




Support-Guessing Decoding Algorithms in the Sum-Rank Metric

Thomas Jerkovits , *Member, IEEE*, Hannes Bartz , *Member, IEEE*, Antonia Wachter-Zeh , *Member, IEEE*

Abstract

The sum-rank metric generalizes the Hamming and rank metric by partitioning vectors into blocks and defining the total weight as the sum of the rank weights of these blocks, based on their matrix representation. In this work, we explore support-guessing algorithms for decoding sum-rank-metric codes. Support-guessing involves randomly selecting candidate supports and attempting to decode the error under the assumption that it is confined to these supports. While previous works have focused on worst-case scenarios, we analyze the average case and derive an optimal support-guessing distribution in the asymptotic regime. We show that this distribution also performs well for finite code lengths. Our analysis provides exact complexity estimates for unique decoding scenarios and establishes tighter bounds beyond the unique decoding radius.

Additionally, we introduce a randomized decoding algorithm for Linearized Reed–Solomon (LRS) codes. This algorithm extends decoding capabilities beyond the unique decoding radius by leveraging an efficient error-and-erasure decoder. Instead of requiring the entire error support to be confined to the guessed support, the algorithm succeeds as long as there is sufficient overlap between the guessed support and the actual error support. As a result, the proposed method improves the success probability and reduces computational complexity compared to generic decoding algorithms.

Our contributions offer more accurate complexity estimates than previous works, which are essential for understanding the computational challenges involved in decoding sum-rank-metric codes. This improved complexity analysis, along with optimized support-guessing distributions, provides valuable insights for the design and evaluation of code-based cryptosystems using the sum-rank metric. This is particularly important in the context of quantum-resistant cryptography.

Index Terms

This work was supported by the German Research Council (DFG) as an ANR-DFG project under Grant no. WA 3907/9-1.

I. INTRODUCTION

The sum-rank metric, introduced for space-time codes in [1], generalizes both the Hamming and rank metrics by partitioning vectors into several blocks and defining the sum-rank weight as the sum of the rank weights of its blocks. Here, the *rank weight* of a block refers to the rank of its matrix representation, as used in rank-metric codes [2].

In recent years, the sum-rank metric has gained significant attention, leading to the development of various code constructions and decoding algorithms [3]–[10]. Linearized Reed–Solomon (LRS) codes, introduced by Martínez-Peñas [11], achieve the Singleton-like bound in the sum-rank metric with equality and include Reed–Solomon and Gabidulin codes as special cases. These codes have found applications in multishot network coding [8], [12], locally repairable codes [11], and space-time codes [1].

The sum-rank metric has also attracted interest in the context of code-based cryptography. It offers a balanced approach between the Hamming and rank metric, making it an appealing choice for constructing secure code-based cryptosystems resilient to quantum and classical attacks. Generic decoding algorithms in the rank metric, such as the GRS algorithm [13], are more complex than generic decoding algorithms in the Hamming metric, which typically rely on (improvements of) Prange’s information set decoding. Consequently, rank-metric-based cryptography can use smaller parameters than in the Hamming metric while maintaining the same level of security, leading to reduced key sizes. However, many rank-metric cryptosystems depend on highly structured codes, which have been vulnerable to structural attacks and, in some cases, have been broken [14]–[18].

Although cryptosystems based on the sum-rank metric have yet to be extensively studied, it is hoped that by carefully choosing the block sizes and the number of blocks, the sum-rank metric can be tuned to achieve a desirable balance between security and key size. This flexibility may offer resistance to attacks that exploit vulnerabilities specific to either the Hamming or rank metric, where many attacks effective in one may prove ineffective in the other [19].

While most code-based cryptosystems, such as the McEliece cryptosystem, focus on unique decoding up to the unique decoding radius, understanding the complexity of decoding in the sum-rank metric beyond the unique decoding radius is also important for cryptographic applications.

Cryptosystems like the Faure-Loidreau (FL) system [18], [20], [21] rely on the difficulty of decoding structured codes beyond the unique decoding radius, even when the attacker knows the code’s structure.

On the other hand, cryptosystems like WAVE [22], BIKE [23], HQC [24], and RQC [25] require solving the syndrome decoding problem for random-like codes. Therefore, analyzing decoding complexities both within and beyond the unique decoding radius is crucial for assessing the security of such systems.

Recently, Puchinger et al. [26] proposed the first non-trivial generic decoding algorithm for arbitrary \mathbb{F}_{q^m} -linear codes in the sum-rank metric. Their algorithm solves the generic decoding problem for error weights up to $n - k$, where n is the code length and k is the code dimension. They derive upper and lower bounds on the expected decoding complexity for the worst-case rank profile, which refers to the distribution of ranks across the individual blocks of the error vector, as defined by the sum-rank metric. However, in deriving the lower bound, they assume a fixed error, meaning that this lower bound is only valid for unique decoding scenarios, such as in the McEliece setting, where the error weight is up to $(d - 1)/2$, with d being the minimum distance of the code. Beyond the unique decoding radius, alternative solutions to the decoding problem may exist, and the lower bound does not account for these, making it loose in this regime. Additionally, the gap between the upper and lower bounds in [26] is significant for some parameters, indicating the potential for improvement. We need a tight lower bound on the decoding complexity to ensure that the system’s best-known attack is as hard as brute-forcing all possible keys. This bound effectively serves as an upper limit on the attacker’s efficiency, ensuring that any potential attack cannot be easier than brute-force. Therefore, improving the lower bound on the expected decoding complexity is essential for accurately assessing the security level and selecting appropriate parameters when designing such cryptosystems.

This paper addresses these limitations by extending the analysis to the average case (over all rank profiles). We derive a support drawing distribution tailored for the average-case scenario, optimizing the expected decoding complexity in the asymptotic setting. This solution also performs well for finite lengths, as numerical evaluations demonstrate. Our derived theoretical complexity can be computed efficiently and is exact for the case of unique decoding, where exactly one solution exists (e.g., in the McEliece setting). For decoding beyond the unique decoding radius, i.e., for error weights larger than $(d - 1)/2$, our exact complexity becomes an upper bound. We introduce a lower bound that accounts for alternative solutions using random coding union bound

(RCU) bound arguments, providing a more accurate estimation of the decoding complexity in this regime.

In our previous work [27], we generalized the randomized decoding algorithm from [28] in the rank metric to the sum-rank metric. In this paper, we extend that analysis by providing additional insights, proofs, and discussions and by considering the asymptotic average case, similar to our analysis of the generic decoding algorithm.

Conceptually, the randomized decoding algorithm works by randomly guessing part of the error's support. Unlike the generic decoder, which requires the guessed support to fully contain the error's support, the randomized decoding algorithm for LRS codes only requires sufficient overlap between the guessed support and the error's support. By using an error-and-erasure decoder (e.g., [29]), successful decoding is possible if enough of the error lies within the guessed support. This relaxation improves the success probability and reduces the expected decoding complexity, particularly for errors with weights only slightly above the unique decoding radius.

We analyze the probability of successful decoding for specific distributions of the guessed supports and propose methods to find the optimal distribution. By exploiting the structure of the underlying LRS code, our approach achieves better expected computational complexity compared to the generic decoding algorithm introduced in [26].

Additionally, we introduce a new Prange-like algorithm for the generic case in the sum-rank metric, specifically designed for the asymptotic average-case setting where the number of blocks tends to infinity and the sizes of the individual blocks are fixed. Similar to the modified Prange algorithm used in the WAVE scheme [22] for the Hamming metric, our algorithm can also find a solution to the generic decoding problem for large error weights in the sum-rank metric.

Organization of the Paper

In Section II, we provide the necessary preliminaries on the sum-rank metric and LRS codes. Section III discusses several decoding problems in the sum-rank metric that are of interest for various applications, highlighting their relevance to cryptography. In Section IV, we extend the analysis of the generic decoding algorithm to the average case and derive optimal support drawing distributions. Additionally, we provide an improved simple closed-form upper bound that enhances the simple bound introduced in [26]. Section V introduces the new Prange-like algorithm for the generic case in the sum-rank metric, specifically designed for the asymptotic average-case setting with a large number of blocks. Section VI presents the extended analysis

of the randomized decoding algorithm for LRS codes, along with its optimization. Finally, in Section VIII, we conclude the paper and discuss potential directions for future work.

II. PRELIMINARIES

This section introduces the notation used throughout the paper, defines the sum-rank metric, and presents the concepts of linear codes and the considered channel model. We also define the error support in the sum-rank metric and LRS codes.

A. Notation

In this paper, we let n and m be integers with $1 \leq m$ and $1 \leq n$. Let q be a prime power and denote the finite field of order q by \mathbb{F}_q . We denote the extension field of \mathbb{F}_q of degree m as \mathbb{F}_{q^m} . We denote the set of all k -dimensional subspaces of \mathbb{F}_q^m as the Grassmannian $\mathcal{G}_k(\mathbb{F}_q^m)$.

Let \mathcal{A} be a discrete set. The cardinality of \mathcal{A} , denoted by $|\mathcal{A}|$, is the number of elements in \mathcal{A} . We use the notation $a \stackrel{\$}{\leftarrow} \mathcal{A}$ to denote an element a drawn uniformly at random from a set \mathcal{A} . We use $[0, 1] \subset \mathbb{R}$ to denote the interval of real numbers between 0 and 1, inclusive. We define the set of all valid probability mass functions (PMFs) over \mathcal{A} as

$$\mathcal{D}(\mathcal{A}) = \left\{ \alpha_s \in [0, 1]^{|\mathcal{A}|} \subset \mathbb{R}^{|\mathcal{A}|} : \sum_{s \in \mathcal{A}} \alpha_s = 1 \right\}. \quad (1)$$

Here, $\alpha_s \in [0, 1]^{|\mathcal{A}|}$ denotes a vector of length $|\mathcal{A}|$ with real values in the interval $[0, 1]$.

Since each component α_s can take any real value in $[0, 1]$ and the sum of these components must be 1, the set $\mathcal{D}(\mathcal{A})$ forms a probability simplex in $\mathbb{R}^{|\mathcal{A}|}$. Therefore, there are uncountably many PMFs over \mathcal{A} .

For a given integer e and f such that $1 \leq e \leq f \leq w$, we use the following notation to denote a *submatrix* of $\mathbf{A} \in \mathbb{F}_q^{v \times w}$ consisting of the selected columns:

$$\mathbf{A}_{[e:f]} := \begin{bmatrix} A_{1,e} & \cdots & A_{1,f} \\ \vdots & \ddots & \vdots \\ A_{v,e} & \cdots & A_{v,f} \end{bmatrix},$$

where $\mathbf{A}_{[e:f]}$ is a submatrix of \mathbf{A} formed by all rows but only columns e to f .

B. Sum-Rank Metric

This section provides a concise overview of the sum-rank weight and sum-rank distance, along with other relevant notations that will be used throughout the paper.

Definition 1 (Sum-Rank Weight and Sum-Rank Distance). *Let $\mathbf{x} = [\mathbf{x}^{(1)} \mid \dots \mid \mathbf{x}^{(\ell)}] \in \mathbb{F}_{q^m}^n$ be a vector, that is partitioned into blocks $\mathbf{x}^{(i)} \in \mathbb{F}_{q^m}^{n_i}$ with respect to a length profile $\mathbf{n} = [n_1, \dots, n_\ell] \in \mathbb{Z}_{\geq 0}^\ell$. The sum-rank weight of \mathbf{x} with respect to the length profile \mathbf{n} is then defined as*

$$\text{wt}_{\Sigma R}^{(\mathbf{n})}(\mathbf{x}) := \sum_{i=1}^{\ell} \text{rk}_q(\mathbf{x}^{(i)}),$$

where $\text{rk}_q(\mathbf{x}^{(i)}) := \dim_q \langle x_1^{(i)}, \dots, x_{n_i}^{(i)} \rangle$ denotes the dimension of the \mathbb{F}_q -span of the entries of $\mathbf{x}^{(i)}$. The sum-rank distance of two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{F}_{q^m}^n$ is then defined by

$$d_{\Sigma R}^{(\mathbf{n})}(\mathbf{x}, \mathbf{y}) := \text{wt}_{\Sigma R}^{(\mathbf{n})}(\mathbf{x} - \mathbf{y}).$$

In some parts of the paper, we restrict our analysis to the case where $n = \ell\eta$ with $\mathbf{n} = [\eta, \dots, \eta] \in \mathbb{Z}_{\geq 0}^\ell$. In this case, we omit \mathbf{n} in the notation and simply write $\text{wt}_{\Sigma R}(\cdot)$ and $d_{\Sigma R}(\cdot, \cdot)$, respectively. Furthermore, we denote the maximum possible rank weight for each block as

$$\mu := \min\{m, \eta\}.$$

Remark 1. *Note that if $n_i = 1$ for all $i \in \{1, \dots, \ell\}$, we have that the sum-rank weight of a vector $\mathbf{x} \in \mathbb{F}_{q^m}^n$ is the number of nonzero entries of \mathbf{x} . This means, that the sum-rank weight coincides with the Hamming weight. For $\ell = 1$ the sum-rank weight coincides with the rank weight. Thus, codes in the Hamming metric and codes in the rank metric can be seen as special cases within the sum-rank metric.*

For a vector $\mathbf{x} = [\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(\ell)}] \in \mathbb{F}_{q^m}^n$ with sum-rank weight $w = \text{wt}_{\Sigma R}(\mathbf{x})$, we are at times interested in the rank profile of \mathbf{x} , which describes the rank weights of the individual blocks. Therefore we define the map $\psi : \mathbb{F}_{q^m}^n \rightarrow \{0, \dots, \mu\}^\ell$ as

$$\psi(\mathbf{x}) = [\text{rk}_q(\mathbf{x}^{(1)}), \dots, \text{rk}_q(\mathbf{x}^{(\ell)})],$$

and call $\psi(\mathbf{x})$ the *rank profile* of \mathbf{x} . Furthermore, we define the set of all possible rank profiles as follows.

Definition 2. Let w , ℓ and μ be non-negative integers such that $w \leq \ell\mu$. We define the set

$$\mathcal{T}_{w,\ell,\mu} := \left\{ \mathbf{w} \in \{0, \dots, \mu\}^\ell : \sum_{i=1}^{\ell} w_i = w \right\},$$

which contains all possible rank profiles of a vector consisting of ℓ blocks and sum-rank weight w .¹

In [26], an upper bound on the cardinality of the set $\mathcal{T}_{w,\ell,\mu}$ has been derived as

$$|\mathcal{T}_{w,\ell,\mu}| \leq \binom{\ell + w - 1}{\ell - 1}.$$

We now define the set that contains only the ordered rank profiles.

Definition 3. Let w , ℓ , and μ be non-negative integers with $w \leq \ell\mu$. We define the set

$$\mathcal{J}_{w,\ell,\mu} := \{ \mathbf{w} \in \mathcal{T}_{w,\ell,\mu} : w_1 \geq w_2 \geq \dots \geq w_\ell \}.$$

Without the restrictions on the maximal length ℓ and the maximum value μ , this set would be identical to the set of integer partitions of w . However, with these restrictions, it forms a special case of integer partitions, constrained in length by ℓ and in value by μ . Note that every element $\mathbf{w} \in \mathcal{J}_{w,\ell,\mu}$ is also an element in $\mathcal{T}_{w,\ell,\mu}$, but not vice versa.

Next, for a given $\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}$, we denote by $\mathfrak{S}_{\ell,\mu}(\mathbf{w})$ the set of all possible permutations of \mathbf{w} . Hence, each element σ in $\mathfrak{S}_{\ell,\mu}(\mathbf{w})$ is an automorphism, i.e., $\sigma : \mathcal{T}_{w,\ell,\mu} \rightarrow \mathcal{T}_{w,\ell,\mu}$.

Additionally, let $\tilde{\mathbf{w}} = [\tilde{w}_1, \dots, \tilde{w}_\ell]$ be the vector obtained by sorting the entries of \mathbf{w} in decreasing order. We denote this operation by $\text{sort}(\mathbf{w})$, so that $\tilde{\mathbf{w}} = \text{sort}(\mathbf{w})$. In other words, $\text{sort}(\mathbf{w})$ represents the permutation of the entries of \mathbf{w} that yields the sorted vector $\tilde{\mathbf{w}}$, such that

$$\tilde{w}_1 \geq \dots \geq \tilde{w}_\ell.$$

This leads to the following two relations between the two sets $\mathcal{J}_{w,\ell,\mu}$ and $\mathcal{T}_{w,\ell,\mu}$:

$$\begin{aligned} \mathcal{J}_{w,\ell,\mu} &= \{ \text{sort}(\mathbf{w}) : \forall \mathbf{w} \in \mathcal{T}_{w,\ell,\mu} \}, \\ \mathcal{T}_{w,\ell,\mu} &= \bigcup_{\mathbf{w} \in \mathcal{J}_{w,\ell,\mu}} \{ \sigma(\mathbf{w}) : \forall \sigma \in \mathfrak{S}_{\ell,\mu}(\mathbf{w}) \}. \end{aligned}$$

¹Rank profiles are closely related to the concept known in the literature as *weak integer compositions*, where an integer is expressed as the sum of non-negative integers. In contrast to weak integer compositions, rank profiles have the additional constraint that each part (rank weight) does not exceed a fixed upper bound, which in our case is μ .

For a given rank profile $\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}$ we also define the restricted set

$$\mathcal{T}_{w,\ell,\mu}^{\geq \mathbf{w}} := \{\mathbf{w}' \in \mathcal{T}_{w,\ell,\mu} : w'_1 \geq w_1, \dots, w'_\ell \geq w_\ell\}.$$

For an element $\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}$ denote by $|\mathfrak{S}_{\ell,\mu}(\mathbf{w})|$ the number of possible permutations of \mathbf{w} which is given by the multinomial as

$$|\mathfrak{S}_{\ell,\mu}(\mathbf{w})| = \binom{\ell}{\lambda_1, \lambda_2, \dots, \lambda_\mu} = \frac{\ell!}{\lambda_1! \lambda_2! \cdots \lambda_\mu!},$$

where λ_i is the number of occurrences of the integer i in \mathbf{w} for all $i = 0, \dots, \mu$. Hence, from all permutations from all elements in $\mathcal{T}_{w,\ell,\mu}$ we obtain the set $\mathcal{T}_{w,\ell,\mu}$.

The set of all vectors in $\mathbb{F}_{q^m}^n$ of sum-rank weight $\text{wt}_{\Sigma R}(\mathbf{e}) = w$ is denoted as

$$\mathcal{E}_{q,\eta,m,\ell}(w) := \{\mathbf{e} \in \mathbb{F}_{q^m}^n \mid \text{wt}_{\Sigma R}(\mathbf{e}) = w\}. \quad (2)$$

The cardinality of this set is given by (see [26])

$$|\mathcal{E}_{q,\eta,m,\ell}(w)| = \sum_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \prod_{i=0}^{\ell-1} \text{NM}_q(m, \eta, w_i), \quad (3)$$

where $\mathbf{w} = [w_0, \dots, w_{\ell-1}]$ denotes the decomposition of the sum-rank weight w into ℓ non-negative integers w_i such that $\sum_{i=0}^{\ell-1} w_i = w$ and $w_i \leq \mu$ for all i . The set of all such decompositions is denoted by $\mathcal{T}_{w,\ell,\mu}$. The cardinality expression in (3) can be efficiently computed using a dynamic programming approach, as described in [26].

The term $\text{NM}_q(m, \eta, w_i)$ represents the number of matrices of size $m \times \eta$ of rank w_i over the finite field \mathbb{F}_q . It can be calculated using the following formula [30, Chapter 13] [31]

$$\begin{aligned} \text{NM}_q(m, \eta, w_i) &= \prod_{j=0}^{w_i-1} \frac{(q^m - q^j)(q^\eta - q^j)}{q^{w_i} - q^j} \\ &= \begin{bmatrix} \eta \\ w_i \end{bmatrix}_q \prod_{j=0}^{w_i-1} (q^m - q^j) \\ &= \begin{bmatrix} m \\ w_i \end{bmatrix}_q \prod_{j=0}^{w_i-1} (q^\eta - q^j), \end{aligned}$$

where $\begin{bmatrix} a \\ b \end{bmatrix}_q$ denotes the q -binomial coefficient, also known as the Gaussian binomial coefficient, which is defined as

$$\begin{bmatrix} a \\ b \end{bmatrix}_q := \prod_{i=1}^b \frac{q^{a-b+i} - 1}{q^i - 1} = \prod_{i=0}^{b-1} \frac{(q^{a-i} - 1)}{(q^{b-i} - 1)}.$$

Here, a and b are integers with $a \geq b \geq 0$, and q is a prime power.

The Gaussian binomial coefficient satisfies the following inequality [32]

$$q^{(a-b)b} \leq \begin{bmatrix} a \\ b \end{bmatrix}_q \leq \gamma_q q^{(a-b)b},$$

with γ_q defined as

$$\gamma_q := \prod_{i=1}^{\infty} (1 - q^{-i})^{-1}.$$

In the following, we present expressions for the probability that two subspaces intersect in a fixed-dimensional space and the probability of one subspace being a subspace of another.

Let \mathcal{A} and \mathcal{B} be two subspaces of \mathbb{F}_q^μ with dimensions a and b , respectively.

We define the conditional probability $P_{q,\mu,a,b}^\cap(j)$ as the probability that the intersection of \mathcal{A} and \mathcal{B} has dimension exactly j , given their dimensions a and b . This probability is given by (see [28])

$$P_{q,\mu,a,b}^\cap(j) := \Pr[\dim(\mathcal{A} \cap \mathcal{B}) = j \mid a, b] = \frac{\begin{bmatrix} \mu-a \\ b-j \end{bmatrix}_q \begin{bmatrix} a \\ j \end{bmatrix}_q q^{(a-j)(b-j)}}{\begin{bmatrix} \mu \\ b \end{bmatrix}_q}. \quad (4)$$

Next, we define the probability that \mathcal{A} is a subspace of \mathcal{B} , denoted by $P_{q,\mu}^\subseteq(a \subseteq b)$, where $a \leq b$. This probability is given by [32]

$$P_{q,\mu}^\subseteq(a \subseteq b) := \Pr[\mathcal{A} \subseteq \mathcal{B} \mid a, b] = \frac{\begin{bmatrix} b \\ a \end{bmatrix}_q}{\begin{bmatrix} \mu \\ a \end{bmatrix}_q}. \quad (5)$$

Remark 2. The probabilities $P_{q,\mu,a,b}^\cap(j)$ and $P_{q,\mu}^\subseteq(a \subseteq b)$ hold whether \mathcal{A} and \mathcal{B} are both drawn uniformly at random from $\mathcal{G}_q(\mathbb{F}_q^\mu)$, or one of them is fixed and the other is drawn uniformly at random from $\mathcal{G}_q(\mathbb{F}_q^\mu)$.

