All Polynomial Generators Preserve Distance with Mutual Correlated Agreement

Sarah Bordage
EPFL, Lausanne, Switzerland

Alessandro Chiesa
EPFL, Lausanne, Switzerland

Ziyi Guan
EPFL, Lausanne, Switzerland

Ignacio Manzur
EPFL, Lausanne, Switzerland

Abstract

A generator is a function that maps a random seed to a list of coefficients. We study generators that preserve distance to a linear code: the linear combination of any list of vectors using coefficients sampled by the generator has distance to the code no smaller than that of the original vectors, except for a small error. Distance preservation plays a central role in modern probabilistic proofs, and has been formalized in several ways. We study *mutual correlated agreement*, the strongest known form of distance preservation.

We initiate a systematic study of mutual correlated agreement, aiming to characterize the class of generators with this property. Towards this, we study polynomial generators, a rich class that includes all examples of generators considered in the distance preservation literature. Our main result is that *all polynomial generators guarantee mutual correlated agreement for every linear code*. This improves on prior work both in generality (the class of generators covered) and in parameters (the error bounds).

We additionally provide new results for the case where the linear code is a Reed–Solomon code, which is of particular interest in applications. We prove that all polynomial generators satisfy mutual correlated agreement for Reed–Solomon codes up to the Johnson bound. In particular, we improve upon the state-of-the-art by Ben-Sasson, Carmon, Ishai, Kopparty, and Saraf (FOCS 2020) and answer a question posed by Arnon, Chiesa, Fenzi, and Yogev (Eurocrypt 2025).

Along the way we develop a flexible and general toolbox for mutual correlated agreement, and are the first to establish distance preservation for generators that lie beyond polynomial generators.

2012 ACM Subject Classification Mathematics of computing → Coding theory

Keywords and phrases proximity testing, distance preservation, mutual correlated agreement

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.24

Related Version *Full Version*: `https://eprint.iacr.org/2025/2051`

Funding The authors are supported in part by the Ethereum Foundation, the Sui Foundation and Polygon Labs.

Acknowledgements We thank Giacomo Fenzi for pointing out relevant references.

1 Introduction

We study probabilistic reductions that preserve the distance of a set of vectors to a linear code. The reductions are captured via generators that, given a random seed, output coefficients used to linearly combine the vectors. We recall notions of distance preservation and then introduce the coding theory problem that we study.

Distance preservation. A *generator* G over a field $\mathbb{F}$ with output size ℓ is a function that maps a seed $\mathbf{x}$ in a set S to a coefficient vector in $\mathbb{F}^\ell$. In its simplest form, distance preservation for G states the following: for every $\mathbf{u}_1, \ldots, \mathbf{u}_\ell \in \mathbb{F}^n$ where at least one $\mathbf{u}_i$ is far (in Hamming distance) from a linear code $\mathcal{C} \subseteq \mathbb{F}^n$, the linear combination $\mathbf{u}_\mathbf{x} := \sum_{j\in[\ell]} G(\mathbf{x})_j \cdot \mathbf{u}_j$ is far from $\mathcal{C}$ with high probability over a random seed $\mathbf{x} \in S$. For example, [43] prove for the identity generator $G\colon \mathbb{F}^\ell \to \mathbb{F}^\ell$ where $G(\mathbf{x}) := \mathbf{x}$ (the ℓ coefficients are random elements in $\mathbb{F}$) that the probability that the maximum distance drops by more than half is at most $\frac{1}{|\mathbb{F}|}$; they use this to design interactive proofs of proximity. Statements about distance preservation such as these are fundamental coding-theoretic properties that play an essential role in various *batch proximity testing problems*, which involve protocols for testing whether $\mathbf{u}_1, \ldots, \mathbf{u}_\ell$ are all close to $\mathcal{C}$. Applications, however, often demand forms of distance preservation that are *more powerful and flexible* than the result in [43].

Improved distance preservation. Statements about distance preservation can be superior in several ways.

