Unique Decoding of Reed–Solomon and Related Codes for Semi-Adversarial Errors

Joshua Brakensiek
Department of Electrical Engineering and Computer Sciences,
University of California, Berkeley, CA, USA

Yeyuan Chen
Department of EECS, University of Michigan, Ann Arbor, MI, USA

Manik Dhar
Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, USA

Zihan Zhang
Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, USA

Abstract

Motivated by recent developments in coding theory, particular in list-decoding, we introduce a new error model which we call **semi-adversarial errors**. This error model bridges between fully random errors and fully adversarial errors by allowing some symbols of a message to be corrupted by an adversary while others are replaced with uniformly random symbols.

As our main quest, we seek to understand optimal efficient **unique** decoding algorithms in the semi-adversarial model. For interleaved Reed–Solomon (IRS), folded Reed–Solomon (FRS) and univariate multiplicity codes, we design decoding algorithms running in near-linear time for most mixtures of random and adversarial errors. Our analysis matches the information-theoretic optimum for semi-adversarial errors.

Our algorithm for interleaved Reed–Solomon codes is an improved implementation of the decoding algorithm by Bleichenbacher–Kiayias–Yung (BKY) for fully random errors. We use a novel monomial-tracking technique to analyze its performance in this new semi-adversarial errors. Inspired by the BKY algorithm, we use novel interpolations to extend our approach to the settings of folded Reed–Solomon and multiplicity codes, resulting in fast algorithms for unique decoding against semi-adversarial errors. Our new decoders for FRS and multiplicity codes replace the sophisticated root-finding step in traditional algorithms, such as the Guruswami–Wang algorithm, with a straightforward polynomial long division. Analysis of these algorithms requires more robust monomial-tracking arguments than IRS codes.

2012 ACM Subject Classification Mathematics of computing → Coding theory; Computing methodologies → Algebraic algorithms

Keywords and phrases coding theory, interleaved codes, Reed-Solomon codes, semi-random models, unique decoding

Digital Object Identifier 10.4230/LIPIcs.ICALP.2026.43

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2504.10399`

Funding *Joshua Brakensiek*: Supported in part by the Simons Investigator award of Venkatesan Guruswami and NSF grants CCF-2211972 and DMS-2503280.
Yeyuan Chen: Partially supported by the National Science Foundation grant No. CCF-2236931.
Zihan Zhang: Partially supported by the National Science Foundation grant No. CCF-2440926.

Acknowledgements We thank Sivakanth Gopi and Venkatesan Guruswami for many helpful discussions and much encouragement while writing this paper. We thank Matteo Abbondati, Eleonora Guerrini, and Romain Lebreton for taking the time to explain to us the contributions of [1–3, 24]. Zihan Zhang also thanks Gabrielle Beck for some helpful discussions. We thank Rishabh Kothary for some valuable corrections to the paper.

53rd International Colloquium on Automata, Languages, and Programming (ICALP 2026).
Editors: Sayan Bhattacharya, Danupon Nanongkai, Michael Benedikt, and Gabriele Puppis;
Article No. 43; pp. 43:1–43:19

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

The primary goal of this investigation is to better understand the error-correcting properties of Reed–Solomon (RS) codes [45], the most fundamental class of algebraic error-correcting codes. To define a Reed–Solomon code, we consider a finite field $\mathbb{F}_q$ as well as distinct evaluation points $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$. An $[n, k]$ RS code over $\mathbb{F}_q$ is then

$$\mathsf{RS}_{n,k,q}(\alpha_1, \ldots, \alpha_n) := \big\{\big(f(\alpha_1), \ldots, f(\alpha_n)\big) \,:\, f(X) \in \mathbb{F}_q[X],\ \deg(f) < k\big\} \subseteq \mathbb{F}_q^n.$$

One of the most fundamental questions one can ask about Reed–Solomon codes is their decodability. In particular, consider a protocol in which a sender picks $\vec{c} \in \mathsf{RS}_{n,k,q}(\alpha_1, \ldots, \alpha_n)$ and transmits $\vec{c}$ across a channel, which is received as a corrupted message $\vec{y} \in \mathbb{F}_q^n$. We seek to understand under what circumstances $\vec{c}$ can be recovered from $\vec{y}$. For instance, the Berlekamp–Welch decoder [55] can efficiently recover $\vec{c}$ if and only if the Hamming distance (i.e., the number of distinct symbols) between $\vec{c}$ and $\vec{y}$ is at most $(n-k)/2$. This algorithm is information-theoretically optimal because $\mathsf{RS}_{n,k,q}$ has distance $n-k+1$, i.e., it attains the *Singleton bound* [51] and thus is an *MDS code*.

However, there are many other questions one can ask about the structure of Reed–Solomon codes. For example, what if the corruptions are sampled uniformly at random? Or what if the number of corruptions exceeds the unique-decoding radius of $(n-k)/2$, and we can only decode to a list? For decoding problems like these, a central open problem is to find *efficient* decoding algorithms which match the information-theoretic optimum for these problems. For instance, the Guruswami-Sudan [29] can efficiently list-decode any Reed–Solomon code up to the *Johnson radius* of $1 - \sqrt{R}$, where $R := k/n$ is the rate of the Reed–Solomon code. However, a recent line of work has surprisingly established that most Reed–Solomon codes can be list-decoded up to the generalized Singleton bound, which asymptotically tends to the channel capacity of $1 - R$ [5, 16, 26, 50].

Although closing the gap between such combinatorial and algorithmic results currently seems out of reach, we note that an analogous result has been known for decades in the *random* error setting. More precisely, for a special class of Reed–Solomon codes known as *subfield evaluation* RS codes or *interleaved* RS codes, the result of Bleichenbacher, Kiayias, and Yung [13] proved that for an s-interleaved RS code one can efficiently uniquely decode $\frac{s}{s+1}(n-k)$ random errors. In particular, as the interleaving grows, one can efficiently uniquely decode random errors up to a radius approaching capacity.[1] Given the success of the BKY algorithm in the random error model, is there any hope of generalizing it to improve our understanding of the adversarial model?

The main mission of this paper is to lay the groundwork of a pathway for translating results into the random error model back into a setting with adversarial errors. As a primary contribution, we introduce a new error model, which we call the *semi-adversarial* error model which hybridizes the fully random and fully adversarial error regimes. Studying such a model allows us to integrate approaches used in the decoding literature for fully adversarial and fully random errors, with the benefit that we are able to understand *tight* tradeoffs between information-theoretic bounds and fast (i.e., near-linear time) decoding algorithms.

[1] Of note, a contemporaneous algorithm by Coppersmith and Sudan [19] is *believed* to also attain capacity, but its current analysis has a decoding radius of only $1 - 2R$ in the limit. We discuss this open problem in Section 2.

1.1 The Semi-Adversarial Error Model

We now proceed to define the error models (i.e., channels) we examine in this paper. Formally, given a code $C \subseteq \Sigma^n$, we define a *channel* to be a function Ψ which takes as input a codeword $\vec{c} \in C$ and returns as output a probability distribution $\Psi(\vec{c})$ over messages $\vec{y} \in \Sigma^n$. We let $\Delta(\Sigma^n)$ denote the set of probability distributions over Σ^n, so a channel can be viewed as a function $\Psi : C \to \Delta(\Sigma^n)$. We then define an *error model* $\mathcal{E}$ to be a set of channels. We say that an algorithm $\mathcal{A}$ can decode $\mathcal{E}$ with probability p, if for every $\vec{c} \in C, \Psi \in \mathcal{E}$, the algorithm $\mathcal{A}$ when given input a sample $\vec{y} \sim \Psi(\vec{c})$ outputs $\vec{c}$ with probability at least p.

For each $i \in [n]$, we let y_i denote the i-th symbol of $\vec{y}$. Given $I \subseteq [n]$, we let $\Psi(\vec{c})|_I$ denote the marginal distribution where we restrict to the coordinates of the output string indexed by I. We start by defining the well-studied adversarial and random error models.

▶ **Definition 1** (Adversarial Errors). *We say that a channel $\Psi : C \to \Delta(\Sigma^n)$ has at most e adversarial errors if for every transmitted string $\vec{c} \in C$, every $\vec{y} \in \operatorname{supp} \Psi(\vec{c})$ (i.e., the strings with nonzero probability) have Hamming distance at most e from $\vec{c}$.*

▶ **Definition 2** (Random Errors). *We say that a channel $\Psi : C \to \Delta(\Sigma^n)$ has e random errors if for every $\vec{c}$, there exists $I \subseteq [n]$ with $|I| \geq n - e$ such that $\Psi(\vec{c})$ has the following properties.*
(1) *For every $\vec{y} \in \operatorname{supp} \Psi(\vec{c})$, $c_i = y_i$ for all $i \in I$.*
(2) *$\Psi(\vec{c})|_{[n]\setminus I}$ is the uniform distribution on $\Sigma^{[n]\setminus I}$.*

▶ Remark 3. The ability for an algorithm to decode a family of channels with success probability p is equivalent to the algorithm being able to decoding with probability p any convex combination of these channels (i.e., sample a random channel then sample the output of that channel).

▶ Remark 4. In our random error model, we assume that the set I of non-errors can be adversarially chosen. Note this coincides with the random error model of Coppersmith–Sudan [19], but deviates from Bleichenbacher et al. [13] where the set I is chosen uniformly at random. However, as we shall see, the decoding algorithm considered in [13] applies equally well in this stronger error model.