Remark 3. The probability $P_{q,\mu}^\subseteq(a \subseteq b)$ is equal to the intersection probability $P_{q,\mu,a,b}^\cap(b)$ (or $P_{q,\mu,a,b}^\cap(a)$) when the dimension of the intersection is equal to the dimension of the smaller subspace \mathcal{A} , i.e.,

$$\begin{aligned} P_{q,\mu}^\subseteq(a \subseteq b) &= P_{q,\mu,a,b}^\cap(b) = \frac{\begin{bmatrix} \mu-a \\ b-a \end{bmatrix}_q}{\begin{bmatrix} \mu \\ b \end{bmatrix}_q} = \frac{\begin{bmatrix} b \\ a \end{bmatrix}_q}{\begin{bmatrix} \mu \\ a \end{bmatrix}_q} && \text{if } a \leq b, \\ P_{q,\mu}^\subseteq(b \subseteq a) &= P_{q,\mu,a,b}^\cap(a) = \frac{\begin{bmatrix} a \\ b \end{bmatrix}_q}{\begin{bmatrix} \mu \\ b \end{bmatrix}_q} && \text{if } b \leq a. \end{aligned}$$

This relationship holds because \mathcal{A} is a subspace of \mathcal{B} if and only if the dimension of their intersection is the same as that of \mathcal{A} .

C. Linear Codes

A linear code over \mathbb{F}_{q^m} with dimension k and length n is a k -dimensional subspace of $\mathbb{F}_{q^m}^n$, denoted as $\mathcal{C}[n, k]_{\mathbb{F}_{q^m}}$. The minimum sum-rank distance of $\mathcal{C}[n, k]_{\mathbb{F}_{q^m}}$ w.r.t to \mathbf{n} is defined as

$$d_{\min} = \min_{\substack{\mathbf{c}_1, \mathbf{c}_2 \in \mathcal{C}[n, k]_{\mathbb{F}_{q^m}} \\ \mathbf{c}_1 \neq \mathbf{c}_2}} \left\{ d_{\Sigma R}^{(\mathbf{n})}(\mathbf{c}_1, \mathbf{c}_2) \right\}.$$

When the minimum distance d_{\min} of a linear code $\mathcal{C}[n, k]_{\mathbb{F}_{q^m}}$ is known, we refer to $\mathcal{C}[n, k]_{\mathbb{F}_{q^m}}$ as an $\mathcal{C}[n, k, d_{\min}]_{\mathbb{F}_{q^m}}$ code. We define the *unique decoding radius* τ of a code with minimum distance d_{\min} as

$$\tau := \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor.$$

This is the maximum weight of errors (in the respective metric) that can be uniquely decoded by the code. In this paper, we concentrate on codes with distance properties in the sum-rank metric. To emphasize that the sum-rank distance is computed with respect to the length profile $\mathbf{n} = [n_1, n_2, \dots, n_\ell]$, we denote such codes as $\mathcal{C}[\mathbf{n}, k]_{\mathbb{F}_{q^m}}$. In the special case where all blocks have the same length η , resulting in a total length of $n = \eta\ell$, we use the notation $\mathcal{C}[n, k]_{\mathbb{F}_{q^m}}$.

A matrix $\mathbf{G} \in \mathbb{F}_{q^m}^{k \times n}$ is called a generator matrix of the code $\mathcal{C}[\mathbf{n}, k]_{\mathbb{F}_{q^m}}$ if and only if the rows of \mathbf{G} form a basis for $\mathcal{C}[\mathbf{n}, k]_{\mathbb{F}_{q^m}}$. In other words, the linear combinations of the rows of \mathbf{G} generate all the codewords in $\mathcal{C}[\mathbf{n}, k]_{\mathbb{F}_{q^m}}$.

Additionally, a matrix $\mathbf{H} \in \mathbb{F}_{q^m}^{(n-k) \times n}$ is called a parity-check matrix of $\mathcal{C}[\mathbf{n}, k]_{\mathbb{F}_{q^m}}$ if its rows form a basis for the right kernel (i.e., the null space) of the generator matrix \mathbf{G} .

D. Channel Model

Consider a codeword $\mathbf{c} \in \mathcal{C}[n, k]_{\mathbb{F}_{q^m}}$ that is corrupted by an error \mathbf{e} of sum-rank weight w . The received word \mathbf{y} is then given by

$$\mathbf{y} = \mathbf{c} + \mathbf{e}.$$

The error vector $\mathbf{e} \in \mathbb{F}_{q^m}^n$ with sum-rank weight $\text{wt}_{\Sigma R}(\mathbf{e}) = w$ can be partitioned into blocks $\mathbf{e}^{(1)}, \dots, \mathbf{e}^{(\ell)}$ according to the length profile \mathbf{n} . This partitioned vector can be decomposed as

$$\mathbf{e} = \mathbf{a}\mathbf{B} = [\mathbf{a}^{(1)} \mid \dots \mid \mathbf{a}^{(\ell)}] \cdot \text{diag}(\mathbf{B}^{(1)}, \dots, \mathbf{B}^{(\ell)}), \quad (6)$$

where $\mathbf{a}^{(i)} \in \mathbb{F}_{q^m}^{w_i}$, $\mathbf{B}^{(i)} \in \mathbb{F}_{q^m}^{w_i \times n_i}$ with $\text{rk}_q(\mathbf{a}^{(i)}) = \text{rk}_q(\mathbf{B}^{(i)}) = w_i$ for all $i \in \{1, \dots, \ell\}$, and $w = \sum_{i=1}^{\ell} w_i$. It follows that $\mathbf{e}^{(i)} = \mathbf{a}^{(i)}\mathbf{B}^{(i)}$ for $i \in \{1, \dots, \ell\}$, where the entries of $\mathbf{a}^{(i)}$ form a basis over \mathbb{F}_q of the column space of $\mathbf{e}^{(i)}$ and the rows of $\mathbf{B}^{(i)}$ form a basis over \mathbb{F}_q of its row space.

E. Error Support

In the following, we define the notions of row support and column support for vectors in the sum-rank metric.

Definition 4 (Row and Column Support). *Let $e \in \mathbb{F}_q^n$ be of sum-rank weight w .*

- **Row Support:** *The row support of e is defined as*

$$\mathcal{E}_R := \mathcal{E}_R^{(1)} \times \mathcal{E}_R^{(2)} \times \cdots \times \mathcal{E}_R^{(\ell)},$$

where $\mathcal{E}_R^{(i)} \subseteq \mathbb{F}_q^{n_i}$ is the \mathbb{F}_q -row space of $\mathbf{B}^{(i)} \in \mathbb{F}_q^{w_i \times n_i}$ and thus of $e^{(i)}$ as in (6) for all $i \in \{1, \dots, \ell\}$.

- **Column Support:** *The column support of e is defined as*

$$\mathcal{E}_C := \mathcal{E}_C^{(1)} \times \mathcal{E}_C^{(2)} \times \cdots \times \mathcal{E}_C^{(\ell)},$$

where $\mathcal{E}_C^{(i)} \subseteq \mathbb{F}_q^m$ is the \mathbb{F}_q -column space of $e^{(i)}$. Specifically, each entry of $e^{(i)} \in \mathbb{F}_q^m$ is expanded as a vector in \mathbb{F}_q^m using a fixed basis of \mathbb{F}_q^m over \mathbb{F}_q . The column support is then defined as the \mathbb{F}_q -span of these expanded vectors, for all $i \in \{1, \dots, \ell\}$.

Assume \mathcal{E} to be either a row or column support of the error as defined in Definition 4. We denote by $\dim_\Sigma(\mathcal{E})$ the *sum dimension* of an error support:

$$\dim_\Sigma(\mathcal{E}) := \sum_{i=1}^{\ell} \dim(\mathcal{E}^{(i)}).$$

The intersection of two supports \mathcal{E}_1 and \mathcal{E}_2 is defined as

$$\mathcal{E}_1 \cap \mathcal{E}_2 := \left(\mathcal{E}_1^{(1)} \cap \mathcal{E}_2^{(1)} \right) \times \cdots \times \left(\mathcal{E}_1^{(\ell)} \cap \mathcal{E}_2^{(\ell)} \right).$$

Given two supports \mathcal{E}_1 and \mathcal{E}_2 , we say that \mathcal{E}_2 is a *row super-support* of \mathcal{E}_1 , denoted by $\mathcal{E}_1 \subseteq \mathcal{E}_2$, if and only if $\mathcal{E}_1^{(i)} \subseteq \mathcal{E}_2^{(i)}$ for all $i \in \{1, \dots, \ell\}$.

Definition 5. *Let μ be a positive integer and $0 \leq w \leq \ell\mu$. For $\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}$, we define the set of all supports as*

$$\Xi_{q,\mu}(\mathbf{w}) := \left\{ \mathcal{F}_1 \times \cdots \times \mathcal{F}_\ell : \mathcal{F}_i \subseteq \mathbb{F}_q^\mu : \dim(\mathcal{F}_i) = w_i \right\}.$$

F. Linearized Reed–Solomon Codes

LRS codes, introduced by Martínez-Peñas in [7], are a class of codes in the sum-rank metric. An LRS code of length n , dimension k , and length partition $\mathbf{n} = [n_1, \dots, n_\ell]$ over \mathbb{F}_{q^m} is denoted by $\mathcal{C}_{\text{LRS}}[\mathbf{n}, k]$.

The key properties of LRS codes are:

- LRS codes achieve the Singleton-like bound in the sum-rank metric, i.e., their minimum sum-rank distance is $d_{\min} = n - k + 1$ [7], [33]. Therefore, LRS codes are called maximum sum-rank distance (MSRD) codes.
- Consequently, LRS codes can uniquely decode errors of sum-rank weight up to

$$\tau = \left\lfloor \frac{n - k}{2} \right\rfloor.$$

- LRS codes have some restrictions on their parameters: the number of blocks ℓ must satisfy $\ell \leq q - 1$, and the length of each block n_i must satisfy $n_i \leq m$ for all $i \in \{1, \dots, \ell\}$ (see [7]).

III. PROBLEM DESCRIPTION

This section presents and categorizes decoding problems in the sum-rank metric, relevant to various coding theory and cryptography applications. Starting with the most generic problem, we introduce progressively specific scenarios.

A. Sum-Rank Syndrome Decoding Problem

The *sum-rank syndrome decoding problem* generalizes both the syndrome decoding problem in the Hamming metric and the rank syndrome decoding problem, thereby encompassing a wide range of cryptosystems such as WAVE [22], BIKE [23], HQC [24] (which are based on the Hamming metric), and RQC [25] (which is based on the rank metric).

Problem 1 (Sum-Rank Syndrome Decoding Problem).

- **Instance:**

- A linear sum-rank metric code $\mathcal{C}[\mathbf{n}, k]_{\mathbb{F}_{q^m}} \subseteq \mathbb{F}_{q^m}^n$ with parity-check matrix $\mathbf{H} \in \mathbb{F}_{q^m}^{(n-k) \times n}$.
- A syndrome $\mathbf{s} \in \mathbb{F}_{q^m}^{n-k}$ and an integer $t > 0$.

- **Objective:** Find an error vector \mathbf{e} such that $\mathbf{s} = \mathbf{e}\mathbf{H}^\top$ and $\text{wt}_{\Sigma R}(\mathbf{e}) = t$.

The sum-rank syndrome decoding problem is particularly interesting due to its applicability across different metrics and cryptosystems. A solution to this problem is not guaranteed.

The algorithm introduced in [26] provides a generic decoding approach that addresses the sum-rank syndrome decoding problem for error weights up to $n - k$.

B. Decoding Beyond the Unique Radius

The following problem can be seen as a special case of the sum-rank syndrome decoding problem (see Problem 1). In this case, we assume that a specific codeword, corrupted by an error of known weight, has been received. Under this assumption, the weight of the error, denoted by w , is known, and we set the decoding radius accordingly to w . This ensures that at least one solution exists, although it may not be unique, similar to the general syndrome decoding problem.

We first consider the problem of decoding beyond the unique radius for arbitrary sum-rank metric codes.

Problem 2 (Beyond Unique Decoding for Sum-Rank Metric Codes).

- **Instance:**

- Linear sum-rank metric code $\mathcal{C}[\mathbf{n}, k]_{\mathbb{F}_{q^m}} \subseteq \mathbb{F}_{q^m}^n$ with unique decoding radius τ .
- Error vector $\mathbf{e} \stackrel{\S}{\leftarrow} \mathcal{E}_w$ with $\text{wt}_{\Sigma R}(\mathbf{e}) = w \geq \tau$.
- Received vector $\mathbf{y} = \mathbf{c} + \mathbf{e}$ with $\mathbf{c} \in \mathcal{C}[\mathbf{n}, k]_{\mathbb{F}_{q^m}}$.

- **Objective:** Find a codeword $\mathbf{c} \in \mathcal{C}[\mathbf{n}, k]_{\mathbb{F}_{q^m}}$, such that $\text{wt}_{\Sigma R}(\mathbf{y} - \mathbf{c}) = w$.

Problem 2 extends the unique decoding problem by allowing error weights that exceed the unique decoding radius. Unlike unique decoding, multiple codewords may satisfy the decoding condition, and a solution is not guaranteed to be unique. The generic decoder introduced in [26] can efficiently address this problem for any linear sum-rank metric code without relying on its structure, handling error weights up to $n - k$.

Next, we specialize this problem for LRS codes.

Problem 3 (Beyond Unique Decoding for LRS codes).

- **Instance:**

- LRS code $\mathcal{C}_{\text{LRS}}[\mathbf{n}, k] \subseteq \mathbb{F}_{q^m}^n$, $\mathbf{y} \in \mathbb{F}_{q^m}^n$.
- Error vector $\mathbf{e} \stackrel{\S}{\leftarrow} \mathcal{E}_w$ with $\text{wt}_{\Sigma R}(\mathbf{e}) = w \geq \tau$.

- Received vector $\mathbf{y} = \mathbf{c} + \mathbf{e} \in \mathbb{F}_{q^m}^n$ with $\mathbf{c} \in \mathcal{C}_{\text{LRS}}$.
- **Objective:** Find a codeword $\mathbf{c} \in \mathcal{C}_{\text{LRS}}$, such that $\text{wt}_{\Sigma R}(\mathbf{y} - \mathbf{c}) = w$.

Problem 3 is a specialization of Problem 2 for LRS codes. This specialization allows us to exploit the inherent structure of LRS codes, potentially leading to more efficient decoding algorithms. We compare the complexity of the generic decoder for Problem 2, as introduced in [26], with our proposed randomized decoder (see Section VI) tailored for LRS codes.

The FL system is a rank-metric code-based cryptosystem that uses Gabidulin codes, which are a special case of LRS codes. Problem 3 is itself a generalization of the decoding problem considered in [28], which focused on Gabidulin codes. Our proposed randomized decoder generalizes the decoder introduced in [28] to the sum-rank metric. We first introduced this generalization in [27].

Our proposed decoder and complexity analysis for the sum-rank metric could be valuable for future cryptosystems that are similar to the FL system but operate in the sum-rank metric, or for different cryptosystems that rely on problems like Problem 3 in the sum-rank metric.

C. Unique Decoding Problem

The *unique decoding problem* is a specific case of Problem 2, where the error weight does not exceed the unique decoding radius, ensuring that there is exactly one solution within this radius.

Problem 4 (Unique Decoding Problem).

- **Instance:**
 - A linear sum-rank metric code $\mathcal{C}[\mathbf{n}, k]_{\mathbb{F}_{q^m}} \subseteq \mathbb{F}_{q^m}^n$ with unique decoding radius τ .
 - An error vector $\mathbf{e} \stackrel{\$}{\leftarrow} \mathcal{E}_w$ with $w = \text{wt}_{\Sigma R}(\mathbf{e}) \leq \tau$.
 - A received vector $\mathbf{y} = \mathbf{c} + \mathbf{e}$ where $\mathbf{c} \in \mathcal{C}[\mathbf{n}, k]_{\mathbb{F}_{q^m}}$.
- **Objective:** Determine the unique codeword $\mathbf{c} \in \mathcal{C}[\mathbf{n}, k]_{\mathbb{F}_{q^m}}$ such that $\text{wt}_{\Sigma R}(\mathbf{y} - \mathbf{c}) \leq \tau$.

In this case, the decoder is guaranteed to find exactly one solution as long as the error weight is within the unique decoding radius. Efficient polynomial-time decoders are available for several well-known algebraic codes, such as Reed–Solomon codes, Bose–Chaudhuri–Hocquenghem (BCH) codes, and Goppa codes in the Hamming metric [30], as well as Gabidulin codes in the

rank metric [2], and LRS codes in the sum-rank metric [7], provided that the code structure is known.

However, in cryptosystems such as McEliece-like cryptosystems, where the code structure is intentionally obfuscated to increase security, decoding becomes much more challenging for unauthorized parties. This further highlights the importance of efficient decoders, particularly in cases where the error weight stays within the unique decoding radius.

D. Channel Model

In Problem 2, Problem 3 and Problem 4, the error vector e is drawn uniformly at random from \mathcal{E}_w , the set of all vectors in $\mathbb{F}_{q^m}^n$ with sum-rank weight w . This is done by first determining a rank profile $\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}$ according to the distribution

$$\Pr[\mathbf{w}] = \frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \prod_{i=1}^{\ell} \text{NM}_q(m, \eta, w_i). \quad (7)$$

Then, for each $i \in \{1, \dots, \ell\}$, an element $e^{(i)} \in \mathbb{F}_{q^m}^{\eta_i}$ of rank weight w_i is drawn independently and uniformly at random from the set of all elements in $\mathbb{F}_{q^m}^{\eta_i}$ of rank weight w_i , as described in [26]. The resulting error vector e satisfies $\text{wt}_{\Sigma R}(e) = w$ and is uniformly distributed in \mathcal{E}_w . The received word \mathbf{y} is then considered to be of the form $\mathbf{y} = \mathbf{c} + e$ with $\mathbf{c} \in \mathcal{C}[\mathbf{n}, k]_{\mathbb{F}_{q^m}} \subseteq \mathbb{F}_{q^m}^n$.

IV. GENERIC DECODING IN THE SUM-RANK METRIC

In this section, we summarize the generic decoding algorithm for the sum-rank metric introduced by Puchinger et al. [26], [34]. Their analysis focuses on solving a special instance of Problem 1, where the syndrome is chosen such that at least one solution exists. This scenario applies to both Problem 2 and Problem 4. In their derivation of the lower bound, they assume that at most one solution exists within the decoding radius, with no alternative solutions possible. As a result, the lower bound applies only to Problem 4.

The decoding algorithm proceeds by guessing possible error supports according to a probability distribution, which is a key design criterion that can be optimized to maximize the decoder's success probability. In each iteration of the decoding loop, a new support is guessed, and the decoder attempts to correct the error based on that support. If the guess is incorrect, the loop continues with another guess. The success probability for each iteration depends on the support-guessing distribution. The worst-case complexity is determined by evaluating all possible error

rank profiles, with the worst-case profile, denoted by \mathbf{w}_{wc} , representing the maximum number of operations needed for the decoder to succeed.

Remark 4. In [26, Remark 17], it was shown that an error \mathbf{e} with row support \mathcal{E}_R and column support \mathcal{E}_C can be uniquely recovered if either a row super support $\mathcal{F}_R \supseteq \mathcal{E}_R$ or a column super support $\mathcal{F}_C \supseteq \mathcal{E}_C$ is found, both with sum-rank weight v , such that $w \leq v < d_{\min}$. The sum-rank weight v cannot exceed

$$v_{\max} := \min \left\{ n - k, \left\lfloor \frac{m}{\eta} (n - k) \right\rfloor \right\}. \quad (8)$$

Decoding beyond the unique erasure decoding radius (i.e. error weights larger than $d_{\min} - 1$) is possible up to v_{\max} . This situation is analogous to the classical information-set-decoding (ISD) algorithm in the Hamming metric, where a support of size $n - k$ is selected. Successful decoding occurs if the corresponding submatrix of the parity-check matrix formed by these columns is of full rank, ensuring a unique solution to the system of linear equations. In the Hamming case, the probability that the submatrix is full rank is typically assumed to be close to 1, which is especially true for large field sizes. However, as shown in [35], for the sum-rank metric, the probability of finding a valid decoding support under these conditions can be significantly lower, depending on parameter choices. Within this paper, we similarly assume this probability to be close to 1 in our analysis, though for certain parameter sets, it might be advisable to check whether this assumption holds in practice.

Algorithm 1 describes the decoding process, which selects the appropriate type of (row or column) support. When m is smaller than η , the algorithm selects the row support; otherwise, it chooses the column support. For the sake of simplicity, we will henceforth refer to the selected support type as simply “support” throughout the remainder of this paper, with the understanding that the decoding algorithm makes this choice based on the given parameters.

A. Improved Simple Bound on the Worst-Case Success Probability

Recall that we denoted the worst-case rank profile as \mathbf{w}_{wc} . In [26, Theorem 16], the authors derive lower and upper bounds on the expected run time W_{gen} of Algorithm 1 to decode an error pattern with rank profile \mathbf{w}_{wc} .

Algorithm 1: Generic Sum-Rank Decoder [26]

Input : Parameters: q, m, n, k, ℓ, w and v with $w \leq v \leq v_{\max}$

Received vector $\mathbf{y} \in \mathbb{F}_{q^m}^n$

Parity-check matrix $\mathbf{H} \in \mathbb{F}_{q^m}^{(n-k) \times n}$ of an \mathbb{F}_{q^m} -linear sum-rank metric code \mathcal{C}

Output : Vector $\mathbf{c}' \in \mathcal{C}$ such that $\text{wt}_{\Sigma R}(\mathbf{y} - \mathbf{c}') = w$

```

1  $\mathbf{e}' \leftarrow \mathbf{0}$ 
2  $\eta \leftarrow n/\ell$ 
3  $\mu \leftarrow \min\{m, \eta\}$ 
4 while  $\mathbf{H}(\mathbf{r} - \mathbf{e}')^\top \neq \mathbf{0}$  or  $\text{wt}_{\Sigma R}(\mathbf{e}') \neq w$  do
5    $\mathcal{F} \leftarrow$  Draw random support  $\mathcal{F} \subseteq \mathbb{F}_q^\mu \times \dots \times \mathbb{F}_q^\mu$  of sum dimension  $v$ 
6   if  $\eta < m$  then
7      $\mathbf{e}' \leftarrow$  Column erasure decoding w.r.t.  $\mathcal{F}, \mathbf{H}, \mathbf{y}$  (cf. [26, Theorem 13])
8   else
9      $\mathbf{e}' \leftarrow$  Row erasure decoding w.r.t.  $\mathcal{F}, \mathbf{H}, \mathbf{y}$  (cf. [26, Theorem 14])
10 return  $\mathbf{y} - \mathbf{e}'$ 

```

Given an error rank profile $\mathbf{w} \in \mathcal{T}_{w, \ell, \mu}$ and a super-support rank profile $\mathbf{v} \in \mathcal{T}_{v, \ell, \mu}$, we compute the probability $\varphi_{q, \mu}(\mathbf{v}, \mathbf{w})$ that the error support lies within the guessed super support as [26]

$$\varphi_{q, \mu}(\mathbf{v}, \mathbf{w}) := \prod_{i=1}^{\ell} P_{q, \mu}^{\subseteq}(\mathbf{w}_i \subseteq \mathbf{v}_i), \quad (9)$$

where $P_{q, \mu}^{\subseteq}(\mathbf{w}_i \subseteq \mathbf{v}_i)$ is the probability as in (5).