- *No distance drop:* [3, 12, 11] show that for the identity generator $G(\mathbf{x}) := \mathbf{x}$ studied in [43] the maximum distance *does not drop at all* (except for a small error over the choice of seed $\mathbf{x}$).
- *More generators:* [10] proves distance preservation for the univariate powers generator $G\colon \mathbb{F} \to \mathbb{F}^{d+1}$ where $G(x) := (1, x, \ldots, x^d)$. Several works [15, 14, 26, 41, 8] establish distance preservation properties for specific generators based on low-degree polynomials. These other specific generators are chosen to best fit the application and setting studied in those papers.
- *Quality of distance preservation:* [10] introduces distance preservation with *correlated agreement* (CA), which states that if $\mathbf{u}_\mathbf{x}$ is close to $\mathcal{C} \subseteq \mathbb{F}^n$ for many $\mathbf{x} \in S$ then there is a large subset of indices $T \subseteq [n]$ over which $\mathbf{u}_1, \ldots, \mathbf{u}_\ell$ agree with corresponding codewords in $\mathcal{C}$. [10] shows that the identity generator and the univariate powers generator have correlated agreement if $\mathcal{C}$ is a Reed–Solomon code. [5] introduces distance preservation with *mutual correlated agreement* (MCA), a strengthening that we study in this paper and discuss below. These stronger forms of distance preservation are useful in applications because they provide information on *how* the vectors agree with the code $\mathcal{C}$.

Mutual correlated agreement. We study distance preservation with *mutual correlated agreement* (MCA) [5]; informally, for many seeds $\mathbf{x} \in S$, if $\mathbf{u}_\mathbf{x} := \sum_{j\in[\ell]} G(\mathbf{x})_j \cdot \mathbf{u}_j$ agrees with an element in $\mathcal{C}$ over a large subset $T \subseteq [n]$ of indices, then each $\mathbf{u}_j$ agrees with a corresponding codeword in $\mathcal{C}$ *over the same set of indices* T. The fraction of $\mathbf{x} \in S$ for which this does not hold is the *MCA error of* G *for the code* $\mathcal{C}$.

▶ **Definition 1.** *$G\colon S \to \mathbb{F}^\ell$ has* **mutual correlated agreement** *for a linear code $\mathcal{C} \subseteq \mathbb{F}^n$ with error ϵ_{MCA} if*

$$\forall\, \mathbf{U} = \begin{pmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_\ell \end{pmatrix} \in \mathbb{F}^{\ell\times n},\ \forall\, \gamma \in [0,1], \quad \Pr_{\mathbf{x}\in S}\left[\exists\, T \subseteq [n] : \begin{array}{l} |T| \geq n\cdot(1-\gamma) \\ \wedge\, (G(\mathbf{x})\cdot\mathbf{U})|_T \in \mathcal{C}|_T \\ \wedge\, \vee_{j\in[\ell]}\, \mathbf{u}_j|_T \notin \mathcal{C}|_T \end{array}\right] \leq \epsilon_{\mathsf{MCA}}(\gamma)\ .$$

Of course, any generator has MCA for any code with the trivial error bound $\epsilon_{\mathsf{MCA}}(\gamma) := 1$; hence when we say that a generator "has MCA" we mean that the error bound $\epsilon_{\mathsf{MCA}}(\gamma)$ is not trivial.

MCA is related to mathematically elegant properties of linear codes. For example, MCA implies a natural *list-decoding commutativity*: if G has MCA for the code $\mathcal{C}$ then, except for the MCA error over a random seed $\mathbf{x}$, the list of codewords close to the combined word $\mathbf{u}_{\mathbf{x}} := \sum_{j\in[\ell]} G(\mathbf{x})_j \cdot \mathbf{u}_j$ equals the list obtained by taking the linear combination according to $G(\mathbf{x})$ of the codewords in the lists of each word $\mathbf{u}_j$; that is, "linearly combine and then list decode" equals "list decode and then linearly combine".

Properties such as list-decoding commutativity are at the heart of the power of MCA for the design and analysis of testing protocols like interactive oracle proofs (IOPs) and interactive oracle reductions (IORs). MCA has thus rapidly become the "gold standard" for distance preservation, by enabling highly efficient protocols and facilitating simpler security analyses (see, e.g., [5, 18, 8, 31]).

Unfortunately, MCA remains *poorly understood*. The MCA error of the affine space generator $G\colon \mathbb{F}^s \to \mathbb{F}^{s+1}$ where $G(x_1,\ldots,x_s) := (1, x_1, \ldots, x_s)$ for every linear code is studied in [30, 29]. However, which generators G have small MCA error and how this error depends on properties of the code $\mathcal{C}$ remains wide open. In fact, the study of distance preservation for several specific generators has led to a medley of different and suboptimal analyses (see [15, 14, 26, 41, 8]).