▶ Remark 5. To simplify the notation later in the paper, we often keep the choice of Ψ implicit and exclusively refer to the transmitted codeword $\vec{c}$ and the sampled codeword $\vec{y} \sim \Psi(\vec{c})$. For example, we will say that $\vec{y}$ is received with at most e random errors if the implicit channel Ψ comes from the at most e random errors model.

Note the random error channel is weaker than the corresponding adversarial error channel in the sense that for the same number of errors, the random error channel can be viewed as a special case of the adversarial error channel where the adversary chooses to corrupt each symbol uniformly at random. However, we have a tighter understanding of the random error model – the unique decoding radius in the random error model (approximately) coincidences with the capacity of the channel. This was proven combinatorially by Rudra and Uurtamo [47] who showed that any MDS code (i.e., code attaining the Singleton bound) is uniquely decodable up to $(1 - R - \varepsilon)n$ random errors.[2]

For any alphabet Σ, radius $e \geq 0$ and length-n string $\vec{y} \in \Sigma^n$, we define the Hamming ball $\mathcal{B}(\vec{y}, e)$ with center $\vec{y}$ as the set of all length-n strings that have Hamming distance at most e from $\vec{y}$. That is, they differ with $\vec{y}$ on at most e positions.

[2] This is not restricted to MDS codes. Generally, for a code with relative distance δ, it is possible to uniquely decode it from $(\delta - \varepsilon)n$ random errors. For simplicity we state the result according to MDS codes here.

▶ **Proposition 6** ([47, Theorem 1]). *Consider parameters $0 < \varepsilon, R < 1$, $n > \Omega(1/\varepsilon)$, any alphabet Σ with size at least $2^{\Omega(1/\varepsilon)}$, and any MDS code $\mathcal{C} \subseteq \Sigma^n$ with rate R. For any codeword $\vec{c} \in \mathcal{C}$, let $\vec{y}$ denote the received word after applying $e = (1 - R - \varepsilon)n$ random errors to $\vec{c}$. Then, with probability at least $1 - |\Sigma|^{-\Omega(\varepsilon n)}$, we have $\mathcal{B}(\vec{y}, e) \cap \mathcal{C} = \{\vec{c}\}$.*

However, if we inspect the proof of Rudra and Uurtamo [47], we can observe that, in fact, unique-decoding is still possible even when the errors are a mixture of adversarial errors and random errors, which implies similar unique-decoding bounds (and as we shall soon discuss, corresponding algorithms) could be further generalized to stronger error models. This motivates us to consider a partial-adversarial/partial-random error model, which we call the semi-adversarial error model.

1.1.1 Semi-Adversarial Errors

We now present the primary error model we study in this paper: the semi-adversarial error model. It particular, it can be viewed as a *composition* of the channels defined in the adversarial and random error models.

▶ **Definition 7** (Semi-Adversarial Errors). *We say that a channel $\Psi : C \to \Delta(\Sigma^n)$ has (e_0, e)-semi-adversarial errors (where $0 \le e_0 \le e$) if for every $\vec{c} \in \Sigma^n$ there exist a set of non-adversarial indices $I \subseteq [n]$ with $|I| = n - e_0$, a set of agreement indices $J \subseteq I$ with $|J| \ge n - e$, and an adversarial corruption string $\vec{z} \in \Sigma^{n-|I|}$ with the following properties.*

(Agreement indices uncorrupted): *For every $\vec{y} \in \operatorname{supp} \Psi(\vec{c})$, $\vec{c}|_J = \vec{y}|_J$.*

(Adversarial corruptions applied) *For every $\vec{y} \in \operatorname{supp} \Psi(\vec{c})$, $\vec{y}|_{[n]\setminus I} = \vec{z}$.*

(Random errors after adversarial corruptions): *The distribution $\Psi(\vec{c})|_{I\setminus J}$ is uniform over $\Sigma^{|I\setminus J|}$.*

▶ Remark 8. In our analysis of decoding algorithms for semi-adversarial errors, we make use of some more fine-grained notation. In particular, for the given $I \subseteq [n]$ with $|I| = n - e_0$ we assume that the "true" adversarial errors occur in some subset $K \subseteq [n]\setminus I$ with $e_A := |K| \le e_0$. Formally, we define K such that for any position $j \in [n] \setminus I$, $j \in K$ iff $y_j \ne c_j$. Note that it may be the case $K \ne [n] \setminus I$, as some of the "corruptions" $\vec{z}$ can equal the corresponding coordinates in the original coordinates $\vec{c}$. We also often use the notation $K' := I \setminus J$ with $e_R := |K'|$ to denote the coordinates for which random errors are applied. We then let $\overline{e} := e_A + e_R \le e$ denote the number of positions which are truly corrupted during transmission.

▶ Remark 9. We note that (e, e)-semi-adversarial errors are equivalent to exactly e adversarial errors, while $(0, e)$-semi-adversarial errors are equivalent to e random errors.

▶ Remark 10 (Comparison to recent and concurrent work.). A recent paper [24] has studied hybrid error models in connection to the closely related problem of rational function decoding. Some follow-up papers [1, 3] deal with the same problem over number fields as opposed to function fields. The stated results in [24] are for semi-adversarial errors where the adversarial errors are for *valuation errors* which do not exist in the IRS case so the statement does not directly imply results in our setting. The results for number fields (like $\mathbb{Q}$) in [3] do work for any kind of adversarial errors, but this work was done concurrently and independently of us. A later work [2] generalizes techniques of [3, 24] and gives results for IRS codes in the semi-adversarial model. Although [2] was posted later than our work, it is developed independently. We compare their results with ours.

- For decoding (e_0, e) semi-adversarial errors for IRS codes, [2] requires $e_0 \le n - k - e - (e - e_0)/s$, while our bound $e_0 \le n - k - e$ is better and actually optimal. This difference could be ignored if one allows s being arbitrarily large, with the cost of larger alphabet.
- Unlike our work, [2] does not include results for folded RS codes and multiplicity codes, but they can handle a family of interleaved rational function codes that we do not consider.
- The success probability of decoding algorithms in [2] is larger than ours, but both two papers achieve $1 - o_n(1)$ success probability when the field size $q = \omega(n)$.

For any number of total errors e, when the number e_0 of adversarial errors gets larger, the task of unique-decoding (with high success probability) gets more difficult. As our first nontrivial observation about the semi-adversarial error model, we show that in any MDS code the number e_0 of adversarial errors must be bounded by

$$e_0 \le \min(e, (1 - R)n - e).$$

In particular, to uniquely decode e errors in the semi-adversarial model, the maximum number of adversarial errors that could occur $e_0 = \min(e, n - k - e)$. Our formal result is as follows.

▶ **Proposition 11.** *For any MDS code $\mathcal{C} \subseteq \Sigma^n$ over alphabet Σ with rate R, block length n, $e_0 > (1 - R)n - e$, there exists some word $\vec{z} \in \Sigma^n, |\mathcal{B}(\vec{z}, e_0) \cap \mathcal{C}| \ge 1$ and $K \subseteq [n], |K| = e - e_0$ such that any received word $\vec{y}$ that differs with $\vec{z}$ only on positions in K satisfies $|\mathcal{B}(\vec{y}, e) \cap \mathcal{C}| \ge 2$.*

We should regard $\vec{y}$ as the real received word by adding $e - e_0$ random errors upon the adversarial received word $\vec{z}$. From Proposition 11 we can see that if $e_0 > (1 - R)n - e$, then no matter the content of random errors the unique-decoding is impossible. As a side remark, Proposition 11 is a special case of Proposition 17.

On the other hand, by slightly adapting the proof of Rudra and Uurtamo [47], we can actually get the following positive combinatorial bound that states with high probability it is possible to uniquely decode from any $e_0 \le \min(e, (1 - R - \varepsilon)n - e)$ adversarial errors, which approaches the optimal bound $(1 - R)n$ for diminishing constant ε. Formally, we have the following combinatorial unique-decodability.

▶ **Proposition 12.** *Consider parameters $0 < \varepsilon, R < 1$, $n > \Omega(1/\varepsilon)$, any alphabet Σ with size at least $2^{\Omega(1/\varepsilon)}$, and any MDS code $\mathcal{C} \subseteq \Sigma^n$ with rate R. For any codeword $\vec{c} \in \mathcal{C}$, let $\vec{y}$ denote the received word after applying (e_0, e)-semi-adversarial errors to $\vec{c}$ where $e_0 \le \min(e, (1 - R - \varepsilon)n - e), e \le (1 - R)n$. Then, with probability at least $1 - |\Sigma|^{-\Omega(\varepsilon n)}$, we have $\mathcal{B}(\vec{y}, e) \cap \mathcal{C} = \{\vec{c}\}$.*

Motivated by these observations, a very natural question to ask is how to design unique-decoding algorithms in semi-adversarial error model that match the combinatorial bound $(1 - R - \varepsilon)n - e$ from Proposition 12 or even achieve the optimal bound $(1 - R)n - e$. In this paper, we provide unique-decoding algorithms from semi-adversarial errors for (a subclass of) Reed–Solomon and related codes that match the counting-based or even the optimal bound. In particular, our algorithm for interleaved Reed–Solomon codes builds upon the approach of Bleichenbacher, Kiayias, and Yung [13] and Alekhnovich [4] and our algorithms for folded Reed–Solomon and multiplicity codes are entirely new.