Using this probability, we define $\varphi_{q, \mu, v}^{(\max)}(\mathbf{w})$ as the maximum probability over all super-support rank profiles $\mathbf{v} \in \mathcal{T}_{v, \ell, \mu}$

$$\varphi_{q, \mu, v}^{(\max)}(\mathbf{w}) := \max_{\mathbf{v} \in \mathcal{T}_{v, \ell, \mu}} \varphi_{q, \mu}(\mathbf{v}, \mathbf{w}).$$

Further let $\text{vcomp}_{\mu}(\mathbf{w}, v)$ be a function $\text{vcomp}_{\mu} : \mathcal{T}_{w, \ell, \mu} \times \mathbb{Z}_{\geq 0} \rightarrow \mathcal{T}_{v, \ell, \mu}$ that for a given integer $v \in \mathbb{Z}_{\geq 0}$ returns a rank profile such that

$$\varphi_{q, \mu, v}^{(\max)}(\mathbf{w}) = \varphi_{q, \mu}(\text{vcomp}_{\mu}(\mathbf{w}, v), \mathbf{w}).$$

The function $\text{vcomp}_{\mu}(\mathbf{w}, v)$ can be implemented efficiently, see [26].

With these definitions in place, Puchinger et al. [26, Theorem 16] provide the following bounds on the expected runtime W_{gen} of Algorithm 1 for Problem 4

$$W_{\text{gen,wc}}^{(\text{LB})} = W_{\text{gen}}^{(\text{iter})} |\mathcal{T}_{w,\ell,\mu}|^{-1} Q_{w,\ell,\mu}, \quad (10)$$

$$W_{\text{gen,wc}}^{(\text{UB})} = W_{\text{gen}}^{(\text{iter})} Q_{w,\ell,\mu}, \quad (11)$$

$$\widetilde{W}_{\text{gen,wc}}^{(\text{UB})} = W_{\text{gen}}^{(\text{iter})} \binom{\ell + w - 1}{\ell - 1} \gamma_q^\ell q^{w(\mu - \frac{v}{\ell})}, \quad (12)$$

with

$$Q_{w,\ell,\mu} := \sum_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v}^{(\text{max})}(\mathbf{w})^{-1}, \quad (13)$$

where $W_{\text{gen}}^{(\text{iter})}$ represents the complexity of one iteration of the generic decoding algorithm. To be more precise, $W_{\text{gen}}^{(\text{iter})}$ is the sum of the complexities of two main components: drawing an error super support and performing the row/column erasure decoding, as outlined in Algorithm 1. The complexity of drawing an error super support is in the order of $\widetilde{O}(n^3 m^2 \log_2(q))$ bit operations. Meanwhile, the row/column-erasure decoding involves $O((n-k)^3 m^3)$ operations over \mathbb{F}_q . While these complexities are asymptotic approximations and neglect constant factors, they provide useful estimates for large input sizes. For finite lengths, the exact complexities depend on the specific implementation details of the underlying algorithms. Therefore, for the purpose of plotting and practical considerations, we approximate $W_{\text{gen}}^{(\text{iter})} \approx n^3 m^3$, combining the dominant terms of both components.

The previous upper bound $\widetilde{W}_{\text{gen,wc}}^{(\text{UB})}$ on the expected runtime of Algorithm 1 can be rather loose when the parameters are closer to the Hamming metric, i.e., when $\ell \rightarrow n$ for fixed η and/or $\eta \rightarrow 1$ (see Figure 1). To address this, we introduce a new, tighter upper bound in Theorem 1.

Theorem 1. *Let \mathbf{c} be a codeword of a sum-rank-metric code $\mathcal{C}[\mathbf{n}, k]_{\mathbb{F}_{q^m}}$ with minimum sum-rank distance d_{\min} . Additionally, let \mathbf{e} be an error of sum-rank weight $w < d_{\min}$ with a rank profile corresponding to the worst-case rank profile \mathbf{w}_{wc} . Then Algorithm 1 in the context of Problem 1 returns a solution w.r.t. an error $\mathbf{e}' \in \mathbb{F}_{q^m}^n$ with the weight w . Each iteration of Algorithm 1 has complexity $W_{\text{gen}}^{(\text{iter})}$. The overall expected worst-case runtime, also referred to as the complexity $W_{\text{gen,wc}}$ of Algorithm 1 is upper bounded by*

$$W_{\text{gen,wc}} \leq \widetilde{W}_{\text{gen,wc}}^{(\text{UB,improved})}, \quad (14)$$

with

$$\widetilde{W}_{\text{gen,wc}}^{(\text{UB,improved})} := W_{\text{gen}}^{(\text{iter})} \binom{\ell + w - 1}{\ell - 1} q^{w(\mu - \frac{v}{\ell})} \cdot \min \left(\gamma_q^\ell, \left(\frac{1 - q^{-\mu}}{1 - q^{-1}} \right)^w \right).$$

Proof. To prove the theorem, we will show that

$$W_{\text{gen,wc}} \leq W_{\text{gen}}^{(\text{iter})} \binom{\ell + w - 1}{\ell - 1} q^{w(\mu - \frac{v}{\ell})} \left(\frac{1 - q^{-\mu}}{1 - q^{-1}} \right)^w.$$

Starting from the bound in (11), it suffices to show that

$$Q_{w,\ell,\mu} \leq \binom{\ell + w - 1}{\ell - 1} q^{w(\mu - \frac{v}{\ell})} \left(\frac{1 - q^{-\mu}}{1 - q^{-1}} \right)^w.$$

By the definition of $Q_{w,\ell,\mu}$ in (13), we can bound $Q_{w,\ell,\mu}$ as

$$Q_{w,\ell,\mu} \leq \underbrace{|\mathcal{T}_{w,\ell,\mu}|}_{\leq \binom{\ell+w-1}{\ell-1} \text{ (cf. [26])}} \cdot \max_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu}^{(\max)}(\mathbf{w})^{-1},$$

where

$$\max_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu}^{(\max)}(\mathbf{w})^{-1} = \max_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \left\{ \varphi_{q,\mu}(\mathbf{v}', \mathbf{w})^{-1} : \mathbf{v}' = \text{vcomp}_\mu(\mathbf{w}, v) \right\}. \quad (15)$$

Next, we can bound $\varphi_{q,\mu}(\mathbf{v}', \mathbf{w})^{-1}$ as

$$\varphi_{q,\mu}(\mathbf{v}', \mathbf{w})^{-1} = \prod_{i=1}^w \frac{\begin{bmatrix} \mu \\ w_i \end{bmatrix}_q}{\begin{bmatrix} v'_i \\ w_i \end{bmatrix}_q} \leq \left(\frac{1 - q^{-\mu}}{1 - q^{-1}} \right)^w \cdot \prod_{i=1}^w q^{w_i(\mu - v'_i)}, \quad (16)$$

where the inequality follows from

$$\frac{\begin{bmatrix} a \\ b \end{bmatrix}_q}{\begin{bmatrix} c \\ b \end{bmatrix}_q} = \frac{q^{b(a-b)}}{q^{b(c-b)}} \cdot \prod_{i=1}^w \frac{(1 - q^{-a})(1 - q^{-a+1}) \dots (1 - q^{-a+b-1})}{(1 - q^{-c})(1 - q^{-c+1}) \dots (1 - q^{-c+b-1})} \leq q^{b(a-c)} \cdot \frac{1 - q^{-a}}{1 - q^{-1}}.$$

Substituting (16) into (15) yields

$$Q_{w,\ell,\mu} \leq \binom{\ell + w - 1}{\ell - 1} \cdot \left(\frac{1 - q^{-\mu}}{1 - q^{-1}} \right)^w \cdot \max_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \left\{ q^{\sum_{i=1}^w w_i(\mu - v'_i)} : \mathbf{v}' = \text{vcomp}_\mu(\mathbf{w}, v) \right\}.$$

The remainder of the proof follows from the proof in [26, Proposition 21], which leads to the desired result

$$W_{\text{gen,wc}} \leq \widetilde{W}_{\text{gen,wc}}^{(\text{UB,improved})}.$$

□

Figure 1 illustrates the existing bounds from Puchinger et al. [26] alongside the new improved bound $\widetilde{W}_{\text{gen,wc}}^{(\text{UB,improved})}$ given by (14). The figure shows the complexity as a function of the number of blocks ℓ for a fixed code length n , where the code length is defined as $n = \ell\eta$. This

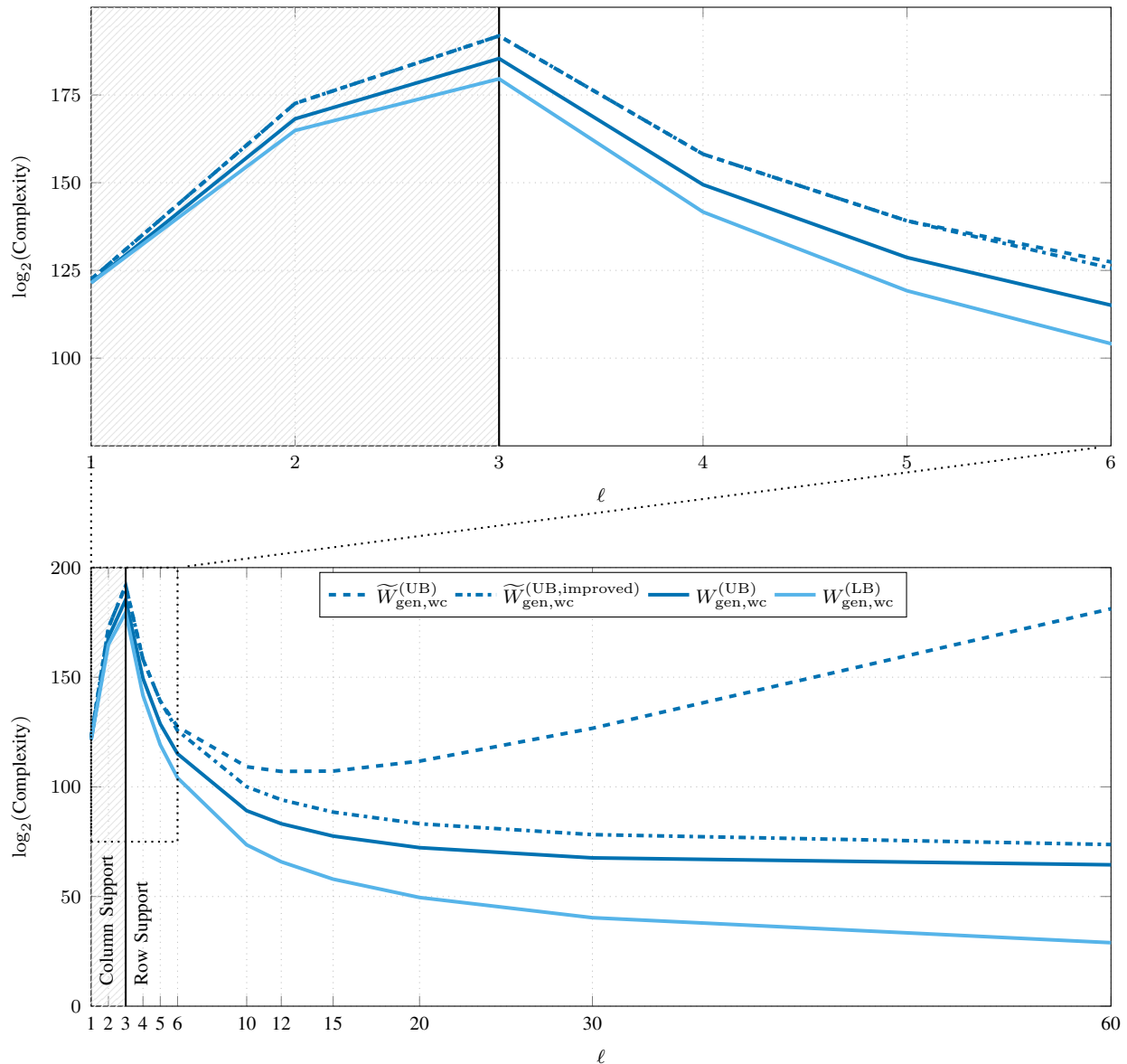


Fig. 1. Illustration of the improved upper bound on the average complexity of Algorithm 1 for $q = 2$, $m = 20$, $n = 60$, $k = 30$, $t = 9$, $s = 10$. At $\ell = 3$, the algorithm transitions from guessing the column support to guessing the row support.

representation allows for the analysis of how the bounds and the improved bound behave as the number of blocks varies while keeping the code length and rate constant.

The existing bounds include the lower bound $W_{\text{gen,wc}}^{(\text{LB})}$, the upper bound $W_{\text{gen,wc}}^{(\text{UB})}$, and the simplified upper bound $\widetilde{W}_{\text{gen,wc}}^{(\text{UB})}$, as defined in (10), (11), and (12), respectively. For a fair comparison, we used the same parameters as those presented in one of the figures from the original paper. The improved bound $\widetilde{W}_{\text{gen,wc}}^{(\text{UB,improved})}$ is significantly tighter and closer to the

upper bound $W_{\text{gen,wc}}^{(\text{UB})}$, particularly in the region near the Hamming metric, which corresponds to cases where $\ell \rightarrow n$ and $\eta \rightarrow 1$. This suggests that the new simplified bound provides a better approximation of the algorithm's complexity compared to the previous simplified upper bound $\widetilde{W}_{\text{gen,wc}}^{(\text{UB})}$, especially when the sum-rank metric closely resembles the Hamming metric.

B. Success Probability Analysis for the Average Case

We now consider the channel model described in Section III-D and begin by deriving the success probability for the case of unique decoding, where exactly one solution exists. Additionally, we derive an upper bound on the success probability for decoding beyond the unique decoding radius, using RCU arguments. This upper bound accounts for alternative solutions that the decoder in Algorithm 1 may return in this scenario.

In this analysis, we assume that the support drawing distribution is known. In the subsequent section, we will use the probabilities derived here to formalize an optimization problem with respect to the support drawing distribution.

In Line 5 of Algorithm 1, we need to draw a suitable super support $\mathcal{F} = \mathcal{F}_1 \times \dots \times \mathcal{F}_\ell$, where $\mathcal{F} \subseteq \mathbb{F}_q^\mu \times \dots \times \mathbb{F}_q^\mu$, each \mathcal{F}_i has dimension v_i for $i \in \{1, \dots, \ell\}$, and $\sum_{i=1}^{\ell} v_i = v$. The distribution from which these super supports are drawn is a critical design parameter of the algorithm and requires careful optimization to maximize the algorithm's performance.

To draw the super support \mathcal{F} , we first draw a suitable rank profile $\mathbf{v} = [v_1, v_2, \dots, v_\ell] \in \mathcal{T}_{v,\ell,\mu}$. Let S be a discrete random variable over $\mathcal{T}_{v,\ell,\mu}$, and denote the probability distribution of S as $\alpha_{\mathbf{v}}$, i.e.,

$$\alpha_{\mathbf{v}} := \Pr[S = \mathbf{v}].$$

Moreover, let $\boldsymbol{\alpha}$ denote the probability vector for S , such that $\boldsymbol{\alpha} = [\alpha_{\mathbf{v}_1}, \dots, \alpha_{\mathbf{v}_{|\mathcal{T}_{v,\ell,\mu}|}}]$, where $\mathbf{v}_1, \dots, \mathbf{v}_{|\mathcal{T}_{v,\ell,\mu}|} \in \mathcal{T}_{v,\ell,\mu}$ and $\boldsymbol{\alpha} \in \mathcal{D}(\mathcal{T}_{v,\ell,\mu})$.

After drawing the rank profile \mathbf{v} according to the distribution $\alpha_{\mathbf{v}}$, the next step is to construct the super support \mathcal{F} . We draw \mathcal{F} uniformly from the set $\Xi_{q,\mu}(\mathbf{v})$, which contains all valid super supports for the given rank profile \mathbf{v} . Each block \mathcal{F}_i is drawn independently from the set of subspaces of \mathbb{F}_q^μ with dimension v_i . By controlling the distribution $\alpha_{\mathbf{v}}$, we influence the distribution of the super support \mathcal{F} and aim to minimize the expected complexity of the decoding process.

The following theorem gives the success probability for uniquely decoding a solution using a single iteration of Algorithm 1 under the average-case setting.

Theorem 2. Let $\mathcal{C} \subseteq \mathbb{F}_{q^m}^n$ be a linear sum-rank metric code with length n and dimension k . Let $\mathbf{c} \in \mathcal{C}$ be a codeword and consider a channel model as described in Section II-D, where the error \mathbf{e} is drawn uniformly at random from $\mathcal{E}_{q,\eta,m,\ell}(w)$, as defined in (2). Let $\text{wt}_{\Sigma R}(\mathbf{e}) = w$, and assume that $\mathbf{y} = \mathbf{c} + \mathbf{e}$. Let v_{\max} denote the maximum sum-rank weight of the guessed super support, as defined in (8). Define the event $\mathcal{E}_{\text{unique}}$ as the event that Algorithm 1 outputs \mathbf{c} in a single iteration for the scenario of Problem 2. The probability of $\mathcal{E}_{\text{unique}}$ is given by

$$\Pr[\mathcal{E}_{\text{unique}}] = \frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{v'=w}^{v_{\max}} \sum_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v'}(\mathbf{w}) \cdot \prod_{i=1}^{\ell} \text{NM}_q(m, \eta, w_i), \quad (17)$$

where $\varphi_{q,\mu,v'}(\mathbf{w})$ is the average probability defined in (18).

Proof. The average probability of decoding success can be expressed as a sum over all possible super space dimensions v' and error weight decompositions \mathbf{w} , weighted by the probability of each error weight decomposition

$$\Pr[\mathcal{E}_{\text{unique}}] = \sum_{v'=w}^{v_{\max}} \sum_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \Pr[\mathbf{w}] \cdot \varphi_{q,\mu,v'}(\mathbf{w}).$$

Substituting the expression for $\Pr[\mathbf{w}]$ from (7), we arrive at the expression given in (17)

$$\Pr[\mathcal{E}_{\text{unique}}] = \frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{v'=w}^{v_{\max}} \sum_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v'}(\mathbf{w}) \cdot \prod_{i=1}^{\ell} \text{NM}_q(m, \eta, w_i),$$

completing the proof. \square

An upper bound on the probability of obtaining alternative solutions when using random linear sum-rank-metric codes is provided by the upcoming theorem. For the analysis, we define $\varphi_{q,\mu,v}(\mathbf{w})$ as the average probability over all super-support rank profiles

$$\varphi_{q,\mu,v}(\mathbf{w}) := \sum_{\mathbf{v} \in \mathcal{T}_{v,\ell,\mu}} \alpha_{\mathbf{v}} \cdot \varphi_{q,\mu,v}(\mathbf{v}, \mathbf{w}), \quad (18)$$

where $\alpha_{\mathbf{v}}$ is the probability distribution over the super-support rank profiles.

Theorem 3 (Random Coding Union Bound). Let \mathcal{C} be a random code of length n and cardinality $|\mathcal{C}| = q^{mk}$ over \mathbb{F}_{q^m} , where each codeword is drawn uniformly at random from the ambient space $\mathbb{F}_{q^m}^n$. Suppose that the received word $\mathbf{y} \in \mathbb{F}_{q^m}^n$ is a noisy version of a codeword $\mathbf{c} \in \mathcal{C}$, corrupted by an error vector $\mathbf{e} \in \mathbb{F}_{q^m}^n$ of sum-rank weight w , i.e., $\mathbf{y} = \mathbf{c} + \mathbf{e}$. Let v be an integer satisfying $w \leq v \leq v_{\max}$.

The probability $\Pr[\mathcal{E}_{\text{RCU}}]$ of one iteration of Algorithm 1 to output an alternative solution \mathbf{c}' with $\mathbf{c}' \neq \mathbf{c}$ is upper bounded by

$$\Pr[\mathcal{E}_{\text{RCU}}] \leq \text{P}_{\text{RCU}}^{(\text{UB,gen})},$$

where

$$\text{P}_{\text{RCU}}^{(\text{UB,gen})} := q^{m(k-n)} \sum_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v}(\mathbf{w}) \prod_{i=1}^{\ell} \text{NM}_q(m, \eta, w_i),$$

and $\varphi_{q,\mu,v}(\mathbf{w})$ is the average probability defined in (18).

Proof. By assumption, each codeword in the codebook \mathcal{C} is drawn uniformly at random over $\mathbb{F}_{q^m}^n$. Let $\mathbf{c}_j \in \mathcal{C}$ with $\mathbf{c}_j \neq \mathbf{c}$ be one such alternative codeword with $j \in \{1, \dots, q^{mk} - 1\}$, and define \mathcal{X}_j as the event that Algorithm 1 can decode this codeword. Then

$$\Pr[\mathcal{X}_j] = \sum_{\substack{\mathbf{e}' \in \mathbb{F}_{q^m}^n \\ \text{wt}_{\Sigma R}(\mathbf{e}') = w}} \frac{1}{q^{mn}} \cdot \varphi_{q,\mu,v}(\psi(\mathbf{e}')).$$

Since $\varphi_{q,\mu,v}(\psi(\mathbf{e}'))$ only depends on the rank profile of \mathbf{e}' , we can change the sum to be over all rank profiles $\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}$ and multiply by the number of error vectors that have the same rank profile

$$\Pr[\mathcal{X}_j] = \sum_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \frac{1}{q^{mn}} \cdot \varphi_{q,\mu,v}(\mathbf{w}) \prod_{i=1}^{\ell} \text{NM}_q(m, \eta, w_i).$$

The total probability of successful decoding is given by the union of the events $\mathcal{X}_1, \dots, \mathcal{X}_{q^{mk}-1}$, which can be upper bounded by

$$\Pr\left[\bigcup_{j=1}^{q^{mk}-1} \mathcal{X}_j\right] \leq \sum_{j=1}^{q^{mk}-1} \Pr[\mathcal{X}_j] \leq q^{mk} \Pr[\mathcal{X}_j] = \text{P}_{\text{RCU}}^{(\text{UB,gen})}.$$

Substituting the expression for $\Pr[\mathcal{X}_j]$ yields the desired upper bound on the success probability. \square

Combining Theorem 3 and Theorem 2, we can derive bounds on the success probability of Algorithm 1 for one iteration to return at least one solution. We state these bounds in the following lemma.