The prominent practical applications of MCA motivate a deeper mathematical understanding of MCA. For example, all known analyses that establish distance preservation for generators (including those in this paper) are limited by the list-decoding bound of the given code $\mathcal{C}$. Yet, establishing MCA beyond this bound (if at all possible) for codes such as Reed–Solomon codes is closely related to desirable efficiency improvements in practical hash-based succinct arguments (e.g., see [5]), so much so that Ethereum Foundation has allocated over $1M towards resolving certain conjectures about MCA [27].

Which generators have MCA? We initiate a systematic study of generators with MCA: what are the necessary and sufficient properties for a generator $G\colon S \to \mathbb{F}^\ell$ to have MCA? An initial observation is that G must be *zero-evading*, namely, the zero-evading error $\epsilon_{\mathsf{ZE}} := \max_{\mathbf{v}\in\mathbb{F}^\ell\setminus\{\mathbf{0}\}} \Pr_{\mathbf{x}\in S}[G(\mathbf{x})\cdot\mathbf{v}^\intercal = 0]$ (maximum probability of obtaining a zero for a non-zero vector) must be small because $\epsilon_{\mathsf{ZE}} \le \min_{\gamma\in[0,1]} \epsilon_{\mathsf{MCA}}(\gamma)$. We wish to understand the reverse direction:

Do all zero-evading generators have mutual correlated agreement?

In other words, can one upper bound the MCA error $\epsilon_{\mathsf{MCA}}(\gamma)$ of a generator G in terms of its zero-evading error ϵ_{ZE} (perhaps up to some loss beyond ϵ_{ZE})? We seek a fundamental understanding of why and how a zero-evading generator G has a non-trivial MCA error $\epsilon_{\mathsf{MCA}}(\gamma)$.

In this paper we focus on the distance preservation properties of *polynomial generators*, a large and flexible class of zero-evading generators that includes all examples of generators previously studied in the distance preservation literature.

▶ **Definition 2.** *For $S_1,\ldots,S_s \subseteq \mathbb{F}$, $G\colon S_1\times\cdots\times S_s \to \mathbb{F}^\ell$ is a* **polynomial generator** *if $G(\mathbf{x}) = (P_j(\mathbf{x}))_{j\in[\ell]}$ for every $\mathbf{x}\in S_1\times\cdots\times S_s$, where $P_1,\ldots,P_\ell$ are linearly independent polynomials in $\mathbb{F}[\mathtt{X}_1,\ldots,\mathtt{X}_s]$.*

1.1 Our results

Our main result is that **all polynomial generators have MCA for every linear code**. Moreover, for every polynomial generator we establish **better MCA errors for Reed–Solomon codes**, linear codes whose MCA properties have significant practical implications.

Tables 1–3 compare our results and prior art in the distance preservation literature; our error bounds **match or improve all bounds for every generator** that was previously studied. We obtain our results thanks to a flexible and general framework of tools for studying MCA that we deem of independent interest. Next we discuss our results in more detail.

MCA for arbitrary linear codes. The polynomial generator $G\colon S_1 \times \cdots \times S_s \to \mathbb{F}^\ell$ where $G(\mathbf{x}) = (P_j(\mathbf{x}))_{j\in[\ell]}$ has zero-evading error $\epsilon_{\mathsf{ZE}} \leq \sum_{i\in[s]} \frac{d_i}{|S_i|}$ and $d_i := \max_{j\in[\ell]}\{\deg_{\mathtt{X}_i}(P_j)\}$ is the maximum degree of $\mathtt{X}_i$ in $(P_j)_{j\in[\ell]}$. Hence for G we expect that $\min_{\gamma\in[0,1]} \epsilon_{\mathsf{MCA}}(\gamma) \geq \sum_{i\in[s]} \frac{d_i}{|S_i|}$. The theorem below bounds the MCA error ϵ_{MCA} of this polynomial generator *for every linear code* $\mathcal{C}$, with a bound corresponding to a unique-decoding analysis and another corresponding to a list-decoding analysis. (While in introductory sections we use big-O notation, all technical sections include explicit constants in the upper bounds.)