1.2 Our Results

We now proceed to discuss our results on the unique decoding of variants of Reed–Solomon codes in the semi-adversarial error model. All of our results are algorithmically efficient, running in $\tilde{O}(n)$ time. Moreover, for most tradeoff regimes between adversarial and random

errors, our results are either information-theoretically optimal or near-optimal. In particular, our results pertain to three variant of Reed–Solomon codes: *interleaved* RS codes (equivalently subfield evalaution RS codes), *folded* RS codes, and *multiplicity* codes. For each result, we briefly define the variant of RS codes we are considering along with the guarantees of our algorithm.

1.2.1 Interleaved Reed–Solomon Codes

Interleaving is a common technique in coding theory that was initially used to design codes and decoders with enhanced burst error-correction capabilities. By dispersing these errors, interleaving transforms them into random errors that are easier to correct. An important subfamily of Reed–Solomon codes is that of interleaved Reed–Solomon (IRS) codes (e.g., [13, 19, 36, 49]).

Given polynomials $f_1, \ldots, f_s \in \mathbb{F}_q[x]$, we define the s-interleaved encoding map $\mathcal{I}(f_1, f_2, \ldots, f_s)$ to be $(\mathcal{C}_1^f, \mathcal{C}_2^f, \ldots, \mathcal{C}_n^f) \in \left(\mathbb{F}_q^s\right)^n$, where $\mathcal{C}_i^f := \Big(f_1(\alpha_i), f_2(\alpha_i), \ldots, f_s(\alpha_i)\Big)$. With this encoder, we define a $[n, k]$ s-interleaved RS code for any distinct $\alpha_1, \ldots \alpha_n \in \mathbb{F}_q$ as follows.

$$\mathsf{IRS}_{n,k,q,s}(\alpha_1, \ldots, \alpha_n) := \Big\{\mathcal{I}(f_1, f_2, \ldots, f_s) : f_i \in \mathbb{F}_q[x],\ \deg f_i < k,\ \text{and } i \in [s]\Big\} \subseteq \left(\mathbb{F}_q^s\right)^n,$$

where $[s] := \{1, 2, \ldots, s\}$. Although an IRS code is technically not an RS code, we show in the full version that IRS codes have decoding properties identical to those of a class of *subfield* Reed–Solomon codes. That is, if we let $\mathbb{F}_{q^s}$ be the s-degree extension of $\mathbb{F}_q$, then $\mathsf{RS}_{n,k,q^s}(\alpha_1, \ldots, \alpha_n) \subseteq \mathbb{F}_{q^s}^n$ has identical decoding properties in the semi-adversarial model.

We now state our main decoding theorem for IRS codes in the semi-adversarial model. See Figure 1 for a visual summary of our result.

▶ **Theorem 13** (Informal Version). *Consider integers $s, n, k \geq 1$. For any interleaved Reed–Solomon code $\mathsf{RS}_{n,k,q^s}(\alpha_1, \ldots, \alpha_n)$ with $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$, there exists an efficient unique decoding algorithm that, with high probability, can correct (e_0, e)-semi-adversarial errors for any*

$$e \leq \frac{s}{s+1}(n-k) \quad \textit{and} \quad e_0 \leq \min(e, n-k-e).$$

The running time of the algorithm is at most $O\left(s^{O(1)} n \log^2(n) \log\log(n)\right)$.

In particular, note that we smoothly interpolate between the algorithms of Berlekamp–Welch for adversarial errors and the algorithm of BKY for random errors. Furthermore, note that as long as $e \leq \frac{s}{s+1}(n-k)$, our decoding bound *perfectly* matches the combinatorial bound of Proposition 11 and Proposition 12. Furthermore, for $e > n - \sqrt{nk}$ we "break" the Johnson bound[3] in the sense that our result is not implied by a combination of Guruswami-Sudan with a combinatorial bound on the list size.

1.2.2 Folded Reed–Solomon (RS) Codes

Another common variant of RS codes which we study is that of *folded* Reed–Solomon (FRS) codes [28]. In contrast to IRS codes which bundle the evaluations of the same point $\alpha \in \mathbb{F}_q$ at s different polynomials, FRS codes bundle s correlated evaluations of the same polynomial.

[3] In this paper "Johnson bound" refers to the error/rate tradeoff of $e \leq n - \sqrt{nk}$. We do *not* improve on Guruswami-Sudan algorithm in the adversarial model.

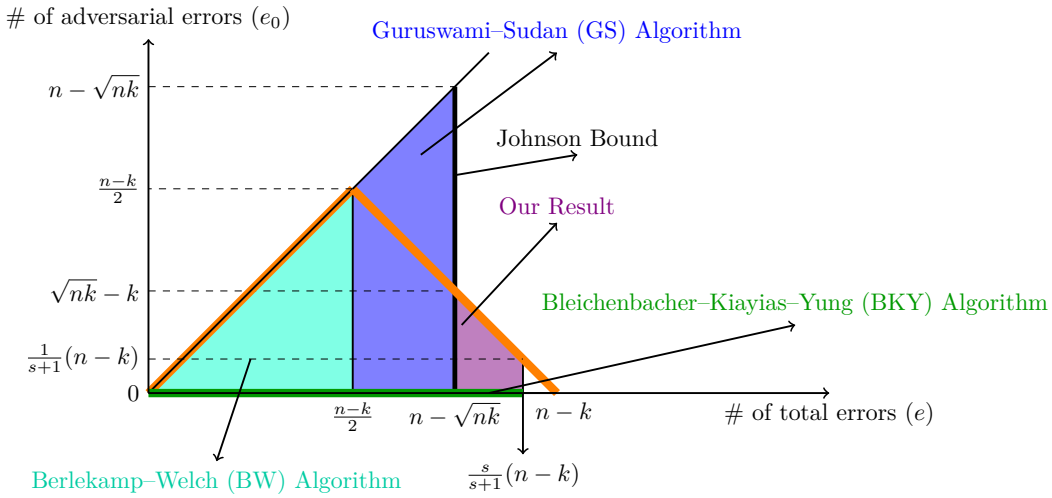

Figure 1 This figure is a comparison between our result with BW Algorithm, GS Algorithm, and the original BKY Algorithm for $[n, k]$ **Reed–Solomon codes** over alphabet $\mathbb{F}_{q^s}$ with n evaluation points chosen from $\mathbb{F}_q$. The horizontal axis depicts the total number of errors e allowed in the semi-adversarial model, and the vertical axis depicts the number of adversarial errors e_0. The region below the two orange lines is where unique-decoding is combinatorially possible by Proposition 12. It is impossible to achieve unique decoding above the two orange lines by Proposition 11.

More precisely, let γ be a multiplicative generator of $\mathbb{F}_q$ and consider $s \geq 1$. We say that $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$ are *appropriate* if the sn values $\{\gamma^{j-1}\alpha_i : i \in [n], j \in [s]\}$ are distinct [18]. Given $f \in \mathbb{F}_q[x]$, we define our evaluation map to be $\mathcal{C}(f) := (\mathcal{C}_1^f, \mathcal{C}_2^f, \ldots, \mathcal{C}_n^f) \in \left(\mathbb{F}_q^s\right)^n$, with $\mathcal{C}_i^f := \Big(f(\alpha_i), f(\gamma\alpha_i), \ldots, f(\gamma^{s-1}\alpha_i)\Big)$. Then, for parameters n, k and appropriate evaluation points $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$, we define our (s, γ)-folded RS code to be

$$\mathsf{FRS}_{n,k,q}^{s,\gamma}(\alpha_1, \alpha_2, \ldots, \alpha_n) := \left\{\mathcal{C}(f) : f \in \mathbb{F}_q[x], \ \deg f < k\right\} \subseteq \left(\mathbb{F}_q^s\right)^n.$$

We now state our results which apply to folded Reed–Solomon codes.

▶ **Theorem 14** (Informal Version). *Consider integers $s, L, n, k \geq 1$ and a generator γ of $\mathbb{F}_q^\times$ where $s \geq L$. For any appropriate (s, γ)-folded Reed–Solomon code $\mathsf{FRS}_{n,k}^{s,\gamma}(\alpha_1, \ldots, \alpha_n)$ over the alphabet $\mathbb{F}_q^s$, there exists an efficient unique decoding algorithm that, with high probability, can correct (e_0, e)-semi-adversarial errors for any*

$$e \leq \frac{L}{L+1}\left(n - \frac{k}{s-L+1}\right) \quad \text{and} \quad e_0 \leq \min\left(e, n - e - \frac{k}{s-L+1}\right).$$

The running time of the algorithm is at most $O\left(s^{O(1)} n \log^2(n) \log\log(n)\right)$.

Compared to the well-known Guruswami–Wang algorithm [23,27,30,34] , a key advantage of our novel algorithm is that it replaces the less-efficient root-finding step with a straightforward polynomial long division, resulting in a substantial efficiency boost. However, this is in tradeoff to the fact that the Guruswami–Wang algorithm can handle fully adversarial errors, whereas our approach is limited to semi-adversarial errors.

