Low-Complexity Algebraic Soft Decoding of Hermitian Codes With Re-Encoding Transform

Jiwei Liang, *Student Member, IEEE*, and Li Chen, *Senior Member, IEEE*

***Abstract*—Algebraic codes are important in communication and storage systems where soft information is limited and decoding efficiency is critical. Among them, algebraic-geometric (AG) codes are promising candidates to replace the widely used Reed-Solomon (RS) codes. However, their more sophisticated algebraic structure results in high decoding complexity, hindering their practical application. This paper proposes a novel interpolation-based algebraic soft decoding (ASD) for one of the most important AG codes - Hermitian codes. Given a designed maximum decoding output list size (OLS), the interpolation module is constructed, and the decoding is formulated as finding a desired Gröbner basis of this module that contains the interpolation polynomial. The re-encoding transform (ReT) specifically designed for ASD is proposed, enabling the modified basis reduction (BR) interpolation to achieve lower complexity. The ReT is formulated by defining Lagrange interpolation polynomials over the Hermitian function fields and appropriately choosing the re-encoding points. By exploiting the linear code property, the ReT transforms a subset of interpolation points to have zero z-coordinates, which allows a common factor to be extracted from the module basis polynomials, thereby simplifying the interpolation. This paper further proposes an improved ReT employing more re-encoding points. It enables the common factor to have a greater degree, yielding a more significant complexity reduction. This also leads to an early termination for the decoding. Theoretical analysis and numerical results demonstrate that the two proposed ReT schemes efficiently facilitate the ASD for Hermitian codes.**

***Index Terms*—Algebraic-geometric codes, algebraic soft decoding, basis reduction, Hermitian codes, re-encoding transform.**

I. Introduction

ALGEBRAIC-GEOMETRIC (AG) codes are linear block codes constructed based on algebraic curves [1]. AG codes comprise a large family, including the widely used Reed-Solomon (RS) codes, elliptic codes and Hermitian codes. Among them, RS codes can be considered as a special case of AG codes since they are constructed from a straight line. However, the length of an RS code cannot exceed the size of finite field in which it is defined, limiting its error-correction capability. Compared with RS codes, general AG codes have greater codeword lengths, thereby offering greater error-correction capabilities. It should be pointed out that despite modern codes that employ probabilistic decoding can approach or even achieve channel capacity, algebraic codes remain beneficial for applications where soft information is limited, or even unavailable. In such settings, where probabilistic decoding is not applicable, conventional code features including the code's minimum Hamming distance remain crucial for system performance.

Similar to RS codes, AG codes can be decoded using the syndrome-based decoding algorithms, which correct errors by determining the error locations and magnitudes. The Berlekamp-Massey (BM) algorithm [2] and the Sakata algorithm [3] are well known syndrome-based decoding algorithms for RS codes and Hermitian codes, respectively. They exhibit a complexity of $O(n^2)$ and $O(n^{7/3})$[1], respectively, where n is the codeword length. But they cannot correct errors beyond half of the code's minimum Hamming distance[2]. The Guruswami-Sudan (GS) algorithm can correct errors beyond this limit by formulating the decoding as a curve-fitting problem [4]. The GS decoding consists of interpolation and root-finding, where the former dominates the complexity. The interpolation can be realized by Kötter's interpolation [5] which constructs the interpolation polynomial in an iterative manner. It can also be realized by the basis reduction (BR) approach that treats the decoding as computing a desired Gröbner basis that contains the interpolation polynomial. The Gröbner basis can be obtained by reducing an interpolation module basis. The BR interpolation-based GS decoding was first proposed for RS codes [6], [7]. It can be generalized to decode other AG codes, such as Hermitian codes [8] and elliptic codes [9], [10]. Based on an efficient weighted row reduction technique, an efficient realization of BR interpolation-based GS decoding for Hermitian codes was presented in [11], yielding a complexity that is sub-quadratic in n. This approach was later generalized and improved for general AG codes [12], [13]. Although GS decoding can yield a better error-correction capability, its interpolation remains computationally expensive. Addressing

Received 1 August 2025; revised 28 January 2026; accepte 19 March 2026. Date of publication 27 March 2026; date of current version 21 May 2026. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62471503 and in part by the Natural Science Foundation of Guangdong Province under Grant 2024A1515010213. An earlier version of this paper was presented in part at the 2025 IEEE International Symposium on Information Theory (ISIT), Ann Arbor, USA [DOI: 10.1109/ISIT63088.2025.11195257]. *(Corresponding author: Li Chen.)*

Jiwei Liang is with the School of System Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China (e-mail: liangjw59@mail2.sysu.edu.cn).

Li Chen is with the School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China, and also with Guangdong Provincial Key Laboratory of Information Security Technology, Guangzhou 510006, China (e-mail: chenli55@mail.sysu.edu.cn).

Communicated by K. R. Duffy, Associate Editor for Coding and Decoding.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIT.2026.3678324.

Digital Object Identifier 10.1109/TIT.2026.3678324

[1]In this paper, the "big-O" notation ignores the logarithmic factors.

[2]For AG codes, this is often called the designed distance which is the lower bound of their minimum Hamming distance.

this problem, the re-encoding transform (ReT) was proposed. It simplifies the interpolation by transforming the received word into one with more zero symbols. It was first proposed to facilitate the decoding of RS codes [14]. Recently, it was generalized to decode general AG codes [15], reducing the interpolation complexity from $O(l^4n(n-k))$ to $O(l^4(n-k)^2)$, where k and l are dimension of the code and the designed maximum decoding output list size (OLS), respectively.

The error-correction capability of GS decoding can be further improved by utilizing soft information. By formulating test-vectors and exploiting their similarities, the low-complexity Chase (LCC) decoding was applied to decode RS codes [16] and elliptic codes [17], [18]. The LCC decoding of Hermitian codes was first proposed in [19] and was recently improved using the ReT-assisted BR interpolation and the fast root-finding [20]. The algebraic soft decoding (ASD) is another soft-decision decoding approach that enhances the decoding performance by transforming soft information into interpolation multiplicities. It achieves a more flexible multiplicity assignment, leading to a more effective utilization of the soft information. It was first proposed for RS codes in [21]. Later, the ASD of Hermitian codes through Kötter's interpolation and the BR interpolation were proposed in [22] and [23], respectively. Following these developments, the BR interpolation-based ASD of elliptic codes was proposed in [24], where the BR interpolation is simplified by the ReT. However, due to the more complex algebraic structure, the ReT was not yet developed for the ASD of Hermitian codes. Addressing this challenge, the ReT-based ASD of Hermitian codes was recently proposed by the authors in [25]. But due to space limitations, major theorems, lemmas, and their proofs cannot be provided.

Major contributions of this paper are summarized as follows:

1) A novel ReT approach for the ASD of Hermitian codes is proposed. By defining Lagrange interpolation polynomials over the Hermitian function fields and appropriately choosing the re-encoding points, the ReT transforms a subset of the interpolation points, setting their z-coordinates to zero. Consequently, the module basis polynomials have a common factor which can be further extracted for simplifying the BR interpolation.
2) An improved ReT is further proposed. It iteratively generates a valid re-encoding codeword using erasure decoding, aiming to produce a common factor of higher degree for the module basis polynomials. As the channel condition improves, this improved variant results in a more significant complexity reduction. The re-encoded codeword can also be used to achieve an early decoding termination.
3) The modified BR interpolation based on both the ReT and its improved variant is proposed. Extracting the common factor of module basis polynomials yields a module basis isomorphism. The isomorphic module basis polynomials are constructed and reduced into a Gröbner basis. The interpolation polynomial can then be restored from the minimum candidate of the Gröbner basis. Both the ReT and its improved variant significantly reduce the basis reduction complexity.
4) A comprehensive complexity analysis for the proposed algorithms is presented. It is shown that both of the two proposed ReT facilitated ASD exhibit a remarkably lower complexity than ASD without ReT. The ASD performance of Hermitian codes is also presented. Compared with RS codes defined in the same finite field and with a similar code rate, Hermitian codes offer significant error-correction performance advantages. The performance advantage of ASD over the GS decoding for Hermitian codes is also demonstrated.

The rest of this paper is organized as follows. Section II provides background knowledge for Hermitian codes and GS decoding. To better contextualize our contribution, Section III introduces the BR interpolation-based ASD for Hermitian codes and explicitly identifies its high complexity as an important challenge. To address this, Section IV details the mathematical derivation of the ReT and its algorithmic formulation. An improved variant that employs more re-encoding points is also proposed, leading to more significant complexity reduction under better channel conditions. Based on the proposed ReT schemes, Section V describes the modified BR interpolation. Numerical results of decoding complexity and performance are demonstrated in Section VI. Finally, Section VII concludes the paper.

II. Background Knowledge

This section provides the background knowledge of this work. It begins with a review of Hermitian codes, followed by an overview of GS decoding. This establishes the technical foundation for the ASD and the subsequent improvements proposed in this paper.

A. Hermitian Codes

Let $\mathbb{F}_q = \{\sigma_0, \sigma_1, \ldots, \sigma_{q-1}\}$ denote the finite field of size q, where q is a square as required for Hermitian codes. Let $\mathbb{F}_q[X, Y]$ denote the bivariate polynomial ring defined over $\mathbb{F}_q$. An affine Hermitian curve defined over $\mathbb{F}_q$ can be written as [26]

$$H_w = Y^w + Y - X^{w+1}, \tag{1}$$

where $w = \sqrt{q}$ and the curve has a genus $g = w(w-1)/2$. There are w^3 affine points $P_j = (x_j, y_j)$ that satisfy $H_w(x_j, y_j) = 0$, and a point at infinity P_∞. Note that the w^3 affine points can be grouped into w^2 classes, each class having a distinct x-coordinate. They form the set of $\mathbb{F}_q$-rational points on H_w. Let $\mathcal{P} = \{P_j = (x_j, y_j) \mid H_w(x_j, y_j) = 0\}$ denote the set of affine points and $|\mathcal{P}| = w^3$. The coordinate ring of H_w is

$$\mathcal{R} = \mathbb{F}_q[X, Y]/ < Y^w + Y - X^{w+1} > . \tag{2}$$

Let x and y denote the residue classes of X and Y, respectively. The pole basis $\mathcal{L}_w$ comprises a set of bivariate monomials $\phi_a(x, y) = x^{i_x}y^{i_y}$. For a nonzero polynomial $S \in \mathcal{R}$, its order at a rational point P is denoted as $\nu_P(S)$. Let $\nu_{P_\infty}(\phi_a^{-1})$ denote the pole order at P_∞ and $\nu_{P_\infty}(\phi_a^{-1}) = \nu_{P_\infty}((x^{i_x}y^{i_y})^{-1}) = w \cdot i_x + (w+1) \cdot i_y$, where $0 \le i_x \le w$ and $i_y \ge 0$. Monomials of the pole basis exhibit an increasing pole order as $\mathcal{L}_w = \{\phi_a | \nu_{P_\infty}(\phi_a^{-1}) < \nu_{P_\infty}(\phi_{a+1}^{-1})\}$. For each P_j, there also exists a zero basis comprising polynomials with an increasing zero order (or multiplicity) at the point. The zero basis polynomials are [27]

$$\psi_{P_j,v}(x, y) = (x - x_j)^\lambda[(y - y_j) - x_j^w(x - x_j)]^\delta, \tag{3}$$

where $(\lambda, \delta) \in \mathbb{N}$ and $v = \lambda + (w+1)\delta$.

Definition 1: Let $\mu = k + g - 1$ and $\mathcal{L}(\mu P_\infty)$ denote the Riemann-Roch space defined by μ and P_∞. For an (n, k) Hermitian code of length n and dimension k, given a message $\underline{f} = (f_0, f_1, \ldots, f_{k-1}) \in \mathbb{F}_q^k$ where $f_0, f_1, \ldots, f_{k-1}$ are the message symbols, its corresponding message polynomial $f(x, y) \in \mathcal{L}(\mu P_\infty)$ is defined as

$$f(x, y) = f_0\phi_0 + f_1\phi_1 + \cdots + f_{k-1}\phi_{k-1}. \tag{4}$$

The codeword $\underline{c} \in \mathbb{F}_q^n$ is generated by the following evaluation

$$\begin{aligned}\underline{c} &= (c_0, c_1, \ldots, c_{n-1}) \\ &= (f(P_0), f(P_1), \ldots, f(P_{n-1})),\end{aligned} \tag{5}$$

where $\{P_0, P_1, \ldots, P_{n-1}\} \subseteq \mathcal{P}$. Let $[a] = \{0, 1, \ldots, a-1\}$, where $a \in \mathbb{Z}$ and $a \geq 0$. Hence, the index set of a codeword is $[n]$.

For an (n, k) Hermitian code, Riemann-Roch theorem [28] provides the relationship between μ and $\upsilon_{P_\infty}(\phi_{k-1}^{-1})$ as

$$\upsilon_{P_\infty}(\phi_{k-1}^{-1}) \leq \mu. \tag{6}$$

Example 1: Consider the Hermitian curve $H_2 = Y^2 + Y - X^3$ defined over $\mathbb{F}_4 = \{0, 1, \alpha, \alpha^2\}$, where α is the primitive element of this field with the primitive polynomial of $x^2 + x + 1$. There are eight affine points on H_2: $P_0 = (0, 0)$, $P_1 = (0, 1)$, $P_2 = (1, \alpha)$, $P_3 = (1, \alpha^2)$, $P_4 = (\alpha, \alpha)$, $P_5 = (\alpha, \alpha^2)$, $P_6 = (\alpha^2, \alpha)$, $P_7 = (\alpha^2, \alpha^2)$. Let $\mu = 5$. Riemann-Roch space $\mathcal{L}(5P_\infty)$ is spanned by the basis $\{1, x, y, x^2, xy\}$. Given the message $\underline{f} = (\alpha, 1, \alpha^2, 1, 0)$, its corresponding message polynomial is

$$f(x, y) = \alpha + x + \alpha^2 y + x^2.$$

It is encoded as

$$\begin{aligned}\underline{c} &= (c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7) \\ &= (f(P_0), f(P_1), f(P_2), f(P_3), f(P_4), f(P_5), f(P_6), f(P_7)) \\ &= (\alpha, 1, \alpha^2, 0, \alpha, 1, \alpha, 1).\end{aligned}$$

B. The GS Decoding

For GS decoding of an (n, k) Hermitian code, the following definition is needed.