▶ **Theorem 3** (informal). *Let $G\colon S_1 \times S_2 \times \cdots \times S_s \to \mathbb{F}^\ell$ be a polynomial generator where $G(\mathbf{x}) = (P_j(\mathbf{x}))_{j\in[\ell]}$; let $d_i := \max_{j\in[\ell]}\{\deg_{\mathtt{X}_i}(P_j)\}$ for every $i \in [s]$. Then for every linear code $\mathcal{C} \subseteq \mathbb{F}^n$ with relative distance $\delta_{\mathcal{C}}$ and tradeoff parameter $\eta \in (0,1)$, G has MCA for $\mathcal{C}$ with error*

$$\epsilon_{\mathsf{MCA}}(\gamma) := \begin{cases} \max\{n\cdot\gamma, 1\} \cdot \sum_{i\in[s]} \frac{d_i}{|S_i|} & \text{if } \gamma \leq \delta_{\mathcal{C}}/(\max_{i\in[s]} d_i + 2) \\ O\left(\frac{n}{\eta}\right) \cdot \sum_{i\in[s]} \frac{d_i}{|S_i|} & \text{if } \gamma \leq 1 - (1-\delta_{\mathcal{C}}+\eta)^{\frac{1}{\max_{i\in[s]} d_i + 2}} \end{cases}.$$

Both upper bounds are the zero-evading error of G with a multiplicative overhead; the two bounds are incomparable, both in applicability (whether the bound holds based on γ) and the achieved error bound. Theorem 3 *matches or improves all bounds previously established for specific generators*; see Tables 1 and 2 for a comparison. We prove that the error bound in the unique-decoding setting (the upper one) is *tight* for the univariate powers generator $G(x) = (1, x, \ldots, x^s)$ (see the discussion in Section 2.3).

The affine space generator $G\colon \mathbb{F}^s \to \mathbb{F}^{s+1}$ where $G(x_1, \ldots, x_s) := (1, x_1, \ldots, x_s)$ is a notable special case; its zero-evading error is $\epsilon_{\mathsf{ZE}} = \frac{1}{|\mathbb{F}|}$. Numerous works obtain error bounds on its distance preservation properties [43, 3, 11, 10, 30, 29]. For any given linear code $\mathcal{C}$, directly applying Theorem 3 to this generator recovers the best bounds in prior work [30]:

$$\epsilon_{\mathsf{MCA}}(\gamma) := \begin{cases} \max\{n\cdot\gamma, 1\} \cdot \frac{s}{|\mathbb{F}|} & \text{if } \gamma \leq \delta_{\mathcal{C}}/3 \\ O\left(\frac{n}{\eta}\right) \cdot \frac{s}{|\mathbb{F}|} & \text{if } \gamma \leq 1 - (1-\delta_{\mathcal{C}}+\eta)^{\frac{1}{3}} \end{cases}.$$

In fact, for any linear code $\mathcal{C} \subseteq \mathbb{F}^n$, we show that the MCA error of the affine space generator is equal to the MCA error of the affine line generator $\mathbb{F} \to \mathbb{F}^2$; $x \mapsto (1, x)$ up to a small, constant multiplicative overhead.

▶ **Lemma 4** (informal). *Let $\mathcal{C} \subseteq \mathbb{F}^n$ be a linear code. Then the generator $G\colon \mathbb{F}^s \to \mathbb{F}^{s+1}$ such that $G(x_1, \ldots, x_s) := (1, x_1, \ldots, x_s)$ has MCA for $\mathcal{C}$ with error*

$$\epsilon_{\mathsf{MCA}}(\gamma) := \begin{cases} O(\max\{n\cdot\gamma, 1\}) \cdot \frac{1}{|\mathbb{F}|} & \text{if } \gamma \leq \delta_{\mathcal{C}}/3 \\ O\left(\frac{n}{\eta}\right) \cdot \frac{1}{|\mathbb{F}|} & \text{if } \gamma \leq 1 - (1-\delta_{\mathcal{C}}+\eta)^{\frac{1}{3}} \end{cases}.$$

This gives an MCA error for the affine space generator that is *independent of the number of variables* s, and leads to the tightest bounds on the MCA error for this generator. This agrees with the fact that $\epsilon_{\mathsf{ZE}} = \frac{1}{|\mathbb{F}|}$ is independent of s. Moreover, our reduction does not impose a lower bound on the field size, contrary to prior analyses for CA [33].

[11] constructs a linear code $\mathcal{C}$ and words $\mathbf{u}_1, \mathbf{u}_2$ that are far from $\mathcal{C}$ such that their linear combination via the affine line generator $x \mapsto (1, x)$ drops with probability almost one. This gives intuition for why a result that holds for every linear code, such as the above lemma, holds only up to some distance γ and incurs errors that get close to 1 in some parameter settings (e.g., when the blocklength n of $\mathcal{C}$ approaches $|\mathbb{F}|$).