Theorem 4. *Let \mathcal{C} be a random code of length n and size q^{mk} over \mathbb{F}_{q^m} , where each codeword is drawn uniformly at random from the ambient space $\mathbb{F}_{q^m}^n$. Suppose that the received word $\mathbf{y} \in \mathbb{F}_{q^m}^n$ is a noisy version of a codeword $\mathbf{c} \in \mathcal{C}$, corrupted by an error vector $\mathbf{e} \in \mathbb{F}_{q^m}^n$ with*

$\text{wt}_{\Sigma R}(\mathbf{e}) = w$, i.e., $\mathbf{y} = \mathbf{c} + \mathbf{e}$. The success probability of Algorithm 1 to output at least one solution satisfies

$$\Pr[\text{success}] \geq \frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{v'=w}^{v_{\max}} \sum_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v'}(\mathbf{w}) \cdot \prod_{i=1}^{\ell} \text{NM}_q(m, \eta, w_i),$$

and

$$\Pr[\text{success}] \leq \left(\frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} + q^{m(k-n)} \right) \sum_{v'=w}^{v_{\max}} \sum_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v'}(\mathbf{w}) \cdot \prod_{i=1}^{\ell} \text{NM}_q(m, \eta, w_i).$$

Proof. First, we prove the lower bound on the success probability. Recall that we assume the received word $\mathbf{y} \in \mathbb{F}_{q^m}^n$ is a noisy version of a codeword $\mathbf{c} \in \mathcal{C}$, corrupted by an error vector $\mathbf{e} \in \mathbb{F}_{q^m}^n$ with $\text{wt}_{\Sigma R}(\mathbf{e}) = w$, i.e., $\mathbf{y} = \mathbf{c} + \mathbf{e}$. This implies that the codeword \mathbf{c} is always within the decoding radius of the received word \mathbf{y} . Using the expression for $\Pr[\mathcal{E}_{\text{unique}}]$ from Theorem 2, we have

$$\Pr[\text{success}] \geq \Pr[\mathcal{E}_{\text{unique}}] = \frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{v'=w}^{v_{\max}} \sum_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v'}(\mathbf{w}) \cdot \prod_{i=1}^{\ell} \text{NM}_q(m, \eta, w_i).$$

Next, we prove the upper bound on the success probability. Using union bound arguments and the expressions for $\Pr[\mathcal{E}_{\text{RCU}}]$ and $\Pr[\mathcal{E}_{\text{unique}}]$ from Theorem 3 and Theorem 2, respectively, we obtain

$$\begin{aligned} \Pr[\text{success}] &\leq \Pr[\mathcal{E}_{\text{unique}}] + \Pr[\mathcal{E}_{\text{RCU}}] \\ &= \frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{v'=w}^{v_{\max}} \sum_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v'}(\mathbf{w}) \cdot \prod_{i=1}^{\ell} \text{NM}_q(m, \eta, w_i) \\ &\quad + q^{m(k-n)} \sum_{v'=w}^{v_{\max}} \sum_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v'}(\mathbf{w}) \cdot \prod_{i=1}^{\ell} \text{NM}_q(m, \eta, w_i) \\ &= \left(\frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} + q^{m(k-n)} \right) \sum_{v'=w}^{v_{\max}} \sum_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \varphi_{q,\mu,v'}(\mathbf{w}) \cdot \prod_{i=1}^{\ell} \text{NM}_q(m, \eta, w_i), \end{aligned}$$

which concludes the lemma. \square

From Theorem 4 we get that to find an optimal distribution $\alpha_{\mathbf{v}}$ to draw \mathbf{v} from $\mathcal{T}_{v,\ell,\mu}$ we need to maximize the term

$$\max_{\alpha \in \mathcal{D}(\mathcal{T}_{v,\ell,\mu})} \sum_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \sum_{\mathbf{v} \in \mathcal{T}_{v,\ell,\mu}} \alpha_{\mathbf{v}} \cdot \varphi_{q,\mu}(\mathbf{v}, \mathbf{w}) \cdot \prod_{i=1}^{\ell} \text{NM}_q(m, \eta, w_i),$$

where $\mathcal{D}(\mathcal{T}_{v,\ell,\mu})$ is the set of all valid PMFs over $\mathcal{T}_{v,\ell,\mu}$ as defined in (1).

C. Optimizing the Support-Drawing Distribution via Linear Programming

The process of drawing a super support from a known distribution can be further broken down. Instead of drawing a rank profile $\mathbf{v} \in \mathcal{T}_{v,\ell,\mu}$ according to $\alpha_{\mathbf{v}}$, we can draw an ordered rank profile $\mathbf{v}' \in \mathcal{J}_{v,\ell,\mu}$ according to a distribution $\tilde{\alpha}_{\mathbf{v}'}$, where $\mathbf{v}' = \text{sort}(\mathbf{v})$ is obtained by sorting the elements of \mathbf{v} in non-increasing order. This simplification is possible due to symmetry, as the probability of drawing a particular rank profile remains the same for all permutations of that profile.

After drawing the ordered rank profile \mathbf{v}' , we perform a uniformly random permutation to obtain the final rank profile \mathbf{v} . The relation between the two probability distributions is given by

$$\alpha_{\mathbf{v}} = \frac{\tilde{\alpha}_{\text{sort}(\mathbf{v})}}{|\mathfrak{S}_{\ell,\mu}(\text{sort}(\mathbf{v}))|} = \frac{\tilde{\alpha}_{\text{sort}(\mathbf{v})}}{|\mathfrak{S}_{\ell,\mu}(\mathbf{v})|}. \quad (19)$$

By reducing the problem to optimizing the distribution $\tilde{\alpha}_{\mathbf{v}'}$ of ordered rank profiles, we have reduced the number of unknowns since we have $|\mathcal{J}_{v,\ell,\mu}| \leq |\mathcal{T}_{v,\ell,\mu}|$.

In summary, the process of drawing a suitable super support \mathcal{F} can be broken down into three steps:

- 1) Draw an ordered rank profile $\mathbf{v}' \in \mathcal{J}_{v,\ell,\mu}$ according to a distribution $\tilde{\alpha}_{\mathbf{v}'}$, which is the criterion we need to optimize, and then apply a uniformly random permutation to obtain the rank profile \mathbf{v} .
- 2) For each $i \in \{1, \dots, \ell\}$, draw \mathcal{F}_i from the set of all spaces of dimension v_i , independently for all blocks.
- 3) Combine the individual blocks \mathcal{F}_i to form the overall super support $\mathcal{F} = \mathcal{F}_1 \times \dots \times \mathcal{F}_\ell$.

By making use of (19) we can reduce the number of unknowns and instead maximize

$$\begin{aligned} & \max_{\tilde{\alpha} \in \mathcal{D}(\mathcal{J}_{v,\ell,\mu})} \sum_{\mathbf{w} \in \mathcal{J}_{w,\ell,\mu}} \sum_{\mathbf{v} \in \mathcal{T}_{v,\ell,\mu}^{\geq \mathbf{w}}} \frac{|\mathfrak{S}_{\ell,\mu}(\mathbf{w})|}{|\mathfrak{S}_{\ell,\mu}(\mathbf{v})|} \tilde{\alpha}_{\text{sort}(\mathbf{v})} \varphi_{q,\mu}(\mathbf{v}, \mathbf{w}) \prod_{i=1}^{\ell} \text{NM}_q(m, \eta, w_i) \\ & = \max_{\tilde{\alpha}_{\mathbf{v}'} \in \mathcal{D}(\mathcal{J}_{v,\ell,\mu})} \sum_{\mathbf{v}' \in \mathcal{J}_{v,\ell,\mu}} \tilde{\alpha}_{\mathbf{v}'} \cdot f(\mathbf{v}'), \end{aligned} \quad (20)$$

where

$$f(\mathbf{v}') := \sum_{\mathbf{w} \in \mathcal{J}_{w,\ell,\mu}} \frac{|\mathfrak{S}_{\ell,\mu}(\mathbf{w})|}{|\mathfrak{S}_{\ell,\mu}(\mathbf{v}')|} \left(\sum_{\mathbf{v}'' \in \mathfrak{S}_{\ell,\mu}(\mathbf{v}')} \varphi_{q,\mu,v}(\mathbf{v}'', \mathbf{w}) \right) \prod_{i=1}^{\ell} \text{NM}_q(m, \eta, w_i).$$

This optimization problem can be solved via linear program (LP) methods, where the objective function is (20) with $|\mathcal{J}_{v,\ell,\mu}|$ unknowns. Although we have reduced the number of unknowns by

restricting the optimization to the ordered set $\mathcal{J}_{v,\ell,\mu}$, it is important to note that the cardinality of this set, and consequently the number of unknowns, can still grow super-polynomially with the parameters v , ℓ , and μ . Furthermore, computing the coefficients of the constraints requires summing over the set $\mathcal{J}_{w,\ell,\mu}$, which can be computationally demanding due to its potentially large cardinality. Even if we successfully derive the optimal support-drawing distribution through this process, implementing an efficient algorithm to sample from this distribution poses another significant challenge. This limitation motivates the need for alternative approaches to simplify the optimization problem and develop more practical sampling algorithms.

D. Efficient Optimization of the Support-Drawing Distribution in Generic Decoding

In this section, we propose an efficient method to optimize the support-drawing distribution, addressing the computational challenges discussed earlier. By assuming independence between the sum-rank metric blocks, we greatly simplify the problem. Instead of drawing a complete rank profile vector $\mathbf{v} \in \mathcal{T}_{w,\ell,\mu}$ with a fixed total rank v , we independently draw the rank v_i for each of the ℓ blocks. As a result, the sum rank $v = \sum_{i=1}^{\ell} v_i$ becomes a random variable. To prevent it from becoming unbounded, we constrain its expected value, $\mathbb{E}[v]$, to match a predetermined relative sum-rank weight v/ℓ . This assumption reduces the complexity of the optimization problem by focusing on the distributions for individual blocks, and it enables efficient sampling from the optimized distribution, overcoming the practical limitations of the previous approach.

Although this heuristic approach may not always yield the optimal solution that accounts for the dependencies between the ranks of the guessed supports across different blocks, it still provides a good approximation. We demonstrate this numerically in Section IV-E by comparing the performance of the heuristic solution with solutions that consider these dependencies, for parameters where the more complex optimization method is feasible.

Let $\alpha_i^{(m)}$ denote the marginal probability of drawing a super support \mathcal{F}_i with dimension v_i , where $0 \leq v_i \leq \mu$ for $i \in \{1, \dots, \ell\}$, and let $\boldsymbol{\alpha}^{(m)} = [\alpha_0^{(m)}, \dots, \alpha_\mu^{(m)}]$ represent the marginal probability vector. Assuming that the dimension of each subspace \mathcal{F}_i is drawn independently according to $\boldsymbol{\alpha}^{(m)}$, the probability of a given rank profile $\mathbf{v} = [v_1, \dots, v_\ell] \in \mathcal{T}_{v,\ell,\mu}$ is given by

$$\alpha_{\mathbf{v}} = \prod_{i=1}^{\ell} \alpha_{v_i}^{(m)}. \quad (21)$$

We define the following two quantities

$$\tilde{B}_{q,m,\eta}(w, v, \ell) := \sum_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \sum_{\mathbf{v} \in \mathcal{T}_{v,\ell,\mu}} \alpha_{\mathbf{v}} \text{NM}_q(m, \eta, w_i) \mathbf{P}_{q,\mu}^{\subseteq}(w_i \subseteq v_i), \quad (22)$$

and

$$B_{q,m,\eta}(w, v, \ell) := \sum_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \sum_{\mathbf{v} \in \mathcal{T}_{v,\ell,\mu}} \prod_{i=1}^{\ell} \alpha_{v_i}^{(m)} \text{NM}_q(m, \eta, w_i) \mathbf{P}_{q,\mu}^{\subseteq}(w_i \subseteq v_i), \quad (23)$$

where (23) is a special case of (22) using our independence assumption. In Appendix A, we show that (23) is efficiently computable in polynomial time. Using the definition of $B_{q,m,\eta}(w, v, \ell)$ from (23) and the relaxation in (21), we can restate Theorem 2, Theorem 3, and Theorem 4 in the following corollaries, respectively.

Corollary 1. *The probability $\Pr[\mathcal{E}_{\text{RCU}}]$ of having an alternative solution in Algorithm 1 for a random linear code of length n and cardinality $M = q^{mk}$ over \mathbb{F}_{q^m} can be upper bounded as*

$$\Pr[\mathcal{E}_{\text{RCU}}] \leq q^{m(k-n)} \sum_{v=w}^{v_{\max}} \tilde{B}_{q,m,\eta}(w, v, \ell).$$

Corollary 2. *The probability $\Pr[\mathcal{E}_{\text{unique}}]$ that Algorithm 1 outputs a unique solution \mathbf{c} for a random linear code of length n and cardinality $M = q^{mk}$ over \mathbb{F}_{q^m} is given by*

$$\Pr[\mathcal{E}_{\text{unique}}] = \frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{v=w}^{v_{\max}} \tilde{B}_{q,m,\eta}(w, v, \ell).$$

Corollary 3. *The success probability of Algorithm 1 to output at least one solution satisfies*

$$\Pr[\text{success}] \geq \frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{v=w}^{v_{\max}} \tilde{B}_{q,m,\eta}(w, v, \ell),$$

and

$$\Pr[\text{success}] \leq \left(\frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} + q^{m(k-n)} \right) \sum_{v=w}^{v_{\max}} \tilde{B}_{q,m,\eta}(w, v, \ell).$$

From Corollary 3, under our independence assumption, the success probability is proportional to the term

$$\sum_{v=w}^{v_{\max}} \tilde{B}_{q,m,\eta}(w, v, \ell) = \sum_{v=w}^{v_{\max}} B_{q,m,\eta}(w, v, \ell), \quad (24)$$

which we aim to maximize over all possible $\boldsymbol{\alpha}^{(m)} \in \mathcal{D}(\{0, \dots, \mu\})$.

In the following, we further upper bound the expression in (24) and propose a method to maximize this upper bound, to obtain a valid solution for $\alpha^{(m)} \in \mathcal{D}(\{0, \dots, \mu\})$. That is

$$\begin{aligned}
\sum_{v=w}^{v_{\max}} B_{q,m,\eta}(w, v, \ell) &= \sum_{v=t}^{v_{\max}} \sum_{\mathbf{w} \in \mathcal{T}_{t,\ell,\mu}} \sum_{\mathbf{v} \in \mathcal{T}_{v,\ell,\mu}} \prod_{i=1}^{\ell} \alpha_{v_i}^{(m)} \text{NM}_q(m, \eta, w_i) \text{P}_{q,\mu}^{\subseteq}(w_i \subseteq v_i) \\
&\leq \sum_{\mathbf{w} \in \{0, \dots, \mu\}^{\ell}} \sum_{\mathbf{v} \in \{0, \dots, \mu\}^{\ell}} \prod_{i=1}^{\ell} \alpha_{v_i}^{(m)} \text{NM}_q(m, \eta, w_i) \text{P}_{q,\mu}^{\subseteq}(w_i \subseteq v_i) \\
&= \left(\sum_{w=0}^{\mu} \sum_{v=0}^{\mu} \alpha_v^{(m)} \text{NM}_q(m, \eta, w) \text{P}_{q,\mu}^{\subseteq}(w \subseteq v) \right)^{\ell}, \tag{25}
\end{aligned}$$

where the last equality follows from the fact that we're summing over all possible ℓ -tuples of w_i and v_i , and for each tuple, we're computing the product of functions that depend only on w_i and v_i . Since each (w_i, v_i) pair is independent and the summations are over the same finite ranges, the combined sum over all vectors can be expressed as the ℓ -th power of a single sum over w and v . This is due to the distributive property of multiplication over addition and the independence of each component in the tuples, allowing us to factor the multiple sums and products into a single term raised to the power of ℓ .

To maximize the right-hand side of (25), it suffices to maximize the expression

$$\sum_{w'=0}^{\mu} \sum_{v'=0}^{\mu} \alpha_{v'}^{(m)} \text{NM}_q(m, \eta, w') \text{P}_{q,\mu}^{\subseteq}(w' \subseteq v'). \tag{26}$$

This expression is closely related to the average probability that a randomly drawn super space \mathcal{F}_i contains the error space \mathcal{E}_i in a single block, averaged over all possible rank weights w' . The average single-block success probability is given as

$$\sum_{w'=0}^{\mu} \text{Pr}[w'] \sum_{v'=0}^{\mu} \alpha_{v'}^{(m)} \text{P}_{q,\mu}^{\subseteq}(w' \subseteq v'), \tag{27}$$

where $\text{Pr}[w']$ is the marginal probability of an error of rank weight w' occurring in a single block.

In the asymptotic setting, where η and m are fixed and the number of blocks $\ell \rightarrow \infty$, the assumption of independence between blocks becomes valid due to the law of large numbers and the concept of typical sequences from statistical mechanics. In this regime, the empirical distribution of error weights in the blocks converges to the marginal distribution $\text{Pr}[w']$, which can be approximated by the Boltzmann distribution [36]

$$\Pr[w'] = \frac{\text{NM}_q(m, \eta, w')}{\sum_{w''=0}^{\mu} \text{NM}_q(m, \eta, w'') e^{-\lambda w''}} \cdot e^{-\lambda w'}, \quad (28)$$

where λ is the unique solution to the weight constraint

$$\mathbb{E}[w'] = \sum_{w''=0}^{\mu} w'' \cdot \Pr[w''] = \frac{w}{\ell}.$$

By substituting (28) into (27), we obtain the single-block success probability under the asymptotic error-weight distribution.

Maximizing the single-block success probability in (27) effectively maximizes the overall success probability in the asymptotic regime. Although (26) represents an upper bound on the success probability, this upper bound becomes tight as $\ell \rightarrow \infty$ due to the convergence properties established by the law of large numbers. Therefore, optimizing this upper bound is justified because it aligns with maximizing the actual success probability in the asymptotic setting.

This connection reveals that optimizing (26) to obtain an optimal marginal distribution $\alpha_{v'}^{(m)}$ for the guessed super-support dimensions is beneficial for maximizing (24).

To optimize (26), our approach focuses on the marginal distribution $\alpha^{(m)} \in \mathcal{D}(\{0, \dots, \mu\})$ rather than directly optimizing α , aiming to approximate the optimal average rank profile for the super support. Since directly optimizing $\alpha^{(m)}$ results in a distribution independent of the number of blocks ℓ , we impose the constraint $\alpha_i^{(m)} = \frac{x_i}{\ell}$, where $x_i \in \mathbb{Z}_{\geq 0}$ represents the number of occurrences of rank i across the ℓ blocks.

We then maximize the objective in (25) using linear integer programming with appropriate constraints and non-negativity conditions. This method assumes independence of rank weights across the ℓ blocks, which holds asymptotically as $\ell \rightarrow \infty$ for fixed η and m .

By applying this method, we obtain a solution $\mathbf{x} = [x_0, \dots, x_\mu]$, from which we construct the ordered rank profile $\hat{\mathbf{v}} \in \mathcal{J}_{w, \ell, \mu}$ as

$$\hat{\mathbf{v}} = [\underbrace{\mu, \dots, \mu}_{x_\mu \text{ times}}, \underbrace{\mu - 1, \dots, \mu - 1}_{x_{\mu-1} \text{ times}}, \dots, \underbrace{1, \dots, 1}_{x_1 \text{ times}}, \underbrace{0, \dots, 0}_{x_0 \text{ times}}], \quad (29)$$

where each element $i \in \{0, \dots, \mu\}$ appears exactly x_i times in the vector $\hat{\mathbf{v}}$. We then have that

$$\tilde{\alpha}_{v'}^{(\text{heu})} := \begin{cases} 1 & \text{if } v' = \hat{\mathbf{v}}, \\ 0 & \text{otherwise.} \end{cases} \quad (30)$$

Using the relation in (19), we obtain the overall probability for \mathbf{v} , i.e., $\alpha_{\mathbf{v}}^{(\text{heu})}$. For this specific PMF we can write (22) as

$$\tilde{B}_{q,m,\eta}(w, v, \ell) = \sum_{w \in \mathcal{T}_{w,\ell,\mu}} \prod_{i=1}^{\ell} \text{NM}_q(m, \eta, w_i) \mathbf{P}_{q,\mu}^{\subseteq}(w_i \subseteq \hat{v}_i).$$

As shown in Appendix A, this expression can be efficiently computed in polynomial time since it is a special case of (23) for a fixed vector $\mathbf{v} = \hat{\mathbf{v}}$.

Thus, the bounds on the overall expected runtime provided in the following theorem, which are general for any support-guessing distribution, can be efficiently computed for our specific support-guessing distribution given by (29) and (30).

Theorem 5. *Under the same assumptions as in Corollary 3, the overall expected runtime $W_{\text{gen,RCU}}$ of Algorithm 1 to output at least one solution is bounded by*

$$W_{\text{gen,RCU}}^{(\text{LB})} \leq W_{\text{gen,RCU}} \leq W_{\text{gen,RCU}}^{(\text{UB})},$$

with

$$W_{\text{gen,RCU}}^{(\text{LB})} := W_{\text{erasure-dec}} \left(\left(\frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} + q^{m(k-n)} \right) \sum_{v=w}^{v_{\max}} \tilde{B}_{q,m,\eta}(w, v, \ell) \right)^{-1}, \quad (31)$$

and

$$W_{\text{gen,RCU}}^{(\text{UB})} := W_{\text{erasure-dec}} \left(\frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{v=w}^{v_{\max}} \tilde{B}_{q,m,\eta}(w, v, \ell) \right)^{-1}, \quad (32)$$

where $W_{\text{erasure-dec}}$ denotes the cost of one iteration of Algorithm 1 and $W_{\text{erasure-dec}} \in O((n-k)^3 m^3)$ operations over \mathbb{F}_q and we neglect the complexity of drawing from $\alpha_{\mathbf{v}}$.

Proof. For the lower bound on the complexity, we consider the worst-case scenario for the complexity of each iteration, denoted by $W_{\text{erasure-dec}}$. The expected number of iterations until success is the reciprocal of the success probability. Using the upper bound on the success probability from Corollary 3, the lower bound on the overall expected runtime satisfies

$$W_{\text{gen,RCU}}^{(\text{LB})} = W_{\text{erasure-dec}} \left(\left(\frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} + q^{m(k-n)} \right) \sum_{v=w}^{v_{\max}} \tilde{B}_{q,m,\eta}(w, v, \ell) \right)^{-1}.$$

For the upper bound on the complexity, we use the cost of one iteration $W_{\text{erasure-dec}}$, which is $O((n-k)^3 m^3)$ operations over \mathbb{F}_q according to [26, Theorem 13 and Theorem 14]. Using the heuristic probability distribution for the guessing super support as in (29), we can neglect the complexity of drawing the rank profile of the guessing support. The expected number of

iterations until success is the reciprocal of the success probability. Using the lower bound on the success probability from Corollary 3, the upper bound on the overall expected runtime satisfies

$$W_{\text{gen,RCU}}^{(\text{UB})} = W_{\text{erasure-dec}} \left(\frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{v=w}^{v_{\max}} B_{q,m,\eta}(w, \mathbf{v}, \ell) \right)^{-1},$$

which concludes the theorem. \square

Note that the assumption made in Theorem 5, which neglects the complexity of drawing from α_v , is valid since, in our solution, we only need to permute the support-guessing rank profile \hat{v} uniformly at random.

E. Numerical Results for the Generic Decoding Algorithm

In this section, we compare the complexity analysis of using a support-drawing distribution derived from the method described in Section IV-D for the average case against the worst-case bounds from [26]. We evaluate the average complexity over all error patterns for a specific sum-rank weight and plot the logarithmic complexity (base 2) versus the number of blocks ℓ , while keeping the code parameters and field size q^m constant. The length of each individual block η is adjusted as ℓ varies.

Figure 2 shows the complexity for generic decoding beyond the unique decoding radius with parameters $q = 2$, $m = 20$, $n = 60$, $k = 30$, $t = 9$, and $v = 10$ while in Figure 3, we increase v to v_{\max} . We include the upper bound $W_{\text{gen,RCU}}^{(\text{UB})}$ (32) and the lower bound $W_{\text{gen,RCU}}^{(\text{LB})}$ (31) for the expected complexity of Algorithm 1. The lower bound reflects the effect of decoding beyond the unique decoding radius and accounts for alternative solutions.

The figures show that the effect of alternative solutions, indicated by the divergence between the upper and lower bounds, becomes more prominent near the rank metric (i.e., when ℓ approaches 1) for the chosen parameters. However, this behavior can vary depending on the parameters, as seen in Figure 4, which uses the parameters $q = 2$, $m = 6$, $n = 36$, $k = 22$, $t = 10$, and $v = 10$. In this case, a significant difference between the lower and upper bounds persists even when the number of blocks does not correspond to the rank metric.

