected hypergraphs are nonsingular when viewed as a matrix over $\mathbb{F}_q(X_1, \ldots, X_n)$. This was essentially conjectured by Shangguan and Tamo [ST20] and essentially established by Brakensiek, Gopi, and Makam [BGM23], who conjectured and showed, respectively, nonsingularity of the (non-reduced) intersection matrix under similar conditions. By the same union bound argument as in [ST20, Theorem 5.8], Theorem 2.10 already implies list-decodability of Reed–Solomon codes up to the generalized Singleton bound over exponentially large fields sizes, which is [BGM23, Theorem 1.5]. For completeness, and to demonstrate how the hypergraph perspective more directly connects the list-decoding problem to the GM-MDS theorem, we include a proof of Theorem 2.10 in Appendix A.

Theorem 2.10 (Full column rank. Implicit from Theorem A.2 of [BGM23])**.** *Let n and k be positive integers and $\mathbb{F}_q$ be a finite field. Let $\mathcal{H}$ be a k-weakly-partition-connected hypergraph with n hyperedges and at least 2 vertices. Then $\mathsf{RIM}_{\mathcal{H}}$ has full column rank over the field $\mathbb{F}_q(X_1, \cdots, X_n)$.*

Remark 2.11. We note that, [BGM23] assumes throughout their paper that the alphabet size q is sufficiently large, but Theorem 2.10 follows from the weaker "q sufficiently large" version: For any fixed field size q, take Q to be a sufficiently large power of q. Then, by the "q sufficiently large" version of Theorem 2.10, matrix $\mathsf{RIM}_{Q,\mathcal{H}}$ has full column rank over the field $\mathbb{F}_Q(X_1, \ldots, X_n)$. Hence, the determinant of some square full-rank submatrix of $\mathsf{RIM}_{Q,\mathcal{H}}$ is a nonzero polynomial in $\mathbb{F}_Q[X_1, \ldots, X_n]$. The entries of $\mathsf{RIM}_{Q,\mathcal{H}}$ can all be viewed as polynomials over $\mathbb{F}_q$, so the corresponding full-rank submatrix of $\mathsf{RIM}_{q,\mathcal{H}}$ has a determinant that is a nonzero polynomial in $\mathbb{F}_q[X_1, \ldots, X_n]$ — symbolically, the determinants are the same polynomials, as $\mathbb{F}_q$ and $\mathbb{F}_Q$ have the same characteristic. Hence, the matrix $\mathsf{RIM}_{q,\mathcal{H}}$ has full column rank over the field $\mathbb{F}_q(X_1, \ldots, X_n)$.

2.5 Reduced intersection matrix: row deletions

As in [GZ23], we consider row deletions from the reduced intersection matrix. The goal of this section is to establish Lemma 2.13, that the full-column-rank-ness of reduced intersection matrices are robust to row deletions.

Definition 2.12 (Row deletion of reduced intersection matrix)**.** Given a hypergraph $\mathcal{H} = ([t], (e_1, \ldots, e_n))$ and set $B \subseteq [n]$, define $\mathsf{RIM}_{\mathcal{H}}^B$ to be the submatrix of $\mathsf{RIM}_{\mathcal{H}}$ obtained by deleting all rows containing a variable X_i with $i \in B$.

The next lemma appears in a weaker form in [GZ23]. It roughly says that, given a reduced intersection matrix $\mathsf{RIM}_{\mathcal{H}}$ with some constant factor "slack" in the combinatorial constraints, we can omit a constant fraction of the rows without compromising the full-column-rank-ness of the matrix. Our version of this lemma saves roughly a factor of $t \sim L$ compared to the analogous lemma [GZ23, Lemma 3.11]. The reason is that the k-weakly-partition-connected condition is more robust to these row deletions (by a factor of roughly t) than the condition in [GZ23]. As such, our proof is also more direct.

Lemma 2.13 (Robustness to deletions. Similar to Lemma 3.11 of [GZ23])**.** *Let $\mathcal{H} = ([t], \mathcal{E})$ be a $(k + \varepsilon n)$-weakly-partition-connected hypergraph with $t \geq 2$. For all sets $B \subset [n]$ with $|B| \leq \varepsilon n$, we have that $\mathsf{RIM}_{\mathcal{H}}^B$ is nonempty and has full column rank.*

Proof. By definition of the reduced intersection matrix $\mathsf{RIM}_{\mathcal{H}}$, the matrix with row deletions $\mathsf{RIM}_{\mathcal{H}}^B$ is the matrix $\mathsf{RIM}_{\mathcal{H}'}$, where $\mathcal{H}' = ([t], \mathcal{E}')$ is the hypergraph obtained from $\mathcal{H}$ by deleting e_i for $i \in B$. By Theorem 2.10, it suffices to prove that $\mathcal{H}'$ is k-weakly-partition connected. Indeed, consider any partition $\mathcal{P}$ of $[t]$. We have

$$\begin{aligned}\sum_{e \in \mathcal{E}'} \max\{|\mathcal{P}(e)| - 1, 0\} &= \sum_{i \in [n]} \max\{|\mathcal{P}(e)| - 1, 0\} - \sum_{i \in B} \max\{|\mathcal{P}(e)| - 1, 0\} \\ &\geq (k + \varepsilon n) \cdot (|\mathcal{P}| - 1) - |B| \cdot (|\mathcal{P}| - 1) = k \cdot (|\mathcal{P}| - 1), \qquad (7)\end{aligned}$$

as desired. The first inequality holds because $\mathcal{H}$ is $(k + \varepsilon n)$-weakly-partition-connected, and, trivially, any edge e_i touches at most $|\mathcal{P}|$ parts of $\mathcal{P}$. □

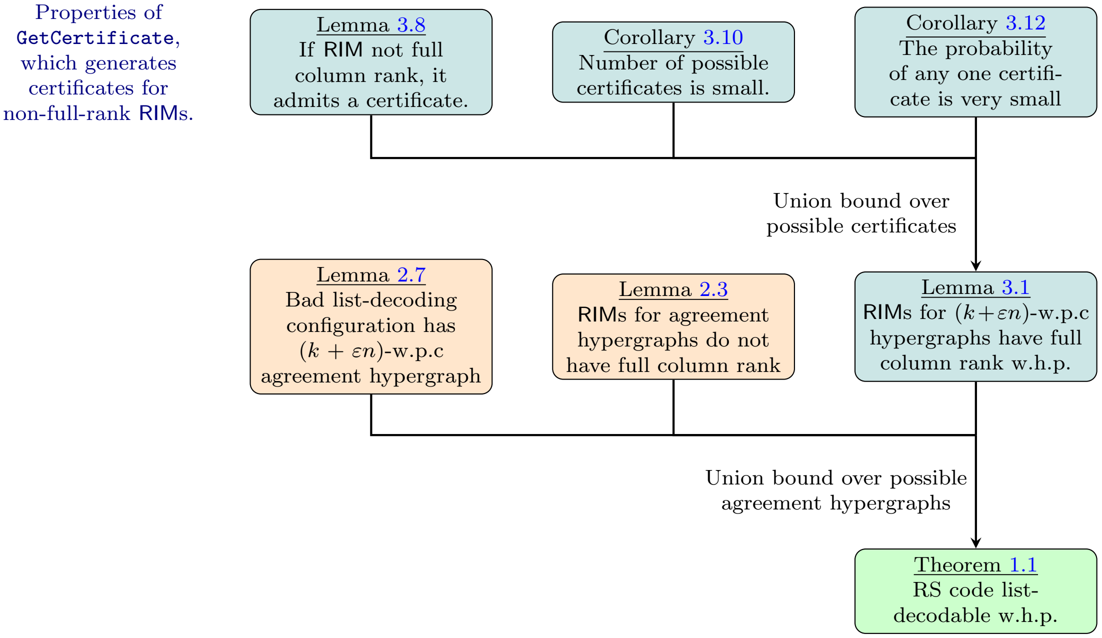

Properties of GetCertificate, which generates certificates for non-full-rank RIMs.
Lemma 3.8
If RIM not full column rank, it admits a certificate.
Corollary 3.10
Number of possible certificates is small.
Corollary 3.12
The probability of any one certificate is very small
Union bound over possible certificates
Lemma 2.7
Bad list-decoding configuration has $(k+\varepsilon n)$-w.p.c agreement hypergraph
Lemma 2.3
RIMs for agreement hypergraphs do not have full column rank
Lemma 3.1
RIMs for $(k+\varepsilon n)$-w.p.c hypergraphs have full column rank w.h.p.
Union bound over possible agreement hypergraphs
Theorem 1.1
RS code list-decodable w.h.p.

Figure 2: A roadmap of our proof. The orange boxes are preliminaries, and the blue-green boxes are the meat of the proof address in Section 3. All probabilities are over the random choice of evaluation points $\alpha_1, \dots, \alpha_n$ for our Reed–Solomon code.

3 Proof of list-decodability with linear-sized alphabets

3.1 Overview of the proof

By Lemma 2.7 and Lemma 2.3, every bad list-decoding configuration admits a weakly-partition-connected agreement hypergraph whose reduced intersection matrix does not have full column rank. Thus, to prove Theorem 1.1, it suffices to show that, with high probability, every such reduced intersection matrix has full column rank. The main technical lemma for this section is the one stated below. Our main result, Theorem 1.1, follows by applying Lemma 2.3 and Lemma 2.7 with Lemma 3.1, and taking a union bound over all $\sum_{t=2}^{L+1} 2^{tn}$ possible agreement hypergraphs.

Lemma 3.1. *Let k be a positive integer and $\varepsilon > 0$. For each $(k+\varepsilon n)$-weakly-partition-connected hypergraph $\mathcal{H} = ([t], (e_1, \dots, e_n))$ with $t \geq 2$, we have, for $r = \lfloor \varepsilon n/2 \rfloor$,*

$$\Pr_{\alpha_1,\dots,\alpha_n \sim \mathbb{F}_q \text{ distinct}} \left[\mathsf{RIM}_{\mathcal{H}}(X_{[n]} = \alpha_{[n]}) \text{ does not have full column rank}\right] \leq \binom{n}{r} 2^{tr} \cdot \left(\frac{(t-1)k}{q-n}\right)^r . \quad (8)$$

At the highest level, the proof of Lemma 3.1 follows the same outline as [GZ23]. For every sequence of evaluation points $(\alpha_1, \dots, \alpha_n) \in \mathbb{F}_q^n$ for which $\mathsf{RIM}_{\mathcal{H}}$ does not have full column rank, we show that there is a *certificate* $(i_1, \dots, i_r) \in [n]^r$ consisting of distinct indices in $[n]$ (Lemma 3.8), which intuitively "attests" to the failure of the matrix $\mathsf{RIM}_{\mathcal{H}}$ to be full column rank. We then show that, for any certificate $(i_1, \dots, i_r)$, the probability that $(\alpha_1, \dots, \alpha_n)$ has certificate $(i_1, \dots, i_r)$ is exponentially small. (More precisely, it will at most be $(\frac{(t-1)k}{q-n})^r$. See Corollary 3.12). We then show that there are not too many certificates (Corollary 3.10), and then union bound over the number of possible certificates to obtain the desired result (Lemma 3.1).

Our argument differs from [GZ23] in how we choose our certificates. The argument of [GZ23] allowed for up to n^r certificates. Our argument instead only needs $\binom{n}{r}2^{tr}$ many certificates, which is much smaller when $r = \Omega(n)$ (the parameter regime of interest here) and overall allows us to save a factor of n in the alphabet size. Our savings comes from leveraging that there are at most 2^t different "types" of hyperedges (see Remark 2.8), and thus at most 2^t different types of variables X_i in the reduced intersection matrix $\mathsf{RIM}_{\mathcal{H}}$. This observation was alluded to in [GZ23].[8] With this observation in mind, we assume, without loss of generality, that the edges of $\mathcal{H}$ are ordered by their respective type (we can relabel the edges of $\mathcal{H}$, which effectively permutes the rows of $\mathsf{RIM}_{\mathcal{H}}$).