1.2.3 Multiplicity Codes

Our final main result is for another popular variant of RS codes known as *(univariate) multiplicity codes* (e.g., [30, 33, 35]). Again, multiplicity codes are similar to IRS codes except this time instead of evaluating a point α at s different polynomials, multiplicity codes evaluate s Hasse derivatives of the same polynomial at α. In particular, given $f \in \mathbb{F}_q[x]$ and symbolic variable z, we define $f^{(i)}(x)$ (the i-th Hasse derivative) to be the coefficient of z^i in the expansion of $f(x+z)$. From this, we can define our evaluation function to be $\mathcal{M}(f) := (\mathcal{C}_1^f, \mathcal{C}_2^f, \ldots, \mathcal{C}_n^f) \in \left(\mathbb{F}_q^s\right)^n$, with $\mathcal{C}_i^f := \left(f(\alpha_i), f^{(1)}(\alpha_i), \ldots, f^{(s-1)}(\alpha_i)\right)$. Then, for parameters n, k, s and distinct evaluation points $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$, our multiplicity code is defined to be

$$\mathsf{MULT}^s_{n,k}(\alpha_1, \alpha_2, \ldots, \alpha_n) := \left\{ \mathcal{M}(f) : f \in \mathbb{F}_q[x],\ \deg f < k \right\} \subseteq \left(\mathbb{F}_q^s\right)^n.$$

▶ Remark 15. RS, IRS, FRS, and MULT codes all coincide when $s = 1$.

▶ **Theorem 16** (Informal Version)**.** *Consider integers $s, L, n, k \geq 1$ and n distinct evaluation points $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$. If $\mathrm{Char}(\mathbb{F}_q) > s$, then for any multiplicity code $\mathsf{MULT}^s_{n,k}(\alpha_1, \ldots, \alpha_n)$ over the alphabet $\mathbb{F}_q^s$, there exists an efficient unique decoding algorithm that, with high probability, can correct (e_0, e)-semi-adversarial errors for any*

$$e \leq \frac{L}{L+1}\left(n - \frac{k}{s-L+1} - 1\right) \quad \text{and} \quad e_0 \leq \min\left(e, n - e - \frac{k}{s-L+1}\right).$$

The running time of the algorithm is at most $O\left(s^{O(1)} n \log^2(n) \log\log(n)\right)$.

As in Theorem 14, Theorem 16 has the advantage of the work of Guruswami–Wang algorithm [23, 27, 30, 34] of replacing the less-efficient root-finding step with a straightforward polynomial long division, resulting in a substantial efficiency boost.

1.3 Literature Overview

In this subsection, we provide an overview of previous work on both unique and list decoding of variants of Reed–Solomon (RS) codes, including RS codes themselves. In particular, we discuss the state-of-the-art decoding algorithms in both the adversarial and random error models.

1.3.1 Decoding Reed–Solomon Codes

Since the 1960s, efficiently decoding Reed–Solomon (RS) codes has become a central subject of study in coding theory, due to its importance in both practical error-correction applications and fundamental theoretical research in algorithms, complexity, and cryptography (e.g., [6, 8–10]). For an $[n, k]$ Reed–Solomon code, it can uniquely correct up to $(n-k)/2$ adversarial errors. During the past half century, numerous algorithms have been developed to uniquely decode Reed–Solomon codes up to this bound in the adversarial error model (e.g., [12, 20, 22, 39, 42, 55]). Among these many algorithms, two of the most popular are the Berlekamp–Welch algorithm [55] and the Berlekamp–Massey algorithm [39].

The Berlekamp–Welch algorithm, with a running time complexity of $O(n^3)$, employs a clever algebraic technique to decode Reed–Solomon codes, using polynomial interpolation to identify and correct errors. More concretely, [55] approaches the decoding problem by transforming it into a task of finding two polynomials: an error locator polynomial $E(x)$ and

a numerator polynomial $Q(x)$. For a received word represented as points (α_i, y_i), where α_i are distinct evaluation points and y_i are the received (corrupted) values, the algorithm sets up a system of linear equations based on the condition $Q(\alpha_i) = y_i E(\alpha_i)$ for all i. Here, $E(x)$ has degree equal to the number of errors e, and $Q(x)$ has degree less than $k + e$, where k is the dimension of the code. When the number of errors e is at most $(n - k)/2$, this system has a unique solution up to scaling. Solving it allows the decoder to determine $E(x)$, whose roots indicate the error locations, and subsequently recover the original message polynomial as $Q(x)/E(x)$.

Unlike an interpolation-based framework, the Berlekamp–Massey algorithm [39], with a better running time complexity of $O(n^2)$, takes a different approach by directly processing the syndrome sequence to find the error locator polynomial, taking advantage of shift registers for multiplication. Their method is recognized as a standard syndrome-based decoding framework. By building on these two algorithms, more efficient algorithms have been developed with a running time complexity of $O(n \cdot \text{poly}(\log n))$ (e.g., [38, 44, 54]).

Beyond unique decoding, to correct more than $(n - k)/2$ errors in the adversarial error model, list decoding algorithms are involved (e.g., [4, 29, 46, 53, 56]). In 1997, inspired by the Berlekamp–Welch algorithm, Sudan [53] provided the first list decoding algorithm of Reed–Solomon codes capable of correcting up to $n - \sqrt{2nk}$ errors. Later, by introducing the notion of multiplicity, Guruswami and Sudan [29] designed an improved list decoding algorithm that can correct up to $n - \sqrt{nk}$ errors, matching the Johnson bound [32]. Two follow-up works by Roth–Ruckenstein [46] and Alekhnovich [4] showed some fast implementations of the Guruswami–Sudan algorithm in time $O(n^2)$ and $O(n \cdot \text{poly}(\log n))$ respectively. Most recently, a long line of works [5, 16, 26, 48, 50] have shown that Reed–Solomon codes can achieve list decodability beyond the Johnson bound, reaching up to list decoding capacity. However, it remains an open problem to design a list decoding algorithm for any subclasses of Reed–Solomon codes capable of correcting adversarial errors beyond the Johnson bound. We emphasize that our result on interleaved Reed–Solomon codes (see Theorem 13) addresses this question in the semi-adversarial error model.

1.3.2 Decoding Interleaved Reed–Solomon Codes

Over the past two decades, considerable attention has been devoted to develop (unique) decoder (e.g., [13, 19, 36, 43, 49, 57]) for interleaved RS codes – an important subclass of RS codes – especially under the random error model. In particular, two independent seminal works – one by Coppersmith and Sudan [19] and the other by Bleichenbacher, Kiayias, and Yung [13] – proposed two remarkable (probabilistic) unique decoding algorithms that demonstrate surprisingly large error correction capabilities under the random error model.[4]

Coppersmith–Sudan is in some sense the dual of the multivariate interpolation decoding algorithm developed by Parvaresh Vardy [41] (although Coppersmith Sudan predates the Parvaresh Vardy algorithm).

The well-known Guruswami-Sudan algorithm does bivariate interpolation (i.e., a polynomial in (X, Y)) on the space of points of the form $\{(\alpha_i, y_i) : i \in [n]\}$, where α_i is the i-th evaluation point and y_i is the i-th received symbol. They use this interpolation to list decode Reed Solomon codes up to an error threshold of $1 - \sqrt{R}$. Multivariate interpolation decoding extends this idea to interleaved Reed-Solomon codes where one interpolates a $s + 1$

[4] Although BKY's result is only stated in the weaker model in which the errors locations are randomly located, the proof can be adapted to also handle adversarial error locations. This fact was recently observed by [1, 24], where they adapted the BKY algorithm for rational evaluation codes.

variate polynomial over $(X, Y_1, \ldots, Y_s)$, where $(Y_1, \ldots, Y_s)$ corresponds to the s components of each received symbol. If the error threshold is $1 - R^{s/(s+1)}$ then one can show that the interpolated polynomials will lie in the ideal $\langle Y_1 - f_1, \ldots, Y_s - f_s \rangle$. In general, this is not enough for decoding as $Y_i - f_i(x), i \in [n]$ may have many algebraic dependencies, but experimentally that occurs with negligible probability. Parvaresh and Vardy conjecture that for the setting of random errors multivariate interpolation decoding will succeed up to the error rate $1 - R^{s/(s+1)}$ with high probability (around $1 - n^{-cn}$). The only theoretical result studying this is a follow up work of, Parvaresh, Taghavi, and Vardy [40] which shows that this algorithm succeeds for $s = 2$ with error rate $1 - (6R)^{2/3} - O(R^{5/3})$ and with failure probability at most n^{-cn}.

Interpolating over $X, Y_1, \ldots, Y_s$ for n evaluation points $\alpha_1, \ldots, \alpha_n$ without multiplicity can be set up as finding an element in the row-kernel of a matrix where the columns are indexed by $i \in [n]$ and the rows are indexed by monomials m in $X, Y_1, \ldots, Y_s$ with the entries being m evaluated at $X = \alpha_i$ and $(Y_1, \ldots, Y_s)$ equals the i-th letter in the received word. Coppersmith-Sudan [19] picks an element in the *column kernel* of this matrix and uses its support (the coordinates $j \in [n]$ which are non-zero in the chosen vector) to help decode the codeword. Using multiplicities, they get further improvements. In the end, they obtain a (unique) decoder capable of correcting $1 - R^{\frac{s}{s+1}} - R$ fraction of random errors with a failure probability at most $O(n^s/q)$.

In contrast, Bleichenbacher, Kiayias, and Yung [13] draw inspiration from the original Berlekamp–Welch unique decoder [55], providing a (unique) decoder capable of correcting $\frac{s}{s+1}(1 - R)$ fraction of random errors with a failure probability at most $O(n/q)$. The failure probability of [13] was further reduced to $O(1/q)$ by Brown, Minder, and Shokrollahi [17].