With the increase of v to v_{\max} in Figure 3, the upper and lower bounds for the worst-case scenario become even looser. The upper bound, in particular, deviates significantly from the complexity of the Prange algorithm, denoted as W_{Prange} , and given by [37]

$$W_{\text{Prange}} = W_{\text{gen}}^{(\text{iter})} \frac{\binom{n}{w}}{\binom{v}{w}},$$

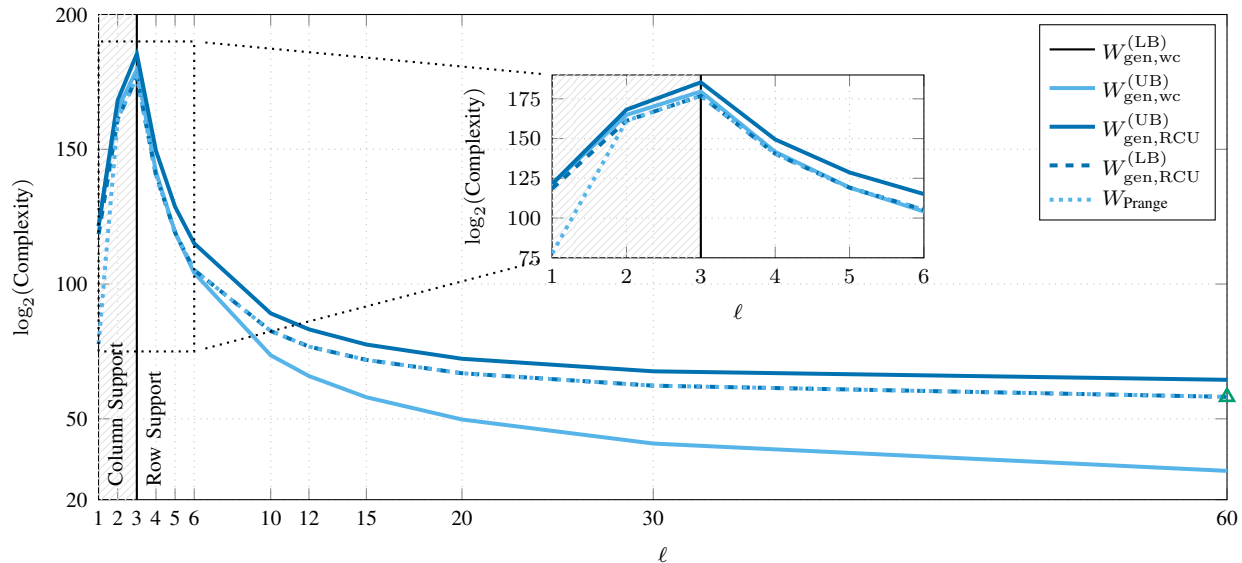


Fig. 2. Complexity comparison for generic decoding with parameters: $q = 2$, $m = 20$, $n = 60$, $k = 30$, $t = 9$, and $v = 10$.

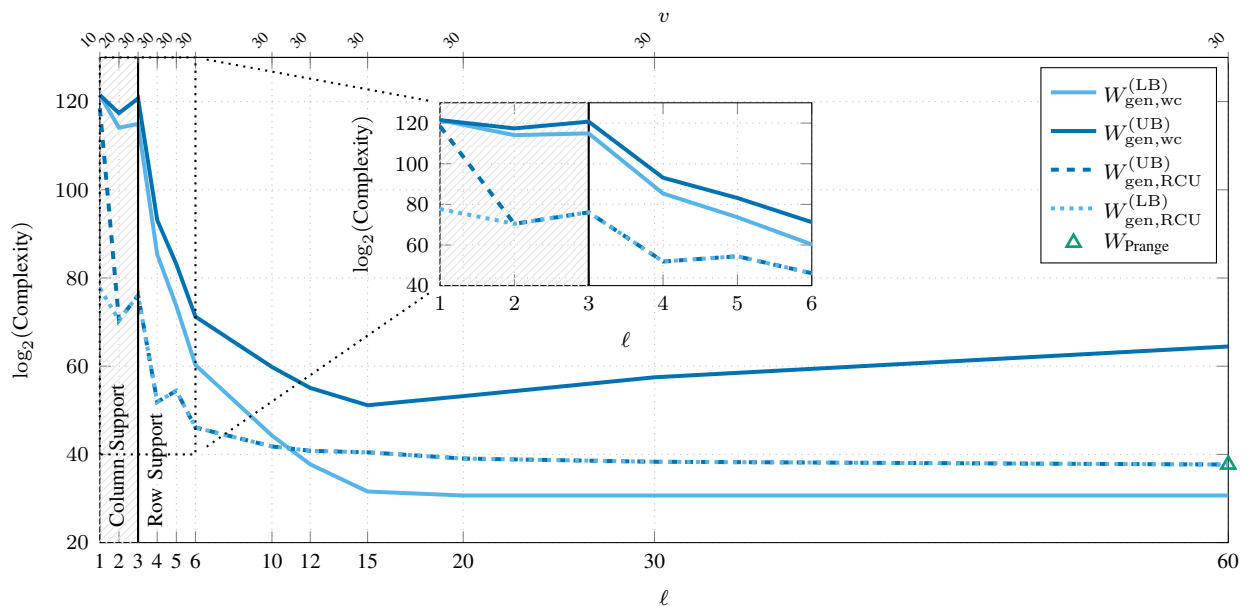


Fig. 3. Complexity comparison for generic decoding with parameters: $q = 2$, $m = 20$, $n = 60$, $k = 30$, $t = 9$ and $v = v_{\max}$.

especially at $\ell = 60$, which corresponds to the special case of the Hamming metric. In this case, not only does the complexity of our solution coincide with the Prange algorithm, but the algorithm itself is exactly the Prange algorithm.

Additionally, in Figure 3, for $\ell < 12$, the complexity of our solution falls below the lower bound of the worst-case scenario. This demonstrates that worst-case bounds may not always provide accurate estimates for real-world scenarios. For instance, when selecting parameters

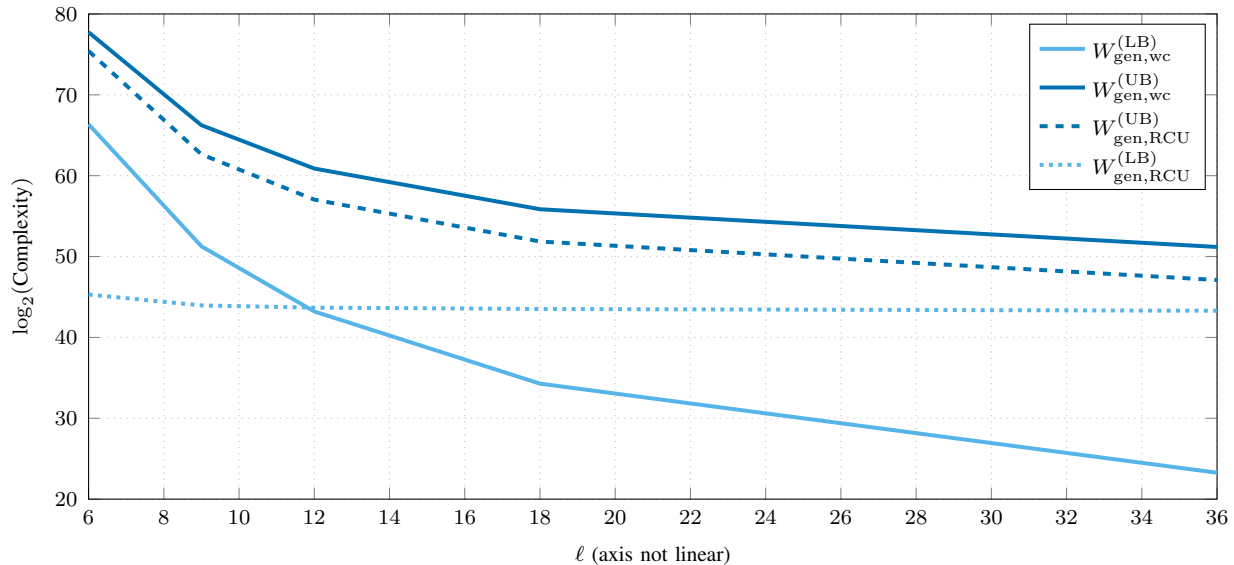


Fig. 4. Complexity comparison for generic decoding with parameters: $q = 2$, $m = 6$, $n = 36$, $k = 22$, $t = 10$ and $v = 10$.

for cryptosystems based on sum-rank metric codes, relying solely on worst-case bounds may lead to underestimating the actual complexity of practical attacks in certain regimes. The lower bound $W_{\text{gen,RCU}}^{(\text{LB})}$ provides a closer approximation to the actual complexity in practice than the worst-case bounds from [26].

We performed extensive computations using LP to account for dependencies between the blocks, as described in Section IV-C, for the parameters used in Figures 2 and 3. For these parameters this approach is feasible for ℓ up to 10 and we obtained the exact same support-guessing distribution corresponding to the rank profiles as our efficient solution. This is particularly interesting, as our efficient method is expected to yield tighter results for larger block sizes, i.e., as $\ell \rightarrow n$. Thus, our findings indicate that for these parameters, the efficient approach remains effective even at lower block counts.

Table I summarizes the individual bounds and their applicable decoding scenarios.

TABLE I
OVERVIEW OF THE BOUNDS AND THEIR APPLICABLE SCENARIOS.

	Bound	Applies to	Conditions
Worst case	$W_{\text{gen,wc}}^{(\text{LB})}$	Problem 4	Exactly one solution exists for the worst-case rank profile channel
	$W_{\text{gen,wc}}^{(\text{UB})}$	Problem 1	At least one solution exists for the worst-case rank profile channel
Average case	$W_{\text{gen,RCU}}^{(\text{UB})}$	Problem 1	At least one solution exists; exact when exactly one solution exists
	$W_{\text{gen,RCU}}^{(\text{LB})}$	Problem 1	At least one solution exists; accounts for alternative solutions

V. GENERIC DECODING FOR LARGE ERROR WEIGHTS

In the previous section, we focused on decoding for low error weights, specifically when the error weight w satisfies $w \leq d_{\min} - 1$, as erasure decoding is not possible beyond this threshold, according to [26, Theorem 13 and Theorem 14]. We explored unique decoding for Problem 4 up to $w \leq d_{\min} - 1$ and going beyond unique decoding is possible for $w \leq \min\{n - k, \frac{m}{\eta}(n - k)\}$ (see [26]).

In this section, we introduce a generic decoding algorithm (Algorithm 2) that aims to solve Problem 1. The proposed algorithm is inspired by the Prange-like algorithm for the Hamming metric, as presented in [22]. Our analysis of the algorithm focuses on the asymptotic case $\ell \rightarrow \infty$ and its average performance. The following proposition establishes the range of relative weights for which solutions to Problem 1 can be found efficiently using Algorithm 2.

Proposition 1. *Consider a \mathbb{F}_{q^m} -linear sum-rank-metric code of length $n = \eta\ell$, dimension $k = \eta\kappa$ with parity check matrix $\mathbf{H} \in \mathbb{F}_{q^m}^{(n-k) \times n}$ and $R := \frac{k}{n} = \frac{\kappa}{\ell}$ where $\kappa \in \mathbb{Z}_{\geq 0}$ and $0 \leq \kappa \leq \ell$. Let the sum-rank weight be defined with respect to the length partition of constant block length, i.e., $\mathbf{n} = [n_1, \dots, n_\ell] = [\eta, \dots, \eta]$. Define*

$$\bar{a} := \frac{\sum_{i=0}^{\mu} i \cdot \text{NM}_q(m, \eta, i)}{q^{m\eta}},$$

as the average rank weight of a single block if drawn uniformly at random. Then, for the relative

Algorithm 2: PrangeSumRank

Input : $H \in \mathbb{F}_{q^m}^{\eta(\ell-\kappa) \times \eta\ell}$, $s \in \mathbb{F}_{q^m}^{\eta(\ell-\kappa)}$ and $w \in \mathbb{Z}_{\geq 0}$

Output : $eH^\top = s$ and $\text{wt}_{\Sigma R}(e) = w$

```

1  $\mu \leftarrow \min\{m, \eta\}$ 
2  $e \leftarrow \mathbf{0} \in \mathbb{F}_{q^m}^{\eta\ell}$ 
3 while  $\text{wt}_{\Sigma R}^{(n)}(e) \neq w$  do
4    $H' \leftarrow \mathbf{0} \in \mathbb{F}_{q^m}^{\eta(\ell-\kappa) \times \eta\ell}$ 
5   while  $\text{rk}_{\mathbb{Q}^m}(H'_{[1:\eta(\ell-\kappa)]}) \neq \eta(\ell - \kappa)$  do
6      $P \xleftarrow{\$}$  Any  $\ell \times \ell$  permutation matrix
7      $P' \leftarrow P \otimes I_\eta$ 
8      $H' \leftarrow HP'$ 
9    $A \leftarrow H'_{[1:\eta(\ell-\kappa)]}$ 
10   $B \leftarrow H'_{[\eta(\ell-\kappa)+1:\eta\ell]}$ 
11   $w_1 \xleftarrow{\$}$   $\{0, \dots, \kappa\mu\}$ 
12   $e' \xleftarrow{\$}$   $\{x \in \mathbb{F}_{q^m}^{\kappa\ell} : \text{wt}_{\Sigma R}^{[n_{\kappa+1}, \dots, n_\ell]}(e') = w_1\}$ 
13   $e \leftarrow ((s - e'B)A^{-\top}, e')(P')^\top$ 
14 return  $e$ 

```

weight $w_{\text{rel}} := w/n$ in the interval $[w_{\text{easy}}^-, w_{\text{easy}}^+]$, where

$$w_{\text{easy}}^- := \frac{1-R}{\eta} \cdot \bar{a},$$

$$w_{\text{easy}}^+ := \frac{1-R}{\eta} \cdot \bar{a} + \frac{R\mu}{\eta},$$

a solution to Problem 1 can be found in probabilistic polynomial time using the Prange-like Algorithm 2.

Proof. To solve Problem 1, we want to find, for a given syndrome s , an error e of sum-rank weight w such that $eH^\top = s$. The matrix H is a full-rank matrix and therefore contains an invertible submatrix $A \in \mathbb{F}_{q^m}^{(n-k) \times (n-k)}$. Without loss of generality, assume that this matrix is formed by the first $n - k = \eta(\ell - \kappa)$ positions, i.e. we have

$$H = [A \mid B]. \tag{33}$$

Assume $e = [e'', e'] \in \mathbb{F}_{q^m}^{\eta\ell}$ with $e'' \in \mathbb{F}_{q^m}^{\eta(\ell-\kappa)}$ and $e' \in \mathbb{F}_{q^m}^{\eta\kappa}$. Then, by (33), we have

$$e'' = (s - e' \mathbf{B}^\top)(\mathbf{A}^{-1})^\top.$$

The idea is to arbitrarily choose e' of length $k = \eta\kappa$. Then, on average, the expected (partial) sum-rank weight of the remaining $\ell - \kappa$ blocks is

$$\mathbb{E} \left[\text{wt}_{\Sigma R}^{([n_{\kappa+1}, \dots, n_\ell])}(e'') \right] = \bar{a} \cdot (\ell - \kappa).$$

The average probability of the sum-rank weight of e is then

$$\mathbb{E} \left[\text{wt}_{\Sigma R}^{(n)}(e) \right] = \underbrace{\mathbb{E} \left[\text{wt}_{\Sigma R}^{([n_1, \dots, n_\kappa])}(e') \right]}_{=: \bar{w}_1} + \mathbb{E} \left[\text{wt}_{\Sigma R}^{([n_{\kappa+1}, \dots, n_\ell])}(e'') \right] = \bar{w}_1 + \bar{a} \cdot (\ell - \kappa),$$

where \bar{w}_1 is determined by the distribution of e' , which we can choose freely. Nonetheless, we have $0 \leq \bar{w}_1 \leq \mu\kappa$, and therefore

$$\underbrace{\bar{a} \cdot (\ell - \kappa)}_{=nw_{\text{easy}}^-} \leq \mathbb{E} \left[\text{wt}_{\Sigma R}^{(n)}(e) \right] \leq \underbrace{\mu\kappa + \bar{a} \cdot (\ell - \kappa)}_{=nw_{\text{easy}}^+}.$$

From this, we deduce that any weight in the interval $w \in [w_{\text{easy}}^- n, w_{\text{easy}}^+ n]$ can be reached probabilistically in polynomial time using a distribution for e' with $\bar{w}_1 = w - w_{\text{easy}}^- n$ such that $\mathbb{E} \left[\text{wt}_{\Sigma R}^{(n)}(e) \right] = w$ and which is sufficiently concentrated around its expectation. Algorithm 2 implements this approach, where in Line 6 to Line 8, the parity-check matrix \mathbf{H} is permuted block-wise among the ℓ blocks, i.e., the permutation is applied to the block indices but not within the blocks. Here, \otimes denotes the Kronecker product, which is used to construct the block-wise permutation matrix. This permutation is reversed in Line 13. \square

The proposition above provides the interval $[w_{\text{easy}}^-, w_{\text{easy}}^+]$ for which a solution to Problem 1 can be found in probabilistic polynomial time using Algorithm 2. Combining this with the Gilbert-Varshamov bound for the sum-rank metric, we have the following summary of the relative weight intervals:

- $w_{\text{rel}} \in [w^-, w^+]$: A solution to Problem 1 is likely to exist (Gilbert-Varshamov bound for the sum-rank metric, see [33])
- $w_{\text{rel}} \in [w_{\text{easy}}^-, w_{\text{easy}}^+]$: A solution to Problem 1 can be found in probabilistic polynomial time using a Prange-like algorithm, as stated in the proposition above.

Figure 5 and Figure 6 show the regions of hardness for finding a solution using Algorithm 2 and the bounds on the relative weight intervals for successful decoding plotted against the code

rate R for different parameters. These results apply asymptotically ($\ell \rightarrow \infty$, for fixed m and μ) and on average. The "no solution", "hard", and "easy" regions indicate the difficulty of finding a solution for different code rates.

In Figure 5, the parameters are set to $m = \eta = 2$, $q = 2$, while in Figure 6, the parameters are $m = \eta = 6$, $q = 2$. Comparing the two figures, we observe that the "hard" region for large relative weights becomes smaller as the values of m and η increase. This indicates that it is easier for Algorithm 2 to decode errors of large relative weight when m and η are larger.

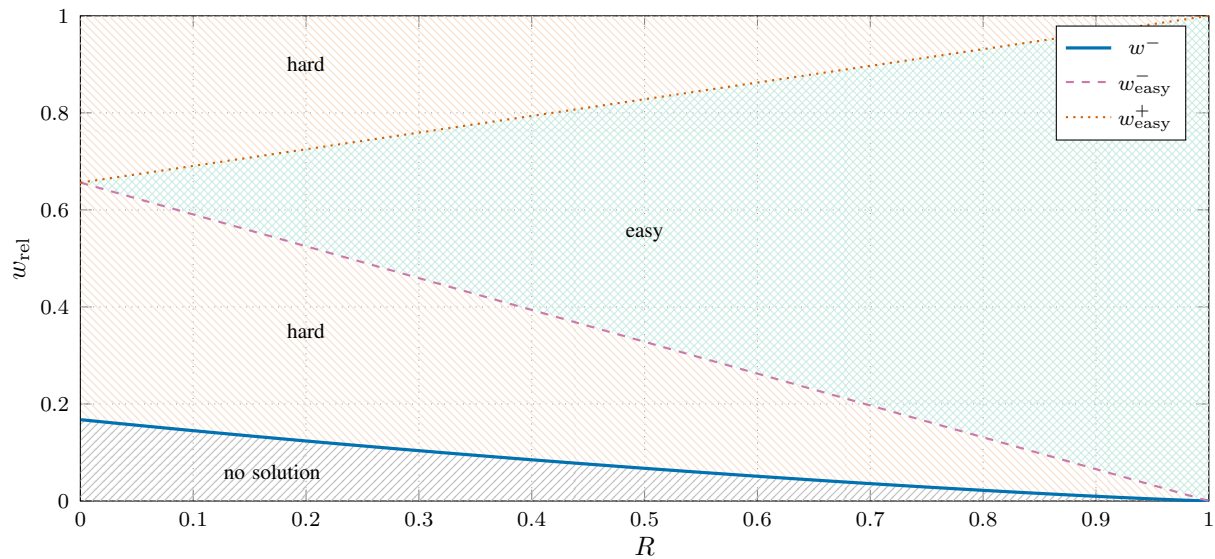


Fig. 5. Regions of hardness for Algorithm 2 and bounds on the relative weight intervals for successful decoding vs code rate $R = k/n$ for parameters: $m = \eta = 2$, $q = 2$ ($\ell \rightarrow \infty$, average-case).

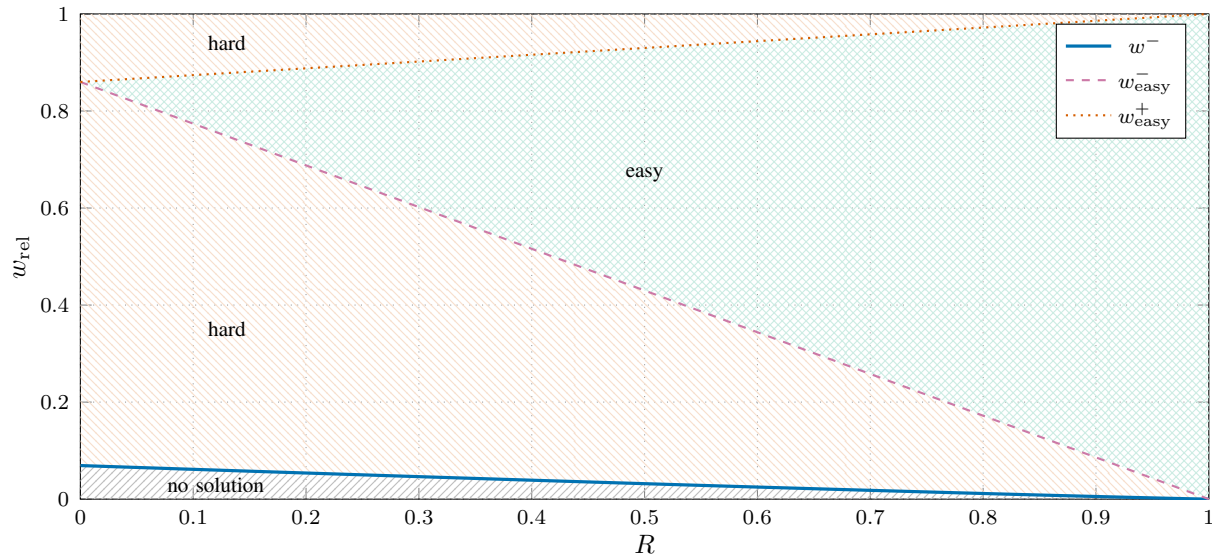


Fig. 6. Regions of hardness for Algorithm 2 and bounds on the relative weight intervals for successful decoding vs code rate $R = k/n$ for parameters: $m = \eta = 6$, $q = 2$ ($\ell \rightarrow \infty$, average-case).