Definition 2: : Let $\mathcal{R}[z]$ denote the polynomial ring defined over $\mathcal{R}$. Monomials $\phi_a z^b \in \mathcal{R}[z]$ are ordered according to their $(1, w_z)$-weighted degrees as

$$\deg_{1,w_z} \phi_a z^b = \nu_{P_\infty}(\phi_a^{-1}) + w_z b, \tag{7}$$

where $w_z = \nu_{P_\infty}(\phi_{k-1}^{-1})$. The $(1, w_z)$-reverse lexicographic (revlex) order can be established as follows. Given two monomials $\phi_{a_1} z^{b_1}$ and $\phi_{a_2} z^{b_2}$, $\mathrm{ord}(\phi_{a_1} z^{b_1}) < \mathrm{ord}(\phi_{a_2} z^{b_2})$, if $\deg_{1,w_z} \phi_{a_1} z^{b_1} < \deg_{1,w_z} \phi_{a_2} z^{b_2}$, or $\deg_{1,w_z} \phi_{a_1} z^{b_1} = \deg_{1,w_z} \phi_{a_2} z^{b_2}$ and $b_1 < b_2$. Given a polynomial $Q(x, y, z) = \sum_{a,b \in \mathbb{N}} Q_{ab}\ \phi_a(x, y)\ z^b$, the $(1, w_z)$-weighted degree of Q is $\deg_{1,w_z} Q = \max\{\deg_{1,w_z} \phi_a z^b \mid Q_{ab} \neq 0\}$ and its leading order is $\mathrm{lod}(Q) = \max\{\mathrm{ord}(\phi_a z^b) \mid Q_{ab} \neq 0\}$.

In decoding an (n, k) Hermitian code, polynomials are organized under the $(1, w_z)$-revlex order. Given two polynomials Q_1 and Q_2, we claim $Q_1 < Q_2$, if $\mathrm{lod}(Q_1) < \mathrm{lod}(Q_2)$.

The interpolation constraint for a polynomial Q in $\mathcal{R}[z]$ is explained as follows. Let $\underline{\omega} = (\omega_0, \omega_1, \ldots, \omega_{n-1}) \in \mathbb{F}_q^n$ denote the received word. Given an interpolation point (P_j, ω_j), if a polynomial Q can be written as

$$Q = \sum_{\alpha,\beta \in \mathbb{N}} Q_{\alpha,\beta}^{(P_j,\omega_j)} \psi_{P_j,\alpha}(z - \omega_j)^\beta, \tag{8}$$

where $Q_{\alpha,\beta}^{(P_j,\omega_j)} \in \mathbb{F}_q$ and $Q_{\alpha,\beta}^{(P_j,\omega_j)} = 0$ for $\alpha + \beta < m$, Q has a zero at (P_j, ω_j) with multiplicity at least m.

Theorem 1 [8]: Given a polynomial $f \in \mathcal{L}(\mu P_\infty)$ in the form of (4) and the polynomial $\mathcal{Q}$ which has a zero of multiplicity at least m over the n interpolation points, if $m(n - |\{j \mid f(P_j) \neq w_j, j \in [n]\}|) > \deg_{1,w_z} \mathcal{Q}$, $\mathcal{Q}(x, y, f) = 0$, or equivalently $(z - f) | \mathcal{Q}$.

When decoding Hermitian codes, the intended interpolation polynomial $\mathcal{Q}$, which is the minimum candidate under the $(1, w_z)$-revlex order satisfying the interpolation constraints, should be computed. The z-roots of $\mathcal{Q}$ are the estimated message polynomials.

III. The ASD and Its BR Interpolation

In this section, we introduce the ASD which can be considered as the soft decision variant of GS decoding, including the reliability transform, BR interpolation and root-finding.

A. Reliability Transform

Assume that a Hermitian codeword $\underline{c}$ is transmitted and $\underline{r} = (r_0, r_1, \ldots, r_{n-1})$ is the channel output. The reliability matrix $\mathbf{\Pi} \in \mathbb{R}^{q \times n}$ with entries $\pi_{ij} = \Pr(r_j \mid c_j = \sigma_i)$ can be obtained. Parametrized by the decoding OLS l, $\mathbf{\Pi}$ will be transformed into a multiplicity matrix $\mathbf{M}$ of the same size, where its entry $m_{ij} \in \mathbb{Z}_0^+$ is the multiplicity for interpolation point (P_j, σ_i). Let i_j denote the index of σ_i that yields $\sigma_i = c_j$. The codeword score of $\mathbf{M}$ is defined as

$$s_{\mathbf{M}}(\underline{c}) = \sum_{j=0}^{n-1} m_{i_j j}. \tag{9}$$

B. Interpolation and Root-Finding

Let $\mathrm{mult}_{(P_j,\sigma_i)}(Q)$ denote the multiplicity of polynomial Q at point (P_j, σ_i). Given $\mathbf{M}$, an ideal $\mathcal{I}_{\mathbf{M}}$ can be defined as

$$\begin{aligned}\mathcal{I}_{\mathbf{M}} = \{&Q \in \mathcal{R}[z] \mid \mathrm{mult}_{(P_j,\sigma_i)}(Q) \geq m_{ij}, \\ &\text{for all } 0 \leq i \leq q-1 \text{ and } 0 \leq j \leq n-1\}.\end{aligned} \tag{10}$$

Let $\mathcal{R}[z]_l = \{Q \in \mathcal{R}[z] \mid \deg_z Q \leq l\}$. The interpolation module $\mathcal{I}_{\mathbf{M},l}$ is defined as

$$\mathcal{I}_{\mathbf{M},l} = \mathcal{I}_{\mathbf{M}} \cap \mathcal{R}[z]_l. \tag{11}$$

With $\mathbf{M}$, interpolation constructs the minimum polynomial $\mathcal{Q}$ in $\mathcal{I}_{\mathbf{M},l}$ with respect to the $(1, w_z)$-revlex order. The following theorem reveals the sufficient condition of a successful ASD.

Theorem 2 [23]: Given a polynomial $f \in \mathcal{L}(\mu P_\infty)$ in the form of (4) and the interpolation polynomial $\mathcal{Q} \in \mathcal{I}_{\mathbf{M},l}$, if $s_{\mathbf{M}}(\underline{c}) > \deg_{1,w_z} \mathcal{Q}$, $\mathcal{Q}(x, y, f) = 0$, or equivalently $(z - f) | \mathcal{Q}$.

The computation of $s_{\mathbf{M}}(\underline{c})$ and $\deg_{1,w_z} \mathcal{Q}$ will be exemplified later in *Example 3*. The interpolation polynomial $\mathcal{Q}$ can be determined by BR interpolation. It first constructs a basis of the interpolation module $\mathcal{I}_{\mathbf{M},l}$, which will then be reduced into a Gröbner basis [8]. Its minimum candidate is $\mathcal{Q}$. For the construction of the module basis, a series of multiplicity matrices

need to be generated based on $\mathbf{M}$. Let $\eta = \max_j \left\{\sum_{i=0}^{q-1} m_{ij}\right\}$. These intermediate multiplicity matrices are denoted as $\mathbf{M}^{(u)}$, where $u = 0, 1, \ldots, \eta$. The first intermediate multiplicity matrix is initiated as $\mathbf{M}^{(0)} = \mathbf{M}$. Let $i_j^{(u)} = \arg\max_i\{m_{ij}^{(u)}\}$. The entries of $\mathbf{M}^{(1)}$ to $\mathbf{M}^{(\eta)}$ are determined by

$$m_{ij}^{(u+1)} = \begin{cases} m_{ij}^{(u)} - 1, \text{if } i = i_j^{(u)} \text{ and } m_{ij}^{(u)} \neq 0, \\ m_{ij}^{(u)}, \text{if } i \neq i_j^{(u)} \text{ or } m_{ij}^{(u)} = 0. \end{cases} \tag{12}$$

The following definition is further needed for constructing the module basis of $\mathcal{I}_{\mathbf{M},l}$.

Definition 3: : Given an index set $\mathcal{J} \subseteq [n]$, let $\mathbb{A}(\mathcal{J}) = \{x_j \mid j \in \mathcal{J}\}$. For $\sigma \in \mathbb{A}(\mathcal{J})$, let $\mathbb{B}_\sigma(\mathcal{J}) = \{y_j \mid (\sigma, y_j) \in \mathcal{P},\ j \in \mathcal{J}\}$, $\mathbb{C}(j) = \{j' \mid x_{j'} = x_j\}$ and $\mathbb{S}(\mathcal{J}) = \{j \mid |\mathbb{B}_{x_j}(\mathcal{J})| = w\}$. The affine points defined by $\mathcal{J}$ form a maximum semi-grid if $|\mathbb{B}_{x_j}(\mathcal{J})| = w$, for all $x_j \in \mathbb{A}(\mathcal{J})$.

The following example illustrates the above definitions.

Example 2: Consider the Hermitian curve in *Example 1*. Given $\mathcal{J} = \{0, 1, 2\}$, $\mathbb{A}(\mathcal{J}) = \{0, 1\}$, $\mathbb{B}_0(\mathcal{J}) = \{0, 1\}$, $\mathbb{B}_1(\mathcal{J}) = \{\alpha\}$ and $\mathbb{S}(\mathcal{J}) = \mathbb{C}(0) = \mathbb{C}(1) = \{0, 1\}$. Since $|\mathbb{B}_0(\mathcal{J})| = 2$ but $|\mathbb{B}_1(\mathcal{J})| = 1$, the affine points defined by $\mathcal{J}$ do not form a maximum semi-grid. Given an affine point index set $\mathcal{J} \subseteq [n]$, the Lagrange interpolation polynomial defined by $\mathcal{J}$ is written as

$$L_{\mathcal{J},j}(x, y) = \prod_{\alpha \in \mathbb{A}(\mathcal{J}) \setminus \{x_j\}} \frac{x - \alpha}{x_j - \alpha} \prod_{\beta \in \mathbb{B}_{x_j}(\mathcal{J}) \setminus \{y_j\}} \frac{y - \beta}{y_j - \beta}. \tag{13}$$

Note that $L_{\mathcal{J},j}(P_j) = 1$ and $L_{\mathcal{J},j}(P_{j'}) = 0$, if $j \neq j'$, where $j, j' \in \mathcal{J}$. Let $\omega_j^{(u)} = \sigma_{i_j^{(u)}}$ and $\underline{\omega}^{(u)} = (\omega_0^{(u)}, \omega_1^{(u)}, \ldots, \omega_{n-1}^{(u)})$. The following polynomial can be defined

$$K_{\underline{\omega}^{(u)}}(x, y) = \sum_{j \in [n]} \omega_j^{(u)} L_{[n],j}(x, y), \tag{14}$$

then $K_{\underline{\omega}^{(u)}}(P_j) = \omega_j^{(u)}$. Let $\mathfrak{m}_j^{(u)} = \max_i\{m_{ij}^{(u)}\}$. Let $\mathcal{E}_{\mathbf{M}^{(u)}}$ be an ideal in $\mathcal{R}$ and $\mathcal{E}_{\mathbf{M}^{(u)}} = \{S \in \mathcal{R} \mid \nu_{P_j}(S) \geq \mathfrak{m}_j^{(u)}, \text{ for all } j\}$. The interpolation module $\mathcal{I}_{\mathbf{M},l}$ can be computed using $\mathcal{E}_{\mathbf{M}^{(u)}}$ and polynomial $K_{\underline{\omega}^{(u)}}$ [23]. In order to compute $\mathcal{E}_{\mathbf{M}^{(u)}}$, indices of the w^3 affine points need to be reassigned by grouping them into w^2 classes with different x-coordinates. Thus, P_j is reassigned to $P_{a,b}$ with $a = \lfloor j/w \rfloor$ and $b = j \mod w$. If the point P_j is reassigned to $P_{a,b}$, $v_{a,b}^{(u)} = \mathfrak{m}_j^{(u)}$. It is also assumed that for each a, index b has been arranged such that $v_{a,b}^{(u)}$ are written in a decreasing order as

$$v_{a,0}^{(u)} \geq v_{a,1}^{(u)} \geq \ldots \geq v_{a,w-1}^{(u)}. \tag{15}$$

Given $B_{b,c}^{(u)} \in \mathbb{F}_q[x]$ satisfying $\nu_{P_{a,b}}(y - B_{b,c}^{(u)}) \geq v_{a,b}^{(u)} - v_{a,c}^{(u)}$, we further define the following polynomial

$$T_{u,c}(x, y) = \prod_{a=0}^{q-1} (x - \sigma_a)^{v_{a,c}^{(u)}} \prod_{0 \leq b \leq c-1} (y - B_{b,c}^{(u)}), \tag{16}$$

where $0 \leq u \leq \eta$ and $0 \leq c \leq w-1$, as the generator polynomial of $\mathcal{E}_{\mathbf{M}^{(u)}}$. The basis of $\mathcal{I}_{\mathbf{M},l}$ can be computed as the following theorem.