Later, inspired by another famous unique decoder of Berlekamp and Massey [11, 39], Schmidt, Sidorenko, and Bossert [49] proposed a faster syndrome-based unique decoder[5] capable of correcting $\frac{s}{s+1}(1 - R)$ fraction of random errors with a failure probability at most $O(1/q)$. The running time of their algorithm is $O_s(n^2)$ (i.e., n^2 times some function of s). More recently, using power decoding techniques, Puchinger and Nielsen [43] proposed an improved algorithm with a conjectured error correction capability that can correct up to $1 - R^{\frac{s}{s+1}}$ fraction of random errors. However, although [43] has demonstrated numerous simulations for a wide variety of parameters that support the heuristic correctness of their approach, a formal proof of the theoretical correctness of their proposed algorithm remains open. Therefore, from a theoretical perspective, for $R > 1/3$ and $s > 2$, the relative decoding radius $\frac{s}{s+1}(1 - R)$ for s-interleaved RS codes of rate R remains state-of-the-art, as demonstrated in [13, 36, 49].

1.3.3 Decoding Folded Reed–Solomon and Multiplicity Codes

Folded Reed–Solomon (RS) codes and multiplicity codes are two prominent variants of RS codes, with list decoders (e.g., [23, 27, 28, 30, 33, 34]) that achieve the list decoding capacity even in the presence of adversarial errors. The first folded RS code list decoder was introduced in the seminal work of Guruswami and Rudra [28], building on the ideas of the Guruswami–Sudan list decoder. The first list decoder for multiplicity codes was introduced independently by Guruswami–Wang [30] and Kopparty [33], and it remains the most widely used framework in this domain.

[5] The IRS codes considered in [49] require a cyclic restriction on their evaluation points.

In fact, Guruswami and Wang [27, 30] proposed a unified linear algebraic list decoding algorithm for folded RS and multiplicity codes capable of correcting adversarial errors up to capacity. However, for codes of rate R and block length n, the list size guarantee from [30] is $O\big(n^{1/\varepsilon}\big)$ when decoding up to radius $1 - R - \varepsilon$. A follow-up work by Kopparty, Ron-Zewi, Saraf, and Wootters [34] introduced a randomized algorithm for the root-finding step under the Guruswami–Wang decoding framework, significantly improving its efficiency and reducing the list size guarantee to $\big(1/\varepsilon\big)^{O(1/\varepsilon)}$. Another follow-up work by Goyal, Harsha, Kumar, and Shankar [23] adapted the approach of Alekhnovich [4] into the Guruswami–Wang decoding framework, improving the running time to quasi-linear, which is $\widetilde{O}\big(2^{\mathrm{poly}(1/\varepsilon)} \cdot n\big)$.

As a side remark, very recently, two concurrent and independent works by Srivastava [52] and Chen–Zhang [18] provided dramatic improvements, reducing the list size from $\big(1/\varepsilon\big)^{O(1/\varepsilon)}$ to $O(1/\varepsilon^2)$ and $O\big(1/\varepsilon\big)$, respectively. Moreover, the Chen–Zhang bound almost matches the generalized Singleton bound [50] up to a constant multiplicative factor. It is worth noting that both proofs of Srivastava and Chen–Zhang are purely combinatorial, and turning them into efficient algorithms remains an open problem. For further details, the reader is referred to the most recent comprehensive survey by Garg, Harsha, Kumar, Saptharishi, and Shankar [21].

Meanwhile, current efficient decoding algorithms for folded Reed–Solomon and multiplicity codes are heavily based on the Guruswami–Wang decoder framework [27, 30]. It is an interesting open problem to design alternative, more efficient decoding algorithms for folded RS and multiplicity codes that do not rely on the Guruswami–Wang framework, and a primary contribution of this paper is making progress toward this exact question.

As previously mentioned, we define folded Reed–Solomon codes in the sense of Guruswami–Rudra [28]. However, the term "folded Reed–Solomon codes" was first coined in a paper by Krachkovsky [37]. The codes defined in both papers are constructed by bundling s classical Reed–Solomon codes, but they choose evaluation points in different ways. Concretely, recall that for the folded Reed–Solomon codes defined in [28], for each $i \in [n]$ and message polynomimal f, the i-th symbol in the encoding of f is $\big(f(\alpha_i), f(\gamma\alpha_i) \ldots, f(\gamma^{s-1}\alpha_i)\big) \in \mathbb{F}_q^s$, where $\alpha_i \in \mathbb{F}_q$ and γ is a generator of $\mathbb{F}_q^\times$. On the other hand, [37] defined it as $\big(f(\alpha^i), f(\alpha^{i+n}) \ldots, f(\alpha^{i+(s-1)n})\big) \in \mathbb{F}_q^s$, where α has order exactly sn in $\mathbb{F}_q^\times$. This special choice on evaluation points enables [37] to perform FFT-related algorithms and obtain a syndrome-based decoding algorithm for random "phased burst" errors.

Subsequent to the first posting of our work, Ashvinkumar, Habib, and Srivastava [7] presented new algorithms for decoding folded Reed-Solomon codes in the adversarial error model.

1.4 Proof Overview

We now give an overview of the main technical contributions of our paper.

1.4.1 Interleaved Reed–Solomon Codes

We begin by discussing our new algorithm for interleaved Reed–Solomon codes (Theorem 13). Our algorithm and analysis builds on the techniques used by Bleichenbacher, Kiayias, and Yung [13], but we deviate from their methods in a few significant ways. In this IRS decoding algorithm, our main technical contribution is that we use a novel monomial-tracking argument to analyze the performance of this algorithm in the new semi-adversarial error setting.

To describe these improvements, we first discuss the interpolation algorithm of [13] (henceforth the "BKY algorithm"). Consider the s-interleaved Reed–Solomon code $\mathcal{C} := \mathsf{IRS}_{n,k,q,s}(\alpha_1, \ldots, \alpha_n)$ with distinct $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q \setminus \{0\}$. We choose polynomials

$f_1(X), \ldots, f_s(X)$ of degree at most $k-1$ as our message, encoding them according to $\mathcal{C}$ and transmit them across the channel. Assume that our received word $\vec{y} := (\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_n) \in (\mathbb{F}_q^s)^n$, where $\vec{y}_i := (y_{i,1}, y_{i,2}, \ldots, y_{i,s}) \in \mathbb{F}_q^s$.

The BKY algorithm begins by assuming $\bar{e}$ of the n coordinates of $\vec{y}$ are corrupted. To identify these $\bar{e}$ coordinates, the BKY algorithm adapts the Berlekamp-Welch algorithm by solving the following interpolation problems. Let $E(X)$ be a polynomial of degree at most $\bar{e}$ with unknown coefficients and let $A_1(X), \ldots, A_s(X)$ be polynomials of degree at most $k-1+\bar{e}$ with unknown coefficients. We seek to solve the following interpolation equations:

$$A_h(\alpha_i) = y_{i,h} E(\alpha_i) \qquad \forall i \in \{1, 2, \ldots, n\}, \forall h \in \{1, 2, \ldots, s\}. \tag{1}$$

To analyze this system of equations, we consider the block matrix

$$\mathbf{B} := \begin{pmatrix} \mathbf{M} & & & -\mathbf{N}_1 \\ & \mathbf{M} & & -\mathbf{N}_2 \\ & & \ddots & \vdots \\ & & \mathbf{M} & -\mathbf{N}_s \end{pmatrix}, \tag{2}$$

where $\mathbf{M} \in \mathbb{F}_q^{n \times (k+\bar{e})}$ such that $\mathbf{M}_{i,j} = \alpha_i^{j-1}$, and for each $h \in [s]$ we have that $\mathbf{N}_h \in \mathbb{F}_q^{n \times \bar{e}}$ with $(\mathbf{N}_h)_{i,j} = y_{i,h}\alpha_i^j$. For the purposes of this overview, assume for simplicity that $\mathbf{B}$ is a square matrix. That is, $s(k+\bar{e}) + \bar{e} = sn$. In this scenario, one can prove that the interpolation (1) succeeds if and only if, we have that $\det \mathbf{B} \neq 0$. That is, it suffices to bound the probability that $\det \mathbf{B} \neq 0$ over the random errors in the channel. By the Schwartz-Zippel lemma, it suffices to replace the random errors $y_{i,h}$ with corresponding symbolic variables $Y_{i,h}$ to create a symbolic matrix $\mathbf{B}^{\mathsf{SYM}}$ and argue that $\det \mathbf{B}^{\mathsf{SYM}} \neq 0$.

Bleichenbacher, Kiayias, and Yung [13] proceeded to argue this by substituting fixed values for some of the $Y_{i,h}$'s and then using an ad-hoc analysis of the determinant to argue that it is nonzero. We improve on the BKY algorithm and analysis in two significant ways. First, we improve the runtime of the algorithm from roughly $O(n^4)$ time to near-linear time by using the interpolation framework of Alekhnovich [4] to solve (1) efficiently.[6] In particular, the interpolation framework automatically finds the optimal choice of $\bar{e}$ rather than doing a brute-force search like in the BKY algorithm.

Second, we streamline the analysis of the determinant of (2). By applying a series of column operations to $\mathbf{B}$, we can transform the rightmost blocks $\mathbf{N}_1, \ldots, \mathbf{N}_s$ into *diagonal* matrices $\mathbf{D}_1, \ldots, \mathbf{D}_s$, where each $\mathbf{D}_h$ corresponds precisely to the errors which occur in the transmission of the codeword. The exact sequence of column operations needed to reveal these errors depends on the original polynomials $f_1, \ldots, f_s$, but we can assume polynomials are known for the purposes of analyzing the rank of $\mathbf{B}$.