Our method of generating a certificate $(i_1, \ldots, i_r)$ for the evaluation sequence $(\alpha_1, \ldots, \alpha_n)$ (Algorithm 2) is similar to that of [GZ23] at a high level—with each certificate $i_1, \ldots, i_r$, we associate a sequence of $(t-1)k \times (t-1)k$ submatrices $M_1, \ldots, M_r$ of $\mathsf{RIM}_{\mathcal{H}}$ (Algorithm 1) that are entirely specified by $i_1, \ldots, i_r$ as follows: since evaluating $X_{[n]} = \alpha_{[n]}$ forces $\mathsf{RIM}_{\mathcal{H}}$ to not be full rank, then so will all of its $(t-1)k \times (t-1)k$ submatrices. Thus if we sequentially 'reveal' $X_1 = \alpha_1, X_2 = \alpha_2, \ldots$, then at some point, M_j becomes singular exactly when we set $X_{i_j} = \alpha_{i_j}$ — in fact, i_j is defined as such, so that we select $M_1, i_1, M_2, i_2, \ldots$, in that order, but we emphasize that M_j can be computed from $i_1, \ldots, i_{j-1}$ without knowing $\alpha_1, \ldots, \alpha_n$. Conditioned on M_j being non-singular with $X_1 = \alpha_1, \ldots, X_{i_j-1} = \alpha_{i_j-1}$, the probability that M_j becomes singular when setting $X_{i_j} = \alpha_{i_j}$ is at most $\frac{(t-1)k}{q-n}$: α_{i_j} is uniformly random over at least $q-n$ field elements, and the degree of X_{i_j} in the determinant of M_j is at most $(t-1)k$ (and the determinant is nonzero by definition). Running conditional probabilities in the correct order, we conclude that the probability that a particular certificate $i_1, \ldots, i_r$ is generated is at most $(\frac{(t-1)k}{q-n})^r$, just as in [GZ23].

Whereas [GZ23] pick any matrix M_j that is obtained after removing the variables $X_{i_1}, \ldots, X_{i_{j-1}}$, we do a more deliberate choice of matrices by leveraging the symmetries of $\mathsf{RIM}_{\mathcal{H}}$ (Remark 2.8). First, we ensure that we can keep a "bank" of $\Omega_t(r)$ unused variables of each of the $O_t(1)$ types. Then, starting with a full column rank submatrix M of $\mathsf{RIM}_{\mathcal{H}}$ devoid of all variables in the "bank," we start sequentially applying the evaluations $X_1 = \alpha_1, X_2 = \alpha_2, \ldots$. Whenever $M(X_{\le i_1} = \alpha_{\le i_1})$ turns singular, we find that the evaluation $X_{i_1} = \alpha_{i_1}$ is what 'caused' it to become singular. We then go to the "bank" to find a variable $X_{i'_1}$ of the same type as X_{i_1} and "re-indeterminate" M by replacing all instances of X_{i_1} in M with $X_{i'_1}$. That way, we ensure that M is, in a sense, "reused." Furthermore, we ensure $i'_1 > i_1$, so that the matrix $M(X_{\le i_1} = \alpha_{\le i_1})$ is now nonsingular, so we can keep going. Of course, if we end up reaching the end (i.e. $M(X_{[n]} = \alpha_{[n]})$ is full column rank), then in fact, $\mathsf{RIM}_{\mathcal{H}}(X_{[n]} = \alpha_{[n]})$ is full column rank, and so the evaluations $(\alpha_1, \ldots, \alpha_n)$ were 'good' after all.

Otherwise, if the evaluations $(\alpha_1, \ldots, \alpha_n)$ were 'bad', then the submatrix M couldn't have reached the end, and that can only happen if some specific type was completely exhausted from the bank. However, given the size of our initial bank, that must have meant that M must have been "re-indeterminated" at least $\Omega_t(r)$ times. When that happens, we collect the indices $i_1, \ldots, i_\ell$ that we gathered from this round, remove them from $\mathsf{RIM}_{\mathcal{H}}$, and repeat the process again with a refreshed bank. Since we only need r indices, then we end up doing at most $O_t(1)$ rounds. Because each round yields a strictly increasing sequence of indices of length at least $\Omega_t(r)$, then we up getting a certificate consisting of at most $O_t(1)$ strictly increasing runs of total length r, of which there are at most $\binom{n}{r} \cdot O_t(1)^r$.

To be more concrete, when we generate the submatrix $M = M_1$, we ensure that any variable appearing in M_1 has the same type as $\Omega_t(r)$ variables that are *not* in M_1 (but still in $\mathsf{RIM}_{\mathcal{H}}$). This creates a "bank" of variables of each type. Then, if $X_{\le i_1} = \alpha_{\le i_1}$ was the point that made M_1 singular, we can get M_2 by replacing all copies of X_{i_1} with some $X_{i'_1}$ that is of the same type and in the "bank." Since variables i_1 and i'_1 are of the same type, they have analogous rows in the reduced intersection matrix $\mathsf{RIM}_{\mathcal{H}}$, so this new matrix M_2 is still a submatrix of $\mathsf{RIM}_{\mathcal{H}}$. Therefore, we can pick up where we left off with M_1 but with M_2 instead. That is, M_2 will in fact be full rank when we apply the evaluations $X_{\le i_1} = \alpha_{\le i_1}$. Thus the next index i_2 on which M_2 turns singular will be strictly greater than i_1. We then repeat the process in M_2, replacing X_{i_2} with some $X_{i'_2}$ that is in the "bank" and of the same type, getting M_3, and so on. We can continue this process for $\Omega_t(r)$ steps because of the size of the bank of each type, so we get an increasing run of length $\Omega_t(r)$ in our certificate. After we run out of some type in our bank, we remove the used indices $i_1, \ldots, i_\ell$ from $\mathsf{RIM}_{\mathcal{H}}$ and repeat the process again with a refreshed bank. This continues for $O_t(1)$ times only, as we only need r indices in the end.

[8] Guo and Zhang [GZ23] write "It is possible that achieving an alphabet size linear in n would require establishing and exploiting other properties of intersection matrices or reduced intersection matrices, such as an appropriate notion of exchangeability." We found this prediction to be insightful and true.

We now finish the proof of Theorem 1.1, assuming Lemma 3.1. The rest of this section is devoted to proving Lemma 3.1.

Proof of Theorem 1.1, assuming Lemma 3.1. By Lemma 2.3, if $RS_{n,k}(\alpha_1,\dots,\alpha_n)$ is not $\left(\frac{L}{L+1}(1-R-\varepsilon),L\right)$ average-radius list-decodable, then there exists a vector y and pairwise distinct codewords $c^{(1)},\dots,c^{(t)}$ with $t\ge 2$ such that the agreement hypergraph $\mathcal{H}=([t],\mathcal{E})$ is $(R+\varepsilon)n=(k+\varepsilon n)$-weakly-partition-connected. By Lemma 2.7, the matrix $\mathsf{RIM}_{\mathcal{H}}(X_{[n]}=\alpha_{[n]})$ is not full column rank. Now, the number of possible agreement hypergraphs $\mathcal{H}$ is at most $\sum_{t=2}^{L+1}2^{tn}\le 2^{(L+2)n}$. Thus by the union bound over possible agreement hypergraphs $\mathcal{H}$ with Lemma 3.1, we have, for $r=\lfloor\frac{\varepsilon n}{2}\rfloor$,

$$\begin{aligned}
&\mathop{\mathbf{Pr}}_{\alpha_{[n]}}\left[RS_{n,k}(\alpha_1,\dots,\alpha_n)\text{ not }\left(\frac{L}{L+1}(1-R-\varepsilon),L\right)\text{ list-decodable}\right]\\
&\le \mathop{\mathbf{Pr}}_{\alpha_{[n]}}\left[\exists\ (k+\varepsilon n)\text{-w.p.c. agreement hypergraph }\mathcal{H}\text{ such that }\mathsf{RIM}_{\mathcal{H}}(X_{[n]}=\alpha_{[n]})\text{ not full column rank}\right]\\
&\le 2^{(L+2)n}\max_{(k+\varepsilon n)\text{-w.p.c. }\mathcal{H}}\ \mathop{\mathbf{Pr}}_{\alpha_{[n]}}\left[\mathsf{RIM}_{\mathcal{H}}(X_{[n]}=\alpha_{[n]})\text{ not full column rank}\right]\\
&\le 2^{(L+2)n}\cdot\binom{n}{r}2^{(L+1)r}\left(\frac{Lk}{q-n}\right)^r\le\left(2^{(L+2)n/r}\cdot\frac{en}{r}\cdot 2^{L+1}\frac{Lk}{q-n}\right)^r\le 2^{-Ln},
\end{aligned}\tag{9}$$

as desired. Here, we used that $q=n+k\cdot 2^{10L/\varepsilon}$. □

3.2 Setup for proof of Lemma 3.1

We now devote the rest of this Section to proving Lemma 3.1.

Types. For a hypergraph $\mathcal{H}=([t],(e_1,\dots,e_n))$, the *type* of an index i (or, by abuse of notation, the type of the variable X_i, or the edge e_i) is simply the set $e_i\subset[t]$. There are 2^t types, and by abuse of notation, we identify the types by the numbers $1,2,\dots,2^t$ in an arbitrary fixed order with a bijection $\tau:(\text{subsets of }[t])\to[2^t]$. We say a hypergraph is *type-ordered* if the hyperedges $e_1,\dots,e_n$ are sorted according to their type: $\tau(e_1)\le\tau(e_2)\le\cdots\le\tau(e_n)$. Since permuting the labels of the edges of $\mathcal{H}$ preserves the rank of $\mathsf{RIM}_{\mathcal{H}}$ (it merely permutes the rows of $\mathsf{RIM}_{\mathcal{H}}$), we can without loss of generality assume in Lemma 3.1 that $\mathcal{H}$ is type-ordered.

Global variables. Throughout the rest of the section, we fix a positive integer k, parameter $\varepsilon>0$, and $\mathcal{H}=([t],(e_1,\dots,e_n))$, a type-ordered $(k+\varepsilon n)$-weakly-partition-connected hypergraph with $t\ge 2$. We also fix

$$r\stackrel{\text{def}}{=}\left\lfloor\frac{\varepsilon n}{2}\right\rfloor.\tag{10}$$

3.3 `GetCertificate` and `GetMatrixSequence`: Basic properties

As mentioned at the beginning of this section, we design an algorithm, Algorithm 2, that attempts to generate a certificate $(i_1,\dots,i_r)\in[n]^r$ for evaluation points $\alpha_1,\dots,\alpha_n$. It uses Algorithm 1, a helper function that generates the associated square submatrices $M_1,\dots,M_r$ of $\mathsf{RIM}_{\mathcal{H}}$. Below, we establish some basic properties of these algorithms.

First, we establish that the matrices outputted by `GetMatrixSequence` are well-defined.

Lemma 3.2 (Output is well-defined)**.** *For all sequence of indices $i_1,\dots,i_{j-1}$, if $M_1,\dots,M_j$ is the output of the function* `GetMatrixSequence`$(i_1,\dots,i_{j-1})$*, then $M_1,\dots,M_j$ are well-defined.*

Proof. If ℓ is a refresh index, then we have $|B\cup\{i_1,\dots,i_{\ell-1}\}|<|B|+r\le 2r\le\varepsilon n$, so by Lemma 2.13, $\mathsf{RIM}_{\mathcal{H}}^{B\cup\{i_1,\dots,i_{\ell-1}\}}$ is nonempty and has full column rank. Thus M_ℓ exists in Line 13. If ℓ is not a refresh index, M_ℓ is always well-defined by definition. □

```
Algorithm 1: GetMatrixSequence
   Input: indices $i_1, \dots, i_{j-1} \in [n]$ for some $j \geq 1$.
   Output: $M_1, \dots, M_j$, which are $(t-1)k \times (t-1)k$ matrices over $\mathbb{F}_q(X_1, X_2, \dots, X_n)$.
1  $B \leftarrow \emptyset$, $i_0 \leftarrow \perp$, $\ell_0 \leftarrow \perp$
2  for $\ell = 1, \dots, j$ do
       // $M_\ell$ depends only on $i_1, \dots, i_{\ell-1}$
3      if $\ell > 1$ then
           // Fetch new index from bank $B$
4          $\tau \leftarrow$ the type of $i_{\ell-1}$
5          $s \leftarrow$ number of indices among $i_{\ell_0}, i_{\ell_0+1}, \dots, i_{\ell-1}$ that are type $\tau$
6          $i'_{\ell-1} \leftarrow$ the $s$-th smallest element of $B$ that has type $\tau$
7          if $i'_{\ell-1}$ is defined then
8              $M_\ell \leftarrow$ the matrix obtained from $M_{\ell-1}$ by replacing all copies of $X_{i_{\ell-1}}$ with $X_{i'_{\ell-1}}$
9      if $M_\ell$ not yet defined then
           // Refresh bank $B$
10         $B \leftarrow \emptyset$
11         for $\tau = 1, \dots, 2^t$ do
12             $B \leftarrow B \cup \{$largest $\lfloor r/2^t \rfloor$ indices of type $\tau$ in $[n] \setminus \{i_1, \dots, i_{\ell-1}\}\}$ (if there are less than
                 $\lfloor r/2^t \rfloor$ indices of type $\tau$, then $B$ contains all such indices)
13         $M_\ell \leftarrow$ lexicographically smallest nonsingular $(t-1)k \times (t-1)k$ submatrix of $\mathsf{RIM}_{\mathcal{H}}^{B \cup \{i_1, \dots, i_{\ell-1}\}}$
14         $\ell_0 \leftarrow \ell$ // new refresh index
15
16 return $M_1, \dots, M_j$
```

Next, we observe that `GetMatrixSequence` is an "online" algorithm.