Theorem 3 [23]: $\mathcal{I}_{\mathbf{M},l}$ can be generated by

$$\mathcal{M} = \left\{ M_{u,c} \mid M_{u,c} = T_{u,c} \prod_{r=0}^{u-1} (z - K_{\underline{\omega}^{(r)}}) \right\}, \tag{17}$$

where $0 \leq u \leq \eta$ and $0 \leq c \leq w - 1$.

Note that the interpolation constraints with respect to $\mathbf{M}^{(u)}$ and $\mathbf{M} \backslash \mathbf{M}^{(u)}$ are satisfied by $T_{u,c}$ and $\prod_{r=0}^{u-1}(z - K_{\underline{\omega}^{(r)}})$, respectively. The Mulders-Storjohann (MS) algorithm [29][3] can be applied to reduce $\mathcal{M}$ into a Gröbner basis, in which the minimum polynomial is chosen as the interpolation polynomial $\mathcal{Q}$. Afterwards, the estimated message polynomial f_{est} will be decoded by finding the z-roots of $\mathcal{Q}$. This can be realized by the recursive coefficient search (RCS) algorithm [30], [31]. If multiple z-roots are found, the one whose corresponding codeword yields the minimum Euclidean distance to $\underline{\omega}$ will be chosen as the decoding output. Interpolation can be interpreted as first constructing the module basis that is defined by satisfying all interpolation constraints implied by $\mathbf{M}$. The following *Example 3* illustrates the above mentioned decoding.

Example 3: Suppose that the codeword in *Example 1* is transmitted over a noisy channel. The reliability matrix $\mathbf{\Pi}$, shown at the bottom of the next page. Given $l = 3$, $\mathbf{\Pi}$ is transformed into $\mathbf{M}$. This transform is parameterized by l such that $\mathbf{M}$ can sustain an interpolation polynomial with $\deg_z \mathcal{Q} \leq l$ [22], [23]. Algorithm A in [21] is used for multiplicity assignment. The resulting matrix is

$$\mathbf{M} = \begin{bmatrix} 0\,1\,0\,3\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,3\,0\,3 \\ 3\,0\,0\,0\,3\,0\,1\,0 \\ 0\,0\,3\,0\,0\,0\,1\,0 \end{bmatrix}.$$

Based on (17), the basis polynomials of $\mathcal{I}_{\mathbf{M},l}$ are

$$\begin{aligned}
M_{0,0} &= x^3 + x^6 + x^9 + x^{12}, \\
M_{0,1} &= \alpha^2 xy + \alpha^2 x^3 + \alpha x^3 y + \alpha^2 x^4 + \alpha^2 x^4 y + \alpha x^5 \\
&\quad + x^5 y + x^6 + \alpha x^6 y + \alpha x^7 + \alpha^2 x^8 + x^8 y \\
&\quad + \alpha x^9 + x^{10} + x^{11}, \\
M_{1,0} &= \alpha x^2 + \alpha x^2 y + x^3 + x^4 + x^5 y + \alpha x^8 \\
&\quad + \alpha x^8 y + x^9 + x^{10} + x^{11} y + (x^2 + x^8) z, \\
M_{1,1} &= \alpha xy + x^2 + \alpha x^3 + \alpha x^5 y + \alpha x^6 + \alpha^2 x^6 y \\
&\quad + \alpha^2 x^7 + \alpha^2 x^7 y + \alpha x^8 y + x^9 y + \alpha^2 x^{10} + \alpha x^{10} y \\
&\quad + x^{11} + (\alpha y + \alpha^2 xy + \alpha^2 x^2 + x^2 y + x^3 \\
&\quad + \alpha x^3 y + \alpha x^4 + \alpha^2 x^4 y + \alpha^2 x^5 + x^5 y + x^6 + \alpha x^7) z, \\
M_{2,0} &= \alpha^2 x + xz^2 + \alpha^2 xy + \alpha x^2 + \alpha x^2 y + \alpha^2 x^3 \\
&\quad + x^3 y + \alpha x^4 + \alpha^2 x^4 y + x^5 + \alpha^2 x^5 y + x^6 \\
&\quad + x^6 y + \alpha^2 x^7 + \alpha^2 x^7 y + x^8 + x^8 y + x^9 \\
&\quad + \alpha x^{10} + \alpha^2 x^{10} y + \alpha x^{11} + \alpha^2 x^{12} + (x^2 + \alpha x^2 y \\
&\quad + \alpha x^3 + \alpha^2 x^3 y + \alpha^2 x^4 + x^4 y + x^5 + \alpha x^5 y \\
&\quad + \alpha x^6 + \alpha^2 x^6 y + \alpha^2 x^7 + x^7 y) z + x^4 z^2, \\
M_{2,1} &= x + xy + x^2 + \alpha x^2 y + \alpha x^3 + \alpha^2 x^3 y \\
&\quad + \alpha^2 x^4 + \alpha^2 x^4 y + x^5 + x^6 + \alpha x^6 y + \alpha x^7 \\
&\quad + \alpha x^7 y + \alpha^2 x^8 + \alpha^2 x^8 y + \alpha x^9 + \alpha^2 x^9 y + \alpha x^{10} \\
&\quad + \alpha^2 x^{10} y + (xy + \alpha x^2 + \alpha^2 x^2 y + \alpha^2 x^3 y + \alpha x^4 \\
&\quad + \alpha x^4 y + \alpha x^5 y + x^6 + x^6 y + x^8) z + (\alpha y \\
&\quad + \alpha x + \alpha^2 xy + \alpha^2 x^2 + x^2 y + x^3) z^2,
\end{aligned}$$

[3]Although there are other basis reduction algorithms that have better asymptotic complexity, the MS algorithm is more efficient for handling codes of practical length.

$$\begin{aligned}M_{3,0} &= 1 + x + y + xy + x^2 + x^2y \\ &\quad + x^3 + x^6 + x^6y + x^8 + x^8y + \alpha^2x^9 \\ &\quad + \alpha x^{10} + \alpha x^{10}y + \alpha x^{11}y + \alpha x^{12} + (\alpha^2 + \alpha^2y \\ &\quad + \alpha^2x^2 + x^2y + \alpha^2x^3 + \alpha^2x^4y + x^5 + \alpha^2x^6 \\ &\quad + \alpha^2x^6y + \alpha^2x^7 + \alpha^2x^9)z + (\alpha + \alpha y + \alpha x \\ &\quad + xy + \alpha x^2y + \alpha x^3 + \alpha x^3y)z^2 + z^3, \\ M_{3,1} &= x^3 + x^3y + x^4 + x^5 + x^9 + \alpha^2x^9y \\ &\quad + x^{11} + \alpha x^{11}y + \alpha^2x^{12}y + \alpha x^{13} + \alpha x^{14} + (\alpha x^2y \\ &\quad + \alpha^2x^3 + \alpha^2x^3y + \alpha^2x^4y + x^5 + x^5y + \alpha^2x^7 \\ &\quad + \alpha^2x^7y + \alpha^2x^9 + \alpha^2x^9y)z + (\alpha^2xy + \alpha x^2y + \alpha x^3 \\ &\quad + x^4 + \alpha x^5 + \alpha x^6)z^2 + yz^3.\end{aligned}$$

Applying the MS algorithm yields the interpolation polynomial

$$\begin{aligned}\mathcal{Q} &= y + \alpha xy + \alpha x^3 + \alpha x^4 + \alpha x^4y + \alpha^2x^5 + \alpha^2x^5y \\ &\quad + \alpha x^6 + \alpha^2x^6y + \alpha^2x^7 + (\alpha^2 + \alpha y + x^2 \\ &\quad + \alpha x^3 + x^4)z + (\alpha + \alpha y + \alpha xy + \alpha^2x^2 \\ &\quad + \alpha^2x^2y + \alpha^2x^3)z^2 + z^3.\end{aligned}$$

Since $s_{\mathbf{M}}(\underline{c}) = 20$ and $\deg_{1,w_z} \mathcal{Q} = 17$, based on *Theorem 2*, the message polynomial can be obtained, i.e.,

$$f_{\text{est}}(x, y) = \alpha + x + \alpha^2y + x^2.$$

Note that decoding procedure requires 1758 finite field additions and 1293 multiplications.

IV. The Re-Encoding Transform and Its Improved Variant

This section presents the ReT for facilitating the following BR interpolation. For Hermitian codes, the ReT is derived based on a sufficient condition for choosing the re-encoding points, while its improved variant is designed based on iterative erasure decoding to employ more re-encoding points. Consequently, a greater complexity reduction can be achieved under better channel conditions.

A. The Re-Encoding Transform

The ReT shifts the interpolation points with a regenerated codeword based on the code's linear property. A subset of the interpolation points is chosen for re-encoding. Let Γ denote the index set of re-encoding points. The re-encoding points are denoted by $\boldsymbol{P}_\Gamma = \{(P_j, \omega_j^{(0)}) \mid j \in \Gamma\}$. The re-encoding polynomial is further defined as

$$K_\Gamma(x, y) = \sum_{j \in \Gamma} \omega_j^{(0)} L_{\Gamma,j}(x, y). \tag{18}$$

The following lemma presents the sufficient condition for generating a valid re-encoded Hermitian codeword through evaluating K_Γ over the points of $\mathcal{P}$.

Lemma 1 [20]: If $|\Gamma| \le w\lfloor (k-g)/w \rfloor$ and $\mathbb{S}(\Gamma) = \Gamma$, $K_\Gamma \in \mathcal{L}(\mu P_\infty)$.

Based on (18), the re-encoded codeword is generated by

$$\begin{aligned}\underline{h} &= (K_\Gamma(P_0), K_\Gamma(P_1), \ldots, K_\Gamma(P_{n-1})) \\ &= (h_0, h_1, \ldots, h_{n-1}).\end{aligned} \tag{19}$$

Note that $h_j = \omega_j^{(0)}$ for all $j \in \Gamma$. With $\underline{h}$, the interpolation points (P_j, σ_i) are transformed as

$$(P_j, \sigma_i) \mapsto (P_j, \sigma_{\widehat{i}}) : \sigma_{\widehat{i}} = \sigma_i - h_j. \tag{20}$$

Accordingly, the multiplicity matrix $\mathbf{M}$ can be transformed into $\widehat{\mathbf{M}}$, whose entries are denoted by $\widehat{m}_{ij}$, where

$$\widehat{m}_{\widehat{i}j} = m_{ij}. \tag{21}$$

The interpolation module with respect to $\widehat{\mathbf{M}}$ is defined as

$$\begin{aligned}\mathcal{I}_{\widehat{\mathbf{M}},l} = \{Q \in \mathcal{R}[z] \mid &\text{mult}_{(P_j,\sigma_{\widehat{i}})}(Q) \ge \widehat{m}_{\widehat{i}j} \text{ and } \deg_z Q \le l, \\ &\text{for all } \ 0 \le i \le q-1 \text{ for } 0 \le j \le n-1\}.\end{aligned} \tag{22}$$

Recalling the intermediate multiplicity matrices defined in Section III-B, let $\widehat{\mathbf{M}}^{(u)}$ denote the intermediate matrices that are elaborated from $\widehat{\mathbf{M}}$, and $\widehat{m}_{ij}^{(u)}$ denote the entries of $\widehat{\mathbf{M}}^{(u)}$. The following lemma establishes the relationship between the minimum polynomials of $\mathcal{I}_{\mathbf{M},l}$ and $\mathcal{I}_{\widehat{\mathbf{M}},l}$.

Lemma 2 [20]: $\mathcal{Q}$ is the minimum polynomial of $\mathcal{I}_{\mathbf{M},l}$ if and only if $\widehat{\mathcal{Q}}(x, y, z) = \mathcal{Q}(x, y, z + K_\Gamma)$ is the minimum polynomial of $\mathcal{I}_{\widehat{\mathbf{M}},l}$.

Let us further define the following polynomials

$$G(x) = \prod_{\sigma_a \in \mathbb{A}(\Gamma)} (x - \sigma_a)^{\nu_{a,w-1}^{(0)}} \tag{23}$$

and

$$G_\Gamma(x) = \prod_{\sigma_a \in \mathbb{A}(\Gamma)} (x - \sigma_a). \tag{24}$$

In order to show G is a common factor of the interpolation module basis polynomials, we define

$$\Gamma_u = \{j \mid \omega_{w\lfloor j/w \rfloor + b}^{(u)} = h_{w\lfloor j/w \rfloor + b}, j \in \Gamma, \text{ for all } b \in [w]\}. \tag{25}$$

Based on Γ_u, we further define

$$G_{\Gamma_u}(x) = \prod_{\sigma_a \in \mathbb{A}(\Gamma_u)} (x - \sigma_a) \tag{26}$$

and

$$G_{\Gamma \setminus \Gamma_u}(x) = \prod_{\sigma_a \in \mathbb{A}(\Gamma \setminus \Gamma_u)} (x - \sigma_a). \tag{27}$$

Note that $G_\Gamma(x) = G_{\Gamma_u}(x) \cdot G_{\Gamma \setminus \Gamma_u}(x)$. Polynomial G satisfies

$$G(x) = \prod_{u=0}^{\eta} G_{\Gamma_u}(x). \tag{28}$$

Let $\widehat{\mathfrak{m}}_j^{(u)} = \max_i \{\widehat{m}_{ij}^{(u)}\}$ and $\mathcal{E}_{\widehat{\mathbf{M}}^{(u)}} = \{S \in \mathcal{R} \mid \nu_{P_j}(S) \ge \widehat{\mathfrak{m}}_j^{(u)}\}$. The following lemma reveals that G is a common factor of the basis polynomials of $\mathcal{I}_{\widehat{\mathbf{M}},l}$.

$$\boldsymbol{\Pi} = \begin{bmatrix} 0.0054 & 0.3836 & 0.0000 & 0.9868 & 0.0082 & 0.0448 & 0.0029 & 0.0049 \\ 0.0000 & 0.2367 & 0.0006 & 0.0006 & 0.0003 & 0.9517 & 0.0021 & 0.9843 \\ 0.9901 & 0.2348 & 0.0012 & 0.0126 & 0.9527 & 0.0002 & 0.5793 & 0.0001 \\ 0.0045 & 0.1449 & 0.9982 & 0.0000 & 0.0388 & 0.0033 & 0.4157 & 0.0107 \end{bmatrix}$$

Lemma 3: : If $Q(x,y,z) \in \mathcal{I}_{\widehat{\mathbf{M}},l}$, $G|Q(x,y,zG_\Gamma)$.