By having the errors be presented transparently as diagonal matrices, arguing that $\det \mathbf{B}^{\mathsf{SYM}} \neq 0$ can be done in a much more straightforward manner. In particular, it suffices to identify an explicit monomial $Y_{i_1,h_1} \cdots Y_{i_\ell,h_\ell}$ with the following properties:

P1. There is exactly one expansion of the last $\bar{e}$ columns of $\mathbf{B}^{\mathsf{SYM}}$ which produces that monomial.

P2. The minor induced by that monomial choice has nonzero determinant.

In the purely random error model, finding such a choice of monomials is straightforward–any choice of monomials from distinct columns for which an equal number of variables appear in

[6] After the initial posting of our work, we also learned that the use of Alekhnovich [4] and similar techniques for fast polynomial interpolation have appeared in related works [25, 58].

each row block will work. However, the true power of this perspective is that it gives us a foothold for analyzing a mixture of adversarial and random errors, i.e., the semi-adversarial error model. More precisely, depending on the exact choices for the adversarial errors, the set of monomials which satisfy both P1 and P2 can change significantly. In particular, there is no "oblivious" choice for the monomial. Instead, we give a procedure which constructs a monomial based on the pattern of the adversarial errors–which we may assume we know for the purposes of the analysis. At a high level this procedure is a greedy algorithm, we iterated through the row blocks of $\mathbf{B}$ and in each block we pick the maximal number of monomials possible such that P1 and P2 can still hold.

1.4.2 Folded Reed–Solomon Codes

Next, we describe the algorithm and analysis leading to Theorem 14. The new decoding algorithm is inspired by the previous IRS algorithm plus the Guruswami-Wang [30] interpolation. The analysis for semi-adversarial errors follows the similar framework as in the IRS case. However, the monomial-tracking is more difficult for FRS codes and we therefore use a new "greedy algorithm" to find the special monomial. This is the main challenge of extending the analysis for IRS to FRS.

Let $\mathcal{C} := \mathsf{FRS}^{s,\gamma}_{n,k,q}(\alpha_1, \alpha_2, \ldots, \alpha_n) \subseteq (\mathbb{F}_q^s)^n$ and consider a polynomial $f(X)$ of degree at most $k-1$. After encoding f according to $\mathcal{C}$ and transmitting it through the $(e_0, \overline{e})$-semi-adversarial channel, let $\vec{y} := (\vec{y}_1, \vec{y}_2, \ldots, \vec{y}_n) \in (\mathbb{F}_q^s)^n$, where $\vec{y}_i := (y_{i,1}, y_{i,2}, \ldots, y_{i,s}) \in \mathbb{F}_q^s$ is the received message. Pick parameters $\overline{e} \in [n]$ and $L \in [s]$ and consider the following L-*length* interpolation equations:

$$A_h(\gamma^{i-1}\alpha_j) = y_{j,i+h-1} E\left(\gamma^{i-1}\alpha_j\right) \quad i \in [s-L+1],\ j \in [n],\ h \in [L]. \tag{3}$$

Like for BKY interpolation, these interpolation equations can be solved in near-linear time using the techniques of Alekhnovich [4]. To the best of our knowledge, our interpolation is novel, blending characteristics of both BKY and Guruswami–Wang [30] interpolation. In particular, compared to other list-decoding algortihms for folded Reed–Solomon codes (such as Guruswami–Wang), the lower-order terms in the runtime for our approach are much smaller. More concretely, our running time is $\tilde{O}(\mathrm{poly}(s)n)$, while the current best implementation of the Guruswami–Wang algorithm takes time $\tilde{O}(2^{\mathrm{poly}(s)}n)$ [23]. A key advantage of our algorithm is that it eliminates the need for the root-finding step required in the Guruswami–Wang algorithm. Instead, after our new interpolation, only a simple polynomial long division is necessary.

To analyze these interpolation equations, we construct a block matrix similar to that of (2). Now, the number of rows in each block is $(s-L+1)n$, which we index by a pair $(i,j) \in [s-L+1] \times [n]$. Then, $\mathbf{M} \in \mathbb{F}_q^{(s-L+1)n \times (k+\overline{e})}$ has the structure $\mathbf{M}_{(i,j),\ell} = (\gamma^{i-1}\alpha_j)^{\ell-1}$ and for each $h \in [s]$, we have that $(\mathbf{N}_h)_{(i,j),\ell} = y_{j,i+h-1}(\gamma^{i-1}\alpha_j)^{\ell}$.

Using the techniques of Bleichenbacher, Kiayias, and Yung [13], this matrix would be very difficult to analyze directly, but by applying analogous simplifications to the ones we used for interleaved Reed–Solomon codes, we can again transform the last block-column of $\mathbf{B}$ to consist only of diagonal matrices. However, since our interpolation is L-length, we now have the extra challenge that each error can appear in up to L diagonal matrices (since each $y_{j,i+h-1}$ can appear up to L times).

Like in the IRS case, we transform $\mathbf{B}$ into a symbolic matrix $\mathbf{B}^{\mathsf{SYM}}$ with symbolic variables as substitutes for the random variables. We then seek to find a monomial in $\mathbf{B}^{\mathsf{SYM}}$ satisfying the aforementioned properties P1 and P2. With the combination of repeated symbolic variables as well as the presence of adversarial errors in the semi-adversarial model, the

procedure of identifying a valid monomial becomes significantly more complex. It is still fundamentally a greedy algorithm like in the IRS case, but much more bookkeeping is needed to ensure that P1 and P2 remain satisfied.

1.4.3 Multiplicity Codes

Next, we describe the adaptations of Theorem 14 which led to Theorem 16. Fundamentally, the interpolation is similar to that of (3), except we now replace multiplication by γ with taking a derivative of the message polynomial f. That is, our length-L interpolation equations are of the form,

$$A_h^{(i-1)}(\alpha_j) = \sum_{\ell=0}^{i-1} \binom{i+h-2-\ell}{h-1} y_{j,i+h-1-\ell} E^{(\ell)}(\alpha_j) \quad i \in [s-L+1],\ j \in [n],\ h \in [L]. \tag{4}$$

where E again has degree at most $\overline{e}$. Again, this interpolation can be done in near-linear time by using the interpolation algorithm of Alekhnovich [4]. Note that due to the chain rule for derivatives, the error term in (4) turns into a sum. As a result, if we build an interpolation matrix $\mathbf{B}$ similar to that of (2) to analyze (4), we cannot use column operations to simplify the final column block to only consist of diagonal matrices. Instead, we will have block-diagonal matrices for which each block is a *lower triangular* matrix. The corresponding symbolic matrix $\mathbf{B}^{\mathsf{SYM}}$ will have each variable appear numerous times, with repetition coming both from the "length-L" of the interpolation as well as the repeated terms arising from the chain rule.

Again, we seek to identify a symbolic monomial in the expansion of $\det \mathbf{B}^{\mathsf{SYM}}$ which satisfies properties P1 and P2. The key observation is that within one of these lower triangular matrices, variable appearing in the lower left corner appears exactly once. Thus, by using a greedy algorithm which starts by taking monomials from these lower left corners, we have much more control over the uniqueness of the monomial selected.

2 Conclusions and Future Directions

In this paper, we proposed a new error model, *semi-adversarial errors*, for studying the efficient decodability of error-correcting codes. Bridging between the adversarial and fully random error models, our interpolation and analysis techniques leveraged ideas from both models to obtain new results in the semi-adversarial model. First, we improved the analysis of Bleichenbacher–Kiayias–Yung's decoding algorithm for interleaved Reed–Solomon codes to show that the algorithms achieves the optimal unique decoding radius for most combinations of adversarial and random errors in the semi-adversarial model. Second, we found a novel application of Alekhnovich's interpolation algorithm to extend BKY's interpolation algorithm to the settings of folded Reed–Solomon codes and multiplicity codes. By extending our analysis methods for interleaved Reed–Solomon codes, we were able to show near information-theoretically optimal decoding for folded Reed–Solomon and multiplicaty codes as well. We conclude the paper by discussing some directions for future research.

2.1 Coppersmith–Sudan in the semi-adversarial model

This paper primarily focused on understanding the limits of the Bleichenbacher–Kiayias–Yung algorithm [13] in the semi-adversarial model. However, contemporaneous to the BKY algorithm, Coppersmith and Sudan [19] independently came up with a different interpolation

algorithm – the CS algorithm – for uniquely decoding interleaved Reed–Solomon codes. As discussed in the literature overview, the CS algorithm uses the interpolation matrix for a generalization of the Guruswami Sudan algorithm for interleaved Reed-Solomon codes. It has non-linear terms in the received word in this matrix unlike BKY which only has linear terms. This linearity was very helpful in the analysis of our algorithm, and the lack of it poses a challenged for analyzing CS in the semi-adversarial model. Furthermore, CS algorithm only decodes to the distance $1 - R^{s/(s+1)} - R$, which is incomparable with BKY (CS beats the BKY rate $\frac{s}{s+1}(1-R)$ for small R for a fixed s). In future work, we plan to tackle these challenges and explore the limits of the CS algorithm in the semi-adversarial model.

2.2 Unique Decoding up to $1 - R^{s/(s+1)}$ distance under random errors

A related problem would be to analyze the Multivariate Interpolation Decoding algorithm of Parvaresh and Vardy [41]. They conjecture that it should be able to decode up to a $1-R^{s/(s+1)}$ distance with high probability. Power decoding algorithms [43] are also conjectured to decode with high probability to this radius. The CS algorithm is also conjectured to decode to a $1 - R^{s/(s+1)}$ distance. Currently, no algorithm has been theoretically proven to achieve this guarantee.