Lemma 3.3 (Online)**.** *Furthermore,* `GetMatrixSequence` *is a deterministic function of $i_1, \dots, i_{j-1}$, and it computes M_ℓ "online", meaning M_ℓ depends only on $i_1, \dots, i_{\ell-1}$ for all $\ell = 1, \dots, j$ (and M_1 is always the same matrix). In particular,* `GetMatrixSequence`$(i_1, \dots, i_{j-1})$ *is a prefix of* `GetMatrixSequence`$(i_1, \dots, i_j)$*.*

Proof. By definition and Lemma 3.2. □

Definition 3.4 (Refresh index)**.** In `GetMatrixSequence`, in the outer loop over ℓ, we say a *refresh index* is an index ℓ obtained at Line 14 (i.e. when M_ℓ is defined on Line 13). For example, $\ell = 1$ is a refresh index.

Our first lemma shows that the new indices we are receiving from `GetMatrixSequence` are in fact new.

Lemma 3.5 (New Variable)**.** *In* `GetMatrixSequence`*, in the outer loop iteration over ℓ at Line 2, if we reach Line 8 of* `GetMatrixSequence`*, variable $X_{i'_{\ell-1}}$ does not appear in $M_{\ell_0}, M_{\ell_0+1}, \dots, M_{\ell-1}$, where ℓ_0 is the largest refresh index less than ℓ.*

Proof. Let B be the set defined in Line 12 at iteration ℓ_0. In iterations $\ell' = \ell_0, \ell_0 + 1, \dots, \ell$, the set B is the same, and $i'_{\ell-1}$ is in this set B by definition. Thus, the variable $X_{i'_{\ell-1}}$ does not appear in M_{ℓ_0} by definition. For $\ell' = \ell_0, \ell_0 + 1, \dots, \ell$, the (τ, s) pairs generated at Line 4 and Line 5 are pairwise distinct, so $X_{i'_{\ell-1}}$ is not added to $M_{\ell'}$ for $\ell' = \ell_0 + 1, \dots, \ell - 1$ and thus is not in $M_{\ell_0}, M_{\ell_0+1}, \dots, M_{\ell-1}$. □

To show that the probability of a particular certificate $(i_1, \dots, i_r)$ is small (Lemma 3.11, Corollary 3.12), we crucially need that $i_1, \dots, i_r$ are pairwise distinct. The next lemma proves that this is always the case.

Lemma 3.6 (Distinct indices)**.** *For any sequence of evaluation points $(\alpha_1, \dots, \alpha_n) \in \mathbb{F}_q^n$, the output of* `GetCertificate`$(\alpha_1, \dots, \alpha_n)$ *is a sequence $(i_1, \dots, i_r) \in [n]^r$ of pairwise distinct indices.*

```
Algorithm 2: GetCertificate
  Input: Evaluation points $(\alpha_1, \dots, \alpha_n) \in \mathbb{F}_q^n$.
  Output: A "certificate" $(i_1, \dots, i_r) \in [n]^r$.
1 for $j = 1, \dots, r$ do
      // $M_1, \dots, M_{j-1}$ stay the same, $M_j$ is now defined
2     $M_1, \dots, M_j =$ GetMatrixSequence$(i_1, \dots, i_{j-1})$
3     $i_j \leftarrow$ smallest index $i$ such that $M_j(X_{\leq i} = \alpha_{\leq i})$ is singular
4     if $i_j$ not defined then
5         return $\perp$
6 return $(i_1, \dots, i_r)$
```

Proof. By definition of i_ℓ at Line 3 of `GetCertificate`, variable X_{i_ℓ} must be in M_ℓ, so suffices to show that M_ℓ never contains any variable X_i for $i \in \{i_1, \dots, i_{\ell-1}\}$. We induct on ℓ. If ℓ is a refresh index, this is true by definition. If not, let ℓ_0 be the largest refresh index less than ℓ. By induction, $i_1, \dots, i_{\ell-2}$ are not in $M_{\ell-1}$, so we just need to show $i'_{\ell-1}$ (the new index replacing $i_{\ell-1}$ in M_ℓ at Line 8) is not any of $i_1, \dots, i_{\ell-1}$. It is not any of $i_1, \dots, i_{\ell_0-1}$ because none of those indices are in B by definition. It is not any of $i_{\ell'}$ for $\ell' = \ell_0, \dots, \ell - 1$, because $X_{i_{\ell'}}$ is in $M_{\ell'}$, but $X_{i'_{\ell-1}}$ is not, by Lemma 3.5 . □

3.4 Bad evaluation points admit certificates

Here, we establish Lemma 3.8, that if some evaluation points make $\mathsf{RIM}_{\mathcal{H}}$ not full column rank, then `GetCertificate` outputs a certificate. To do so, we first justify our matrix constructions, showing that the matrices in `GetMatrixSequence` are in fact submatrices of $\mathsf{RIM}_{\mathcal{H}}$.

Lemma 3.7 (`GetMatrixSequence` gives submatrices of $\mathsf{RIM}_{\mathcal{H}}$)**.** *For all sequence of indices $i_1, \dots, i_{j-1}$, if $M_1, \dots, M_j$ is the output of* `GetMatrixSequence`*$(i_1, \dots, i_{j-1})$, then $M_1, \dots, M_j$ are $(t-1)k \times (t-1)k$ submatrices of $\mathsf{RIM}_{\mathcal{H}}$.*

Proof. We proceed with induction on $\ell = 1, \dots, j$. First, if ℓ is a refresh index, then M_ℓ is a submatrix of $\mathsf{RIM}_{\mathcal{H}}$ by definition. In particular, M_1 is a submatrix of $\mathsf{RIM}_{\mathcal{H}}$, so the base case holds. Now suppose ℓ is not a refresh index and $M_{\ell-1}$ is a submatrix of $\mathsf{RIM}_{\mathcal{H}}$. Matrix M_ℓ is defined by replacing all copies of $X_{i_{\ell-1}}$ with $X_{i'_{\ell-1}}$. To check that M_ℓ is a submatrix of $\mathsf{RIM}_{\mathcal{H}}$, it suffices to show that

(i) for each row of $\mathsf{RIM}_{\mathcal{H}}$ containing $X_{i_{\ell-1}}$, replacing all copies of $X_{i_{\ell-1}}$ with $X_{i'_{\ell-1}}$ gives another row of $\mathsf{RIM}_{\mathcal{H}}$, and

(ii) the variable $X_{i'_{\ell-1}}$ does not appear in $M_{\ell-1}$.

The first item follows from the fact that indices $i_{\ell-1}$ and $i'_{\ell-1}$ are of the same type, so (i) holds by definition of types and $\mathsf{RIM}_{\mathcal{H}}$ (see also Remark 2.8). The second item is Lemma 3.5. Thus, M_ℓ is a submatrix of $\mathsf{RIM}_{\mathcal{H}}$, completing the induction. □

We now show that any n-tuple of bad evaluation points admits a certificate.

Lemma 3.8 (Bad evaluations points admit certificates)**.** *If $(\alpha_1, \dots, \alpha_n) \in \mathbb{F}_q^n$ are evaluation points such that $\mathsf{RIM}_{\mathcal{H}}(X_{[n]} = \alpha_{[n]})$ does not have full column rank,* `GetCertificate`*$(\alpha_1, \dots, \alpha_n)$ returns a certificate $(i_1, \dots, i_r) \in [n]^r$ (rather than $\perp$).*

Proof. Suppose for contradiction that `GetCertificate` returns $\perp$ at iteration j in the loop. Then there is no index i such that $M_j(X_{\leq i} = \alpha_{\leq i})$ is singular, so in particular, $M_j(X_{[n]} = \alpha_{[n]})$ is nonsingular and thus has full column rank. By Lemma 3.7, M_j is a submatrix of $\mathsf{RIM}_{\mathcal{H}}$, so we conclude $\mathsf{RIM}_{\mathcal{H}}$ has full column rank. □

3.5 Bounding the number of possible certificates

In this section, we upper bound the number of possible certificates. The key step is to prove the following structural result about certificates.

Lemma 3.9 (Certificate structure)**.** *Given a sequence of evaluation points $(\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_q^n$ such that $\mathsf{RIM}_{\mathcal{H}}(X_{[n]} = \alpha_{[n]})$ is not full column rank, the return value $(i_1, \ldots, i_r) = \texttt{GetCertificate}(\alpha_1, \ldots, \alpha_n)$ satisfies $i_{j-1} < i_j$ for all but at most 2^t values $j = 2, \ldots, r$.*

Proof. Let $(i_1, \ldots, i_r)$ be the return of `GetCertificate`, and let $M_1, \ldots, M_r$ be the associated matrix sequence. By Lemma 3.3, we have $M_1, \ldots, M_j = \texttt{GetMatrixSequence}(i_1, \ldots, i_{j-1})$ for $j = 1, \ldots, r$. Recall an index $\ell \in [r]$ is a *refresh index* if M_ℓ is defined on Line 13 rather than Line 8. The lemma follows from two claims:

(i) If $\ell > 1$ is not a refresh index, then $i_{\ell-1} < i_\ell$.

(ii) Any two refresh indices differ by at least $r/2^t$.

To see claim (i), let ℓ_0 be the largest refresh index less than ℓ. By definition of a refresh index, the set B stays constant between when M_{ℓ_0} is defined and when M_ℓ is defined. From the definition of i_j at Line 3 in `GetCertificate`, we know that

- For $i < i_{\ell-1}$ the matrix $M_{\ell-1}(X_{\leq i} = \alpha_{\leq i})$ is nonsingular.
- The matrix $M_\ell(X_{\leq i_\ell} = \alpha_{\leq i_\ell})$ is singular.

Suppose for contradiction that $i_\ell < i_{\ell-1}$. (Note that $i_{\ell-1} \neq i_\ell$ by Lemma 3.6.) We contradict the first item by showing, using the second item, that $M_{\ell-1}(X_{\leq i_\ell} = \alpha_{\leq i_\ell})$ is also singular. By the definition of `GetMatrixSequence`, since ℓ is not a refresh index, M_ℓ is defined in Line 8. By construction of B and $i'_{\ell-1}$, we know that $i'_{\ell-1} > i_{\ell-1} > i_\ell$. Thus, not only is M_ℓ obtained from $M_{\ell-1}$ by replacing all copies of $X_{i_{\ell-1}}$ with $X_{i'_{\ell-1}}$, but $M_\ell(X_{\leq i_\ell} = \alpha_{\leq i_\ell})$ is also obtained by replacing all copies of $X_{i_{\ell-1}}$ with $X_{i'_{\ell-1}}$ in $M_{\ell-1}(X_{\leq i_\ell} = \alpha_{\leq i_\ell})$. Moreover, the variable $X_{i'_{\ell-1}}$ does not appear in $M_{\ell-1}$ by Lemma 3.5. So we conclude that, as $M_\ell(X_{\leq i_\ell} = \alpha_{\leq i_\ell})$ is singular, so is $M_{\ell-1}(X_{\leq i_\ell} = \alpha_{\leq i_\ell})$.