Proof: Let $\widehat{i}_j^{(u)} = \arg\max_i\{\widehat{m}_{ij}^{(u)}\}$, $z_j^{(u)} = \sigma_{\widehat{i}_j^{(u)}}$ and $\underline{z}^{(u)} = (z_0^{(u)}, z_1^{(u)}, \ldots, z_{n-1}^{(u)})$. Based on Theorem 3, Q can be generated by

$$\widehat{\mathcal{M}} = \left\{ \widehat{M}_{u,c} \mid \widehat{M}_{u,c} = \widehat{T}_{u,c} \cdot \prod_{r=0}^{u-1} (z - K_{\underline{z}^{(r)}}) \right\}, \tag{29}$$

where $\widehat{T}_{u,c}$ is the basis polynomial of $\mathcal{E}_{\widehat{\mathbf{M}}^{(u)}}$, and $K_{\underline{z}^{(r)}}$ is denoted as

$$K_{\underline{z}^{(r)}}(x,y) = \sum_{j\in[n]} z_j^{(r)} L_{[n],j}, \tag{30}$$

where $0 \le u \le \eta$ and $0 \le c \le w-1$. For each $\widehat{M}_{u,c}(x,y,zG_\Gamma)$, it can be expressed as

$$\widehat{M}_{u,c}(x,y,zG_\Gamma) = \widehat{T}_{u,c} \cdot \prod_{r=0}^{u-1} (zG_\Gamma - K_{\underline{z}^{(r)}}). \tag{31}$$

Since $\prod_{\sigma_a\in\mathbb{A}([n])}(x-\sigma_a)^{v_{a,w-1}^{(u)}} | \widehat{T}_{u,c}$, it follows that $\prod_{r=u}^{\eta} G_{\Gamma_r} | \widehat{T}_{u,c}$. The polynomial $K_{\underline{z}^{(r)}}$ can be written as

$$\begin{aligned}
K_{\underline{z}^{(r)}} &= \sum_{j\in[n]} z_j^{(u)} L_{[n],j} \\
&= \sum_{j\in\Gamma_r} 0 \cdot L_{[n],j} + \sum_{j\in[n]\setminus\Gamma_r} z_j^{(r)} L_{[n],j} \\
&= \sum_{j\in[n]\setminus\Gamma_u} z_j^{(u)} \prod_{\alpha\in\mathbb{A}([n])\setminus\{x_j\}} \frac{x-\alpha}{x_j-\alpha} \prod_{\beta\in\mathbb{B}_{x_j}([n])\setminus\{y_j\}} \frac{y-\beta}{y_j-\beta} \\
&= G_{\Gamma_r} \cdot \sum_{j\in[n]\setminus\Gamma_r} \frac{z_j^{(r)}}{G_{\Gamma_r}(x_j)} \\
&\quad \prod_{\alpha\in\mathbb{A}([n]\setminus\Gamma_r)\setminus\{x_j\}} \frac{x-\alpha}{x_j-\alpha} \prod_{\beta\in\mathbb{B}_{x_j}([n]\setminus\Gamma_r)\setminus\{y_j\}} \frac{y-\beta}{y_j-\beta}.
\end{aligned} \tag{32}$$

Therefore, we have $\prod_{r=0}^{u-1} G_{\Gamma_r} | \prod_{r=0}^{u-1}(zG_\Gamma - K_{\underline{z}^{(r)}})$. Since $G = \prod_{r=0}^{u-1} G_{\Gamma_r} \cdot \prod_{r=u}^{\eta} G_{\Gamma_r}$, $G|\widehat{M}_{u,c}(x,y,zG_\Gamma)$. Therefore, $G|Q(x,y,zG_\Gamma)$. □

Lemma 3 is the cornerstone for extracting the common factor G, which enables the complexity reduction in BR interpolation. The generation of $\widehat{\mathbf{M}}$ and G will be shown later in *Examples 4* and *5*. The extraction of G from the module basis will be demonstrated in *Example 6*. The above description illustrates that as long as the choosing of re-encoding points satisfies *Lemma 1*, we can transform the interpolation points so that the module basis polynomials share a common factor G. Based on (23),

$$\deg G = \sum_{\sigma_a\in\mathbb{A}(\Gamma)} v_{a,w-1}^{(0)}. \tag{33}$$

The degree of G determines the complexity reduction brought by the ReT. Therefore, we aim to maximize $\deg G$. First, $|\Gamma|$ should be maximized. Therefore, we need to ensure that

$$|\Gamma| = w\lfloor (k-g)/w \rfloor. \tag{34}$$

Moreover, we should also maximize $v_{a,w-1}^{(0)}$, for all $\sigma_a \in \mathbb{A}(\Gamma)$, i.e. points of $\boldsymbol{P}_\Gamma$ should correspond to larger multiplicities. By sorting $v_{a,w-1}^{(0)}$ in a descending order, a refreshed x-coordinates index sequence can be obtained. It indicates

$$v_{a_0,w-1}^{(0)} \ge v_{a_1,w-1}^{(0)} \ge \ldots \ge v_{a_{q-1},w-1}^{(0)}. \tag{35}$$

The index set of $\boldsymbol{P}_\Gamma$ corresponds to the first $|\Gamma|/w$ ordered x-coordinates. Therefore,

$$\begin{aligned}
\Gamma = \{ j \mid\ & j = a_\iota + b, \\
& 0 \le \iota \le \lfloor (k-g)/w \rfloor - 1, 0 \le b \le w-1 \}.
\end{aligned} \tag{36}$$

Since ReT is a newly added step in the original decoding process, we need to analyze its asymptotic complexity. The ReT mainly consists of two parts: computing K_Γ and $\underline{h}$. We will analyze the complexity of each part and identify the one that dominates the overall complexity. Since $\deg_x \prod_{\alpha\in\mathbb{A}(\Gamma)\setminus\{x_j\}} \frac{x-\alpha}{x_j-\alpha} \le (n-|\Gamma|-1)/w$ and $\deg_y \prod_{\alpha\in\mathbb{A}(\Gamma)\setminus\{x_j\}} \frac{x-\alpha}{x_j-\alpha} \le w-1$, computing one $L_{\Gamma,j}$ exhibits a complexity of $O(|\Gamma|)$. Since there are $|\Gamma|$ occurrences of $L_{\Gamma,j}$, K_Γ can be computed with $O(|\Gamma|^2)$. The complexity of computing $\underline{h}$ exhibits a complexity of $O(n)$[4]. Therefore, the ReT exhibits a complexity of $O(|\Gamma|^2)$.

Algorithm 1 The ReT for ASD of Hermitian Codes

Input: $\mathbf{M}$;
Output: $\widehat{\mathbf{M}}$;
1: Reassign P_j to $P_{a,b}$;
2: Sort $v_{a,w-1}$ in a descending order as in (35);
3: Generate the index set of re-encoding points Γ as in (36);
4: Compute the re-encoding polynomial K_Γ as in (18);
5: Generate the re-encoded codeword $\underline{h}$ as in (19);
6: Transform $\mathbf{M}$ to $\widehat{\mathbf{M}}$ as in (20).

The ReT for ASD of Hermitian codes is summarized in Algorithm 1.

Example 4: Given the same multiplicity matrix $\mathbf{M}$ of *Example 3*, we reassign the index of eight affine points as

$$\begin{aligned}
&P_{0,0} = P_0, P_{0,1} = P_1, P_{1,0} = P_2, P_{1,1} = P_3, \\
&P_{2,0} = P_4, P_{2,1} = P_5, P_{3,0} = P_7, P_{3,1} = P_6.
\end{aligned}$$

Hence,

$$v_{0,1} = 1, v_{1,1} = 3, v_{2,1} = 3, v_{3,1} = 1.$$

Since $w\lfloor \frac{k-g}{w} \rfloor = 4$, $\Gamma = \{2,3,4,5\}$. Based on (18), the re-encoding polynomial is

$$K_\Gamma(x,y) = \alpha^2 y + \alpha x.$$

Therefore, the re-encoding codeword can be generated by

$$\underline{h} = (0, \alpha^2, \alpha^2, 0, \alpha, 1, 0, \alpha^2).$$

Based on this, the interpolation points are transformed as

$$\begin{aligned}
&(P_0,\alpha) \mapsto (P_0,\alpha),\ (P_1,0) \mapsto (P_1,\alpha^2),\ (P_2,\alpha^2) \mapsto (P_2,0), \\
&(P_3,0) \mapsto (P_3,0),\ (P_4,\alpha) \mapsto (P_4,0),\ (P_5,1) \mapsto (P_5,0), \\
&(P_6,\alpha) \mapsto (P_6,\alpha),\ (P_6,\alpha^2) \mapsto (P_6,\alpha^2),\ (P_7,1) \mapsto (P_7,\alpha).
\end{aligned}$$

Therefore, the transformed matrix is

$$\widehat{\mathbf{M}} = \begin{bmatrix} 0&0&3&3&3&3&0&0 \\ 0&0&0&0&0&0&0&0 \\ 3&0&0&0&0&0&1&3 \\ 0&1&0&0&0&0&1&0 \end{bmatrix}.$$

The common factor can also be computed as in (23), yielding

$$G(x) = 1 + \alpha x + x^3 + \alpha^2 x^5 + x^6.$$

[4]The fast encoding algorithm based on fast multipoint evaluation can be used to achieve a quasi-linear complexity [32].

B. The Improved Variant

As mentioned above, if more re-encoding points are chosen, the common factor has a greater degree. However, the sufficient condition of *Lemma 1* limits $|\Gamma|$ to at most $w\lfloor(k-g)/w\rfloor$. It should be noted that even if $|\Gamma| > w\lfloor(k-g)/w\rfloor$, it is still possible that $K_\Gamma \in \mathcal{L}(\mu P_\infty)$. In this section, we propose an approach to formulate Γ using erasure decoding iteratively, which allows Γ to have a greater cardinality when the channel condition improves.

As demonstrated in the previous subsection, we can compute K_Γ and $\underline{h}$ as in (18) and (19), respectively. This process can be interpreted as Lagrange interpolation-based erasure decoding, where Γ is the index set of preserved symbols. In the following, we will illustrate the iterative formation based on erasure decoding. First, we can choose as many re-encoding points as possible. Hence, Γ is initialized as $\Gamma = \mathbb{S}([n])$. The re-encoding polynomial K_Γ can be computed as in (18). If $\deg_{1,w_z} K_\Gamma \le \mu$, i.e. K_Γ falls into the desired Riemann-Roch space, $\underline{h}$ will be generated as in (19). It will be applied for transforming the interpolation points as in (20). However, if $\deg_{1,w_z} K_\Gamma > \mu$, Γ should be updated. Let $\pi'_j = \max_i\{\pi_{ij}\}$ and $j' = \arg\min_j\{\pi'_j\}$, where $j \in \Gamma$. The index set Γ will be updated by

$$\Gamma = \Gamma \backslash \mathbb{C}(j'). \tag{37}$$

Note that although Γ is updated, the Lagrange interpolation polynomial does not need to be recomputed. It will be updated by

$$L_{\Gamma,j} = L_{\Gamma,j} \cdot \frac{x_j - x_{j'}}{x - x_{j'}}. \tag{38}$$

The polynomial K_Γ will be computed again and checked if $\deg_{1,w_z} K_\Gamma \le \mu$. Let G_1 and G_2 denote common factors generated by the ReT of the previous subsection and the improved ReT of this subsection, respectively. This iterative formulation of Γ terminates if any of the following conditions occurs:

1) $\deg_{1,w_z} K_\Gamma \le \mu$;
2) $|\Gamma| \le w\lfloor(k-g)/w\rfloor$;
3) $\deg G_1 \ge \deg G_2$.

If 1) occurs, it indicates that a valid Γ has been generated. Hence, the interpolation points transform can be performed as in (20). The conditions 2) and 3) both indicate that the improved ReT is not more efficient than the ReT of Section IV-A. As previously mentioned, the essence of the improved ReT lies in its utilization of erasure decoding based on Lagrange interpolation. When the channel condition is good, erasure decoding is more likely to succeed, thereby yielding a valid re-encoded Hermitian codeword. Consequently, this improved variant is more suitable for scenarios with good channel condition. However, when channel quality is insufficient, erasure decoding becomes less likely to succeed while incurring additional computational cost. In such cases, the ReT of Section IV-A is more efficient.

Remark 1: The complexity for an improved ReT iteration is also $O(|\Gamma|^2)$. But it may require multiple iterations. Hence, the actual complexity depends on both the number of iterations and $|\Gamma|$ of each iteration. In fact, the improved ReT has higher complexity than the ReT in the previous subsection. However, it leads to a significant complexity reduction for the subsequent BR interpolation.