2.3 List-decoding in the semi-adversarial model

So far, we have only discussed the limits of unique decodability in the semi-adversarial model. A natural question is to what extent these results can be extended to the setting of list decoding. To approach this question, a natural first task is to explore the *combinatorial* limits of list-decoding in the semi-adversarial model.

Recall that a code $\mathcal{C} \subseteq \Sigma^n$ is (e, L)-list-decodable (in the adversarial model) if for any message $\vec{y} \in \Sigma^n$ we have that at most $|\mathcal{C} \cap \mathcal{B}(\vec{y}, e)| \leq L$. The generalized Singleton bound of Shangguan and Tamo [50] says that for any $L \geq 1$, if a code $\mathcal{C} \subseteq \Sigma^n$ is (e, L)-list decodable then $e \leq \frac{L}{L+1}(n-k)$.

This bound can be extended to the semi-adversarial model. In particular, given $0 \leq e_0 \leq e \leq n$, we say that a code $\mathcal{C}$ is (e_0, e, L)-list-decodable if for any $\vec{c} \in \mathcal{C}$, when sending $\vec{c}$ through the (e_0, e)-semi-adversarial channel, the output message $\vec{y}$ satisfies $|\mathcal{B}(\vec{y}, e)| \leq L$ with high probability, no matter what choices the adversary makes in the channel. We state the generalized semi-adversarial Singleton bound as follows.

▶ **Proposition 17** (GSSB: Generalized Semi-adversarial Singleton Bound)**.** *Consider any code $\mathcal{C} \subseteq \Sigma^n$ with $R := \frac{\log |\mathcal{C}|}{n \log |\Sigma|}$ such that $k := Rn$ is integral. For any (e_0, e, L) with $L + 1 \leq |\Sigma|$ and $\lfloor \frac{e_0}{L} \rfloor > (1-R)n - e$, there exists some word $\vec{z} \in \Sigma^n$ such that $|\mathcal{B}(\vec{z}, e_0) \cap \mathcal{C}| \geq 1$ as well as $K \subseteq [n], |K| = e - e_0$ such that any received word $\vec{y} \in \Sigma^n$ that differs with $\vec{z}$ only on positions in K satisfies $|\mathcal{B}(\vec{y}, e) \cap \mathcal{C}| \geq L + 1$.*

In other words, there exists $\vec{c} \in C$ such that sending $\vec{c}$ through the (e_0, e)-semi-adversarial channel results a message $\vec{y}$ for which there are at least L other codewords of C which are within the minimal ball centered at $\vec{y}$ containing $\vec{c}$. For MDS codes, observe that Proposition 11 is a special case of Proposition 17 when $L = 1$.

Proposition 17 naturally interpolates the known combinatorial decoding bounds for fully adversarial and fully random errors. In particular, when $e_0 = e$, we get precisely the $\left(\frac{L}{L+1}(n-k), L\right)$ generalized Singleton bound (GSB) of Shangguan and Tamo [50]. Further, when $e_0 = 0$, we get an upper bound of $n - k$, which is consistent with the fact that

combinatorially random errors can be uniquely decoded up to capacity.[7] As a result, we can view Proposition 17 as suggesting that one can *break* the generalized Singleton bound when we are promised that the channel is not entirely adversarial.

However, to understand the sharpness of the GSSB, one needs to determine whether there exist codes which attain the GSSB, particularly Reed–Solomon codes. In the adversarial case, through a series of papers, [5, 16, 26, 50], we now that the GSB can be attained by Reed–Solomon codes with random evaluation points, even when the field size is $O(n)$. For arbitrary subfield evaluation points (i.e., s-interleaved RS codes), we only know that $\left(\frac{s}{s+1}(n-k), q^{\gamma sn}\right)$ list decodability is possible [31], where γ is a constant in $(0, 1)$. However, a recent work by Chen and Zhang [18] shows that explicit s-folded Reed-Solomon codes are $\left(\frac{L}{L+1}\left(n - \frac{k}{s-L+1}\right), L\right)$ list-decodable. For random evaluation points, some broader families of codes also attain the GSB [14, 15]. We leave extending these results to the GSSB as the topic of future work.

References

1 Matteo Abbondati, Eleonora Guerrini, and Romain Lebreton. Decoding simultaneous rational evaluation codes. In Jonathan D. Hauenstein, Wen-shin Lee, and Shaoshi Chen, editors, *Proceedings of the 2024 International Symposium on Symbolic and Algebraic Computation, ISSAC 2024, Raleigh, NC, USA, July 16-19, 2024*, pages 153–161. ACM, 2024. doi:10.1145/3666000.3669686.

2 Matteo Abbondati, Eleonora Guerrini, and Romain Lebreton. Simultaneous rational function codes: Improved analysis beyond half the minimum distance with multiplicities and poles, 2025. doi:10.48550/arXiv.2508.05284.

3 Matteo Abbondati, Eleonora Guerrini, and Romain Lebreton. Simultaneous rational number codes: Decoding beyond half the minimum distance with multiplicities and bad primes. *J. Symb. Comput.*, 132:102481, 2026. doi:10.1016/J.JSC.2025.102481.

4 Michael Alekhnovich. Linear Diophantine equations over polynomials and soft decoding of Reed-Solomon codes. *IEEE Trans. Inform. Theory*, 51(7):2257–2265, 2005. doi:10.1109/TIT.2005.850097.

5 Omar Alrabiah, Venkatesan Guruswami, and Ray Li. Randomly punctured Reed-Solomon codes achieve list-decoding capacity over linear-sized fields. In *STOC'24—Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1458–1469. ACM, New York, 2024. doi:10.1145/3618260.3649634.

6 Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. STIR: Reed-Solomon proximity testing with fewer queries. In *Advances in cryptology—CRYPTO 2024. Part X*, volume 14929 of *Lecture Notes in Comput. Sci.*, pages 380–413. Springer, Cham, [2024] ©2024. doi:10.1007/978-3-031-68403-6_12.

7 Vikrant Ashvinkumar, Mursalin Habib, and Shashank Srivastava. Algorithmic improvements to list decoding of folded reed-solomon codes. In Kasper Green Larsen and Barna Saha, editors, *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2026, Vancouver, BC, Canada, January 11-14, 2026*, pages 880–898. SIAM, 2026. doi:10.1137/1.9781611978971.35.

8 Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast reed-solomon interactive oracle proofs of proximity. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, Prague, Czech Republic, July 9-13, 2018*, LIPIcs, pages 14:1–14:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ICALP.2018.14.

[7] This can be seen by taking the decoding radius of $\frac{s}{s+1}(n-k)$ for s-interleaved Reed–Solomon by the BKY algorithm to the limit as $s \to \infty$.

9 Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity gaps for Reed-Solomon codes. *J. ACM*, 70(5):Art. 31, 57, 2023. doi:10.1145/3614423.

10 Eli Ben-Sasson, Lior Goldberg, Swastik Kopparty, and Shubhangi Saraf. DEEP-FRI: Sampling Outside the Box Improves Soundness. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:32. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ITCS.2020.5.

11 Elwyn Berlekamp. *Algebraic coding theory*. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, revised edition, 2015. doi:10.1142/9407.

12 Elwyn R. Berlekamp. Bounded distance+1 soft-decision reed-solomon decoding. *IEEE Trans. Inf. Theory*, 42(3):704–720, 1996. doi:10.1109/18.490539.

13 Daniel Bleichenbacher, Aggelos Kiayias, and Moti Yung. Decoding of interleaved Reed Solomon codes over noisy data. In *Automata, languages and programming*, volume 2719 of *Lecture Notes in Comput. Sci.*, pages 97–108. Springer, Berlin, 2003. doi:10.1007/3-540-45061-0_9.

14 Joshua Brakensiek, Manik Dhar, and Sivakanth Gopi. Generalized GM-MDS: Polynomial Codes Are Higher Order MDS. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 728–739, Vancouver BC Canada, June 2024. ACM. doi:10.1145/3618260.3649637.

15 Joshua Brakensiek, Manik Dhar, Sivakanth Gopi, and Zihan Zhang. AG codes achieve list-decoding capacity over constant-sized fields. *IEEE Trans. Inform. Theory*, 71(8):5935–5956, 2025. doi:10.1109/tit.2025.3577506.

16 Joshua Brakensiek, Sivakanth Gopi, and Visu Makam. Generic Reed-Solomon codes achieve list-decoding capacity. In *STOC'23—Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1488–1501. ACM, New York, [2023] ©2023. doi:10.1145/3564246.3585128.

17 Andrew Brown, Lorenz Minder, and Amin Shokrollahi. Probabilistic decoding of interleaved rs-codes on the q-ary symmetric channel. In *Proceedings of the 2004 IEEE International Symposium on Information Theory, ISIT 2004, Chicago Downtown Marriott, Chicago, Illinois, USA, June 27 - July 2, 2004*, page 327. IEEE, 2004. doi:10.1109/ISIT.2004.1365363.

18 Yeyuan Chen and Zihan Zhang. Explicit folded reed-solomon and multiplicity codes achieve relaxed generalized singleton bounds. In Michal Koucký and Nikhil Bansal, editors, *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, Czechia, June 23-27, 2025*, pages 1–12. ACM, 2025. doi:10.1145/3717823.3718114.

19 Don Coppersmith and Madhu Sudan. Reconstructing curves in three (and higher) dimensional space from noisy data. In Lawrence L. Larmore and Michel X. Goemans, editors, *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA*, pages 136–142. ACM, 2003. doi:10.1145/780542.780563.