Now we show claim (ii). Suppose ℓ_0 and ℓ_1 are consecutive refresh indices. If a variable of type τ appears in the matrix M_{ℓ_0}, there must be exactly $\lfloor r/2^t \rfloor$ indices of type τ in B (if there were fewer, then $B \cup \{i_1, \ldots, i_{\ell-1}\}$ would contain all indices of type τ, and the corresponding variables would not appear in $\mathsf{RIM}_{\mathcal{H}}^{B\cup\{i_1,\ldots,i_{\ell-1}\}}$). Let τ be the type of index i_{ℓ_1-1}. Since ℓ_1 is a refresh index, the number of indices of type τ among $i_{\ell_0}, i_{\ell_0+1}, \ldots, i_{\ell_1-1}$ must therefore be $\lfloor r/2^t \rfloor + 1$. In particular, this means $\ell_1 - \ell_0 \geq \lfloor r/2^t \rfloor + 1 \geq r/2^t$, as desired. □

Corollary 3.10 (Certificate count)**.** *The number of possible outputs to `GetCertificate` is at most $\binom{n}{r}2^{tr}$.*

Proof. The certificate consists of r distinct indices of $[n]$ by Lemma 3.6. We can choose those in $\binom{n}{r}$ ways. These indices are distributed between at most 2^t increasing runs by Lemma 3.9. We can distribute these indices between the 2^t increasing runs in at most $(2^t)^r$ ways. □

3.6 Bounding the probability of one certificate

The goal of this section is to establish Corollary 3.12, which states that the probability of obtaining a particular certificate is at most $(\frac{(t-1)k}{q-n})^r$. The argument is implicit in [GZ23], but we include a proof for completeness.

Lemma 3.11 (Implicit in [GZ23])**.** *Let $i_1, \ldots, i_r \in [n]$ be pairwise distinct indices, and $M_1, \ldots, M_r$ be $(t-1)k \times (t-1)k$ submatrices of $\mathsf{RIM}_{\mathcal{H}}$. Over randomly chosen pairwise distinct evaluation points $\alpha_1, \ldots \alpha_n \in \mathbb{F}_q$, define the following events for $j = 1, \ldots, r$:*

- *E_j is the event that $M_j(X_{\leq i} = \alpha_{\leq i})$ is non-singular for all $i < i_j$.*

- *F_j is the event that $M_j(X_{\le i_j} = \alpha_{\le i_j})$ is singular.*

The probability that all the events hold is at most $(\frac{(t-1)k}{q-n})^r$.

Proof. Note that the set of evaluation points $\alpha_1, \dots, \alpha_n$ for which events E_j and F_j occur depends only on M_j and i_j. Furthermore, each of the events E_j and F_j depends only on M_i, i_j, and the evaluation points. Thus, by relabeling the index j, we may assume without loss of generality that $i_1 < i_2 < \cdots < i_r$. We emphasize that we are *not* assuming that the output of `GetCertificate` satisfies $i_1 < \cdots < i_r$ (this is not true). We are instead just choosing how we 'reveal' our events E_j and F_j: starting with the smallest index in $i_1, \dots, i_r$ and ending with the largest index in it.

We have

$$\Pr_{\alpha_{[n]}}\left[\bigwedge_{j=1}^{r}(E_j \wedge F_j)\right] = \prod_{j=1}^{r}\Pr_{\alpha_{[n]}}\left[E_j \wedge F_j | E_1 \wedge F_1 \wedge \cdots \wedge E_{j-1} \wedge F_{j-1}\right]$$
$$\le \prod_{j=1}^{r}\Pr_{\alpha_{[n]}}\left[F_j | E_1 \wedge F_1 \wedge \cdots \wedge E_{j-1} \wedge F_{j-1} \wedge E_j\right] \tag{11}$$

Note that $E_1 \wedge F_1 \wedge \cdots \wedge E_{j-1} \wedge F_{j-1} \wedge E_j$ depends only on $\alpha_1, \dots, \alpha_{i_j-1}$, and F_j depends only on $\alpha_1, \dots, \alpha_{i_j}$. For any $\alpha_1, \dots, \alpha_{i_j-1}$ for which $E_1 \wedge F_1 \wedge \cdots \wedge E_{j-1} \wedge F_{j-1} \wedge E_j$ holds, we have that $M_j(X_{\le i_j-1} = \alpha_{\le i_j-1})$ is a $(t-1)k \times (t-1)k$ matrix in $\mathbb{F}_q(X_{i_j}, X_{i_j+1}, \dots, X_n)$ whose determinant is a nonzero polynomial of degree at most $(t-1)k$ in each variable (the determinant contains at most $t-1$ rows including X_{i_j}, each time with maximum degree $k-1$). In particular, at most $(t-1)k$ values of α_{i_j} can make the determinant zero since, viewing the determinant as a polynomial in variables $X_{i_j+1}, \dots, X_n$ with coefficients in $\mathbb{F}_q[X_{i_j}]$, any single nonzero coefficient becomes zero on at most $(t-1)k$ values of α_{i_j}. Conditioned on $\alpha_1, \dots, \alpha_{i_j-1}$, the field element α_{i_j} is uniformly random over $q - i_j + 1 \ge q - n$ elements. Thus, we have, for all $\alpha_1, \dots, \alpha_{i_j-1}$ such that $E_1 \wedge F_1 \wedge \cdots \wedge E_{j-1} \wedge F_{j-1} \wedge E_j$,

$$\Pr_{\alpha_{i_j}}\left[F_j | \alpha_1, \dots, \alpha_{i_j-1}\right] \le \frac{(t-1)k}{q-n}. \tag{12}$$

Since $E_1 \wedge F_1 \wedge \cdots \wedge E_{j-1} \wedge F_{j-1} \wedge E_j$ depends only on $\alpha_{\le i_j-1}$ and F_j depends only on $\alpha_{\le i_j}$, we have

$$\Pr_{\alpha_{[n]}}\left[F_j | E_1 \wedge F_1 \wedge \cdots \wedge E_{j-1} \wedge F_{j-1} \wedge E_j\right] \le \frac{(t-1)k}{q-n}. \tag{13}$$

Combining with (11) gives the desired result. □

The key result for this section is a corollary of Lemma 3.11.

Corollary 3.12 (Probability of one certficiate)**.** *For any sequence $i_1, \dots, i_r \in [n]$, over randomly chosen pairwise distinct evaluation points $\alpha_1, \dots, \alpha_n$, we have*

$$\Pr\left[\texttt{GetCertificate}(\alpha_1, \dots, \alpha_n) = (i_1, \dots, i_r)\right] \le \left(\frac{(t-1)k}{q-n}\right)^r. \tag{14}$$

Proof. By Lemma 3.6, we only need to consider pairwise distinct indices $i_1, \dots, i_r$, otherwise the probability is 0. Let $M_1, \dots, M_r = \texttt{GetMatrixSequence}(i_1, \dots, i_r)$. By Lemma 3.7, matrices $M_1, \dots, M_r$ are all submatrices of $\mathsf{RIM}_{\mathcal{H}}$. Thus, Lemma 3.11 applies. Let $E_1, \dots, E_r, F_1, \dots, F_r$ be the events in Lemma 3.11. If `GetCertificate`$(\alpha_1, \dots, \alpha_n) = (i_1, \dots, i_r)$, then the definition of i_j in Line 3 of `GetCertificate` implies that events E_j and F_j both occur. By Lemma 3.11, the probability that all E_j and F_j hold is at most $(\frac{(t-1)k}{q-n})^r$, hence the result. □

3.7 Finishing the proof of Lemma 3.1

Proof of Lemma 3.1. Recall (Section 3.2) that we fixed $\mathcal{H}$ to be a type-ordered $(k+\varepsilon n)$-weakly-partition-connected hypergraph. By Lemma 3.8, if the matrix $\mathsf{RIM}_{\mathcal{H}}(X_{[n]} = \alpha_{[n]})$ does not have full column rank, then $\texttt{GetCertificate}(\alpha_1, \ldots, \alpha_n)$ is some certificate $(i_1, \ldots, i_r)$. The probability that $\texttt{GetCertificate}(\alpha_1, \ldots, \alpha_n) = (i_1, \ldots, i_r)$ is at most $(\frac{(t-1)k}{q-n})^r$ by Corollary 3.12. By Corollary 3.10, there are at most $\binom{n}{r}2^{tr}$ certificates. Taking a union bound over possible certificates gives the lemma. □

4 Random Linear Codes

In this section, we discuss how to modify the proof of Theorem 1.1 to give Theorem 1.3, list-decoding for random linear codes (RLCs). Our proof follows the roadmap in Figure 2. The proof is identical up to a few minor modifications, which we state here for brevity. Below, we state the same lemmas as in the proof for Reed–Solomon codes, adjusted for random linear codes, and we highlight the key differences in purple. We expect that our framework could be applied even more generally to show that other families of random codes — beyond randomly punctured Reed–Solomon codes and random linear codes — achieve list-decoding capacity with small alphabet sizes, assuming such codes satisfy an appropriate GM-MDS theorem.

4.1 Preliminaries: Notation and Definitions

The generator matrix $G \in \mathbb{F}_q^{n\times k}$ of a random linear code has independent uniformly random entries in $\mathbb{F}_q$. To transfer the proof for list-decoding Reed–Solomon codes to list-decoding random linear codes, a key analogy is to think of the generator matrix as a $n \times k$ matrix of nk distinct indeterminates $(X_{i,\ell})_{i\in[n],\ell\in[k]}$, evaluated at nk independent and uniformly random field elements $(\alpha_{i,\ell})_{i\in[n],\ell\in[k]}$.

$$
\begin{aligned}
\mathcal{G} &\stackrel{\text{def}}{=} \begin{bmatrix} X_{1,1} & \cdots & X_{1,k} \\ \vdots & \ddots & \vdots \\ X_{n,1} & \cdots & X_{n,k} \end{bmatrix} \in \mathbb{F}_q(X_{1,1}, \ldots, X_{n,k})^{n\times k}, \\
G &\stackrel{\text{def}}{=} \mathcal{G}|_{X_{[n]\times[n]}=\alpha_{[n]\times[k]}} \\
\mathcal{G}_i &\stackrel{\text{def}}{=} [X_{i,1}, \ldots, X_{i,k}] \text{ (the } i\text{th row of } \mathcal{G}). \qquad (15)
\end{aligned}
$$

We note that our randomly punctured Reed–Solomon code can also be viewed as an evaluation of $\mathcal{G}$, where $X_{i,\ell}$ is assigned $\alpha_i^{\ell-1}$ where $\alpha_1, \ldots, \alpha_n$ are random distinct field elements over $\mathbb{F}$. In this light, one might expect our framework can also apply, and indeed it does.

Accordingly, we use similar indexing shorthand, where the notation $X_{[a]\times[b]}$ represents the $a \cdot b$ indeterminates $X_{1,1}, X_{1,2}, \ldots, X_{a,b}$, and similarly for field elements $\alpha_{[a]\times[b]}$. For field elements $\alpha_{1,1}, \ldots, \alpha_{a,b}$, we write $X_{[a]\times[b]} = \alpha_{[a]\times[b]}$ to denote $X_{i,\ell} = \alpha_{i,\ell}$ for $1 \le i \le a$ and $1 \le b \le \ell$.