Remark 2: Once a valid $\underline{h}$ is obtained, we can validate whether it satisfies the maximum likelihood (ML) criterion [33], [34] or the acceptance criterion [35]. If the criterion is satisfied, the decoding can terminate early, similar to the approach in [20]. Since the improved ReT results in $\underline{h}$ and $\underline{\omega}$ sharing more identical symbols, the improved ReT can trigger early termination, while the ReT of Section IV-A is unlikely to realize it. Although both the improved ReT and the method in [20] support early termination, the former additionally produces a common factor G of a higher degree.

Algorithm 2 The Improved ReT for ASD of Hermitian Codes

Input: $\mathbf{M}$;
Output: $\widehat{\mathbf{M}}$;
1: Initialize Γ as $\Gamma = \mathbb{S}([n])$;
2: Generate K_Γ as in (18);
3: **While** $K_\Gamma \notin \mathcal{L}(\mu P_\infty)$
4: Update Γ as in (37);
5: **If** $|\Gamma| \le w\lfloor(k-g)/w\rfloor$ or $\deg G_1 \ge \deg G_2$
6: Terminate the iteration and deploy Algorithm 1;
7: **End If**
8: Update $L_{\Gamma,j}$ as in (38);
9: Compute K_Γ as in (18);
10: **End While**
11: Generate the re-encoded codeword $\underline{h}$ as in (19);
12: Transform $\mathbf{M}$ to $\widehat{\mathbf{M}}$ as in (20).

The improved ReT is summarized in Algorithm 2.

Example 5: Given the same Hermitian code and the same $\mathbf{M}$ in *Example 3*, the index set Γ is initialized by

$$\begin{aligned}\Gamma &= \mathbb{S}([n])\\ &= \{0, 1, \ldots, 7\}.\end{aligned}$$

The Lagrange interpolation polynomials are computed by

$$\begin{aligned}
L_{\Gamma,0} &= 1 + y + x^3 + x^3\ y,\\
L_{\Gamma,1} &= y + x^3\ y,\\
L_{\Gamma,2} &= \alpha^2 x + xy + \alpha^2 x^2 + x^2 y + \alpha^2 x^3 + x^3\ y,\\
L_{\Gamma,3} &= \alpha x + xy + \alpha x^2 + x^2 y + \alpha x^3 + x^3\ y,\\
L_{\Gamma,4} &= \alpha x + \alpha^2 xy + x^2 + \alpha x^2 y + \alpha x^3 + x^3\ y,\\
L_{\Gamma,5} &= x + \alpha^2 xy + \alpha^2 x^2 + \alpha x^2 y + \alpha x^3 + x^3\ y,\\
L_{\Gamma,6} &= x + \alpha xy + \alpha^2 x^2 + \alpha^2 x^2 y + \alpha x^3 + x^3\ y,\\
L_{\Gamma,7} &= \alpha^2 x + \alpha xy + x^2 + \alpha^2 x^2 y + \alpha x^3 + x^3\ y.
\end{aligned}$$

Since $\deg_{1,w_z} K_\Gamma = 9$, $\Gamma = \{0, 1, \ldots, 7\}$ is not a valid index set for re-encoding points and it should be updated. Based on $\mathbf{M}$,

$$\begin{aligned}\pi'_0 = 0.9901, \pi'_1 = 0.3836, \pi'_2 = 0.9982, \pi'_3 = 0.9868,\\ \pi'_4 = 0.9527, \pi'_5 = 0.9517, \pi'_6 = 0.5793, \pi'_7 = 0.9843.\end{aligned}$$

Hence, $j' = 0$. The index set Γ is updated by

$$\begin{aligned}\Gamma &= \Gamma \backslash \mathbb{C}(j')\\ &= \{2, 3, 4, 5, 6, 7\}.\end{aligned}$$

The Lagrange interpolation polynomials are updated by

$$\begin{aligned}
L_{\Gamma,2} &= \alpha^2 + y + \alpha^2 x + xy + \alpha^2 x^2 + x^2\ y,\\
L_{\Gamma,3} &= \alpha + y + \alpha x + xy + \alpha x^2 + x^2\ y,
\end{aligned}$$

$$L_{\Gamma,4} = \alpha + \alpha^2 y + x + \alpha xy + \alpha^2 x^2 + x^2\ y,$$
$$L_{\Gamma,5} = \alpha + y + x + \alpha^2 xy + \alpha^2 x^2 + \alpha x^2\ y,$$
$$L_{\Gamma,6} = \alpha^2 + y + x + \alpha xy + \alpha x^2 + \alpha^2 x^2\ y,$$
$$L_{\Gamma,7} = \alpha + y + \alpha^2 x + \alpha xy + x^2 + \alpha^2 x^2\ y.$$

Since $\deg_{1,w_z} K_\Gamma = 4$, $\Gamma = \{2,3,4,5,6,7\}$ is a valid index set for re-encoding points. The common factor is

$$G(x) = \alpha^2 + \alpha x^2 + \alpha^2 x^3 + x^4 + \alpha x^5 + x^7.$$

Compared with *Example 4*, this Example employs more re-encoding points and yields a common factor of a higher degree.

V. The Modified BR Interpolation

Based on the ReT schemes of Section IV, this section proposes the modified BR interpolation. The modification exploits the algebraic structure induced by the ReT in reducing interpolation complexity. We begin with introducing modified module basis construction.

A. Modified Module Basis Construction

As described in *Lemma 3*, polynomial G is a common factor of the module basis polynomials. Consequently, this common factor can be extracted, leading to a module isomorphism. Let Φ denote the isomorphic module with respect to $\mathcal{I}_{\widehat{\mathbf{M}},l}$. The module isomorphism between $\mathcal{I}_{\widehat{\mathbf{M}},l}$ and Φ is

$$\mathcal{I}_{\widehat{\mathbf{M}},l} \mapsto \Phi$$
$$\widehat{Q}(x,y,z) = G\widetilde{Q}\left(x,y,\frac{z}{G_\Gamma}\right) \mapsto \widetilde{Q}(x,y,z). \tag{39}$$

The following lemma describes the relationship between $\mathcal{I}_{\widehat{\mathbf{M}},l}$ and Φ.

Lemma 4: Polynomial $\widehat{\mathcal{Q}}(x,y,z)$ is the minimum polynomial with respect to the $(1,w_z)$-revlex order in $\mathcal{I}_{\widehat{\mathbf{M}},l}$ if and only if $\widetilde{\mathcal{Q}} = G^{-1}\widehat{\mathcal{Q}}(x,y,zG_\Gamma)$ is the one with respect to the $(1,w_z-|\Gamma|)$-revlex order in Φ.

Proof: Based on (39), $\widehat{Q}$ can be written as

$$\widehat{Q}(x,y,z) = G\cdot\widetilde{Q}\left(x,y,\frac{z}{G_\Gamma}\right). \tag{40}$$

Since $\deg_{1,w_z} G_\Gamma = |\Gamma|$,

$$\begin{aligned}\deg_{1,w_z}\widehat{Q}(x,y,z) &= \deg_{1,w_z}\left(G\cdot\widetilde{Q}\left(x,y,\frac{z}{G_\Gamma}\right)\right)\\ &= \deg_{1,w_z}G + \deg_{1,w_z}\widetilde{Q}\left(x,y,\frac{z}{G_\Gamma}\right)\\ &= \deg_{1,w_z}G + \deg_{1,w_z-|\Gamma|}\widetilde{Q}(x,y,z).\end{aligned} \tag{41}$$

Given $\widehat{\mathcal{Q}}$ as the minimum polynomial with respect to the $(1,w_z)$-revlex order in $\mathcal{I}_{\widehat{\mathbf{M}},l}$, if there exists a polynomial $\widetilde{\mathcal{S}}$ that satisfies $\deg_{1,w_z-|\Gamma|}\widetilde{\mathcal{S}} < \deg_{1,w_z-|\Gamma|}\widetilde{\mathcal{Q}}$, we have $\widehat{\mathcal{S}}(x,y,z) = G\cdot\widetilde{\mathcal{S}}\left(x,y,\frac{z}{G_\Gamma}\right)$. Based on (41),

$$\begin{aligned}\deg_{1,w_z}\widehat{\mathcal{S}} &= \deg_{1,w_z}G + \deg_{1,w_z-|\Gamma|}\widetilde{\mathcal{S}}\\ &< \deg_{1,w_z}G + \deg_{1,w_z-|\Gamma|}\widetilde{\mathcal{Q}}\\ &= \deg_{1,w_z}\widehat{\mathcal{Q}},\end{aligned} \tag{42}$$

which contradicts $\widehat{\mathcal{Q}}$ being the minimum polynomial. Therefore, if $\widehat{\mathcal{Q}}(x,y,z)$ is the minimum polynomial with respect to the $(1,w_z)$-revlex order in $\mathcal{I}_{\widehat{\mathbf{M}},l}$, $\widetilde{\mathcal{Q}} = G^{-1}\widehat{\mathcal{Q}}(x,y,zG_\Gamma)$ is the one with respect to the $(1,w_z-|\Gamma|)$-revlex order in Φ, and vice versa. □

Lemma 5: Φ can be generated as an $\mathbb{F}_q[x]$-module by

$$\begin{aligned}\widetilde{\mathcal{M}} = \{&\widetilde{M}_{u,c}(x,y,z) \mid \widetilde{M}_{u,c}(x,y,z)\\ &= \widetilde{T}_{u,c}(x,y)\cdot\prod_{r=0}^{u-1}(zG_{\Gamma\setminus\Gamma_r}(x) - \widetilde{K}_{\underline{z}_r})\},\end{aligned} \tag{43}$$

where

$$\widetilde{T}_{u,c} = \frac{\widehat{T}_{u,c}}{\prod_{r=u}^{\eta} G_{\Gamma_r}} \tag{44}$$

and

$$\begin{aligned}\widetilde{K}_{\underline{z}^{(r)}} = &\sum_{j\in[n]\setminus\Gamma_r}\frac{z_j^{(r)}}{G_{\Gamma_r}(x_j)}\\ &\prod_{\alpha\in\mathbb{A}([n]\setminus\Gamma_r)\setminus\{x_j\}}\frac{x-\alpha}{x_j-\alpha}\prod_{\beta\in\mathbb{B}_{x_j}([n]\setminus\Gamma_r)\setminus\{y_j\}}\frac{y-\beta}{y_j-\beta}.\end{aligned} \tag{45}$$

Proof: If $Q(x,y,z)\in\mathcal{I}_{\widehat{\mathbf{M}},l}$, $Q(x,y,z)$ can be constructed using the basis polynomial defined in (29). Thus $Q(x,y,zG_\Gamma)$ can be constructed using the basis polynomial defined in (31). Since

$$\begin{aligned}\widetilde{M}_{u,c}(x,y,z) &= \frac{\widehat{T}_{u,c}}{\prod_{r=u}^{\eta}G_{\Gamma_r}}\cdot\frac{\prod_{r=0}^{u-1}(zG_\Gamma - K_{\underline{z}^{(r)}})}{\prod_{r=0}^{u-1}G_{\Gamma_u}}\\ &= \frac{\widehat{M}_{u,c}(x,y,zG_\Gamma)}{G},\end{aligned} \tag{46}$$

based on the bijective mapping defined in (39), Φ can be generated by $\widetilde{\mathcal{M}}$. □

We can first compute the minimum polynomial $\widetilde{\mathcal{Q}}$ in Φ using the BR interpolation, then restore it into $\widehat{\mathcal{Q}}$ by

$$\widehat{\mathcal{Q}}(x,y,z) = G(x)\cdot\widetilde{\mathcal{Q}}\left(x,y,\frac{z}{G_\Gamma(x)}\right). \tag{47}$$

Afterwards, the RCS algorithm will be applied to find the z-roots of $\widehat{\mathcal{Q}}$. The z-roots of $\widehat{\mathcal{Q}}$, denoted by $\widehat{f}$, can be used to restore the estimated message polynomial f_{est}. Based on *Lemma 2*, f_{est} can be computed by

$$f_{\text{est}}(x,y) = \widehat{f}(x,y) + K_\Gamma(x,y). \tag{48}$$

Algorithm 3 The ReT-Based ASD for Hermitian Codes

Input: $\mathbf{\Pi}, l$;
Output: f_{est};
1: Transform $\mathbf{\Pi}$ to $\mathbf{M}$ that sustains l;
2: Transform $\mathbf{M}$ to $\widehat{\mathbf{M}}$ using Algorithms 1 or 2;
3: Formulate the basis polynomials of $\widetilde{\mathcal{M}}$ as in (43) - (45);
4: Perform the basis reduction using the MS algorithm, obtaining $\tilde{\mathcal{Q}}$;
5: Restore $\tilde{\mathcal{Q}}$ to $\widehat{\mathcal{Q}}$ as in (47);
6: Determine $\widehat{f}$ by finding z-roots of $\widehat{\mathcal{Q}}$;
7: Compute f_{est} as in (48).

The ReT-based ASD for Hermitian codes is summarized in Algorithm 3.

In order to better illustrate the above mentioned ReT-assisted BR interpolation and its subsequent root-finding, the following *Example 6* is further provided.