20 Shuhong Gao. A new algorithm for decoding reed-solomon codes. In *Communications, information and network security*, pages 55–68. Springer, 2003.

21 Abhibhav Garg, Prahladh Harsha, Mrinal Kumar, Ramprasad Saptharishi, and Ashutosh Shankar. An exposition of recent list-size bounds of FRS codes. *CoRR*, abs/2502.14358, 2025. doi:10.48550/arXiv.2502.14358.

22 Daniel Gorenstein and Neal Zierler. A class of error-correcting codes in p^m symbols. *J. Soc. Indust. Appl. Math.*, 9:207–214, 1961.

23 Rohan Goyal, Prahladh Harsha, Mrinal Kumar, and Ashutosh Shankar. Fast list decoding of univariate multiplicity and folded Reed-Solomon codes. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science—FOCS 2024*, pages 328–343. IEEE Computer Soc., Los Alamitos, CA, 2024. doi:10.1109/FOCS61266.2024.00028.

24 Eleonora Guerrini, Kamel Lairedj, Romain Lebreton, and Ilaria Zappatore. Simultaneous rational function reconstruction with errors: Handling multiplicities and poles. *J. Symb. Comput.*, 116:345–364, 2023. doi:10.1016/J.JSC.2022.10.007.

25 Eleonora Guerrini, Romain Lebreton, and Ilaria Zappatore. Polynomial linear system solving with random errors: New bounds and early termination technique. In Frédéric Chyzak and George Labahn, editors, *ISSAC '21: International Symposium on Symbolic and Algebraic Computation, Virtual Event, Russia, July 18-23, 2021*, pages 171–178. ACM, 2021. doi:10.1145/3452143.3465548.

26 Zeyu Guo and Zihan Zhang. Randomly punctured Reed-Solomon codes achieve the list decoding capacity over polynomial-size alphabets. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science—FOCS 2023*, pages 164–176. IEEE Computer Soc., Los Alamitos, CA, 2023. doi:10.1109/FOCS57990.2023.00019.

27 Venkatesan Guruswami. Linear-algebraic list decoding of folded Reed-Solomon codes. In *26th Annual IEEE Conference on Computational Complexity*, pages 77–85. IEEE Computer Soc., Los Alamitos, CA, 2011. doi:10.1109/CCC.2011.22.

28 Venkatesan Guruswami and Atri Rudra. Explicit capacity-achieving list-decodable codes or decoding up to the Singleton bound using folded Reed-Solomon codes. In *STOC'06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 1–10. ACM, New York, 2006. doi:10.1145/1132516.1132518.

29 Venkatesan Guruswami and Madhu Sudan. Improved decoding of reed-solomon and algebraic-geometric codes. In *39th Annual Symposium on Foundations of Computer Science, FOCS 1998, Palo Alto, California, USA, November 8-11, 1998*, pages 28–39. IEEE Computer Society, 1998. doi:10.1109/SFCS.1998.743426.

30 Venkatesan Guruswami and Carol Wang. Linear-algebraic list decoding for variants of Reed-Solomon codes. *IEEE Trans. Inform. Theory*, 59(6):3257–3268, 2013. doi:10.1109/TIT.2013.2246813.

31 Venkatesan Guruswami and Chaoping Xing. Optimal rate list decoding over bounded alphabets using algebraic-geometric codes. *J. ACM*, 69(2):Art. 10, 48, 2022. doi:10.1145/3506668.

32 Selmer M. Johnson. A new upper bound for error-correcting codes. *IRE Trans. Inf. Theory*, 8(3):203–207, 1962. doi:10.1109/TIT.1962.1057714.

33 Swastik Kopparty. List-decoding multiplicity codes. *Theory Comput.*, 11:149–182, 2015. doi:10.4086/TOC.2015.V011A005.

34 Swastik Kopparty, Noga Ron-Zewi, Shubhangi Saraf, and Mary Wootters. Improved list decoding of folded reed-solomon and multiplicity codes. *SIAM J. Comput.*, 52(3):794–840, 2023. doi:10.1137/20M1370215.

35 Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-rate codes with sublinear-time decoding. *J. ACM*, 61(5):28:1–28:20, 2014. doi:10.1145/2629416.

36 V Yu Krachkovsky and Yuan Xing Lee. Decoding for iterative reed-solomon coding schemes. *IEEE Transactions on Magnetics*, 33(5):2740–2742, 1997.

37 Victor Y. Krachkovsky. Reed-Solomon codes for correcting phased error bursts. *IEEE Trans. Inform. Theory*, 49(11):2975–2984, 2003. doi:10.1109/TIT.2003.819333.

38 Sian-Jheng Lin, Tareq Y. Al-Naffouri, and Yunghsiang S. Han. FFT algorithm for binary extension finite fields and its application to Reed-Solomon codes. *IEEE Trans. Inform. Theory*, 62(10):5343–5358, 2016. doi:10.1109/TIT.2016.2600417.

39 James L. Massey. Shift-register synthesis and BCH decoding. *IEEE Trans. Inform. Theory*, IT-15:122–127, 1969. doi:10.1109/tit.1969.1054260.

40 Farzad Parvaresh, Mohammad H. Taghavi, and Alexander Vardy. On the performance of multivariate interpolation decoding of reed-solomon codes. In *2006 IEEE International Symposium on Information Theory*, pages 2027–2031, 2006. doi:10.1109/ISIT.2006.261905.

41 Farzad Parvaresh and Alexander Vardy. Multivariate interpolation decoding beyond the guruswami-sudan radius. In *Proceedings of the 42nd Allerton Conference on Communication, Control and Computing*, 2004.

42 W. Wesley Peterson. Encoding and error-correction procedures for the bose-chaudhuri codes. *IRE Trans. Inf. Theory*, 6(4):459–470, 1960. doi:10.1109/TIT.1960.1057586.

43 Sven Puchinger and Johan Rosenkilde. Decoding of interleaved reed-solomon codes using improved power decoding. In *2017 IEEE International Symposium on Information Theory, ISIT 2017, Aachen, Germany, June 25-30, 2017*, pages 356–360. IEEE, 2017. doi:10.1109/ISIT.2017.8006549.

44 Irving S. Reed, Robert A. Scholtz, T. K. Truong, and Lloyd R. Welch. The fast decoding of Reed-Solomon codes using Fermat theoretic transforms and continued fractions. *IEEE Trans. Inform. Theory*, IT-24(1):100–106, 1978. doi:10.1109/tit.1978.1055816.

45 Irving S. Reed and Gustave Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960. doi:10.1137/0108018.

46 Ron M. Roth and Gitit Ruckenstein. Efficient decoding of reed-solomon codes beyond half the minimum distance. *IEEE Trans. Inf. Theory*, 46(1):246–257, 2000. doi:10.1109/18.817522.

47 Atri Rudra and Steve Uurtamo. Two theorems on list decoding - (extended abstract). In Maria J. Serna, Ronen Shaltiel, Klaus Jansen, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 13th International Workshop, APPROX 2010, and 14th International Workshop, RANDOM 2010, Barcelona, Spain, September 1-3, 2010. Proceedings*, Lecture Notes in Computer Science, pages 696–709. Springer, 2010. doi:10.1007/978-3-642-15369-3_52.

48 Atri Rudra and Mary Wootters. Every list-decodable code for high noise has abundant near-optimal rate puncturings. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 764–773. ACM, 2014. doi:10.1145/2591796.2591797.

49 Georg Schmidt, Vladimir R. Sidorenko, and Martin Bossert. Collaborative decoding of interleaved Reed-Solomon codes and concatenated code designs. *IEEE Trans. Inform. Theory*, 55(7):2991–3012, 2009. doi:10.1109/TIT.2009.2021308.

50 Chong Shangguan and Itzhak Tamo. Generalized Singleton Bound and List-Decoding Reed–Solomon Codes Beyond the Johnson Radius. *SIAM Journal on Computing*, 52(3):684–717, June 2023. doi:10.1137/20M138795X.

51 Richard C. Singleton. Maximum distance q-nary codes. *IEEE Trans. Inform. Theory*, IT-10:116–118, 1964. doi:10.1109/tit.1964.1053661.

52 Shashank Srivastava. Improved list size for folded Reed-Solomon codes. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2040–2050. SIAM, Philadelphia, PA, 2025. doi:10.1137/1.9781611978322.64.

53 Madhu Sudan. Decoding of reed solomon codes beyond the error-correction bound. *J. Complex.*, 13(1):180–193, 1997. doi:10.1006/JCOM.1997.0439.

54 Nianqi Tang and Yunghsiang S. Han. A new decoding method for reed-solomon codes based on FFT and modular approach. *IEEE Trans. Commun.*, 70(12):7790–7801, 2022. doi:10.1109/TCOMM.2022.3215998.

55 Lloyd R Welch and Elwyn R Berlekamp. Error correction for algebraic block codes, December 30 1986. US Patent 4,633,470.

56 Yingquan Wu. New list decoding algorithms for reed-solomon and BCH codes. *IEEE Trans. Inf. Theory*, 54(8):3611–3630, 2008. doi:10.1109/TIT.2008.926355.

57 Jiun-Hung Yu and Hans-Andrea Loeliger. Simultaneous partial inverses and decoding interleaved reed-solomon codes. *IEEE Trans. Inf. Theory*, 64(12):7511–7528, 2018. doi:10.1109/TIT.2018.2868701.

58 Ilaria Zappatore. *Simultaneous Rational Function Reconstruction and applications to Algebraic Coding Theory*. PhD thesis, Université Montpellier, 2020.