We again use the notion of an agreement hypergraph in Section 2.2, and Lemma 2.3 still holds. For each agreement hypergraph $\mathcal{H}$, we consider more general reduced intersection matrix $\mathsf{RIM}_{\mathcal{H},\mathcal{G}}$, where the X_i-Vandermonde-rows are instead the i-th row of $\mathcal{G}$. More precisely,

Definition 4.1 (Reduced intersection matrix, Random Linear Codes, Analogous to Definition 2.5.)**.** The *reduced intersection matrix* $\mathsf{RIM}_{\mathcal{H},\mathcal{G}}$ associated with a hypergraph $\mathcal{H} = ([t], (e_1, \ldots, e_n))$ is a $\mathrm{wt}(\mathcal{E}) \times (t-1)k$ matrix over the field of fractions $\mathbb{F}_q(X_{1,1}, \ldots, X_{n,k})$. For each hyperedge e_i with vertices $j_1 < j_2 < \cdots < j_{|e_i|}$, we add $\mathrm{wt}(e_i) = |e_i| - 1$ rows to $\mathsf{RIM}_{\mathcal{H},\mathcal{G}}$. For $u = 2, \ldots, |e_i|$, we add a row $r_{i,u} = (r^{(1)}, \ldots, r^{(t-1)})$ of length $(t-1)k$ defined as follows:

- If $j = j_1$, then $r^{(j)} = \mathcal{G}_i = [X_{i,1}, X_{i,2}, X_{i,3}, \ldots, X_{i,k}]$
- If $j = j_u$ and $j_u \ne t$, then $r^{(j)} = -\mathcal{G}_i = -[X_{i,1}, X_{i,2}, X_{i,3}, \ldots, X_{i,k}]$
- Otherwise, $r^{(j)} = 0^k$.

4.2 Preliminaries: Properties of RLC Reduced Intersection Matrices

We have similar preliminaries for reduced intersection matrices of random linear codes.

Lemma 4.2 (RIM of agreement hypergraphs are not full column rank, Analogous to Lemma 2.7)**.** *Let $\mathcal{H}$ be an agreement hypergraph for $(y, c^{(1)}, \dots, c^{(t)})$, where $c^{(j)} \in \mathbb{F}_q^n$ are distinct codewords of the code generated by $\mathcal{G}|_{X_{[n]\times[k]}=\alpha_{[n]\times[k]}}$. Then the reduced intersection matrix $\mathsf{RIM}_{\mathcal{H},\mathcal{G}}(X_{[n]\times[k]} = \alpha_{[n]\times[k]})$ does not have full column rank.*

Proof. Analogous to the proof of Lemma 2.7. □

Lemma 4.3 (RIM have full column rank, Analogous to Theorem 2.10)**.** *Let $\mathcal{H}$ be a k-weakly-partition-connected hypergraph with n hyperedges and at least 2 vertices. Then $\mathsf{RIM}_{\mathcal{H},\mathcal{G}}$ has full column rank over the field $\mathbb{F}_q(X_{1,1}, \dots, X_{n,k})$.*

Proof. We note that the Reed–Solomon code reduced intersection matrix $\mathsf{RIM}_{\mathcal{H}}$ can be obtained from the random linear code reduced intersection matrix $\mathsf{RIM}_{\mathcal{H},\mathcal{G}}$ by setting the indeterminates $X_{i,\ell} = X_i^{\ell-1}$, so Lemma 4.3 immediately follows from Theorem 2.10. We emphasize that, while Reed–Solomon codes require large alphabet sizes $q \geq \Omega(n)$, Theorem 2.10 still holds for constant alphabet sizes q (see Remark 2.11), so we can use it here. □

We remark that Lemma 4.3 can be proven directly by following the proof framework of Theorem 2.10 in Appendix A.3, but instead substitute the use of Theorem A.2 with an analogous GM-MDS theorem for Random Linear Codes, which can be found in Lemma 7 of [DSY15] (Lemma 7 of [DSY15] only implies Lemma 4.3 for q to be sufficiently large, but again by Remark 2.11 the q sufficiently large version of Lemma 4.3 implies the lemma for all q). That way, the proof of Theorem 1.3 relies only on the proof framework of Theorem 1.1 and not on any of its lemmas.

We again define row deletions for reduced intersection matrices.

Definition 4.4 (Row deletions, Analogous to Definition 2.12)**.** Given a hypergraph $\mathcal{H} = ([t], (e_1, \dots, e_n))$ and set $B \subseteq [n]$, define $\mathsf{RIM}^B_{\mathcal{H},\mathcal{G}}$ to be the submatrix of $\mathsf{RIM}_{\mathcal{H},\mathcal{G}}$ obtained by deleting all rows containing the row $\mathcal{G}_i$ with $i \in B$.

Now we show that, as for Reed–Solomon codes, the full-column-rankness of reduced intersection matrices is robust to deletions.

Lemma 4.5 (Robustness to deletions, Analogous to Lemma 2.13)**.** *Let $\mathcal{H} = ([t], \mathcal{E})$ be a $(k + \varepsilon n)$-weakly-partition-connected hypergraph with $t \geq 2$. For all sets $B \subset [n]$ with $|B| \leq \varepsilon n$, we have that $\mathsf{RIM}^B_{\mathcal{H},\mathcal{G}}$ is nonempty and has full column rank.*

Proof. The proof is identical to Lemma 2.13, where we instead use the full column rankness of $\mathsf{RIM}_{\mathcal{H},\mathcal{G}}$ for k-weakly-partition-connected $\mathcal{H}$ (Lemma 4.3) rather than the full column rankness of $\mathsf{RIM}_{\mathcal{H}}$ (Theorem 2.10). □

4.3 The proof

The proof of Theorem 1.3 follows similarly to the proof of Theorem 1.1. Our key lemma, analogous to Lemma 3.1 is to show that reduced intersection matrices of weakly-partition-connected hypergraphs are full column rank with high probability.

Lemma 4.6 (Analogous to Lemma 3.1)**.** *Let k be a positive integer and $\varepsilon > 0$. For each $(k + \varepsilon n)$-weakly-partition-connected hypergraph $\mathcal{H} = ([t], (e_1, \dots, e_n))$ with $t \geq 2$, we have, for $r = \lfloor \varepsilon n/2 \rfloor$,*

$$\Pr_{\alpha_{[n]\times[k]}} \left[\mathsf{RIM}_{\mathcal{H},\mathcal{G}}(X_{[n]\times[k]} = \alpha_{[n]\times[k]}) \textit{ does not have full column rank}\right] \leq \binom{n}{r} 2^{tr} \cdot \left(\frac{t-1}{q}\right)^r . \tag{16}$$

Algorithm 3: `GetMatrixSequenceRLC`

```
Input: indices $i_1, \dots, i_{j-1} \in [n]$ for some $j \geq 1$.
Output: $M_1, \dots, M_j$, which are $(t-1)k \times (t-1)k$ matrices over $\mathbb{F}_q(X_{1,1}, \dots, X_{n,k})$.
$B \leftarrow \emptyset$, $i_0 \leftarrow \perp$, $\ell_0 \leftarrow \perp$
for $\ell = 1, \dots, j$ do
    // $M_\ell$ depends only on $i_1, \dots, i_{\ell-1}$
    if $\ell > 1$ then
        // Fetch new index from bank $B$
        $\tau \leftarrow$ the type of $i_{\ell-1}$
        $s \leftarrow$ number of indices among $i_{\ell_0}, i_{\ell_0+1}, \dots, i_{\ell-1}$ that are type $\tau$
        $i'_{\ell-1} \leftarrow$ the $s$-th smallest element of $B$ that has type $\tau$
        if $i'_{\ell-1}$ is defined then
            $M_\ell \leftarrow$ the matrix obtained from $M_{\ell-1}$ by replacing all copies of row $\mathcal{G}_{i_{\ell-1}}$ with $\mathcal{G}_{i'_{\ell-1}}$
    if $M_\ell$ not yet defined then
        // Refresh bank $B$
        $B \leftarrow \emptyset$
        for $\tau = 1, \dots, 2^t$ do
            $B \leftarrow B \cup \{$largest $\lfloor r/2^t \rfloor$ indices of type $\tau$ in $[n] \setminus \{i_1, \dots, i_{\ell-1}\}\}$ (if there are less than
              $\lfloor r/2^t \rfloor$ indices of type $\tau$, then $B$ contains all such indices)
        $M_\ell \leftarrow$ lexicographically smallest nonsingular $(t-1)k \times (t-1)k$ submatrix of $\mathsf{RIM}_{\mathcal{H},\mathcal{G}}^{B \cup \{i_1, \dots, i_{\ell-1}\}}$
        $\ell_0 \leftarrow \ell$ // new refresh index

return $M_1, \dots, M_j$
```

We highlight that our probability bound here is better than the one in Lemma 3.1 for Reed–Solomon codes. This is because (i) all indeterminates in our generator matrix (and thus, the reduced intersection matrix) appear with degree 1 (rather than degree up to $k-1$), and (ii) our indeterminates are assigned independently uniformly at random, rather than random *distinct* values. Thus, the probability of any particular square submatrix matrix being made singular with an assignment is at most $\frac{t-1}{q}$, rather than $\frac{(t-1)k}{q-n}$: item (i) improves the numerator from $(t-1)k$ to $t-1$, and item (ii) improves the denominator from $q-n$ to q. This improved probability bound means we can use a smaller alphabet size for random linear codes than for Reed–Solomon codes. Other than this difference, the rest of our proof follows analogously. We include some more details for completeness.

We start with the same setup in Section 3.2, defining types in the same way, and starting with a $(k+\varepsilon n)$-weakly-partition-connected hypergraph $\mathcal{H}$ that we assume without loss of generality is type-ordered. We again fix

$$r \overset{\text{def}}{=} \left\lfloor \frac{\varepsilon n}{2} \right\rfloor \tag{17}$$

To prove Lemma 4.6, we similarly find a certificate $(i_1, \dots, i_r)$ for each singular reduced intersection matrix. This certificate is generated by an analogous algorithm, `GetCertificateRLC`, which uses an analogous helper function `GetMatrixSequenceRLC`. We show this certificate has the same three properties

1. A bad generator matrix, namely a generator matrix for which the reduced intersection matrix is not full column rank, must yield a certificate.
2. There are few possible certificates
3. The probability that a random generator matrix yields a particular certificate is small.

We generate the certificate in a similar way. This time, instead of sequentially revealing the evaluation points, we sequentially reveal rows of the generator matrix, and i_1 indicates.

Algorithm 4: `GetCertificateRLC`

```
Input: Generator matrix entries $\alpha_{1,1}, \ldots, \alpha_{n,k} \in \mathbb{F}_q$.
Output: A "certificate" $(i_1, \ldots, i_r) \in [n]^r$.
for $j = 1, \ldots, r$ do
    // $M_1, \ldots, M_{j-1}$ stay the same, $M_j$ is now defined
    $M_1, \ldots, M_j$ = GetMatrixSequenceRLC($i_1, \ldots, i_{j-1}$)
    $i_j \leftarrow$ smallest index $i$ such that $M_j(X_{[i]\times[k]} = \alpha_{[i]\times[k]})$ is singular
    if $i_j$ not defined then
        return $\perp$
return $(i_1, \ldots, i_r)$
```

The first item is captured in the following Lemma.

Lemma 4.7 (Bad generator matrix admits certificate, Analogous to Lemma 3.8)**.** *If $\alpha_{1,1}, \ldots, \alpha_{n,k} \in \mathbb{F}_q$ are entries for the generator matrix such that $\mathsf{RIM}_{\mathcal{H},\mathcal{G}}(X_{[n]\times[k]} = \alpha_{[n]\times[k]})$ does not have full column rank,* `GetCertificateRLC`$(\alpha_{1,1}, \ldots, \alpha_{n,k})$ *returns a certificate $(i_1, \ldots, i_r) \in [n]^r$ (rather than $\perp$).*

Proof. Analogous to the proof of Lemma 3.8. □

Just as for Reed–Solomon codes, we obtain the same bound on the number of possible certificates.

Lemma 4.8 (Certificate count, Analogous to Corollary 3.10)**.** *The number of possible outputs to* `GetCertificateRLC` *is at most $\binom{n}{r}2^{tr}$.*

Proof. Analogous to the proof of Corollary 3.10. □

Lastly, we obtain an upper bound on the probability of obtaining a particular certificate.