Example 6: Given the transformed multiplicity matrix $\widehat{\mathbf{M}}$ yielding in *Example 4*, we have

$$G(x) = 1 + \alpha x + x^3 + \alpha^2 x^5 + x^6$$

and

$$G_\Gamma(x) = \alpha + \alpha^2 x + x^2.$$

To better illustrate *Lemma 3*, we demonstrate the division relationship for a specific basis polynomial. With the same $\mathbf{M}$ and $G(x)$ in *Example 4*, the module basis polynomial $\widehat{M}_{0,0}(x, y, zG_\Gamma)$ is

$$\widehat{M}_{0,0}(x, y, zG_\Gamma) = x^3 + x^6 + x^9 + x^{12}.$$

It satisfies

$$\frac{x^3 + x^6 + x^9 + x^{12}}{1 + \alpha x + x^3 + \alpha^2 x^5 + x^6} = x^3 + \alpha x^4 + \alpha^2 x^5 + x^6.$$

Hence, $G(x)|\widehat{M}_{0,0}(x, y, zG_\Gamma)$, and the quotient is the isomorphic basis polynomial $\widetilde{M}_{0,0}(x, y, z)$. The same process can also apply to all other module basis polynomials $\widehat{M}_{u,c}(x, y, zG_\Gamma)$. Based on (43), the basis polynomials of Φ are

$$\begin{aligned}
\widetilde{M}_{0,0} &= x^3 + \alpha x^4 + \alpha^2 x^5 + x^6,\\
\widetilde{M}_{0,1} &= \alpha^2 xy + x^2 y + \alpha^2 x^3 + \alpha x^4 + x^5,\\
\widetilde{M}_{1,0} &= \alpha x^2 + x^2 y + \alpha x^3 y + x^4 + \alpha^2 x^4 y + x^5 y\\
&\quad + (\alpha x^2 + x^4) z,\\
\widetilde{M}_{1,1} &= y + xy + x^2 + \alpha x^2 y + \alpha x^4 y + x^5\\
&\quad + (\alpha^2 y + xy + x^2 + \alpha x^3) z,\\
\widetilde{M}_{2,0} &= \alpha^2 x + \alpha^2 xy + \alpha^2 x^2 + \alpha^2 x^2 y + \alpha^2 x^3 + \alpha^2 x^3 y\\
&\quad + \alpha x^4 + x^5 + x^6 + \alpha^2 x^7 + (xy + \alpha x^2\\
&\quad + \alpha^2 x^2 y + \alpha^2 x^3 + \alpha x^3 y) z + (\alpha^2 x + x^2) z^2,\\
\widetilde{M}_{2,1} &= x + xy + \alpha^2 x^2 + \alpha^2 x^2 y + x^3 + \alpha x^3 y\\
&\quad + \alpha^2 x^4 y + \alpha^2 x^5 + \alpha^2 x^5 y + \alpha^2 x^6 + (\alpha y + \alpha^2 xy\\
&\quad + \alpha^2 x^2 + \alpha x^2 y + \alpha x^3 + \alpha x^4) z + (y + x) z^2,\\
\widetilde{M}_{3,0} &= 1 + \alpha x + y + \alpha xy + \alpha^2 x^2 + \alpha^2 x^2 y\\
&\quad + \alpha x^3 + \alpha^2 x^3 y + \alpha^2 x^4 + \alpha^2 x^4 y + \alpha^2 x^5 y + \alpha^2 x^6\\
&\quad + (1 + y + \alpha xy + \alpha^2 x^2 + \alpha x^2 y + x^3\\
&\quad + x^4 + \alpha^2 x^5) z + (1 + \alpha^2 y + \alpha x + \alpha xy) z^2 + z^3,\\
\widetilde{M}_{3,1} &= x^3 + x^3 y + \alpha x^4 + \alpha^2 x^5 + \alpha^2 x^5 y + \alpha^2 x^6\\
&\quad + \alpha^2 x^6 y + \alpha^2 x^7 + \alpha^2 x^8 + (\alpha xy + x^2 y + x^3\\
&\quad + x^3 y + \alpha x^4 + x^4 y + \alpha x^5 + \alpha^2 x^5 y) z + (\alpha y\\
&\quad + \alpha^2 x^3 + \alpha x^4) z^2 + y z^3.
\end{aligned}$$

It can be seen that the x-degrees of $\widetilde{M}_{u,c}$ are smaller than that of $M_{u,c}$ in *Example 3*. Applying the MS algorithm, the interpolation polynomial of Φ can be obtained as

$$\widetilde{\mathcal{Q}} = \alpha^2 y + \alpha^2 x + z + (\alpha^2 y + \alpha^2 x) z^2 + z^3.$$

Based on (47), it can be restored into $\widehat{\mathcal{Q}}$ as

$$\widehat{\mathcal{Q}}(x, y, z) = G(x) \cdot \widetilde{\mathcal{Q}}\left(x, y, \frac{z}{G_\Gamma(x)}\right)$$

$$\begin{aligned}
&= \alpha^2 y + x + xy + x^2 + x^3 + \alpha^2 x^3 y\\
&\quad + \alpha^2 x^4 + \alpha^2 x^5 + \alpha x^5 y + \alpha^2 x^6 + \alpha^2 x^6 y + \alpha^2 x^7\\
&\quad + (\alpha^2 + \alpha x^2 + x^4) z + (\alpha + y + \alpha x\\
&\quad + \alpha xy + \alpha^2 x^2 + \alpha^2 x^2 y + \alpha^2 x^3) z^2 + z^3.
\end{aligned}$$

The z-root of $\widehat{\mathcal{Q}}$ is

$$\widehat{f}(x, y) = \alpha + \alpha^2 x + x^2.$$

Based on (48), the estimated message polynomial f_{est} can be obtained by

$$\begin{aligned}
f_{\text{est}}(x, y) &= \widehat{f}(x, y) + K_\Gamma(x, y)\\
&= \alpha + x + \alpha^2 y + x^2.
\end{aligned}$$

The decoding procedure requires 861 additions and 648 multiplications, reducing the number of finite field arithmetic operations compared to *Example 3*. Moreover, by applying the improved ReT (*Example 5*), the decoding complexity can be further reduced, requiring only 622 additions and 605 multiplications.

B. Complexity Reduction

This subsection provides an asymptotic complexity analysis for the modified BR interpolation. First, we consider the basis construction complexity. The basis construction mainly consists of four steps: computing $\widetilde{T}_{u,c}$, computing $\widetilde{K}_{\underline{z}^{(r)}}$, multiplying $\prod_{r=0}^{u-1}(zG_{\Gamma\setminus\Gamma_r} - \widetilde{K}_{z_r})$ and obtaining $\widetilde{M}_{u,c}$. We will analyze the complexity of each part and identify the one that dominates the overall complexity. Since $\prod_{r=u}^{\eta} G_{\Gamma_r}|\widehat{T}_{u,c}$, $\deg_x \prod_{a=0}^{q-1}(x-\sigma_a)^{v_{a,c}^{(u)}} \le lw^2 - \sum_{r=u}^{\eta}|\Gamma_r|/w$, $\deg_x \prod_{0\le b\le c-1}(y - B_{b,c}^{(u)}) \le lw^3$ and $\deg_y \prod_{0\le b\le c-1}(y - B_{b,c}^{(u)}) \le w - 1$. Hence, $\widetilde{T}_{u,c}$ can be computed with a complexity of $O(l^3 wn(n - |\Gamma|))$. Since $G_{\Gamma_r}|K_{\underline{z}}^{(r)}$, $\deg_x \widetilde{K}_{\underline{z}^{(r)}} \le w^2 - 1 - |\Gamma_r|/w$ and $\deg_y \widetilde{K}_{\underline{z}^{(r)}} \le w - 1$. Hence, the complexity of computing $\widetilde{K}_{\underline{z}^{(r)}}$ is $O(l(n - |\Gamma|)^2)$. Since $\deg_x \prod_{r=0}^{u-1}(zG_{\Gamma\setminus\Gamma_r} - \widetilde{K}_{z_r}) \le \sum_{r=0}^{u-1}(n - |\Gamma_r|)/w - 1$, $\deg_y \prod_{r=0}^{u-1}(zG_{\Gamma\setminus\Gamma_r} - \widetilde{K}_{z_r}) \le (u-1)(w-1)$ and $\deg_z \prod_{r=0}^{u-1}(zG_{\Gamma\setminus\Gamma_r} - \widetilde{K}_{z_r}) \le u - 1$, the multiplication of $\prod_{r=0}^{u-1}(zG_{\Gamma\setminus\Gamma_r} - \widetilde{K}_{z_r})$ exhibits a complexity of $O(l^6(n - |\Gamma|)^2)$. Since $\deg_x \widetilde{T}_{u,c} \le lw^3 + lw^2 - \sum_{r=u}^{\eta}|\Gamma_r|$ and $\deg_y \widetilde{T}_{u,c} \le w$, the multiplication to obtain $\widetilde{M}_{u,c}$ exhibits a complexity of $O(l^5 wn(n - |\Gamma|))$. A comparison of these terms reveals that obtaining $\widetilde{M}_{u,c}$ dominates the complexity of basis construction. Compared with the basis construction without ReT, which exhibits a complexity of $O(l^5 wn^2)$, the complexity is reduced by a factor of $|\Gamma|/n$. Based on (28), $\deg G$ is determined by Γ_r. Therefore, the necessity of maximizing $\deg G$ and $|\Gamma|$ is confirmed by the above analysis, which has been implemented in the improved ReT.

We further consider the basis reduction complexity. With the MS algorithm, the basis reduction complexity is mainly determined by the weighted degree of the maximum polynomial in the module basis [11]. The basis reduction complexity can be bounded by $l^3 w^3 \gamma^2$, where γ is the weighted degree of the maximum polynomial in the module basis. The weighted degree $\gamma = O(lwn)$, which is mainly determined by $\prod_{0\le b\le c-1}(y - B_{b,c}^{(u)})$. Therefore, whether ReT is applied or not, its asymptotic complexity is $O(l^5 w^2 n^3)$. However, $\gamma \le w(ln + 2lw^2 - \deg$

TABLE I
COMPLEXITY OF ASD OF THE (64, 47) HERMITIAN CODE WITH $l = 4$ AND SNR = 8 DB

Algorithm	ReT	Basis Construction	Basis Reduction	Root -finding	Total
ASD	–	6.14×10^5	5.51×10^5	7.61×10^3	1.17×10^6
ASD (ReT)	3.90×10^3	1.35×10^5	2.57×10^5	6.65×10^3	4.02×10^5
ASD (I-ReT)	1.11×10^4	4.89×10^4	9.14×10^4	1.15×10^3	1.53×10^5
ASD (I-ReT-ET)	9.25×10^3	1.59×10^4	3.98×10^4	5.17×10^2	6.86×10^4

TABLE II
COMPLEXITY OF ASD OF THE (512, 409) HERMITIAN CODE WITH $l = 3$ AND SNR = 8 DB

Algorithm	ReT	Basis Construction	Basis Reduction	Root -finding	Total
ASD	–	9.66×10^7	1.70×10^8	1.63×10^6	2.68×10^8
ASD (ReT)	3.54×10^5	2.73×10^7	8.19×10^7	1.59×10^6	1.11×10^8
ASD (I-ReT)	3.00×10^6	1.75×10^7	5.75×10^7	6.42×10^5	7.85×10^7
ASD (I-ReT-ET)	3.00×10^6	1.13×10^7	3.58×10^7	6.31×10^5	5.07×10^7

$G + lw - 2) + l(k + g - |\Gamma| - 1)$. Therefore, we can see that γ is reduced by $\deg G$ and $|\Gamma|$, resulting in a complexity reduction. This also confirms the necessity of maximizing $\deg G$ and $|\Gamma|$. In fact, the difference between ln and $\deg G$ is only at the level of one w factor, which is not significant for the codes of practical length. As can be seen from the numerical results in the later sections, although ReT cannot reduce the asymptotic complexity of basis reduction, it does reduce the empirically measured complexity by a certain proportion in simulations. It can be observed that when l is greater, $v^{(0)}_{a,w-1}$ may be greater accordingly, resulting in a greater $\deg G$ and a more significant complexity reduction. It can also be seen that for codes with a high code-rate, $\deg G$ and $|\Gamma|$ can be greater. Consequently, ReT is particularly advantageous for high-rate codes, which are also of greater practical interest.

VI. DECODING COMPLEXITY AND PERFORMANCE

This section presents numerical results for decoding complexity and performance. The decoding complexity is measured as the average number of finite field multiplications in decoding a codeword. The decoding performance is evaluated in frame error rate (FER). All simulations are conducted over the additive white Gaussian noise (AWGN) channel using binary phase shift keying (BPSK) modulation. The multiplicity matrix is generated using Algorithm A in [21] and parametrized by a predefined OLS l [22], [23]. The ASD with the ReT facilitated BR interpolation and that with the improved ReT facilitated BR interpolation are marked as ASD (ReT) and ASD (I-ReT), respectively. Moreover, the ASD that is facilitated by the improved ReT and early termination is marked as ASD (I-ReT-ET). The ML criterion of [33] and [34] is applied to assess the re-encoded codeword. The prototype ASD of [23] is used as a comparison benchmark.

Tables I and II present the complexity of the four ASD algorithms for the (64, 47) Hermitian code and the (512, 409) Hermitian code, respectively. Their complexities are measured under the maximum decoding OLS of $l = 4$ and $l = 3$, respectively. The simulation signal-to-noise ratio (SNR) is 8 dB. It can be seen that the BR interpolation, which consists of basis construction and reduction, dominates the decoding complexity. Although the ReT introduces extra computational cost, all the three ReT-facilitated ASD exhibit a lower complexity than their prototype ASD. By yielding a greater common factor for the basis polynomials, the improved ReT achieves a more significant complexity reduction. Since the decoding can be terminated earlier by assessing the re-encoded codeword, ASD (I-ReT-ET) further yields a complexity reduction. Note that the decoding can be terminated immediately after the ReT if the re-encoded codeword satisfies the ML criterion.