VI. RANDOMIZED DECODING OF LINEARIZED REED–SOLOMON CODES

In this section, we consider the probabilistic decoding algorithm introduced in [27] to solve Problem 3. In [27], the complexity of this decoder was analyzed for the worst-case rank profile. This algorithm generalizes the randomized decoder for Gabidulin codes from [28] to LRS codes.

We revisit the decoder from [27] and present two new contributions:

- We adapt the methods from [26] to efficiently compute worst-case bounds and sample the support rank profile for the randomized decoder for LRS codes.
- We extend the average-case analysis from Section IV to the randomized decoder, deriving an objective function to optimize the support-drawing distribution, motivated by the asymptotic setting.

The proposed decoder relies on two key aspects. First, we consider an underlying LRS code. Recall that LRS codes are MSRD codes with a minimum sum-rank distance $d_{\min} = n - k + 1$. Efficient algorithms exist to decode LRS codes up to the unique decoding radius $\tau = \frac{n-k}{2}$. We consider LRS codes of length n partitioned into constant block lengths $\mathbf{n} = [n_1, \dots, n_\ell] = [\eta, \dots, \eta]$ and dimension k over \mathbb{F}_{q^m} , denoted by \mathcal{C}_{LRS} . Additionally, LRS codes are restricted to $\ell \leq q - 1$ and $n_i \leq m$ for all $i \in \{1, \dots, \ell\}$.

Second, we focus on decoding beyond the unique decoding radius, where the sum-rank weight of the error \mathbf{w} exceeds τ , i.e., $\text{wt}_{\Sigma R}(\mathbf{w}) = w > \tau$. The error excess beyond the unique decoding radius is defined as

$$\xi := w - \tau.$$

Note that while 2ξ is always an integer, ξ itself does not necessarily need to be an integer.

As discussed in Section III, for errors with weight $w \leq \tau$, Problem 3 has at most one solution. However, for $w > \tau$, multiple solutions may exist. The number of solutions can vary, being either polynomially or exponentially bounded in terms of the code parameters and depending on the structure of the code. This behavior has been studied for LRS codes in [38]. Following the reasoning in [28], we analyze the complexity of finding at least one solution. If the code is list-decodable, this process can be repeated to obtain a list of solutions.

A. Erasures in the Sum-Rank Metric

We consider an error of the form as described in Section II-D. The error \mathbf{e} can be further decomposed into a sum of three types of error vectors

$$\mathbf{e} = \mathbf{e}_F + \mathbf{e}_R + \mathbf{e}_C,$$

where \mathbf{e}_F represents *full errors*, \mathbf{e}_R represents *row erasures*, and \mathbf{e}_C represents *column erasures*. The sum-rank weights of these error vectors are denoted by w_F , w_R , and w_C , respectively, such that $\text{wt}_{\Sigma R}(\mathbf{e}_F) = w_F$, $\text{wt}_{\Sigma R}(\mathbf{e}_R) = w_R$, and $\text{wt}_{\Sigma R}(\mathbf{e}_C) = w_C$ (see [29]).

Each of the three error vectors can be decomposed as in (6)

$$\begin{aligned} \mathbf{e}_F &= \mathbf{a}_F \mathbf{B}_F & \text{with } \mathbf{a}_F &\in \mathbb{F}_q^{w_F} \text{ and } \mathbf{B}_F \in \mathbb{F}_q^{w_F \times n}, \\ \mathbf{e}_R &= \mathbf{a}_R \mathbf{B}_R & \text{with } \mathbf{a}_R &\in \mathbb{F}_q^{w_R} \text{ and } \mathbf{B}_R \in \mathbb{F}_q^{w_R \times n}, \\ \mathbf{e}_C &= \mathbf{a}_C \mathbf{B}_C & \text{with } \mathbf{a}_C &\in \mathbb{F}_q^{w_C} \text{ and } \mathbf{B}_C \in \mathbb{F}_q^{w_C \times n}. \end{aligned}$$

For full errors, neither \mathbf{a}_F nor \mathbf{B}_F are known. For row erasures, \mathbf{a}_R is known but \mathbf{B}_R is unknown. For column erasures, \mathbf{a}_C is unknown but \mathbf{B}_C is known.

An efficient algorithm for LRS codes was proposed in [29], capable of correcting combinations of *full errors*, *row erasures*, and *column erasures* up to

$$2w_F + w_C + w_R \leq n - k, \tag{34}$$

with a complexity of $O(n^2)$ operations over \mathbb{F}_{q^m} . We estimate this complexity, denoted as $W_{\text{ee-dec}}$, over \mathbb{F}_q as $O(m^2n^2)$ and approximate it, similar to the arguments in Section IV, as

$$W_{\text{ee-dec}} \approx m^2n^2. \quad (35)$$

We denote this error-erasure decoder by $\text{DEC}(\mathbf{y}, \mathbf{a}_R, \mathbf{B}_C)$, which takes as input the received word $\mathbf{y} = \mathbf{c} + \mathbf{e}$, along with a basis \mathbf{a}_R of the column support of e_R (row erasures) and/or a basis \mathbf{B}_C of the row support of e_C (column erasures). The decoder outputs a valid codeword $\hat{\mathbf{c}}$ if the condition in (34) is satisfied; otherwise, it returns \emptyset .

B. Randomized Decoding Algorithm

We consider an error e with row support \mathcal{E}_R and column support \mathcal{E}_C . Unlike the generic decoding algorithm, where we guess a super support that must completely contain the actual error support to succeed, the randomized approach aims to guess only parts of the error supports and utilizes an error-and-erasure decoder to succeed with a smaller number of guesses. The steps are as follows:

For each block $i \in \ell$, we have $\mathcal{E}_R^{(i)} \subseteq \mathbb{F}_q^\eta$ and $\mathcal{E}_C^{(i)} \subseteq \mathbb{F}_q^m$. As shown in [28], for Gabidulin codes, guessing a combination of row and column supports does not improve success, and it is more effective to guess from a smaller ambient space. Since this result applies block-wise in the sum-rank metric, we set $\mu = \min\{m, \eta\}$ and guess from \mathcal{E}_R if $\mu = \eta$, otherwise from \mathcal{E}_C .

This ensures the guessed support is always a subspace of \mathbb{F}_q^μ . For simplicity, we use \mathcal{E} to denote the error support, whether from \mathcal{E}_R or \mathcal{E}_C .

To guess parts of \mathcal{E} , we first draw a corresponding rank profile $\mathbf{u} = [u_1, \dots, u_\ell] \in \mathbb{Z}_{\geq 0}^\ell$ according to some PMF denoted as $\beta_{\mathbf{u}} := \Pr[\mathbf{u}]$. Then, a support \mathcal{U} is drawn uniformly at random from $\Xi_{q,\mu}(\mathbf{u})$ with $u := \dim_{\Sigma}(\mathcal{U})$. Define ϵ as the sum dimension of the intersection space between the guessed space \mathcal{U} and the actual error support \mathcal{E} , i.e.,

$$\epsilon := \dim_{\Sigma}(\mathcal{U} \cap \mathcal{E}).$$

The number of full errors w_F is reduced by ϵ , so $w_F = w - \epsilon$, while the number of column or row erasures increases by u , corresponding to the guessed parts of \mathcal{U} . For these guessed parts, we assume knowledge of the column support but not the row support (or vice versa), effectively trading errors for erasures ².

²This approach is reminiscent of the generalized minimum distance (GMD) decoding strategy introduced by Forney [39] for Hamming metric codes, and later extended to the rank metric in [40] by Bossert *et al.*

The error-and-erasure decoder takes as input a vector containing $w - \epsilon$ full errors and u erasures. From the decoding condition in (34), we have

$$2(w - \epsilon) + u \leq n - k,$$

which implies that for successful decoding, we need to have

$$\epsilon \geq \epsilon_{\min} := w + \frac{u - (n - k)}{2} = \xi + \frac{u}{2}. \quad (36)$$

If the intersection between the guessed spaces and the actual error support is sufficiently large, an error-erasure decoder can successfully decode. From (36), the valid range for u is $u \in \{2\xi, \dots, n - k\}$, with the lower bound ensuring $\epsilon \geq 0$ and the upper bound corresponding to the most favorable case, where $\epsilon = w$. In the latter case, $u \leq n - k$, which is the maximum erasure decoding capability for LRS codes. The performance of the randomized decoder depends on the choice of the PMF $\beta_{\mathbf{u}}$ used to draw the rank profile \mathbf{u} . The optimal choice of $\beta_{\mathbf{u}}$ and the analysis of the algorithm's success probability will be discussed in the following sections.

Algorithm 3 outlines the proposed approach. It can be easily generalized to variable block lengths and other sum-rank metric codes that support efficient error-and-erasure decoders.

Algorithm 3: Randomized Sum-Rank Metric Decoder for LRS codes

Input : Parameters: q, m, η, ℓ, w and u with $2\xi \leq u \leq n - k$

Received vector $\mathbf{y} \in \mathbb{F}_{q^m}^n$

LRS code \mathcal{C}_{LRS} of length $n = \ell\eta$ and dimension k

Error-erasure decoder $\text{DEC}(\cdot, \cdot, \cdot)$ for \mathcal{C}_{LRS}

Output : Vector $\mathbf{c}' \in \mathcal{C}_{\text{LRS}}$ such that $\text{wt}_{\Sigma R}(\mathbf{y} - \mathbf{c}') = w$

```

1  $\mathbf{c}' \leftarrow \emptyset$ 
2  $\mu \leftarrow \min\{m, \eta\}$ 
3 while  $\mathbf{c}' = \emptyset$  or  $\text{wt}_{\Sigma R}(\mathbf{y} - \mathbf{c}') \neq w$  do
4    $\mathbf{u} \leftarrow$  Draw rank profile for the guess space according to  $\beta_{\mathbf{u}}$ 
5    $\mathcal{U} \xleftarrow{\$} \Xi_{q,\mu}(\mathbf{u})$ 
6   if  $\eta < m$  then
7      $\mathbf{B}_C \leftarrow$  Basis of  $\mathcal{U}$ 
8      $\mathbf{c}' \leftarrow \text{DEC}(\mathbf{y}, \emptyset, \mathbf{B}_C)$  /* Error-erasure decoding with row erasures */
9   else
10     $\mathbf{a}_R \leftarrow$  Basis of  $\mathcal{U}$ 
11     $\mathbf{c}' \leftarrow \text{DEC}(\mathbf{y}, \mathbf{a}_R, \emptyset)$  /* Error-erasure decoding with column erasures */
12 return  $\mathbf{c}'$ 

```

The following lemma provides a useful result that contributes to the derivation of the average complexity of the randomized approach in Algorithm 3.

Lemma 1. For a fixed error $\mathbf{e} = [e_1, \dots, e_\ell] \in \mathbb{F}_{q^m}^n$ with rank profile $\mathbf{w} = [w_1, \dots, w_\ell] \in \mathbb{Z}_{\geq 0}^\ell$ and a given rank profile $\mathbf{u} = [u_1, \dots, u_\ell] \in \mathbb{Z}_{\geq 0}^\ell$, let \mathcal{E} and \mathcal{U} be the error space and guessed space, respectively, where \mathcal{U} is chosen uniformly from $\Xi_{q,\mu}(\mathbf{u})$. Further, let S_j denote the event that $\dim_{\Sigma}(\mathcal{E} \cap \mathcal{U}) = j$. The probability of S_j given \mathbf{e} and \mathbf{u} is then

$$\Pr[S_j | \mathbf{e}, \mathbf{u}] = \Pr[S_j | \mathbf{w}, \mathbf{u}] = \left(\bigotimes_{i=1}^{\ell} P_{q,\mu,w_i,u_i}^{\cap} \right) (j), \quad (37)$$

with

$$\left(\bigotimes_{i=1}^{\ell} P_{q,\mu,w_i,u_i}^{\cap} \right) (j) := \left(P_{q,\mu,w_1,u_1}^{\cap} \otimes \cdots \otimes P_{q,\mu,w_\ell,u_\ell}^{\cap} \right) (j),$$

being the ℓ -fold discrete convolution (denoted by \circledast) of the probability distributions P_{q,μ,w_i,u_i}^\cap evaluated at j for all $i = 1, \dots, \ell$. Here, P_{q,μ,w_i,u_i}^\cap represents the probability distribution of the intersection dimension between the error space and the guessed space in the i -th shot.

Proof. Given the error $\mathbf{e} = [e_1, \dots, e_\ell] \in \mathbb{F}_{q^m}^n$ with rank profile $\mathbf{w} = [w_1, \dots, w_\ell] \in \mathbb{Z}_{\geq 0}^\ell$ and the rank profile $\mathbf{u} = [u_1, \dots, u_\ell] \in \mathbb{Z}_{\geq 0}^\ell$ of the guessed support, let V_i be a random variable that corresponds to the dimension of the intersection of the i -th guessed space $\mathcal{U}^{(i)}$ with the i -th actual error space $\mathcal{E}^{(i)}$ for $i \in \{1, \dots, \ell\}$. By (4), we have that $P_{q,\mu,w_i,u_i}^\cap(j)$ is the probability of that event, i.e.,

$$\Pr[V_i = j | \mathbf{e}, \mathbf{u}] = P_{q,\mu,w_i,u_i}^\cap(j).$$

Note that the probability $\Pr[V_i = j | \mathbf{e}, \mathbf{u}]$ depends only on the rank weights w_i and u_i of the error and guessed support in the i -th shot, respectively, and not on the specific error vector \mathbf{e} . Thus, we can write

$$\Pr[V_i = j | \mathbf{e}, \mathbf{u}] = \Pr[V_i = j | \mathbf{w}, \mathbf{u}] = P_{q,\mu,w_i,u_i}^\cap(j).$$

Since we are interested in the probability distribution of the sum of random variables, i.e., $V = \sum_{i=1}^\ell V_i$, the resulting probability distribution is given by the ℓ -fold discrete convolution of the probability distributions of the random variables V_i for $i \in \{1, \dots, \ell\}$. Thus,

$$\Pr[V = j | \mathbf{e}, \mathbf{u}] = \Pr[V = j | \mathbf{w}, \mathbf{u}] = \left(\circledast_{i=1}^\ell P_{q,\mu,w_i,u_i}^\cap \right) (j),$$

with

$$\left(P_{q,\mu,w_1,u_1}^\cap \circledast P_{q,\mu,w_2,u_2}^\cap \right) (j) := \sum_{r=-\infty}^{\infty} P_{q,\mu,w_1,u_1}^\cap(r) P_{q,\mu,w_2,u_2}^\cap(j-r). \quad (38)$$

Finally, we have that S_j is the event that $V = j$, which proves the claim. \square

The probability of the event S_j given the rank profiles of the error and the guessed support, as stated in Lemma 1, can be further expanded using the concept of rank profiles. The following proposition expresses this probability.

Proposition 2. *Let $\mathbf{e} = [e_1, \dots, e_\ell] \in \mathbb{F}_{q^m}^n$ be a fixed error with rank profile $\mathbf{w} = [w_1, \dots, w_\ell] \in \mathbb{Z}_{\geq 0}^\ell$, and let $\mathbf{u} = [u_1, \dots, u_\ell] \in \mathbb{Z}_{\geq 0}^\ell$ be a given rank profile. Further, let $\mu = \min\{\eta, m\}$ and $\epsilon \in \{0, \dots, \ell\mu\}$. Then, we can write (37) from Lemma 1 as*

$$\Pr[S_\epsilon | \mathbf{w}, \mathbf{u}] = \sum_{\epsilon \in \mathcal{T}_{\epsilon,\ell,\mu}} \prod_{i=1}^{\ell} P_{q,\mu,w_i,u_i}^\cap(\epsilon_i). \quad (39)$$

Proof. Starting from the right-hand side of (39), we have

$$\begin{aligned}
\sum_{\epsilon \in \mathcal{T}_{\epsilon, \ell, \mu}} \prod_{i=1}^{\ell} P_{q, \mu, w_i, u_i}^{\cap}(\epsilon_i) &= \sum_{\substack{\mu \geq \epsilon_1, \dots, \epsilon_{\ell} \geq 0 \\ \epsilon_1 + \dots + \epsilon_{\ell} = \epsilon}} \prod_{i=1}^{\ell} P_{q, \mu, w_i, u_i}^{\cap}(\epsilon_i) \\
&= \left(\bigotimes_{i=1}^{\ell} P_{q, \mu, w_i, u_i}^{\cap} \right) (\epsilon) \\
&= \Pr[S_{\epsilon} | \mathbf{w}, \mathbf{u}],
\end{aligned}$$

where the first equality follows from the definition of the set of rank profiles, the second equality follows from the definition of discrete convolution as defined in (38), and the last equality follows from the recursive application of the definition of discrete convolution. This completes the proof. \square

The probability of successful decoding is given by the sum of the probabilities of the events S_{ϵ} over all feasible values of the total intersection dimension ϵ . Since these events are mutually exclusive, we define

$$\phi_{\mu}(\mathbf{u}, \mathbf{w}) := \sum_{\epsilon = \epsilon_{\min}}^{\min\{u, w\}} \Pr[S_{\epsilon} | \mathbf{u}, \mathbf{w}], \quad (40)$$

where $\phi_{\mu}(\mathbf{u}, \mathbf{w})$ denotes the probability of successful decoding. The lower bound ϵ_{\min} , defined in (36), ensures that the intersection support has enough dimensions for successful decoding. The upper bound $\min\{u, w\}$ reflects that the intersection cannot have a greater dimension than either of the supports.

Given a probability distribution $\beta_{\mathbf{u}}$ over the rank profiles \mathbf{u} , the overall probability of successful decoding for a fixed error rank profile \mathbf{w} is

$$\phi_{\mu, u}(\mathbf{w}) := \sum_{\mathbf{u} \in \mathcal{T}_{u, \ell, \mu}} \beta_{\mathbf{u}} \cdot \phi_{\mu}(\mathbf{u}, \mathbf{w}). \quad (41)$$

C. Worst-Case Complexity

When decoding beyond the unique decoding radius for LRS codes (Problem 3), multiple solutions may exist, and the proposed decoder (Algorithm 3) lacks a mechanism to identify the original codeword among them. To make the complexity analysis exact, we assume a genie-aided version of the decoder that can identify the correct solution and allows us to stop the decoder at the iteration when the original codeword is found. Consequently, only the upper bound on the expected complexity applies to the non-genie-aided decoder, similar to the bounds derived in [26] for the generic decoder.

We analyze the worst-case expected complexity over all possible rank profiles of errors with $\text{wt}_{\Sigma R}(\mathbf{e}) = w$. We derive lower and upper bounds on this complexity and provide algorithms to efficiently compute these bounds. Additionally, we present methods to optimize the guessing support distribution for this worst-case scenario.

Determining the optimal sum-rank weight $u \in \{2\xi, \dots, n - k\}$ for the guessing support to minimize the expected complexity of Algorithm 3 is not straightforward. The success probability bounds from [28] for the rank metric are convex in u , suggesting that the probability is maximized at $u = 2\xi$ or $u = n - k$. However, this does not guarantee that these values always yield the optimal expected complexity. Similarly, for the sum-rank metric, we cannot be certain, though these extreme values for u remain important points of interest. For these two specific values of u , we can express the success probability from (40) as follows. When $u = 2\xi \leq w$, we have

$$\phi_{\mu}(\mathbf{u}, \mathbf{w}) = \sum_{\epsilon=u}^u \Pr[S_{\epsilon} | \mathbf{u}, \mathbf{w}] = \prod_{i=1}^{\ell} P_{q,\mu}^{\subseteq}(u_i \subseteq w_i). \quad (42)$$

On the other hand, when $u = n - k \geq w$, we have

$$\phi_{\mu}(\mathbf{u}, \mathbf{w}) = \sum_{\epsilon=w}^w \Pr[S_{\epsilon} | \mathbf{u}, \mathbf{w}] = \prod_{i=1}^{\ell} P_{q,\mu}^{\subseteq}(w_i \subseteq u_i). \quad (43)$$

Notably, choosing either $u = 2\xi$ or $u = n - k$ simplifies the expression for the success probability, as the convolution in (40) reduces to a simple product over the blocks. This simplification is advantageous for our analysis of the decoder's performance.

When $u = n - k$, the success probability matches that of the fully generic decoding algorithm, as the expressions in (42) and (9) are identical. In this case, the bounds from [26] apply directly, with the only difference being the per-iteration complexity, which is replaced by that of the error-and-erasure decoder for LRS codes. Therefore, setting $u = n - k$ offers no improvement in success probability over the fully generic approach.

When $u = 2\xi$ and $u \leq w$, the algorithms from [26] require adjustments. We present these modifications below, using the notation

$$\phi'_{\mu}(\mathbf{u}, \mathbf{w}) := \prod_{i=1}^{\ell} P_{q,\mu}^{\subseteq}(u_i \subseteq w_i). \quad (44)$$

From (43) follows the worst-case expected number of iterations of Algorithm 3 for a given PMF of the guessing support, denoted as $\tilde{\beta}_{\mathbf{u}} := \Pr[\mathbf{u}]$ by

$$\max_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \mathbb{E}[\#\text{iterations}] = \max_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \left(\sum_{\mathbf{u} \in \mathcal{T}_{u,\ell,\mu}} \tilde{\beta}_{\mathbf{u}} \phi'_{\mu}(\mathbf{u}, \mathbf{w}) \right)^{-1}. \quad (45)$$

The problem of minimizing the worst-case expected number of iterations over all valid distributions $\tilde{\beta}_{\mathbf{u}}$ on $\mathcal{T}_{u,\ell,\mu}$ can be formulated as a linear program. While this linear program can be solved numerically using standard methods for small values of μ , ℓ , and u , the number of unknowns, i.e., $\tilde{\beta}_{\mathbf{u}} \in [0, 1]$, grows rapidly as these parameters increase. Consequently, solving the linear program directly becomes computationally prohibitive for larger problem instances.

To address this computational challenge, we adopt the approach proposed by Puchinger et al. [26], which introduces a randomized mapping $\text{ucomp}_{\mu}: \mathcal{T}_{w,\ell,\mu} \times \mathbb{Z}_{\geq 0} \rightarrow \mathcal{T}_{u,\ell,\mu}$. This mapping aims to maximize the probability $\phi'_{\mu}(\text{ucomp}_{\mu}(\mathbf{w}, u), \mathbf{w})$ for a given $\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}$ by randomly selecting an output vector from multiple possible candidates for each input, providing a more computationally tractable approach to the problem.

Rather than directly choosing a vector $\mathbf{u} \in \mathcal{T}_{u,\ell,\mu}$, we first select a vector $\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}$ at random according to a designed distribution $\gamma_{\mathbf{w}}$ on $\mathcal{T}_{w,\ell,\mu}$, and then set $\mathbf{u} \leftarrow \text{ucomp}_{\mu}(\mathbf{w}, u)$. For a fixed error e , this allows us to bound the probability as follows

$$\Pr(\mathbf{U} \subseteq \mathcal{E}_e) = \sum_{\mathbf{u} \in \mathcal{T}_{u,\ell,\mu}} \tilde{\beta}_{\mathbf{u}} \cdot \phi'_{\mu}(\mathbf{u}, \mathbf{w}_e) \geq \gamma_{\mathbf{w}_e} \cdot \phi'_{\mu}(\text{ucomp}_{\mu}(\mathbf{w}_e, u), \mathbf{w}_e).$$

Using this bound, we can minimize the following upper bound on the worst-case expected number of iterations, instead of directly minimizing (45)

$$\max_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \mathbb{E}[\#\text{iterations}] \leq \max_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} (\gamma_{\mathbf{w}} \cdot \phi'_{\mu}(\text{ucomp}_{\mu}(\mathbf{w}, u), \mathbf{w}))^{-1},$$

over all valid probability mass functions $\tilde{\beta}_{\mathbf{u}}$ on $\mathcal{T}_{u,\ell,\mu}$.