Lemma 4.9 (Probability of one certficiate, Analogous to Corollary 3.12)**.** *For any sequence $i_1, \ldots, i_r \in [n]$, over independent uniformly random $\alpha_{1,1}, \ldots, \alpha_{n,k}$, we have*

$$\mathbf{Pr}\left[\texttt{GetCertificateRLC}(\alpha_{1,1}, \ldots, \alpha_{n,k}) = (i_1, \ldots, i_r)\right] \leq \left(\frac{t-1}{q}\right)^r. \tag{18}$$

Lemma 4.9 is slightly different from the analogous result for Reed–Solomon codes, Corollary 3.12, so we provide a little more justification here. Similar to Corollary 3.12, Lemma 4.9 follows from a lemma analogous to Lemma 3.11.

Lemma 4.10 (Analogous to Lemma 3.11)**.** *Let $i_1, \ldots, i_r \in [n]$ be pairwise distinct indices, and $M_1, \ldots, M_r$ be $(t-1)k \times (t-1)k$ submatrices of $\mathsf{RIM}_{\mathcal{H},\mathcal{G}}$. Over random generator matrix entries $\alpha_{1,1}, \ldots \alpha_{n,k} \in \mathbb{F}_q$, define the following events for $j = 1, \ldots, r$:*

- *E_j is the event that $M_j(X_{[i]\times[k]} = \alpha_{[i]\times[k]})$ is non-singular for all $i < i_j$.*
- *F_j is the event that $M_j(X_{[i_j]\times[k]} = \alpha_{[i_j]\times[k]})$ is singular.*

The probability that all the events hold is at most $(\frac{t-1}{q})^r$.

Proof of Lemma 4.10. The proof is similar to the proof of Lemma 3.11. Lemma 3.11 follows from combining Eqaution (12) with the appropriate conditional probabilities. This lemma follows the same approach. We again assume without loss of generality $i_1 < i_2 < \cdots, i_r$.

Here, we want, analogous to Equation (12), for all $\alpha_{[i_j-1]\times[k]}$ such that $E_1 \wedge F_1 \wedge \cdots \wedge E_{j-1} \wedge F_{j-1} \wedge E_j$,

$$\mathop{\mathbf{Pr}}_{\alpha_{\{i_j\}\times[k]}}\left[F_j | \alpha_{[i_j-1]\times[k]}\right] \leq \frac{t-1}{q}. \tag{19}$$

To see (19), consider the determinant of $M_j(X_{[i_j-1]\times[k]} = \alpha_{[i_j-1]\times[k]})$, a $(t-1)k \times (t-1)k$ matrix in $\mathbb{F}_q(X_{\{i_j,i_j+1,\ldots,n\}\times[k]})$. View the determinant of $M_j(X_{[i_j-1]\times[k]} = \alpha_{[i_j-1]\times[k]})$ as a polynomial in variables $X_{\{i_j+1,\ldots,n\}\times[k]}$ with coefficients in $\mathbb{F}_q[X_{i_j,1},\ldots,X_{i_j,k}]$. It is nonzero because we assume E_j holds, so there is some coefficient of the form $f(X_{i_j,1},\ldots,X_{i_j,k})$ that is nonzero. Since matrix M_j has at most $t-1$ rows containing any variables among $X_{i_j,1},\ldots,X_{i_j,k}$, each appearing with total degree 1, the total degree of $X_{i_j,1},\ldots,X_{i_j,k}$ in the determinant of M_j is at most $t-1$. Thus, the total degree of $f(X_{i_j,1},\ldots,X_{i_j,k})$ is at most $t-1$. Hence, by the Schwarz-Zippel lemma, f becomes zero with probability at most $\frac{t-1}{q}$ over random $\alpha_{i_j,1},\ldots,\alpha_{i_j,k}$. Thus, the probability that F_j holds is at most $\frac{t-1}{q}$, giving (19).

Combining conditional probabilities as in Lemma 3.11 gives the result. □

Proof of Theorem 1.3. By Lemma 2.3, if our random linear code generated by G is not $\left(\frac{L}{L+1}(1-R-\varepsilon), L\right)$ average-radius list-decodable, then there exists a vector y and codewords $c^{(1)},\ldots,c^{(t)}$ with $t \ge 2$ such that the agreement hypergraph $\mathcal{H} = ([t],\mathcal{E})$ is $(R+\varepsilon)n = (k+\varepsilon n)$-weakly-partition-connected. By Lemma 4.2, the matrix $\mathsf{RIM}_{\mathcal{H},\mathcal{G}}(X_{[n]\times[k]} = \alpha_{[n]\times[k]})$ is not full column rank. Now, the number of possible agreement hypergraphs $\mathcal{H}$ is at most $\sum_{t=2}^{L+1} 2^{tn} \le 2^{(L+2)n}$. Thus by the union bound over possible agreement hypergraphs $\mathcal{H}$ with Lemma 4.6, we have, for $r = \lfloor \frac{\varepsilon n}{2} \rfloor$,

$$\begin{aligned}
&\Pr_{\alpha_{[n]\times[k]}}\left[\text{Code generated by } \mathcal{G}|_{X_{[n]\times[k]}=\alpha_{[n]\times[k]}} \text{ not } \left(\frac{L}{L+1}(1-R-\varepsilon), L\right) \text{ list-decodable}\right] \\
&\le \Pr_{\alpha_{[n]\times[k]}}\left[\exists\ (k+\varepsilon n)\text{-w.p.c. agreement hypergraph } \mathcal{H} \text{ such that } \mathsf{RIM}_{\mathcal{H},\mathcal{G}}(X_{[n]\times[k]} = \alpha_{[n]\times[k]}) \text{ not full column rank}\right] \\
&\le 2^{(L+2)n} \max_{(k+\varepsilon n)\text{-w.p.c. } \mathcal{H}} \Pr_{\alpha_{[n]\times[k]}}\left[\mathsf{RIM}_{\mathcal{H},\mathcal{G}}(X_{[n]\times[k]} = \alpha_{[n]\times[k]}) \text{ not full column rank}\right] \\
&\le 2^{(L+2)n} \cdot \binom{n}{r} 2^{(L+1)r} \left(\frac{L}{q}\right)^r \le \left(2^{(L+2)n/r} \cdot \frac{en}{r} \cdot 2^{L+1} \cdot \frac{L}{q}\right)^r \le 2^{-Ln}, \qquad (20)
\end{aligned}$$

as desired. Here, we used that $q = 2^{10L/\varepsilon}$. □

4.4 Technical comparison with [GZ23]

To prove that random linear codes achieved list-decoding capacity (Theorem 1.3), we extended the framework for showing that (randomly punctured) Reed–Solomon codes achieve list-decoding capacity over linear-sized fields (Theorem 1.1). It is possible to instead use the framework of Guo and Zhang [GZ23] to show a similar result. However, using the framework of Guo and Zhang in the same way would have only worked for alphabet size that is linear in n, rather than, in our case, a near-optimal constant. Below, we explain why our new ideas were necessary for obtaining our near-optimal alphabet size.

In (20), our upper bound on the non-list-decodability probability is

$$2^{(L+2)n} \cdot \binom{n}{r} 2^{(L+1)r} \cdot \left(\frac{L}{q}\right)^r, \qquad (21)$$

where $r = \varepsilon n/2$, where $\varepsilon > 0$ is roughly the gap to capacity. Here, the term $2^{(L+2)n}$ comes from a union bound over the number of possible hypergraphs, the term $\binom{n}{r}2^{(L+1)r}$ comes from a union bound over the number of possible certificates, and the term $\left(\frac{L}{q}\right)^r$ bounds the probability of a single certificate. We saw above that this probability is $o(1)$ as long as $q \ge 2^{10L/\varepsilon}$.

If we applied the framework of [GZ23] to random linear codes, the number of possible certificates would instead be n^r. Our bound on the non-list-decodability probability would then be

$$2^{(L+2)n} \cdot n^r \cdot \left(\frac{L}{q}\right)^r. \qquad (22)$$

For this to bound to be $o(1)$, we need to take $q \ge 2^{L/\varepsilon} \cdot n$, giving an alphabet size of $O(n)$. This would still have been a new result, as, previously, the Reed–Solomon codes of [GZ23] gave the smallest known alphabet size ($O(n^2)$) of any linear code achieving list-decoding capacity with optimal list size $O(1/\varepsilon)$. However, using our framework allows us to achieve a near-optimal constant list size of $2^{O(L/\varepsilon)}$.

Acknowledgements

We thank Mary Wootters and Francisco Pernice for helpful discussions about [BGM23] and the hypergraph perspective of the list-decoding problem. We thank Karthik Chandrasekaran for helpful discussions about hypergraph connectivity notions and for the reference of Theorem A.3 in [Fra11]. We thank Nikhil Shagrithaya and Jonathan Mosheiff for pointing out a mistake in the proof of Lemma 4.3 in an earlier version of the paper.

References

[BDG22] Joshua Brakensiek, Manik Dhar, and Sivakanth Gopi. Improved field size bounds for higher order mds codes. *arXiv preprint arXiv:2212.11262*, 2022.

[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.

[BGM23] Joshua Brakensiek, Sivakanth Gopi, and Visu Makam. Generic reed-solomon codes achieve list-decoding capacity. In *STOC 2023*, page to appear, 2023.

[BKR10] Eli Ben-Sasson, Swastik Kopparty, and Jaikumar Radhakrishnan. Subspace polynomials and limits to list decoding of Reed-Solomon codes. *IEEE Trans. Inform. Theory*, 56(1):113–120, Jan 2010.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003.

[CPS99] Jin-Yi Cai, Aduri Pavan, and D Sivakumar. On the hardness of permanent. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 90–99. Springer, 1999.

[CW07] Qi Cheng and Daqing Wan. On the list and bounded distance decodability of Reed-Solomon codes. *SIAM J. Comput.*, 37(1):195–209, April 2007.

[CX18] Chandra Chekuri and Chao Xu. Minimum cuts and sparsification in hypergraphs. *SIAM Journal on Computing*, 47(6):2118–2156, 2018.

[DSY14] Son Hoang Dau, Wentu Song, and Chau Yuen. On the existence of mds codes over small fields with constrained generator matrices. In *2014 IEEE International Symposium on Information Theory*, pages 1787–1791. IEEE, 2014.

[DSY15] Son Hoang Dau, Wentu Song, and Chau Yuen. On simple multiple access networks. *IEEE Journal on Selected Areas in Communications*, 33(2):236–249, 2015.

[Eli57] Peter Elias. List decoding for noisy channels. *Wescon Convention Record, Part 2, Institute of Radio Engineers*, pages 99–104, 1957.

[Eli91] Peter Elias. Error-correcting codes for list decoding. *IEEE Transactions on Information Theory*, 37(1):5–12, 1991.

[FGKP06] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. New results for learning noisy parities and halfspaces. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 563–574. IEEE, 2006.

[FK09] András Frank and Tamás Király. A survey on covering supermodular functions. *Research Trends in Combinatorial Optimization: Bonn 2008*, pages 87–126, 2009.

[FKK03a] András Frank, Tamás Király, and Zoltán Király. On the orientation of graphs and hypergraphs. *Discrete Applied Mathematics*, 131(2):385–400, 2003.

[FKK03b] András Frank, Tamás Király, and Matthias Kriesell. On decomposing a hypergraph into k connected sub-hypergraphs. *Discrete Applied Mathematics*, 131(2):373–383, 2003.

[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.

[Fra11] András Frank. *Connections in combinatorial optimization*, volume 38. Oxford University Press Oxford, 2011.

[GHK11] Venkatesan Guruswami, Johan Håstad, and Swastik Kopparty. On the list-decodability of random linear codes. *IEEE Trans. Inform. Theory*, 57(2):718–725, Feb 2011.

[GHSZ02] Venkatesan Guruswami, Johan Håstad, Madhu Sudan, and David Zuckerman. Combinatorial bounds for list decoding. *IEEE Trans. Inform. Theory*, 48(5):1021–1034, May 2002.

[GI01] Venkatesan Guruswami and Piotr Indyk. Expander-based constructions of efficiently decodable codes. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 658–667. IEEE, 2001.

[GLM$^+$21] Venkatesan Guruswami, Ray Li, Jonathan Mosheiff, Nicolas Resch, Shashwat Silas, and Mary Wootters. Bounds for list-decoding and list-recovery of random linear codes. *IEEE Transactions on Information Theory*, 68(2):923–939, 2021.