Tables III and IV show how the decoding complexity reduces at different SNRs for the two above mentioned codes. In both tables, the best performing ASD variant for each combination of l and SNR is highlighted in gray. It can be seen that under different SNRs, all three proposed schemes exhibit lower complexity than the prototype ASD. At low SNR, erasure decoding rarely succeeds in producing a valid re-encoded codeword, yet its computational cost remains. Consequently, the ASD (I-ReT) and ASD (I-ReT-ET) exhibit higher complexity than the ASD (ReT). As the SNR increases, erasure decoding becomes more likely to produce a codeword that satisfies the ML criterion. The ASD (I-ReT) and ASD (I-ReT-ET) therefore yield a lower complexity. However, early termination becomes effective only at sufficiently high SNR, leading to complexity reduction for ASD (I-ReT-ET). In this case, additional cost is still required to assess the re-encoded codeword using the ML criterion[5]. Hence, under these conditions, the ASD (I-ReT) is more efficient. As the SNR further increases, the ASD (I-ReT-ET) yields the lowest complexity because of the early termination. Therefore, the three proposed approaches are suitable for different channel conditions. It can also be seen that the ReT can deliver a more significant complexity reduction when the maximum decoding OLS l increases. This verifies the discussion in Section V-B.

Fig. 1 compares the decoding performance of the (64, 47) Hermitian code and the (15, 11) RS code, both of which are defined over $\mathbb{F}_{16}$. Due to its longer codeword length and inherently stronger error-correction capability, the Hermitian code outperforms the RS code, even with a smaller OLS l. Furthermore, compared with GS decoding, ASD demonstrates the advantage of effectively leveraging soft information. For instance, GS decoding with $m = l = 5$ incurs a complexity of 4.19×10^5. The ASD (BR) with only $l = 2$ achieves better performance at a lower complexity of 3.64×10^5. This demonstrates that exploiting soft information can enhance decoding performance without incurring a substantial compu-

[5]Please note that realizing early termination incurs additional computational cost that may not manifest in finite field multiplications. For the ML criterion of [33] and [34], using it would incur additional floating-point computational cost. For example, in Tables III and IV, at SNR = 6 dB, since early termination rarely occurs, ASD (I-ReT) and ASD (I-ReT-ET) exhibit the same count of finite-field multiplications.

TABLE III
ASD COMPLEXITY AT DIFFERENT SNRS IN DECODING THE (64, 47) HERMITIAN CODE

SNR (dB)	ASD		ASD (ReT)		ASD (I-ReT)		ASD (I-ReT-ET)	
	$l=2$	$l=4$	$l=2$	$l=4$	$l=2$	$l=4$	$l=2$	$l=4$
4	4.64×10^5	4.91×10^6	$\mathbf{2.68\times10^5}$	$\mathbf{2.81\times10^6}$	2.99×10^5	3.00×10^6	2.99×10^5	3.00×10^6
6	3.78×10^5	3.20×10^6	1.98×10^5	1.44×10^6	$\mathbf{1.81\times10^5}$	$\mathbf{1.03\times10^6}$	1.81×10^5	1.03×10^6
8	2.17×10^5	1.17×10^6	1.20×10^5	4.02×10^5	5.48×10^4	1.53×10^5	$\mathbf{2.76\times10^4}$	$\mathbf{6.86\times10^4}$
10	1.17×10^5	4.03×10^5	6.95×10^4	1.11×10^5	3.72×10^4	4.62×10^4	$\mathbf{1.06\times10^4}$	$\mathbf{1.06\times10^4}$

TABLE IV
ASD COMPLEXITY AT DIFFERENT SNRS IN DECODING THE (512, 409) HERMITIAN CODE

SNR (dB)	ASD		ASD (ReT)		ASD (I-ReT)		ASD (I-ReT-ET)	
	$l=2$	$l=3$	$l=2$	$l=3$	$l=2$	$l=3$	$l=2$	$l=3$
4	2.01×10^8	6.32×10^8	$\mathbf{1.43\times10^8}$	$\mathbf{4.48\times10^8}$	1.50×10^8	4.49×10^8	1.50×10^8	4.49×10^8
6	9.85×10^7	3.12×10^8	$\mathbf{5.25\times10^7}$	1.58×10^8	6.22×10^7	$\mathbf{1.57\times10^8}$	6.22×10^7	1.57×10^8
8	8.73×10^7	2.68×10^8	4.44×10^7	1.11×10^8	3.50×10^7	7.85×10^7	$\mathbf{1.25\times10^7}$	$\mathbf{5.07\times10^7}$
10	3.27×10^7	9.72×10^7	1.76×10^7	3.42×10^7	1.09×10^7	1.59×10^7	$\mathbf{5.24\times10^5}$	$\mathbf{5.24\times10^5}$

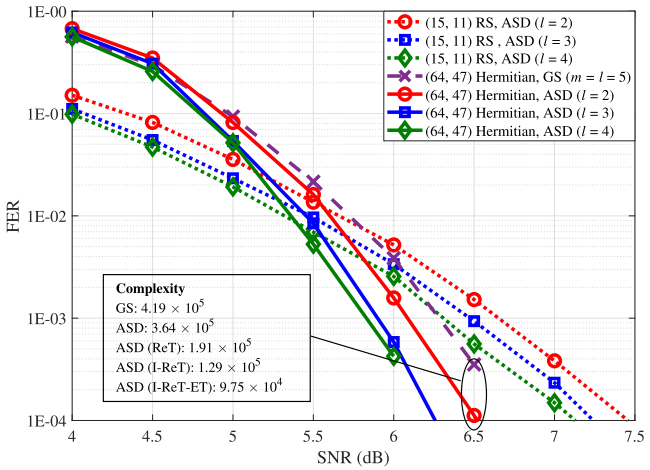

Fig. 1. Decoding performance comparison of the (15, 11) RS code and the (64, 47) Hermitian code.

tational cost. Building on this baseline, the ASD (ReT) and its improved variants achieve further complexity reduction.

The decoding performance advantage observed above is preserved in the proposed ReT-facilitated ASD due to the algebraic equivalence of the decoding. *Lemma 2* ensures that the ReT establishes a one-to-one correspondence between the minimal polynomials of the original and the transformed interpolation modules. This correspondence is then preserved through common factor extraction, as shown by *Lemma 4*. Consequently, the interpolation polynomial derived from the modified BR interpolation is algebraically equivalent to that of the prototype ASD. The estimated message polynomial can then be restored via equations (47) and (48), thus ensuring the overall decoding performance is preserved.

VII. CONCLUSION

This paper has proposed the ReT-facilitated ASD for Hermitian codes. The ReT has been derived using Lagrange interpolation polynomials over Hermitian function fields, along with appropriately chosen re-encoding points. With a designed maximum decoding OLS, the interpolation module basis has been formulated. It has been shown that the ReT results in the module basis polynomials sharing a common factor, which can be extracted to reduce the BR interpolation complexity. Furthermore, an improved ReT has been proposed to yield a common factor of higher degree, further reducing the interpolation complexity. Theoretical analysis and numerical results have shown that both ReT variants can effectively facilitate ASD, enabling advanced decoding of Hermitian codes.

APPENDIX
NOTATION

The primary notation used in this paper is summarized below, grouped by conceptual area.

Finite Fields

- q: Size (cardinality) of the finite field.
- $\mathbb{F}_q$: The finite field of q elements. Its elements are denoted $\sigma_0, \sigma_1, \ldots, \sigma_{q-1}$.
- $\mathbb{F}_q[X, Y]$: The ring of bivariate polynomials over $\mathbb{F}_q$.

Hermitian Curve

- w: The positive integer given by $\sqrt{q}$.
- H_w: An affine Hermitian curve defined over $\mathbb{F}_q$ given by (1).
- g: Genus of H_w, given by $g = w(w-1)/2$.
- $P_j = (x_j, y_j)$: An affine point of H_w (there are w^3 in total).
- P_∞: The point at infinity of H_w.
- $\mathcal{P}$: The set of affine points.
- $\mathcal{R}$: The coordinate ring of $\mathcal{H}$ defined as in (2).
- x, y: The residue classes of X and Y, respectively.
- $\mathcal{L}_w$: The pole basis of the Hermitian curve H_w.
- $\phi_a(x, y)$: A monomial forming the pole basis $\mathcal{L}_w$, of the form $x^{i_x}y^{i_y}$ for integers $0 \le i_x \le w$ and $i_y \ge 0$.
- $\nu_P(S)$: The order of a nonzero polynomial $S \in \mathcal{R}$ at a rational point P.

- $\psi_{P,\nu}(x,y)$: The zero basis at a rational point P, parameterized by $\nu = \lambda + (w+1)\delta$ for nonnegative integers λ, δ and defined as in (3).

Encoding

- n: Codeword length. For Hermitian codes, $n = w^3$.
- k: Dimension of the code.
- $\mathcal{L}(\mu P_\infty)$: The Riemann-Roch space used to define Hermitian codes, where $\mu = k + g - 1$.
- $\underline{f} = (f_0, f_1, \ldots, f_{k-1})$: A message vector in $\mathbb{F}_q^k$ to be encoded.
- $f(x,y)$: The message polynomial corresponding to $\underline{f}$, expressed as in (4).
- $\underline{c} = (c_0, c_1, \ldots, c_{n-1})$: The codeword obtained by evaluating $f(x,y)$ as in (5).

GS Decoding

- $\mathcal{R}[z]$: The polynomial ring in variable z over the coordinate ring $\mathcal{R}$.
- $\phi_a z^b$: A monomial in $\mathcal{R}[z]$, where $\phi_a \in \mathcal{L}_w$ and $b \geq 0$.
- $\deg_{1,w_z}(\cdot)$: The $(1, w_z)$-weighted degree defined as in (7).
- w_z: The weight parameter for the variable z.
- m: Multiplicity.

ASD

- l: Designed maximum decoding OLS.
- $\underline{r} = (r_0, r_1, \ldots, r_{n-1})$: The channel output.
- $\mathbf{\Pi}$: The $q \times n$ reliability matrix with entries $\pi_{ij} = \Pr(r_j \mid c_j = \sigma_i)$.
- $\mathbf{M}$: The $q \times n$ multiplicity matrix with nonnegative integer entries m_{ij}.
- i_j: The index satisfying $\sigma_{i_j} = c_j$, identifying the transmitted symbol at position j.
- $s_{\mathbf{M}}(\underline{c})$: The codeword score of $\underline{c}$ with respect to $\mathbf{M}$ defined as in (9).
- $\mathrm{mult}_{(P_j,\sigma_i)}(Q)$: The multiplicity of a polynomial Q at the interpolation point (P_j, σ_i).
- $\mathcal{I}_{\mathbf{M}}$: The ideal determined by the multiplicity matrix $\mathbf{M}$, defined as in (10).
- $\mathcal{R}[z]_l$: The set of polynomials in $\mathcal{R}[z]$ with z-degree at most l.
- $\mathcal{I}_{\mathbf{M},l}$: The interpolation module defined as in (11).

BR Interpolation

- η: The maximum column sum of the multiplicity matrix $\mathbf{M}$, defined as $\eta = \max_j \sum_{i=0}^{q-1} m_{ij}$.
- $\mathbf{M}^{(u)}$: The u-th intermediate multiplicity matrix with entries $m_{ij}^{(u)}$.
- $i_j^{(u)}$: The row index that attains the maximum entry in column j of $\mathbf{M}^{(u)}$, i.e., $i_j^{(u)} = \arg\max_i \{m_{ij}^{(u)}\}$.
- $\mathbb{A}(\mathcal{J})$: The set of distinct x-coordinates from the points $\{P_j \mid j \in \mathcal{J}\}$.
- $\mathbb{B}_\sigma(\mathcal{J})$: For a given x-coordinate σ, the set of corresponding y-coordinates from points in $\{P_j \mid j \in \mathcal{J}\}$.
- $\mathbb{C}(j)$: The set of indices j' such that $P_{j'}$ shares the same x-coordinate with P_j (i.e., $x_{j'} = x_j$).
- $\mathbb{S}(\mathcal{J})$: The set of indices $j \in \mathcal{J}$ for which $|\mathbb{B}_{x_j}(\mathcal{J})| = w$.
- $L_{\mathcal{J},j}(x,y)$: The Lagrange interpolation polynomial defined as in (13).
- $K_{\omega^{(u)}}(x,y)$: A polynomial defined as in (14).
- $\mathfrak{m}_j^{(u)}$: The maximum entry in column j of $\mathbf{M}^{(u)}$, i.e., $\mathfrak{m}_j^{(u)} = \max_i \{m_{ij}^{(u)}\}$.
- $\mathcal{E}_{\mathbf{M}^{(u)}}$: The ideal in $\mathcal{R}$ defined by $\mathfrak{m}_j^{(u)}$, i.e., $\{S \in \mathcal{R} \mid \nu_{P_j}(S) \geq \mathfrak{m}_j^{(u)}, \forall j\}$.
- $v_{a,b}^{(u)}$: Reassigned multiplicity, s.t. $v_{a,b}^{(u)} = \mathfrak{m}_{j^{(u)}}$ for the corresponding index j with $P_j \mapsto Pa, b$.
- $B_{b,c}^{(u)}$: A polynomial in $\mathbb{F}_q[x]$ satisfying $\nu_{P_{a,b}}(y - B_{b,c}^{(u)}) \geq v_{a,b}^{(u)} - v_{a,c}^{(u)}$ for relevant indices a, b, c.
- $T_{u,c}(x,y)$: A generator polynomial for the ideal $\mathcal{E}_{\mathbf{M}^{(u)}}$, defined as in (16).
- $\mathcal{M} = \{M_{u,c}\}$: The module basis as defined in (17).