The randomized mapping ucomp_{μ} is formally defined in Appendix B-A and its correctness is proofed in Lemma 2.

We adapt the support-drawing algorithm from [26] to handle cases where the sum dimension of the guessed support is smaller than that of the error support. The modified version, shown in Algorithm 4, retains the structure of the original algorithm but is adjusted to account for this dimension difference.

We define the probability distribution $\gamma_{\mathbf{w}}$ as follows

$$\gamma_{\mathbf{w}} := \left(\phi'_{\mu}(\text{ucomp}_{\mu}(\mathbf{w}, u), \mathbf{w}) \cdot \tilde{Q}_{\ell,w,\mu} \right)^{-1} \quad \forall \mathbf{w} \in \mathcal{T}_{w,\ell,\mu}, \quad (46)$$

where $\tilde{Q}_{\ell,w,\mu}$ is defined as

$$\tilde{Q}_{\ell,w,\mu} := \sum_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \phi'_{\mu}(\text{ucomp}_{\mu}(\mathbf{w}, u), \mathbf{w})^{-1}. \quad (47)$$

Algorithm 4: DrawRandomSupport(u, w, μ)

Input : Integers $u, \mu, w \in \mathbb{Z}_{\geq 0}$ with $u \leq w$

Output : \mathcal{U} of sum dimension u

- 1 Draw $\mathbf{w} \in \mathcal{T}_{w, \ell, \mu}$ according to the distribution $\gamma_{\mathbf{w}}$ defined in (46).
 - 2 $\mathbf{u} \leftarrow \text{ucomp}_{\mu}(\mathbf{w}, u)$
 - 3 $\mathcal{U} \xleftarrow{\S} \Xi_{\mu, \zeta}(\mathbf{u})$
 - 4 **return** \mathcal{U}
-

The following proposition presents bounds on the expected number of iterations.

Proposition 3. *Let $\mathbf{e} \in \mathbb{F}_{q^m}^n$ be an error of sum-rank weight w and let u be an integer with $u \leq w$. If \mathcal{U} is a sub-support that is drawn by Algorithm 4 with input u and w , then we have*

$$|\mathcal{T}_{w, \ell, \mu}|^{-1} \tilde{Q}_{\ell, w, \mu} \leq \Pr(\mathcal{U} \subseteq \mathcal{E}_{\mathbf{e}})^{-1} \leq \tilde{Q}_{\ell, w, \mu},$$

where $\tilde{Q}_{\ell, w, \mu}$ is defined as in (47), and $\mathcal{E}_{\mathbf{e}}$ denotes the error support corresponding to the error vector \mathbf{e} .

Proof. Denote by $\gamma_{\mathbf{w}}$ the distribution of $\mathbf{w} = \text{ucomp}_{\mu}(\mathcal{E}_{\mathbf{e}})$, where \mathbf{w} is a random variable with probability mass function $\gamma_{\mathbf{w}}$ as defined in (46). By the law of total probability, we have

$$\begin{aligned} \Pr(\mathcal{U} \subseteq \mathcal{E}_{\mathbf{e}}) &= \sum_{\mathbf{w} \in \mathcal{T}_{w, \ell, \mu}} \gamma_{\mathbf{w}} \cdot \phi'_{\mu}(\text{ucomp}_{\mu}(\mathbf{w}, u), \mathbf{w}_{\mathbf{e}}) \\ &\geq \gamma_{\mathbf{w}_{\mathbf{e}}} \cdot \phi'_{\mu}(\text{ucomp}_{\mu}(\mathbf{w}_{\mathbf{e}}, u), \mathbf{w}_{\mathbf{e}}) \\ &= \tilde{Q}_{\ell, w, \mu}^{-1}, \end{aligned}$$

where the last equality follows from the definition of $\gamma_{\mathbf{w}}$ in (46). This proves the upper bound on $\Pr(\mathcal{U} \subseteq \mathcal{E}_{\mathbf{e}})^{-1}$. For the lower bound, we observe that for all $\mathbf{w} \in \mathcal{T}_{w, \ell, \mu}$

$$\phi'_{\mu}(\text{ucomp}_{\mu}(\mathbf{w}, u), \mathbf{w}_{\mathbf{e}}) \leq \phi'_{\mu}(\text{ucomp}_{\mu}(\mathbf{w}, u), \mathbf{w}),$$

which yields

$$\begin{aligned} \Pr(\mathcal{U} \subseteq \mathcal{E}_{\mathbf{e}}) &= \sum_{\mathbf{w} \in \mathcal{T}_{w, \ell, \mu}} \gamma_{\mathbf{w}} \cdot \phi'_{\mu}(\text{ucomp}_{\mu}(\mathbf{w}, u), \mathbf{w}_{\mathbf{e}}) \\ &\leq \sum_{\mathbf{w} \in \mathcal{T}_{w, \ell, \mu}} \gamma_{\mathbf{w}} \cdot \phi'_{\mu}(\text{ucomp}_{\mu}(\mathbf{w}, u), \mathbf{w}) \\ &= \tilde{Q}_{\ell, w, \mu}, \end{aligned}$$

where the last equality follows from the definitions of $\tilde{Q}_{\ell,w,\mu}$ in (47) and the design distribution γ_w in (46), which proves the claim. \square

Using Proposition 3, we can formulate the following theorem about the expected runtime of the genie-aided version of Algorithm 3.

Theorem 6. *Consider a genie-aided version of Algorithm 3 for an LRS code \mathcal{C}_{LRS} of length n , dimension k , and length partition \mathbf{n} with constant block length $n_i = \eta$ for all $i \in \{1, \dots, \ell\}$. Let $\mathbf{e} \in \mathbb{F}_{q^m}^n$ be an error of sum-rank weight $\tau < w \leq n - k$, and let $\mathbf{c} \in \mathcal{C}_{\text{LRS}}$ be a codeword. We consider the success event of the algorithm returning the originally transmitted codeword \mathbf{c} when given input $\mathbf{y} = \mathbf{e} + \mathbf{c}$ and parameter u with $u = 2\xi$.*

Each iteration of Algorithm 3 costs $W_{\text{rand}}^{(\text{iter})}$. By including also the expected number of iterations, we can bound the overall expected runtime W_{rand} of the genie-aided version of Algorithm 3 by

$$W_{\text{rand}}^{(\text{LB})} \leq W_{\text{rand}} \leq W_{\text{rand}}^{(\text{UB})},$$

where, for $\mu = \min\{\eta, m\}$, we define (see (47) for $\tilde{Q}_{\ell,w,\mu}$)

$$\begin{aligned} W_{\text{rand}}^{(\text{LB})} &:= |\mathcal{T}_{w,\ell,\mu}|^{-1} \cdot \tilde{Q}_{\ell,w,\mu}, \\ W_{\text{rand}}^{(\text{UB})} &:= W_{\text{rand}}^{(\text{iter})} \cdot \tilde{Q}_{\ell,w,\mu}. \end{aligned}$$

Proof. The bounds follow directly from Proposition 3 by multiplying the cost of a single iteration W_{iter} by the expected number of iterations:

$$\begin{aligned} W_{\text{rand}}^{(\text{LB})} &:= |\mathcal{T}_{w,\ell,\mu}|^{-1} \cdot \tilde{Q}_{\ell,w,\mu} \leq W_{\text{rand}} \\ &\leq W_{\text{iter}} \cdot \tilde{Q}_{\ell,w,\mu} =: W_{\text{rand}}^{(\text{UB})}. \end{aligned}$$

\square

Remark 5. *The complexity of one iteration $W_{\text{rand}}^{(\text{iter})}$ in Theorem 6 is determined by two main components. First, the support drawing algorithm from [26] can be easily adapted to our case, which yields a complexity of $\tilde{O}(n^3 m^2 \log_2(q))$ bit operations.*

Second, the overall complexity $W_{\text{rand}}^{(\text{iter})}$ is then the sum of the complexity of the support drawing algorithm and the complexity of the error and erasure decoder, which is of the order of $O(n^2 m^2)$ operations over \mathbb{F}_q (see (35)). Similar to the complexity analysis for the worst-case scenario in the generic decoding algorithm, we approximate $W_{\text{rand}}^{(\text{iter})}$ with the two dominating terms in each of the complexities as $W_{\text{rand}}^{(\text{iter})} \approx n^3 m^2$ when we plot or evaluate the complexities.

To evaluate the bounds from Theorem 6, we must compute $\tilde{Q}_{\ell,w,\mu}$. Direct computation using (47) is infeasible, as the number of summands $|\mathcal{T}_{w,\ell,\mu}|$ can grow super-polynomially with w , depending on ℓ and μ . In Appendix B-B, we show, following [26, Lemma 22], how to compute this efficiently in $\tilde{O}(wun^3\mu^3 \log_2(q))$ bit operations.

D. Average Complexity

We analyze the average complexity of the randomized decoding algorithm over all possible error vectors e with sum-rank weight w , similar to the analysis for the generic decoder in Section IV-B. Although Algorithm 3 operates on an LRS code with significant structure, we use random coding arguments akin to the generic decoding approach to estimate the average success probability when decoding beyond the unique decoding radius. This accounts for the additional codeword solutions that may appear in this regime. Note that since LRS codes are not random codes, applying random coding arguments only yields an approximation for the lower bound.

Adapting Theorem 4, we replace the expression for the success probability of one iteration of the decoding loop to obtain bounds for our randomized decoder.

Corollary 4. *Let \mathcal{C} be a random \mathbb{F}_{q^m} -linear code of length n and dimension k over \mathbb{F}_{q^m} , where each codeword is drawn uniformly at random from $\mathbb{F}_{q^m}^n$. Suppose the received word $\mathbf{y} = \mathbf{c} + \mathbf{e}$, where $\mathbf{c} \in \mathcal{C}$ and $\mathbf{e} \in \mathbb{F}_{q^m}^n$ with $\text{wt}_{\Sigma R}(\mathbf{e}) = w$. Assume we have an error-and-erasure decoder that can correct combinations of errors and erasures up to the condition in (34). Then, the success probability of Algorithm 3 to output at least one solution satisfies:*

$$\Pr[\text{success}] \geq \frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{u'=w}^{v_{\max}} \sum_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \phi_{q,\mu,u'}(\mathbf{w}) \prod_{i=1}^{\ell} \text{NM}_q(m, \eta, w_i),$$

and

$$\Pr[\text{success}] \leq \left(\frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} + q^{m(k-n)} \right) \sum_{u'=w}^{v_{\max}} \sum_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \phi_{q,\mu,u'}(\mathbf{w}) \prod_{i=1}^{\ell} \text{NM}_q(m, \eta, w_i).$$

Proof. The proof follows the same steps as in Theorem 4, replacing $\varphi_{q,\mu,v'}(\mathbf{w})$ with $\phi_{q,\mu,u'}(\mathbf{w})$. \square

E. Optimizing the Support-Drawing Distribution in Randomized Decoding of LRS Codes

We propose a heuristic approach to optimize the support-drawing distribution used in Algorithm 3. Inspired by the asymptotic analysis in Section IV-D, we aim at maximizing the average

intersection between the guessed support and the actual error support, thereby increasing the probability of successful decoding.

To simplify the optimization, we consider an asymptotic setting where the number of blocks ℓ tends to infinity. In this context, we approximate the rank weight distributions of the error and the guessed support by their marginal distributions for a single block, assuming they are independently and identically distributed across blocks.

Let $\beta_u^{(m)} := \Pr[\dim(\mathcal{U}^{(i)}) = u]$ denote the marginal distribution of the rank weight of the guessed support $\mathcal{U}^{(i)}$ for a single block i , where $u \in \{0, \dots, \mu\}$. The joint distribution of the guessing rank profile \mathbf{u} is then

$$\beta_{\mathbf{u}} = \prod_{i=1}^{\ell} \beta_{u_i}^{(m)}. \quad (48)$$

We introduce the random variable Z , representing the dimension of the intersection between the guessed support and the error support for a single block. Our objective is to maximize the expected value $\mathbb{E}[Z]$, as a larger average intersection increases the probability of successful decoding with an error-and-erasure decoder, which requires a sufficiently large intersection dimension (see (36)). The expectation $\mathbb{E}[Z]$ can be computed as

$$\mathbb{E}[Z] = \sum_{\epsilon=0}^{\mu} \sum_{w'=0}^{\mu} \sum_{u'=0}^{\mu} \beta_{u'}^{(m)} \cdot \Pr[w'] \cdot P_{q,\mu,w',u}^{\cap}(\epsilon),$$

where $\Pr[w']$ is the marginal distribution of the error rank weight for a single block.

The distribution $\beta^{(m)}$ can be optimized using LP methods to maximize $\mathbb{E}[Z]$, similar to the approach in Section IV-D.

The following theorem provides bounds on the expected runtime of the randomized sum-rank decoder (Algorithm 3) based on the success probability bounds derived in Corollary 4.

Theorem 7. *Under the same assumptions as in Corollary 4, the overall expected runtime $W_{\text{rand,RCU}}$ of Algorithm 3 to output at least one solution is bounded by*

$$W_{\text{rand,RCU}}^{(\text{LB})} \leq W_{\text{rand,RCU}} \leq W_{\text{rand,RCU}}^{(\text{UB})},$$

with

$$W_{\text{rand,RCU}}^{(\text{LB})} := W_{\text{ee-dec}} \left(\left(\frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} + q^{m(k-n)} \right) \sum_{u=w}^{v_{\max}} C_{q,m,\eta}(\ell, w, u) \right)^{-1},$$

and

$$W_{\text{rand,RCU}}^{(\text{UB})} := W_{\text{ee-dec}} \left(\frac{1}{|\mathcal{E}_{q,\eta,m,\ell}(w)|} \sum_{u=w}^{v_{\max}} C_{q,m,\eta}(\ell, w, u) \right)^{-1},$$

where

$$C_{q,m,\eta}(\ell, w, u) := \sum_{\epsilon=\epsilon_{\min}}^{\min\{u,w\}} \sum_{\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}} \sum_{\mathbf{u} \in \mathcal{T}_{u,\ell,\mu}} \sum_{\epsilon \in \mathcal{T}_{\epsilon,\ell,\mu}} \prod_{i=1}^{\ell} \beta_{u_i}^{(m)} \cdot \mathbf{P}_{q,\mu,w_i,u_i}^{\cap}(\epsilon_i) \cdot \text{NM}_q(m, \eta, w_i),$$

and $W_{\text{ee-dec}} \in O(n^2 m^2)$ over \mathbb{F}_q denotes the complexity of the error and erasure decoder from [29] as defined in Section VI-B.

Proof. The proof follows similar arguments as in Theorem 5. The main difference is that the bounds on the success probability are replaced by the expressions derived in Corollary 4, which involve the definitions from (39), (40), (41), and (48). The complexity of one iteration of Algorithm 3 is given by $W_{\text{ee-dec}}$, which is the complexity of the error and erasure decoder used in the randomized algorithm. \square

Remark 6. The function $C_{q,m,\eta}(\ell, w, u)$ plays a crucial role in determining the complexity bounds of the randomized sum-rank syndrome decoder. It can be computed efficiently using a dynamic programming routine similar to Algorithm 5 in polynomial time.

VII. NUMERICAL RESULTS

We compare the performance of the randomized decoding algorithm for LRS codes with the generic decoder. Figures 7 and 8 illustrate the expected complexities for both algorithms under two different parameter sets, ensuring that the total number of bits, calculated as $m \log_2(q) = 144$, remains constant.

In both figures, we set $n = 48$ and $k = 24$, resulting in a minimum sum-rank distance $d_{\min} = 25$, and a unique decoding radius $\tau = \lfloor \frac{n-k}{2} \rfloor = 12$. We consider errors with sum-rank weight $w = \tau + \xi = 13$, where the error excess is $\xi = 1$.

The results show a significant reduction in complexity for the randomized decoding algorithm compared to the generic decoder across both parameter sets. This improvement highlights the advantage of leveraging the structural properties of LRS codes. Although the relative gain decreases as the parameters approach those of the Hamming metric, the randomized decoder still maintains a lower complexity in both the worst-case and average-case settings.

VIII. CONCLUSION AND OUTLOOK

In this paper, we developed and analyzed algorithms to address the general sum-rank metric decoding problem, with a focus on both worst-case and average-case complexities. We derived

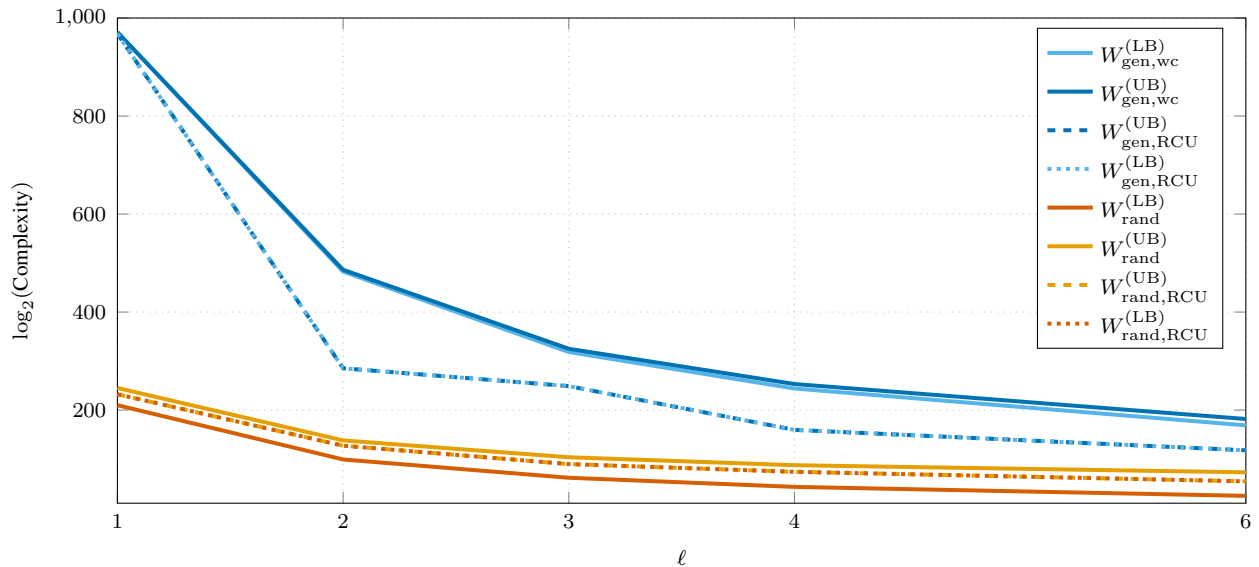


Fig. 7. Complexity comparison of generic decoding vs. randomized decoding beyond the unique decoding radius for parameters: $q = 2^3$, $m = 48$, $n = 48$, $k = 24$, $w = 13$, $u = 2$, $v = 24$.

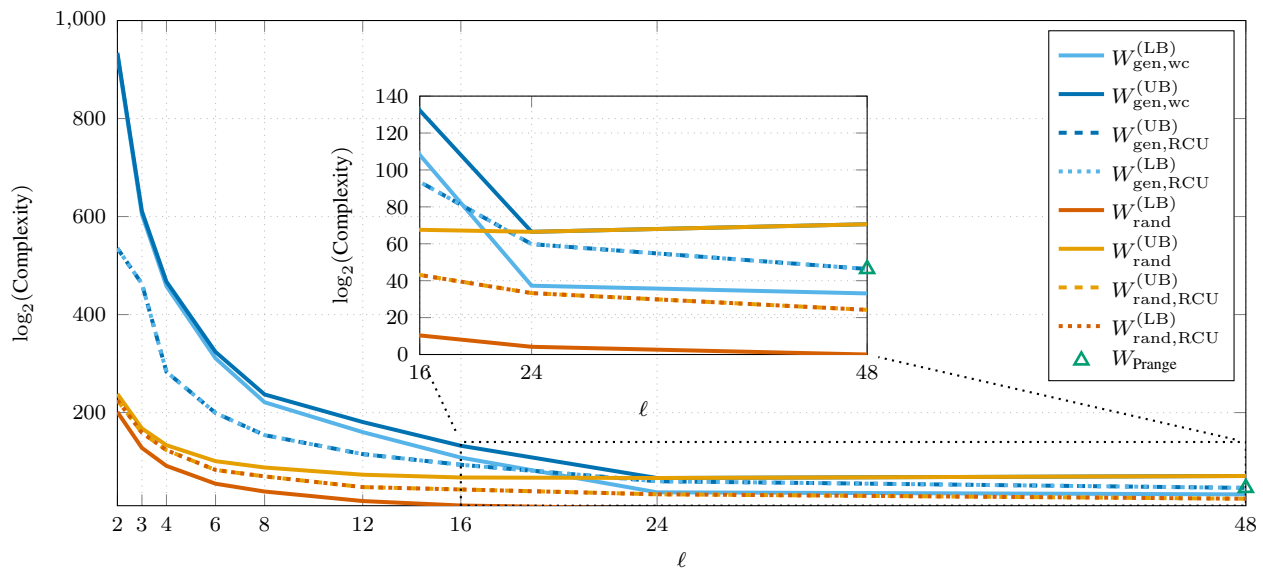


Fig. 8. Complexity comparison of generic decoding vs. randomized decoding beyond the unique decoding radius for parameters: $q = 2^6$, $m = 24$, $n = 48$, $k = 24$, $w = 13$, $u = 2$, $v = 24$.

a tighter upper bound for the results in [26] and extended their work to the average-case scenario, with a particular focus on decoding beyond the unique decoding radius. Additionally, we improved the randomized decoding algorithm for LRS codes, building on our previous work [27]. Furthermore, we introduced a Prange-like algorithm for the sum-rank metric that effectively handles larger error weights in the asymptotic setting, where $\ell \rightarrow \infty$.

Future research could adapt techniques that improved Prange’s original ISD algorithm in the Hamming metric [41]–[44] to the generic and randomized decoding algorithms in the sum-rank metric. Given the hybrid nature of the sum-rank metric, these methods may particularly benefit the Hamming-like error structure and reduce complexity, especially when ℓ is large.

Another direction is to apply improvements from generic decoding algorithms in the rank metric, such as those in [45], or extend algebraic techniques like [46] to the sum-rank metric.

These approaches could yield substantial complexity reductions for specific parameter regimes in the sum-rank metric.

A list decoding algorithm for Gabidulin codes based on Gröbner bases was introduced in [47], enabling error correction beyond the unique decoding radius. This approach can be easily adapted for LRS codes. However, as no upper bound on the list size is known, it is difficult to assess the overall complexity of the algorithm, making it challenging to compare with our approach. Establishing tighter complexity bounds for this algorithm remains an open problem and could be a promising direction for future research.