[GLS$^+$22] Zeyu Guo, Ray Li, Chong Shangguan, Itzhak Tamo, and Mary Wootters. Improved list-decodability and list-recoverability of reed-solomon codes via tree packings. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 708–719. IEEE, 2022.

[GM22] Venkatesan Guruswami and Jonathan Mosheiff. Punctured low-bias codes behave like random linear codes. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 36–45. IEEE, 2022.

[GR06] Venkatesan Guruswami and Atri Rudra. Limits to list decoding Reed–Solomon codes. *IEEE Trans. Inform. Theory*, 52(8):3642–3649, August 2006.

[GR08] Venkatesan Guruswami and Atri Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Transactions on Information Theory*, 54(1):135–150, 2008.

[GRS22] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. Essential coding theory. *Draft available at https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/*, 2022.

[GS99] Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed–Solomon and algebraic-geometry codes. *IEEE Transactions on Information Theory*, 45(6):1757–1767, 1999.

[GST22] Eitan Goldberg, Chong Shangguan, and Itzhak Tamo. Singleton-type bounds for list-decoding and list-recovery, and related results. In *2022 IEEE International Symposium on Information Theory (ISIT)*, pages 2565–2570. IEEE, 2022.

[GV10] Venkatesan Guruswami and Salil Vadhan. A lower bound on list size for list decoding. *IEEE Transactions on Information Theory*, 56(11):5681–5688, 2010.

[GW13] Venkatesan Guruswami and Carol Wang. Linear-algebraic list decoding for variants of reed–solomon codes. *IEEE Transactions on Information Theory*, 59(6):3257–3268, 2013.

[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*, pages 843–852, 2013.

[GZ23] Zeyu Guo and Zihan Zhang. Randomly punctured reed-solomon codes achieve the list decoding capacity over polynomial-size alphabets. *arXiv preprint arXiv:2304.01403*, 2023.

[HW18] Brett Hemenway and Mary Wootters. Linear-time list recovery of high-rate expander codes. *Information and Computation*, 261:202–218, 2018.

[JMS03] Kamal Jain, Mohammad Mahdian, and Mohammad R Salavatipour. Packing steiner trees. In *SODA*, volume 3, pages 266–274, 2003.

[Joh62] Selmer Johnson. A new upper bound for error-correcting codes. *IRE Transactions on Information Theory*, 8(3):203–207, 1962.

[Kir03] Tamás Király. *Edge-connectivity of undirected and directed hypergraphs.* PhD thesis, Eötvös Loránd University, 2003.

[Kop15] Swastik Kopparty. List-decoding multiplicity codes. *Theory of Computing*, 11(1):149–182, 2015.

[KRZSW18] Swastik Kopparty, Noga Ron-Zewi, Shubhangi Saraf Saraf, and Mary Wootters. Improved decoding of folded Reed-Solomon and multiplicity codes. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 212–223. IEEE, 2018.

[Lov18] Shachar Lovett. Mds matrices over small fields: A proof of the gm-mds conjecture. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 194–199. IEEE, 2018.

[LP20] Ben Lund and Aditya Potukuchi. On the list recoverability of randomly punctured codes. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*, volume 176, pages 30:1–30:11, 2020.

[LW20] Ray Li and Mary Wootters. Improved list-decodability of random linear binary codes. *IEEE Transactions on Information Theory*, 67(3):1522–1536, 2020.

[MRRZ$^+$20] Jonathan Mosheiff, Nicolas Resch, Noga Ron-Zewi, Shashwat Silas, and Mary Wootters. Ldpc codes achieve list decoding capacity. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 458–469. IEEE, 2020.

[NW61] Crispin St. J. A. Nash-Williams. Edge-disjoint spanning trees of finite graphs. *Journal of the London Mathematical Society*, 1(1):445–450, 1961.

[PP23] Aaron Putterman and Edward Pyne. Pseudorandom linear codes are list decodable to capacity. *arXiv preprint arXiv:2303.17554*, 2023.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):1–40, 2009.

[Rot22] Ron M Roth. Higher-order mds codes. *IEEE Transactions on Information Theory*, 68(12):7798–7816, 2022.

[RS60] Irving S. Reed and Gustave Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.

[RW14] Atri Rudra and Mary Wootters. Every list-decodable code for high noise has abundant near-optimal rate puncturings. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 764–773. ACM, 2014.

[RW18] Atri Rudra and Mary Wootters. Average-radius list-recoverability of random linear codes. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 644–662. SIAM, 2018.

[Sin64] Richard Singleton. Maximum distance q-nary codes. *IEEE Trans. Inform. Theory*, 10(2):116–118, April 1964.

[ST20] Chong Shangguan and Itzhak Tamo. Combinatorial list-decoding of Reed-Solomon codes beyond the Johnson radius. In *Proceedings of the 52nd Annual ACM Symposium on Theory of Computing*, STOC 2020, pages 538–551, 2020.

[STV01] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the xor lemma. *Journal of Computer and System Sciences*, 62(2):236–266, 2001.

[Tut61] William T. Tutte. On the problem of decomposing a graph into n connected factors. *Journal of the London Mathematical Society*, 1(1):221–230, 1961.

[Woo13] Mary Wootters. On the list decodability of random linear codes with large error rates. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 853–860, New York, NY, USA, 2013. ACM.

[Woz58] John M. Wozencraft. List decoding. *Quarterly Progress Report, Research Laboratory of Electronics, MIT*, 48:90–95, 1958.

[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.

[ZP81] Victor Vasilievich Zyablov and Mark Semenovich Pinsker. List concatenated decoding. *Problemy Peredachi Informatsii*, 17(4):29–33, 1981.

A Alternate presentation of [BGM23]

Here, we include alternate presentations of some ideas from [BGM23]. Algebraically, our presentation is the same, but the hypergraph perspective streamlines combinatorial aspects of their ideas.

A.1 Preliminaries

Dual of Reed–Solomon codes. It is well known that the *dual* of a Reed–Solomon code is a *generalized Reed–Solomon code*: Given positive integers $k \le n$ and evaluation points $\alpha_1, \dots, \alpha_n \in \mathbb{F}_q$, there exists nonzero $\beta_1, \dots, \beta_n \in \mathbb{F}_q$ such that the following matrix, called the *parity-check matrix*,

$$H = \begin{bmatrix} \beta_1 & \beta_2 & \cdots & \beta_n \\ \beta_1\alpha_1 & \beta_2\alpha_2 & \cdots & \beta_n\alpha_n \\ \vdots & \vdots & & \vdots \\ \beta_1\alpha_1^{n-k-1} & \beta_2\alpha_2^{n-k-1} & \cdots & \beta_n\alpha_n^{n-k-1} \end{bmatrix} \tag{23}$$

satisfies $Hc = 0^{n-k}$ if and only if $c \in \mathsf{RS}_{n,k}(\alpha_1, \dots, \alpha_n)$.

Generic Zero Patterns. Following [BGM23], we leverage the GM-MDS theorem to establish list-decodability of Reed–Solomon codes. In this work, we more directly connect the list-decoding problem to the GM-MDS theorem using a hypergraph orientation lemma (introduced in the next section). Here, we review generic zero-patterns and the GM-MDS theorem. To keep the meaning of the variable "k" consistent throughout the paper, we unconventionally state the definition of zero patterns and the GM-MDS theorem with $n-k$ rows instead of k rows.

Definition A.1. Given positive integers $k \le n$, an $(n, n-k)$-*generic-zero-pattern* (GZP) is a collection of sets $S_1, \dots, S_{n-k} \subset [n]$ such that, for all $K \subseteq [n-k]$,

$$\left| \bigcap_{\ell \in K} S_\ell \right| \le n - k - |K|. \tag{24}$$

GM-MDS Theorem. As in [BGM23], we connect the list-decoding problem to the GM-MDS theorem. Here, we make the connection more directly.

Theorem A.2 (GM-MDS Theorem [DSY14, Lov18, YH19])**.** *Given $q \geq 2n-k-1$ and any generic zero-pattern $S_1, \dots, S_{n-k} \subset [n]$, there exists pairwise distinct evaluation points $\alpha_1, \dots, \alpha_n \in \mathbb{F}_q$ and an invertible matrix $M \in \mathbb{F}_q^{(n-k)\times(n-k)}$ such that, if H is the parity-check matrix for $\mathsf{RS}_{n,k}(\alpha_1, \dots, \alpha_n)$ (as in (23)), then MH achieves zero-pattern $S_1, \dots, S_{n-k}$, meaning that $(MH)_{\ell,i} = 0$ whenever $i \in S_\ell$.*

We note that the original GM-MDS theorem shows that the generator matrix of a (non-generalized) Reed Solomon code achieves any generic zero pattern. Here, we state that the generator matrix of a *generalized* Reed–Solomon code achieves any generic zero pattern, which is an immediate corollary of the former result.

A.2 Hypergraph orientations

Our new perspective of the tools from [BGM23] leverages a well-known theorem about orienting weakly-partition-connected hypergraphs, stated below. This theorem is most explicitly stated in [Fra11], but it implicit in [Kir03, FKK03a].

A *directed hyperedge* is a hyperedge with one vertex assigned as the *head*. All the other vertices in the hyperedge are called *tails*. A *directed hypergraph* consists of directed hyperedges. In a directed hypergraph, the *in-degree* of a vertex v is the number of edges for which v is the head. A *path* in a directed hypergraph is a sequence $v_1, e_1, v_2, e_2, \dots, v_{s-1}, e_{s-1}, v_s$ such that for all $\ell = 1, \dots, s-1$, vertex v_ℓ is a tail of edge e_ℓ and vertex $v_{\ell+1}$ is the head of edge e_ℓ. An *orientation* of an (undirected) hypergraph is obtained by assigning a head to each hyperedge, making every hyperedge directed.

Theorem A.3 (Theorems 9.4.13 and 15.4.4 of [Fra11])**.** *A hypergraph $\mathcal{H}$ is k-weakly-partition-connected if and only if it has an orientation such that, for some vertex v (the "root"), every other vertex u has k edge-disjoint paths to v.*[9]

We note that Theorem A.3 is remains true if "to v" is replaced with "from v" and k-weakly-partition-connected is replaced with another hypergraph notion called k-partition-connected. The following corollary essentially captures (the hard direction of) [BGM23, Lemma 2.8].

Corollary A.4. *Let $\mathcal{H} = ([t], \mathcal{E})$ be a k-weakly-partition-connected hypergraph with n hyperedges and $t \geq 2$. Then there exists integers $\delta_1, \dots, \delta_t \geq 0$ summing to $n-k$ such that taking δ_j copies of $S_j \overset{\text{def}}{=} \{i \in [n] : j \notin e_i\} \subset [n]$ gives an $(n, n-k)$-GZP.*

Proof. Take the orientation of $\mathcal{H}$ and root vertex $v \in [t]$ given by Theorem A.3. We now take our δ_j's as follows: for each non-root $j \in [t]$, let $\delta_j \overset{\text{def}}{=} \deg_{in}(j)$ to be the in-degree of vertex j. For the root v, let $\delta_v \overset{\text{def}}{=} \deg_{in}(v) - k$. Note that $\delta_v \geq 0$ as $\mathcal{H}$ has another vertex u with k edge-disjoint paths to v. Since there are n hyperedges, the sum of all δ_j's is thus $n-k$. We now check the generic zero pattern condition (24). Consider any nonempty multiset $K \subset [t]$ such that each vertex $j \in [t]$ appears at most δ_j times. First, observe that $|K| \leq \sum_{j \in K} \delta_j$ by definition of K. Now, we have two cases:

Case 1: K does not contain the root. Then we claim:

$$\left| \bigcap_{\ell \in K} S_\ell \right| \leq \sum_{j \in [t] \setminus K} \delta_j = n - k - \sum_{j \in K} \delta_j \leq n - k - |K|. \tag{25}$$

The left side is exactly the number of hyperedges induced by the vertices $[t] \setminus K$, which is at most the sum of the indegrees of $[t] \setminus K$ which is exactly $\sum_{j \in [t] \setminus K} \delta_j$.