ReT

Notation with a hat $\widehat{(\cdot)}$ denotes variables related to the re-encoding transform using $\underline{h}$.

- Γ: The index set selecting the points used for re-encoding.
- $\boldsymbol{P}_\Gamma$: The set of re-encoding points defined as $\boldsymbol{P}_\Gamma = \{(P_j, \omega_j^{(0)}) \mid j \in \Gamma\}$.
- $K_\Gamma(x,y)$: The re-encoding polynomial defined as in (18).
- $\underline{h}$: The re-encoded codeword defined as in (19).
- $\widehat{\mathbf{M}}$: The $q \times n$ multiplicity matrix (from $\mathbf{M}$ via (20), (21)), with entries $\widehat{m}_{ij} \in \mathbb{Z}_0^+$.
- $\widehat{\mathbf{M}}^{(u)}$: The u-th intermediate multiplicity matrix with respect to $\widehat{\mathbf{M}}$, whose entries are $\widehat{m}_{ij}^{(u)}$.
- $\widehat{\mathfrak{m}}_j^{(u)}$: The maximum entry in column j of $\widehat{\mathbf{M}}^{(u)}$, i.e., $\widehat{\mathfrak{m}}_j^{(u)} = \max_i \{\widehat{m}_{ij}^{(u)}\}$.
- $\mathcal{E}_{\widehat{\mathbf{M}}^{(u)}}$: The ideal in $\mathcal{R}$ defined by $\widehat{\mathfrak{m}}_j^{(u)}$, i.e., $\{S \in \mathcal{R} \mid \nu_{P_j}(S) \geq \widehat{\mathfrak{m}}_j^{(u)}, \forall j\}$.
- $\mathcal{I}_{\widehat{\mathbf{M}},l}$: The interpolation module with respect to $\widehat{\mathbf{M}}$.
- $\widehat{i}_j^{(u)}$: The row index of the maximum entry in column j of $\widehat{\mathbf{M}}^{(u)}$, i.e., $\widehat{i}_j^{(u)} = \arg\max_i \{\widehat{m}_{ij}^{(u)}\}$.
- $z_j^{(u)}$: The finite field element corresponding to $\widehat{i}_j^{(u)}$, defined as $z_j^{(u)} = \sigma_{\widehat{i}_j^{(u)}}$.
- $\underline{z}^{(u)}$: The vector $(z_0^{(u)}, z_1^{(u)}, \ldots, z_{n-1}^{(u)})$.
- $K_{\underline{z}^{(r)}}(x,y)$: A polynomial defined as in (30).
- $\widehat{T}_{u,c}$: A basis polynomial for the ideal $\mathcal{E}_{\widehat{\mathbf{M}}^{(u)}}$.
- $\widehat{\mathcal{M}} = \{\widehat{M}_{u,c}\}$: The module basis as defined in (29) and (31).
- Γ_u: A subset of the re-encoding index set Γ defined as in (25).
- $G(x), G_\Gamma(x), G_{\Gamma_u}(x), G_{\Gamma \setminus \Gamma_u}(x)$: The common factor polynomials defined as in (23), (24), (26) and (27), respectively.

Modified BR Interpolation

Notation with a tilde $\widetilde{(\cdot)}$ denotes concepts in the module isomorphism after extracting the common factor.

- Φ: The isomorphic module with respect to $\mathcal{I}_{\widehat{\mathbf{M}},l}$.
- $\widetilde{T}_{u,c}(x,y)$: A polynomial defined as in (44).
- $\widetilde{K}_{\underline{z}^{(r)}}$: A polynomial defined as in (45).
- $\widetilde{\mathcal{M}} = \{\widetilde{M}_{u,c}\}$: The module basis isomorphism defined as in (43).

Others

- $[a]$: For a nonnegative integer a, the set $\{0, 1, \ldots, a-1\}$.
- $f_{\mathrm{est}}(x,y)$: The estimated message polynomial.
- γ: The weighted degree of the maximum polynomial in the module basis.

REFERENCES

[1] V. Goppa, "Codes associated with divisors," *Problemy Peredachi Informatsii*, vol. 13, no. 1, pp. 33–39, 1977.

[2] J. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inf. Theory*, vol. IT-15, no. 1, pp. 122–127, Jan. 1969.
[3] S. Sakata, J. Justesen, Y. Madelung, H. E. Jensen, and T. Hoholdt, "Fast decoding of algebraic-geometric codes up to the designed minimum distance," *IEEE Trans. Inf. Theory*, vol. 41, no. 6, pp. 1672–1677, Nov. 1995.
[4] V. Guruswami and M. Sudan, "Improved decoding of Reed–Solomon and algebraic-geometric codes," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 1757–1767, Sep. 1999.
[5] R. Kötter, "On algebraic decoding of algebraic-geometric and cyclic codes," Ph.D thesis, Dept. Elect. Eng., Univ. Linköping, Linköping, Sweden, 1996.
[6] M. Alekhnovich, "Linear diophantine equations over polynomials and soft decoding of Reed–Solomon codes," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2257–2265, Jun. 2005.
[7] K. Lee and M. O'Sullivan, "List decoding of Reed–Solomon codes from a Gröbner basis perspective," *J. Symb. Comput.*, vol. 43, no. 9, pp. 645–658, Sep. 2008.
[8] K. Lee and M. E. O'Sullivan, "List decoding of Hermitian codes using Gröbner bases," *J. Symbolic Comput.*, vol. 44, no. 12, pp. 1662–1675, Dec. 2009.
[9] P. Beelen and K. Brander, "Efficient list decoding of a class of algebraic-geometry codes," *Adv. Math. Commun.*, vol. 4, no. 4, pp. 485–518, Nov. 2010.
[10] Y. Wan, L. Chen, and F. Zhang, "Guruswami-sudan decoding of elliptic codes through module basis reduction," *IEEE Trans. Inf. Theory*, vol. 67, no. 11, pp. 7197–7209, Nov. 2021.
[11] J. S. R. Nielsen and P. Beelen, "Sub-quadratic decoding of one-point Hermitian codes," *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3225–3240, Jun. 2015.
[12] P. Beelen, J. Rosenkilde, and G. Solomatov, "Fast decoding of AG codes," *IEEE Trans. Inf. Theory*, vol. 68, no. 11, pp. 7215–7232, Nov. 2022.
[13] P. Beelen and V. Neiger, "Faster list decoding of AG codes," *IEEE Trans. Inf. Theory*, vol. 71, no. 5, pp. 3397–3408, May 2025.
[14] R. Koetter, J. Ma, and A. Vardy, "The re-encoding transformation in algebraic list-decoding of Reed–Solomon codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 633–647, Feb. 2011.
[15] Y. Wan, J. Xing, Y. Huang, T. Wu, B. Bai, and G. Zhang, "The re-encoding transform in algebraic list decoding of algebraic geometric codes," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Taipei, Taiwan, Jun. 2023, pp. 19–24.
[16] J. Bellorado and A. Kavcic, "Low-complexity soft-decoding algorithms for Reed–Solomon codes—Part I: An algebraic soft-in hard-out chase decoder," *IEEE Trans. Inf. Theory*, vol. 56, no. 3, pp. 945–959, Mar. 2010.
[17] Y. Wan, J. Liang, L. Chen, and F. Zhang, "Low-complexity chase decoding of elliptic codes," *IEEE Trans. Commun.*, vol. 73, no. 10, pp. 8574–8586, Oct. 2025.
[18] J. Zhao and L. Chen, "Algebraic chase decoding of elliptic codes with generalized re-encoding transform and improved root-finding," *IEEE Trans. Inf. Theory*, vol. 71, no. 7, pp. 5089–5108, Jul. 2025.
[19] S. Wu, L. Chen, and M. Johnston, "Interpolation-based low-complexity chase decoding algorithms for Hermitian codes," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1376–1385, Apr. 2018.
[20] J. Liang, J. Zhao, and L. Chen, "Low-complexity chase decoding of Hermitian codes with improved interpolation and root-finding," *IEEE Trans. Commun.*, vol. 73, no. 8, pp. 5509–5522, Aug. 2025.
[21] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed–Solomon codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 11, pp. 2809–2825, Nov. 2003.
[22] L. Chen, R. Carrasco, and M. Johnston, "Soft-decision list decoding of Hermitian codes," *IEEE Trans. Commun.*, vol. 57, no. 8, pp. 2169–2176, Aug. 2009.
[23] K. Lee and M. E. O'Sullivan, "Algebraic soft-decision decoding of Hermitian codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2587–2600, Jun. 2010.
[24] Y. Wan, L. Chen, and F. Zhang, "Algebraic soft decoding of elliptic codes," *IEEE Trans. Commun.*, vol. 70, no. 3, pp. 1522–1534, Mar. 2022.
[25] J. Liang and L. Chen, "Algebraic soft decoding of Hermitian codes with re-encoding transform," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2025, pp. 1–6.
[26] I. F. Blake, C. Heegard, T. Høholdt, and V. K. Wei, "Algebraic-geometry codes," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2596–2618, Oct. 1998.
[27] R. Nielsen, "List decoding of linear block codes," Ph.D thesis, Dept. Math., Tech. Univ. Denmark, Lyngby, Denmark, 2001.
[28] H. Stichtenoth, *Algebraic Function Fields and Codes*, 2nd ed., New York, NY, USA: Springer, 2009.
[29] T. Mulders and A. Storjohann, "On lattice reduction for polynomial matrices," *J. Symbolic Comput.*, vol. 35, no. 4, pp. 377–401, Apr. 2003.
[30] R. M. Roth and G. Ruckenstein, "Efficient decoding of Reed–Solomon codes beyond half the minimum distance," *IEEE Trans. Inf. Theory*, vol. 46, no. 1, pp. 246–257, Jan. 2000.
[31] X.-W. Wu and P. H. Siegel, "Efficient root-finding algorithm with application to list decoding of algebraic-geometric codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 6, pp. 2579–2587, Sep. 2001.
[32] S. Li, S. Liu, L. Ma, Y. Wan, and C. Xing, "Encoding of algebraic geometry codes with quasi-linear complexity O(NlogN)," *IEEE Trans. Inf. Theory*, vol. 71, no. 7, pp. 5013–5026, Jul. 2025.
[33] T. Kaneko, T. Nishijima, H. Inazumi, and S. Hirasawa, "An efficient maximum-likelihood-decoding algorithm for linear block codes with algebraic decoder," *IEEE Trans. Inf. Theory*, vol. 40, no. 2, pp. 320–327, Mar. 1994.
[34] X. Ma and S. Tang, "An efficient maximum-likelihood-decoding algorithm for linear block codes with algebraic decoder," *IEEE Trans. Inf. Theory*, vol. 58, no. 6, p. 4073, Jun. 2012.
[35] M. Safieh, D. N. Bailon, and J. Freudenberger, "An acceptance criterion for hybrid algebraic and soft-input decoding," in *Proc. 24th Int. Conf. Inf. Technol. (IT)*, Zabljak, Montenegro, Feb. 2020, pp. 1–5.

Jiwei Liang (Student Member, IEEE) received the B.E. degree in electronic engineering from Southeast University, Nanjing, China, in 2013, and the M.S. degree in communication engineering from Guilin University of Electronic Technology, Guilin, China, in 2017. He is currently pursuing the Ph.D. degree in information and communication engineering with Sun Yat-sen University, Guangzhou, China. In 2013, he was with Avonaco Communication Systems Company Ltd., Suzhou, China, as a DSP Engineer, where he was involved in the development of multimedia communication systems. From 2017 to 2021, he was with Allwinner Technology Company Ltd., Zhuhai, China, as a firmware engineer, where he was involved in the development of WLAN/bluetooth combo chip. His research interests include channel coding and data communications.

Li Chen (Senior Member, IEEE) received the B.Sc. degree in applied physics from Jinan University, China, in 2003, and the M.Sc. degree in communications and signal processing and the Ph.D. degree in communications engineering from Newcastle University, U.K., in 2004 and 2008, respectively. From 2007 to 2010, he was a Research Associate with Newcastle University. In 2010, he returned China, as a Lecturer with the School of Information Science and Technology, Sun Yat-sen University, Guangzhou. From 2011 to 2012, he was a Visiting Researcher with the Institute of Network Coding, The Chinese University of Hong Kong, where he was an Associate Professor and a Professor from 2011 and 2016. Since 2013, he has been the Associate Head of the Department of Electronic and Communication Engineering (ECE). From July 2015 to October 2015, he was a Visitor with the Institute of Communications Engineering, Ulm University, Germany. From October 2015 to June 2016, he was a Visiting Associate Professor with the Department of Electrical Engineering, University of Notre Dame, USA. From 2017 to 2020, he was the Deputy Dean of the School of Electronics and Communication Engineering. His research interests include information theory, error-correction codes, and data communications. He is a Senior Member of Chinese Institute of Electronics (CIE). He is a member of the IEEE Information Theory Society Board of Governors and its External Nomination Committee and the Chair of its Conference Committee. He is also the Chair of the IEEE Information Theory Society Guangzhou Chapter. He has been organizing several international conferences and workshops, including the 2018 IEEE Information Theory Workshop (ITW) at Guangzhou and the 2022 IEEE East Asian School of Information Theory (EASIT) at Shenzhen, for which he is the General Co-Chair. He is also the TPC Co-Chair of the 2022 IEEE/CIC International Conference on Communications in China (ICCC) at Foshan. He was an Associate Editor of IEEE TRANSACTIONS ON COMMUNICATIONS and is currently an Associate Editor of IEEE TRANSACTIONS ON INFORMATION THEORY.