APPENDIX A

EFFICIENT COMPUTATION OF $B_{q,m,\eta}(w, v, \ell)$

The quantity $B_{q,m,\eta}(w, v, \ell)$ defined in (23) can also be computed recursively as follows

$$B_{q,m,\eta}(w, v, \ell) = \begin{cases} \alpha_v^{(m)} \text{NM}_q(m, \eta, w) \text{P}_{q,\mu}^{\subseteq}(w \subseteq v) & \text{if } \ell = 1 \\ \sum_{w'=0}^{\min\{\mu,w\}} \sum_{v'=w'}^{\min\{\mu,v\}} \alpha_{v'}^{(m)} \text{NM}_q(m, \eta, w') \text{P}_{q,\mu}^{\subseteq}(w' \subseteq v') & \text{else} \\ \cdot B_{q,m,\eta}(w - w', v - v', \ell - 1) & \end{cases} .$$

This expression can be computed in polynomial time using dynamic programming; see Algorithm 5.

APPENDIX B

APPENDIX FOR SECTION VI

A. Definition of ucomp and Proof of Correctness

The computation of $\text{ucomp}_\mu(\mathbf{w}, u)$ is described in Algorithm 6. and Lemma 2 proofs its correctness. The randomization step in Line 6 of Algorithm 6 is crucial to avoid bias towards

Algorithm 5: Compute the term $B_{q,m,\eta}(w, v, \ell)$ in the sum (24) for a given v .

Input : Parameters: q, m, n, k, ℓ, w and v with $v \geq 0$

Output : Value of $B_{q,m,\eta}(w, v, \ell)$

Initialize: $N(v', w', \ell') = 0 \quad \forall w' \in \{0, \dots, w\}, v' \in \{0, \dots, v\}, \ell' \in \{0, \dots, \ell\}$

```

1 if  $v < w$  then
2   return 0
3 for  $w' \in \{0, \dots, w\}$  do
4   for  $v' \in \{w', \dots, v\}$  do
5     if  $v' \leq \mu$  then
6        $N(v', w', 1) \leftarrow \alpha_{v'}^{(m)} \text{NM}_q(m, \eta, w') \mathbf{P}_{q,\mu}^{\subseteq}(v' \subseteq w')$ 
7 for  $\ell' \in \{2, \dots, \ell\}$  do
8   for  $w' \in \{0, \dots, w\}$  do
9     for  $v' \in \{w', \dots, v\}$  do
10      
$$N(v', w', \ell') \leftarrow \sum_{w''=0}^{\min\{\mu, w'\}} \sum_{v''=w''}^{\min\{\mu, v'\}} N(v' - v'', w' - w'', \ell' - 1) \cdot \alpha_{v''}^{(m)} \text{NM}_q(m, \eta, w'') \mathbf{P}_{q,\mu}^{\subseteq}(v'' \subseteq w'')$$

11 return  $N(v, w, \ell)$ 

```

specific positions, particularly when ℓ is large. This contrasts with a deterministic choice, which may lead to suboptimal results. In the Hamming case, where $\eta = 1$ and $n = \ell$, such randomization is essential for the effectiveness of Prange's generic decoder. However, our analysis does not explicitly take this randomness property into account and instead relies on $\phi'_\mu(\text{ucomp}_\mu(\mathbf{w}, u), \mathbf{w})$, which is not randomized, despite ucomp_μ being a randomized function.

Lemma 2. Let $\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}$ and let $u \leq w$. Then, $\mathbf{u} = \text{ucomp}_\mu(\mathbf{w}, u)$, with ucomp_μ as in Algorithm 6, maximizes $\phi'_\mu(\mathbf{u}, \mathbf{w})$, i.e.,

$$\phi'_\mu(\text{ucomp}_\mu(\mathbf{w}, u), \mathbf{w}) = \max_{\mathbf{u} \in \mathcal{T}_{u,\ell,\mu}} \phi'_\mu(\mathbf{u}, \mathbf{w}).$$

Algorithm 6: $\text{ucomp}_\mu(\mathbf{w}, u)$

Input : $\mathbf{w} \in \mathcal{T}_{w,\ell,\mu}$ and $u \in \mathbb{Z}_{\geq 0}$ with $u = 2\xi \leq w$
Output : $\mathbf{u} \in \mathcal{T}_{u,\ell,\mu}$ such that $\mathbf{u} = \arg \max_{\mathbf{u}' \in \mathcal{T}_{u,\ell,\mu}} \phi'_\mu(\mathbf{u}', \mathbf{w})$

```

1  $\mathbf{u} = [u_1, \dots, u_\ell] \leftarrow \mathbf{w}$ 
2  $\delta \leftarrow w - u$ 
3 while  $\delta > 0$  do
4    $\mathcal{J}_1 \leftarrow \{i \in \{1, \dots, n\} : u_i > 0\}$ 
5    $\mathcal{J}_2 \leftarrow \{i \in \mathcal{J}_1 : w_i = \min_{j \in \mathcal{J}_1} \{w_j\}\}$ 
6    $\mathcal{J}_3 \leftarrow \{i \in \mathcal{J}_2 : u_i = \max_{j \in \mathcal{J}_2} \{u_j\}\}$ 
7    $h \xleftarrow{\$} \mathcal{J}_3$ 
8    $u_h \leftarrow u_h - 1$ 
9    $\delta \leftarrow \delta - 1$ 
10 return  $\mathbf{u}$ 

```

Proof. By (44) we have that

$$\phi'_\mu(\mathbf{u}, \mathbf{w}) := \prod_{i=1}^{\ell} P_{q,\mu}^{\subseteq}(u_i \subseteq w_i) = \frac{\begin{bmatrix} w_i \\ u_i \end{bmatrix}_q}{\begin{bmatrix} \mu \\ u_i \end{bmatrix}_q}.$$

The factor by which this expression is increased if we decrease u_i by 1 is

$$\begin{aligned} \frac{\begin{bmatrix} w_i \\ u_i-1 \end{bmatrix}_q / \begin{bmatrix} \mu \\ u_i-1 \end{bmatrix}_q}{\begin{bmatrix} w_i \\ u_i \end{bmatrix}_q / \begin{bmatrix} \mu \\ u_i \end{bmatrix}_q} &= \prod_{j=1}^{u_i} \frac{q^{\mu-u_i+j} - 1}{q^{w_i-u_i+j} - 1} \cdot \prod_{j=1}^{u_i-1} \frac{q^{w_i-u_i+1+j} - 1}{q^{\mu-u_i+1+j} - 1} \\ &= \frac{q^\mu - 1}{q^{w_i} - 1} \cdot \underbrace{\prod_{i=j}^{u_i-1} \frac{(q^{w_i-u_i+1+j} - 1)(q^{\mu-u_i+j} - 1)}{(q^{\mu-u_i+1+j} - 1)(q^{w_i-u_i+j} - 1)}}_{= \frac{(q^{w_i-1})(q^{\mu+1}-q^{u_i})}{(q^\mu-1)(q^{w_i+1}-q^{u_i})}} \\ &= \frac{q^{\mu+1} - q^{u_i}}{q^{w_i+1} - q^{u_i}}. \end{aligned}$$

This increase factor is monotonically increasing in u_i for a fixed w_i and μ , and decreasing in w_i for a fixed u_i . Consequently, the maximum increase of (44) is obtained by decreasing the largest u_i among the smallest w_i . By adopting a greedy approach and incrementally adjusting such positions, a global maximum can be reached as this strategy ensures optimal increase

in subsequent steps. Thus, (44) is optimized by incrementally decreasing u_i by one while maintaining $u_i \geq 0$ and ensuring $\sum_{i=1}^{\ell} u_i \geq u$. This method aligns with the operations performed by $\text{ucomp}_{\mu}(\mathbf{w}, u)$ as described in Algorithm 6. \square

B. Efficient Computation of $\tilde{Q}_{\ell, w, \mu}$

Fortunately, we can employ a similar approach to [26, Lemma 22] and compute $\tilde{Q}_{\ell, w, \mu}$ as

$$\tilde{Q}_{\ell, w, \mu} = \ell! \cdot M(w, \ell, \mu, u),$$

where $M(w, \ell, \mu, u)$ can be computed using Algorithm 7. To do so, we first initialize a global table $\{M(w', \ell', \mu', u')\}_{w' \leq w, \ell' \leq \ell}^{\mu' \leq \mu, u' \leq u}$ with $M(w', \ell', \mu', u') = -1$ for all entries. Then, we call Algorithm 7 with input parameters w, ℓ, μ , and u .

By applying arguments similar to those in [26, Proposition 23], we can show that the complexity of computing $\tilde{Q}_{\ell, w, \mu}$ using this approach is $\tilde{O}(wun^3\mu^3 \log_2(q))$ and thus polynomially bounded.

Algorithm 7: Fill Table $\{M(w', \ell', \mu', u')\}_{w' \leq w, \ell' \leq \ell}^{\mu' \leq \mu, u' \leq u}$

Input : Integers $w' \leq w, \ell' \leq \ell, \mu' \leq \mu, u' \leq u$

Global table $\{M(w', \ell', \mu', u')\}_{w' \leq w, \ell' \leq \ell}^{\mu' \leq \mu, u' \leq u}$

Global parameters q and μ

Output : $M(w', \ell', \mu', u')$

```

1 if  $M(w', \ell', \mu', u') = -1$  then
2   if  $\ell' = w' = u' = 0$  then
3      $x \leftarrow 1$ 
4   else
5     if  $\ell' \geq 1$  and  $0 \leq w' \leq \ell' \mu'$  and  $0 \leq u' \leq \min\{\ell' \mu, w'\}$  then
6        $x \leftarrow 0$ 
7       for  $w_1 \in \{\mu, \dots, \lfloor w/\ell \rfloor\}$  do
8          $\delta_{\min} \leftarrow \max\{1, \ell'(w_1 + 1) - w\}$ 
9          $\delta_{\max} \leftarrow \max\{i \in \mathbb{Z}_{\geq 0} : 1 \leq i \leq \ell' + 1, w_1 i \leq w\}$ 
10        for  $\delta \in \{\delta_{\min}, \dots, \delta_{\max}\}$  do
11           $u_1 \leftarrow \max\{u' - (w' - \delta w_1), 0\}$   $\mathbf{u}^{(1)} \leftarrow \text{ucomp}_{\mu}([w_1, \dots, w_1], u_1)$ 
12           $\rho \leftarrow \frac{1}{\delta!} \cdot \prod_{i=1}^{\delta} \left( \left[ \begin{smallmatrix} \mu \\ u_i^{(1)} \end{smallmatrix} \right]_q \cdot \left[ \begin{smallmatrix} w_1 \\ u_i^{(1)} \end{smallmatrix} \right]_q^{-1} \right)$ 
13           $x \leftarrow x + \rho \cdot M(w' - \delta w_1, \ell' - \delta, w_1 + 1, u' - u_1)$ 
14        else
15           $x \leftarrow 0$ 
16 return  $M(w', \ell', \mu', u')$ 

```

REFERENCES

- [1] H. F. Lu and P. V. Kumar, "A Unified Construction of Space-Time Codes with Optimal Rate-Diversity Tradeoff," *IEEE Transactions on Information Theory*, vol. 51, no. 5, 2005.
- [2] E. M. Gabidulin, "Theory of Codes with Maximum Rank Distance," *Probl. Inf. Transm.*, vol. 21, no. 1, pp. 3–16, 1985.
- [3] A. Wachter, V. Sidorenko, M. Bossert, and V. V. Zyablov, "On (Partial) Unit Memory Codes Based on Gabidulin Codes," *Problems of Information Transmission*, vol. 47, no. 2, pp. 117–129, Jun. 2011. [Online]. Available: <https://doi.org/10.1134/S0032946011020049>
- [4] A. Wachter-Zeh and V. Sidorenko, "Rank Metric Convolutional Codes for Random Linear Network Coding," in *2012*

- International Symposium on Network Coding (NetCod)*. Cambridge, MA, USA: IEEE, Jun. 2012, pp. 1–6. [Online]. Available: <http://ieeexplore.ieee.org/document/6261875/>
- [5] A. Wachter-Zeh, M. Stinner, and V. Sidorenko, “Convolutional Codes in Rank Metric With Application to Random Network Coding,” *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3199–3213, Jun. 2015. [Online]. Available: <http://ieeexplore.ieee.org/document/7090997/>
- [6] D. Napp, R. Pinto, J. Rosenthal, and P. Vettori, “MRD Rank Metric Convolutional Codes,” in *2017 IEEE International Symposium on Information Theory (ISIT)*. Aachen, Germany: IEEE, Jun. 2017, pp. 2766–2770. [Online]. Available: <http://ieeexplore.ieee.org/document/8007033/>
- [7] U. Martínez-Peñas, “Skew and Linearized Reed–Solomon Codes and Maximum Sum Rank Distance Codes Over Any Division Ring,” *Journal of Algebra*, vol. 504, pp. 587–612, 2018, publisher: Elsevier.
- [8] U. Martínez-Peñas and F. R. Kschischang, “Reliable and Secure Multishot Network Coding Using Linearized Reed–Solomon Codes,” *IEEE Transactions on Information Theory*, vol. 65, no. 8, pp. 4785–4803, 2019, publisher: IEEE.
- [9] H. Bartz, T. Jerkovits, and J. Rosenkilde, “Fast Kötter–Nielsen–Høholdt Interpolation over Skew Polynomial Rings and its Application in Coding Theory,” *Designs, Codes and Cryptography*, vol. 92, no. 2, pp. 435–465, Feb. 2024. [Online]. Available: <https://link.springer.com/10.1007/s10623-023-01315-4>
- [10] H. Bartz and S. Puchinger, “Decoding of Interleaved Linearized Reed–Solomon Codes with Applications to Network Coding,” in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 160–165.
- [11] U. Martínez-Peñas and F. R. Kschischang, “Universal and Dynamic Locally Repairable Codes With Maximal Recoverability via Sum-Rank Codes,” *IEEE Transactions on Information Theory*, vol. 65, no. 12, pp. 7790–7805, 2019.
- [12] R. W. Nóbrega and B. F. Uchôa-Filho, “Multishot Codes for Network Coding Using Rank-Metric Codes,” in *2010 Third IEEE International Workshop on Wireless Network Coding*. IEEE, 2010, pp. 1–6.
- [13] P. Gaborit, O. Ruatta, and J. Schrek, “On the Complexity of the Rank Syndrome Decoding Problem,” *IEEE Trans. Inform. Theory*, vol. 62, no. 2, pp. 1006–1019, 2016.
- [14] J. K. Gibson, “Severely Denting the Gabidulin Version of the McEliece Public Key Cryptosystem,” *Designs, Codes and Cryptography*, vol. 6, no. 1, pp. 37–45, Jul. 1995. [Online]. Available: <http://link.springer.com/10.1007/BF01390769>
- [15] K. Gibson, “The Security of the Gabidulin Public Key Cryptosystem,” in *Advances in Cryptology — EUROCRYPT ’96*, G. Goos, J. Hartmanis, J. Van Leeuwen, and U. Maurer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, vol. 1070, pp. 212–223, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/3-540-68339-9_19
- [16] R. Overbeck, “Extending Gibson’s Attacks on the GPT Cryptosystem,” in *Coding and Cryptography*, O. Ytrehus, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, vol. 3969, pp. 178–188, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/11779360_15
- [17] H. Rashwan, E. M. Gabidulin, and B. Honary, “Security of the GPT Cryptosystem and Its Applications to Cryptography,” *Security and Communication Networks*, vol. 4, no. 8, pp. 937–946, Aug. 2011. [Online]. Available: <https://onlinelibrary.wiley.com/doi/10.1002/sec.228>
- [18] A. Wachter-Zeh, S. Puchinger, and J. Renner, “Repairing the Faure-Loidreau Public-Key Cryptosystem,” in *IEEE International Symposium on Information Theory (ISIT)*, Jun. 2018, pp. 2426–2430, place: Vail, Colorado, USA.
- [19] F. Hörmann, H. Bartz, and A.-L. Horlemann, “Distinguishing and Recovering Generalized Linearized Reed–Solomon Codes,” in *Code-Based Cryptography*, J.-C. Deneuville, Ed. Cham: Springer Nature Switzerland, 2023, vol. 13839, pp. 1–20, series Title: Lecture Notes in Computer Science. [Online]. Available: https://link.springer.com/10.1007/978-3-031-29689-5_1

- [20] C. Faure and P. Loidreau, “A New Public-Key Cryptosystem Based on the Problem of Reconstructing p -Polynomials,” in *Coding and Cryptography*. Springer, 2006, pp. 304–315.
- [21] J. Renner, S. Puchinger, and A. Wachter-Zeh, “LIGA: A Cryptosystem Based on the Hardness of Rank-Metric List and Interleaved Decoding,” *Designs, Codes, and Cryptography*, vol. 89, pp. 1279–1319, 2021.
- [22] T. Debris-Alazard, N. Sendrier, and J.-P. Tillich, “Wave: A New Family of Trapdoor One-Way Preimage Sampleable Functions Based on Codes,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11921 LNCS, 2019, iSSN: 16113349.
- [23] N. Aragon, P. Barreto, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Gueron, T. Güneysu, C. A. Melchor, R. Misoczki, E. Persichetti, N. Sendrier, J.-P. Tillich, V. Vasseur, and G. Zémor, “BIKE: Bit Flipping Key Encapsulation,” 2020, submission to NIST Post-Quantum Cryptography Standardization Process, Round 3.
- [24] C. A. Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, E. Persichetti, and G. Zémor, “Hamming Quasi-Cyclic (HQC),” 2020, submission to NIST Post-Quantum Cryptography Standardization Process, Round 3.
- [25] C. A. Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, M. Bros, A. Couvreur, J.-C. Deneuville, P. Gaborit, A. Hauteville, and G. Zémor, “Rank Quasi-Cyclic (RQC),” 2020. [Online]. Available: https://pqc-rqc.org/doc/rqc-specification_2020-04-21.pdf
- [26] S. Puchinger, J. Renner, and J. Rosenkilde, “Generic Decoding in the Sum-Rank Metric,” *IEEE Transactions on Information Theory*, vol. 68, no. 8, 2022.
- [27] T. Jerkovits, H. Bartz, and A. Wachter-Zeh, “Randomized Decoding of Linearized Reed–Solomon Codes Beyond the Unique Decoding Radius,” in *2023 IEEE International Symposium on Information Theory (ISIT)*, 2023, pp. 820–825.
- [28] J. Renner, T. Jerkovits, H. Bartz, S. Puchinger, P. Loidreau, and A. Wachter-Zeh, “Randomized Decoding of Gabidulin Codes Beyond the Unique Decoding Radius,” in *Post-Quantum Cryptography*, J. Ding and J.-P. Tillich, Eds. Cham: Springer International Publishing, 2020, pp. 3–19.
- [29] F. Hörmann, H. Bartz, and S. Puchinger, “Error-Erasures Decoding of Linearized Reed–Solomon Codes in the Sum-Rank Metric,” in *2022 IEEE International Symposium on Information Theory (ISIT)*, 2022, pp. 7–12.
- [30] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. North Holland Publishing Co., 1977, published: Hardcover.
- [31] T. Migler, K. E. Morrison, and M. Ogle, “Weight and Rank of Matrices Over Finite Fields,” 2004, arXiv: math/0403314 [math.RA]. [Online]. Available: <https://arxiv.org/abs/math/0403314>
- [32] R. Kötter and F. R. Kschischang, “Coding for Errors and Erasures in Random Network Coding,” *IEEE Transactions on Information theory*, vol. 54, no. 8, pp. 3579–3591, 2008, publisher: IEEE.
- [33] E. Byrne, H. Gluesing-Luerssen, and A. Ravagnani, “Fundamental Properties of Sum-Rank-Metric Codes,” *IEEE Transactions on Information Theory*, vol. 67, no. 10, 2021.
- [34] S. Puchinger, J. Renner, and J. Rosenkilde, “Generic Decoding in the Sum-Rank Metric,” in *IEEE International Symposium on Information Theory - Proceedings*, vol. 2020-June, 2020, iSSN: 21578095.
- [35] T. Jerkovits, F. Hörmann, and H. Bartz, “An Error-Code Perspective on Metzner–Kapturowski-Like Decoders,” 2024, submitted to IEEE Transactions on Information Theory Version Number: 1. [Online]. Available: <https://arxiv.org/abs/2409.18488>
- [36] H. S. Couvée, T. Jerkovits, and J. Bariffi, “Bounds on Sphere Sizes in the Sum-Rank Metric and Coordinate-Additive Metrics,” Apr. 2024, arXiv:2404.10666 [cs, math]. [Online]. Available: <http://arxiv.org/abs/2404.10666>
- [37] E. Prange, “The Use of Information Sets in Decoding Cyclic Codes,” *IEEE Transactions on Information Theory*, vol. 8, no. 5, pp. 5–9, Sep. 1962. [Online]. Available: <http://ieeexplore.ieee.org/document/1057777/>

- [38] S. Puchinger and J. Rosenkilde, “Bounds on List Decoding of Linearized Reed–Solomon Codes,” in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 154–159.
- [39] G. Forney, “Generalized Minimum Distance Decoding,” *IEEE Transactions on Information Theory*, vol. 12, no. 2, pp. 125–131, Apr. 1966. [Online]. Available: <http://ieeexplore.ieee.org/document/1053873/>
- [40] M. Bossert, E. Costa, E. M. Gabidulin, E. Schulz, and M. Weckerle, “Verfahren und Kommunikationsvorrichtung zum Dekodieren von mit einem Rang-Code codierten Daten,” EU Patent EP20040104458, 2003.
- [41] J. Stern, “A Method for Finding Codewords of Small Weight,” in *Coding Theory and Applications*, G. Cohen and J. Wolfmann, Eds. Berlin/Heidelberg: Springer-Verlag, 1989, vol. 388, pp. 106–113, series Title: Lecture Notes in Computer Science. [Online]. Available: <http://link.springer.com/10.1007/BFb0019850>
- [42] D. J. Bernstein, T. Lange, and C. Peters, “Smaller Decoding Exponents: Ball-Collision Decoding,” in *Advances in Cryptology – CRYPTO 2011*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, and P. Rogaway, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, vol. 6841, pp. 743–760, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/978-3-642-22792-9_42
- [43] A. May, A. Meurer, and E. Thomae, “Decoding Random Linear Codes in $O(2^{0.054n})$,” in *Advances in Cryptology – ASIACRYPT 2011*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, D. H. Lee, and X. Wang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, vol. 7073, pp. 107–124, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/978-3-642-25385-0_6
- [44] A. Becker, A. Joux, A. May, and A. Meurer, “Decoding Random Binary Linear Codes in $2^{(n/20)}$: How $1 + 1 = 0$ Improves Information Set Decoding,” in *Advances in Cryptology – EUROCRYPT 2012*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, D. Pointcheval, and T. Johansson, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, vol. 7237, pp. 520–536, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/978-3-642-29011-4_31
- [45] N. Aragon, P. Gaborit, A. Hauteville, and J.-P. Tillich, “Improvement of Generic Attacks on the Rank Syndrome Decoding Problem,” Oct. 2017. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01618464>
- [46] M. Bardet, M. Bros, D. Cabarcas, P. Gaborit, R. Perlner, D. Smith-Tone, J.-P. Tillich, and J. Verbel, “Improvements of Algebraic Attacks for Solving the Rank Decoding and MinRank Problems,” in *Advances in cryptology – ASIACRYPT 2020*, S. Moriai and H. Wang, Eds. Cham: Springer International Publishing, 2020, pp. 507–536.
- [47] A.-L. Horlemann and M. Kuijper, “A Module Minimization Approach to Gabidulin Decoding Via Interpolation,” *Journal of Algebra Combinatorics Discrete Structures and Applications*, vol. 5, pp. 29–43, Dec. 2017.