Case 2: K contains the root. Then we claim:

$$\left| \bigcap_{\ell \in K} S_\ell \right| \leq -k + \sum_{j \in [t] \setminus K} \delta_j = -k + \left(n - k - \left(\sum_{i \in K} \delta_i - k \right) \right) \leq n - k - |K|. \tag{26}$$

[9] In [Fra11, Theorems 9.4.13 and 15.4.4], the property of having k edge-disjoint paths to v is called $(0, k)$-*edge-connected*.

The left side is exactly the number of hyperedges induced by the vertices $[t] \setminus K$. Fix an arbitrary vertex u in K. By our orientation, u has k edge-disjoint paths to v. These paths have k distinct edges that “enter” $[t] \setminus K$, i.e., their head is in $[t] \setminus K$ but they are not induced by $[t] \setminus K$. Thus, the number of edges induced by $[t] \setminus K$ is at most $(\sum_{i \in [t] \setminus K} \delta_i) - k$. Hence, we have the first inequality. This completes the proof. $\square$

A.3 Proof of Theorem 2.10

In this section, we reprove Theorem 2.10, which we need in this work.

Proof of Theorem 2.10. It suffices to prove that $\mathsf{RIM}_{\mathcal{H}}$ has full column rank for some evaluation of $X_1 = \alpha_1, \dots, X_n = \alpha_n$ for $\alpha_1, \dots, \alpha_n \in \mathbb{F}_q$. Furthermore, by Remark 2.11, it also suffices to prove Theorem 2.10 for when $q \geq 2n - k - 1$. Indeed, that would then show there that is a square $(t-1)k \times (t-1)k$ submatrix of $\mathsf{RIM}_{\mathcal{H}}(X_{[n]} = \alpha_{[n]})$ of full column rank, which means that submatrix has nonzero determinant (in $\mathbb{F}_q$), which means the corresponding square submatrix of $\mathsf{RIM}_{\mathcal{H}}$ also has a nonzero determinant (in $\mathbb{F}_q(X_1, \dots, X_n)$), so $\mathsf{RIM}_{\mathcal{H}}$ has full column rank.

Let $e_1, \dots, e_n$ be the edges of our k-weakly-partition-connected hypergraph $\mathcal{H}$. By Corollary A.4, there a generic zero pattern $S_1, \dots, S_{n-k}$ where, for all $\ell = 1, \dots, n-k$, the set S_ℓ is $\{i : j \notin e_i\}$ for some $j \in [t]$. By Theorem A.2, there exists pairwise distinct $\alpha_1, \dots, \alpha_n \in \mathbb{F}_q$ and a nonsingular matrix $M \in \mathbb{F}_q^{(n-k) \times (n-k)}$ such that, for $H \in \mathbb{F}_q^{(n-k) \times n}$ the parity check matrix of $\mathsf{RS}_{n,k}(\alpha_1, \dots, \alpha_n)$, the matrix $M \cdot H \in \mathbb{F}_q^{(n-k) \times n}$ achieves the zero pattern $S_1, \dots, S_{n-k}$, meaning that $(MH)_{\ell,i} = 0$ whenever $i \in S_\ell$.

Suppose for the sake of contradiction there is a nonzero vector $v \in \mathbb{F}_q^{(t-1)k}$ such that $\mathsf{RIM}_{\mathcal{H}}(X_{[n]} = \alpha_{[n]}) \cdot v = 0$. Let $f^{(1)}, \dots, f^{(t)} \in \mathbb{F}_q^k$ be such that $v = [f^{(1)}, f^{(2)}, \dots, f^{(t-1)}]$ and $f^{(t)} = 0$. Define $c^{(1)}, \dots, c^{(t)} \in \mathbb{F}_q^n$ be such that $c^{(i)} = G \cdot f^{(i)}$ where

$$G \stackrel{\text{def}}{=} \begin{bmatrix} 1 & \alpha_1 & \cdots & \alpha_1^{k-1} \\ 1 & \alpha_2 & \cdots & \alpha_2^{k-1} \\ \vdots & \vdots & & \vdots \\ 1 & \alpha_n & \cdots & \alpha_n^{k-1} \end{bmatrix} \tag{27}$$

We next show that, for any $i = 1, \dots, n$, $c_i^{(j)} = c_i^{(j')}$ for all $j, j' \in e_i$. Let $e_i = j_1, \dots, j_{|e_i|}$. Since $\mathsf{RIM}_{\mathcal{H}}(X_{[n]} = \alpha_{[n]}) \cdot v = 0$, we have, by definition of $\mathsf{RIM}_{\mathcal{H}}$, for $u = 2, \dots, |e_i|$,

$$c_i^{(j_1)} - c_i^{(j_u)} = [1, \alpha_i, \dots, \alpha_i^{k-1}] \cdot (f^{(j_1)} - f^{(j_u)})^T = 0. \tag{28}$$

(note this is true even if $j_u = t$, since $f^{(t)} = 0$).

Define a vector $y \in \mathbb{F}_q^n$ such that, for $i = 1, \dots, n$, we have $y_i = c_i^{(j)}$, where j is an arbitrary element of hyperedge e_i (by the previous paragraph, the choice of j does not matter). For each $j = 1, \dots, t$, we must have $(MH \cdot (y - c^{(j)}))_\ell = 0$ for all $\ell \in [n-k]$ such that S_ℓ is a copy of $\{i \in [n] : j \notin e_i\}$; the ℓ'th row of MH is supported only on $\{i \in [n] : j \in e_i\}$, and $y - c^{(j)}$ is zero on $\{i \in [n] : j \in e_i\}$ by definition of y. Since $MHc^{(j)} = M \cdot (Hc^{(j)}) = 0$ for all $j = 1, \dots, t$, we have, for all j and all ℓ such that S_ℓ is a copy of $\{i \in [n] : j \notin e_i\}$,

$$(MHy)_\ell = (MH \cdot (y - c^{(j)}))_\ell = 0. \tag{29}$$

By construction, all S_ℓ are a copy of some set $\{i : j \notin e_i\}$, so we conclude $MHy = 0$. Since M is invertible, we must have $Hy = 0$.

This means $y = G \cdot f$ for some $f \in \mathbb{F}_q^k$, so y is the evaluation of a degree-less-than-k polynomial. Since $\mathcal{H}$ is k-weakly-partition-connected, by considering the partition $\{j\} \sqcup ([t] \setminus \{j\})$, there are at least k hyperedges e_i containing vertex j in $\mathcal{H}$, so $y_i = c_i^{(j)}$ in at least k indices i. Since y and $c^{(j)}$ are the evaluation of degree-less-than-k polynomials, we must have $y = c^{(j)}$. This holds for all j, so we have $y = c^{(1)} = \cdots = c^{(t)} = 0$ (recall $f^{(t)} = 0$), which contradicts our initial assumption that $v \neq 0$. $\square$

B Alphabet size limitations

In this section, we establish Proposition 1.5. For positive integers m, view $\mathbb{F}_{2^m}$ as a vector space of dimension m over base field $\mathbb{F}_2$. For a set $S \subset \mathbb{F}_{2^m}$, let

$$P_S(X) \overset{\text{def}}{=} \prod_{\alpha\in S}(X-\alpha). \tag{30}$$

An *affine subspace* is a set $L+\alpha = \{\alpha+\beta : \beta\in L\}$ for some subspace L of $\mathbb{F}_{2^m}$.

Lemma B.1 (Proposition 3.2 of [BKR10])**.** *Let L be a subspace of $\mathbb{F}_{2^m}$. Then P_L has the form*

$$X^{2^{\dim L}} + \sum_{i=0}^{\dim L-1}\alpha_i X^{2^i}. \tag{31}$$

where $\alpha_i \in \mathbb{F}_{2^m}$

As an immediate corollary, we have

Lemma B.2. *Let L be an affine subspace of $\mathbb{F}^{2^m}$. Then P_L has the form*

$$X^{2^{\dim L}} + \sum_{i=0}^{\dim L-1}\alpha_i X^{2^i} + \beta \tag{32}$$

for $\alpha_i, \beta \in \mathbb{F}_{2^m}$.

Proof. Since L is an affine subspace, there exists γ such that $L-\gamma \overset{\text{def}}{=} \{\alpha-\gamma : \alpha\in L\}$ is a subspace of $\mathbb{F}_{2^m}$. By Lemma B.1, we have $P_{L-\gamma}$ is of the form

$$X^{2^{\dim L}} + \sum_{i=0}^{\dim L-1}\alpha_i X^{2^i} \tag{33}$$

for $\alpha_i \in \mathbb{F}_{2^m}$. In particular, $P_{L-\gamma}$ is $\mathbb{F}_2$-linear, so

$$P_L(X) = P_{L-\gamma}(X+\gamma) = P_{L-\gamma}(X) + P_{L-\gamma}(\gamma). \tag{34}$$

Setting $\beta = P_{L-\gamma}(\gamma)$ gives the desired form for $P_L(X)$. □

Lemma B.3 (Analogous to Lemma 3.5 of [BKR10])**.** *Let S be a subset of $\mathbb{F}_{2^m}$ of size n. Let u and v be integers such that $0\le u\le v\le m$. Then there is a family $\mathcal{L}$ of at least $2^{(u+1)m-v^2}$ affine subspaces of dimension v, such that each affine subspace $L\in\mathcal{L}$ satisfies $|L\cap S|\ge n/2^{m-v}$, and for any two affine subspaces $L, L'\in\mathcal{L}$, the difference $P_L - P_{L'}$ has degree at most 2^u.*

Proof. For every subspace L of dimension v, there exists $\beta_0,\dots,\beta_{2^{m-v}}$ such that the affine subspaces $L+\beta_i$ partition $\mathbb{F}_{2^m}$. By pigeonhole, there exists some β_i such that $|(L+\beta_i)\cap S| \ge |S|/2^{m-v} = n/2^{m-v}$ The number of subspaces of dimension v is

$$\frac{(2^m-1)(2^m-2)\cdots(2^m-2^{v-1})}{(2^v-1)(2^v-2)\cdots(2^v-2^{v-1})} \ge 2^{v(m-v)}, \tag{35}$$

so there are at least $2^{v(m-v)}$ affine-subspaces L with $|L\cap S|\ge n/2^{m-v}$. For all such affine-subspaces L, the polynomial $P_L(X)$ has the form $X^{2^v}+\sum_{i=0}^{v-1}\alpha_i X^{2^i}+\beta$ by Lemma B.2. Among these affine-subspaces L, by the pigeonhole principle, for at least a fraction $2^{-m(v-u-1)}$ of these subspaces, their subspace polynomials $P_L(X)$ have the same α_i for $i=u+1,u+2,\dots,v$. Let $\mathcal{L}$ be this family of subspaces. The number of subspaces is at least $2^{v(m-v)}\times 2^{-m(v-u-1)} = 2^{(u+1)m-v^2}$, so $\mathcal{L}$ is the desired family of affine subspaces. □

Proof of Proposition 1.5. Let $\delta = 2^{-r-1}$ as in the statement of Proposition 1.5. Consider a Reed–Solomon code of length n and rate δ over $\mathbb{F}_q$, where $q = 2^m$ with m sufficiently large. Let $S \subset \mathbb{F}_q$ be the set of n evaluation points. Apple Lemma B.3 with $u = m - \lceil 1.99r \rceil$ and $v = m - r$. This gives a family $\mathcal{L}$ of $2^{m(m-\lceil 1.99r \rceil)-(m-r)^2} = 2^{2rm-\lceil 1.99r \rceil m+r^2} \geq q^{\Omega(\log(1/\delta))}$ affine subspaces $L \leq \mathbb{F}_{2^m}$ for which $|L \cap S| \geq n/2^{m-v} = 2\delta n$. Furthermore, for $L \in \mathcal{L}$, the subspace polynomials P_L each have 2^v roots, and agree on all coefficients of degree larger than 2^u. Let L_0 be an arbitrary element of $\mathcal{L}$. Then the polynomials $\{P_{L_0} - P_L : L \in \mathcal{L}\}$ are each of degree at most $2^u = 2^{-\lceil 1.99r \rceil} q \leq 4\delta^{1.99} q \leq \delta n$, and each agree with $P_{L_0}(X)$ on at least $|L \cap S| \geq 2\delta n$ values of S. Thus, our Reed–Solomon code is not $(1 - 2\delta, n^{\Omega(1/\delta)})$-list-decodable, as desired. □