MCA for Reed–Solomon codes. Numerous highly efficient probabilistic proofs rely on distance preservation properties of Reed–Solomon codes (e.g., [3, 9, 11, 13, 26, 4, 48, 5, 17, 46, 24, 41, 19, 20, 18, 7, 8, 42, 31]). For Reed–Solomon codes, [10] proves *correlated agreement* for the univariate powers generator and the affine space generator, achieving better error bounds than what is known for arbitrary linear codes. We strengthen their result to achieve MCA and use our framework to extend it to hold for all polynomial generators. Below, $\mathsf{RS}[\mathbb{F}, D, k]$ denotes the Reed–Solomon code consisting of evaluations of polynomials over $\mathbb{F}$ of degree less than k over $D \subseteq \mathbb{F}$; we let $n := |D|$ and $\rho := k/n$.

▶ **Theorem 5** (informal). *Let $G\colon \mathbb{F}^s \to \mathbb{F}^\ell$ be a polynomial generator where $G(\mathbf{x}) = (P_j(\mathbf{x}))_{j\in[\ell]}$; let $d_i := \max_{j\in[\ell]}\{\deg_{\mathbf{X}_i}(P_j)\}$ for every $i \in [s]$. Then for every tradeoff parameter $m \geq 3$, G has MCA for $\mathsf{RS}[\mathbb{F}, D, k]$ with error*

$$\epsilon_{\mathsf{MCA}}(\gamma) := O\left(\frac{m^7 \cdot n^2}{\rho^{3/2}}\right) \cdot \frac{\sum_{i\in[s]} d_i}{|\mathbb{F}|} \quad \text{if } \gamma \leq 1 - \left(1 + \frac{1}{2m}\right) \cdot \sqrt{\rho}\ .$$

The error bound in Theorem 5 matches the one for correlated agreement in [10] (when fixing the specific generators studied therein). Table 3 gives a comparison of our bound on Reed–Solomon codes with prior art (and similarly for Reed–Muller codes). As a special case, Theorem 5 resolves a question asked in [5] on the MCA error for the univariate powers generator $x \mapsto (1, x, \ldots, x^d)$ for Reed–Solomon codes; prior to this work [38] resolves this question for the affine line generator $x \mapsto (1, x)$ $(d = 1)$.

List decoding and MCA. If a linear code $\mathcal{C}$ is list-decodable with list size L up to distance δ, then the affine space generator has a small MCA error $\epsilon_{\mathsf{MCA}}(\gamma)$ for $\mathcal{C}$ provided $\gamma < 1 - \sqrt{1-\delta}$ [29]. This connection is particularly interesting for code families that are list-decodable up to capacity, e.g., random linear codes [37, 36, 22, 40], random Reed–Solomon codes [44, 45, 34, 28, 32, 16, 35, 2, 39], folded Reed–Solomon codes, and univariate multiplicity codes [21]. However, for codes in this paper, namely arbitrary Reed–Solomon codes and arbitrary linear codes, the best known list-decoding bound is the Johnson bound, which is $\approx 1-(1-\delta_{\mathcal{C}})^{1/2}$ if the code $\mathcal{C}$ has distance $\delta_{\mathcal{C}}$. Applying [29]'s result would yield a non-trivial MCA error for the affine space generator only for γ within distance $\approx 1-(1-\delta_{\mathcal{C}})^{1/4}$ (the "double Johnson bound"). Our results for the affine space generator (implied by Lemma 4 and Theorem 5) subsume the application of [29] to these codes.

Operations on codes. Most of this work investigates which generators have MCA for general linear codes (and sometimes specific ones like Reed–Solomon codes). An orthogonal question is what happens to MCA when we fix an arbitrary generator and apply various operations on codes. We prove for example that if any generator G has MCA for some linear code $\mathcal{C}$ with error ϵ_{MCA} then G has MCA for the k-interleaving $\mathcal{C}^k$ with error at most $k \cdot \epsilon_{\mathsf{MCA}}$. Similarly we obtain upper bounds on the MCA error of arbitrary generators when we take the direct product of codes, or when we concatenate them.

Open questions. Theorem 3 makes progress towards a characterization of generators with MCA by providing a large class of generators with MCA. Yet we already know that more can be said: we prove that a zero-evading generator in [1], which is *not* a polynomial generator, has MCA (see Section 2.6). This tells us that MCA holds beyond polynomial generators and a characterization of MCA remains open.

The behavior of MCA under natural operations on codes remains to be understood. Of particular interest are tensor products of codes, and distance amplification procedures on codes.

For the special case of Reed–Solomon codes, Theorem 5 states that any polynomial generator has MCA for γ close to the code's list-decoding bound $1 - \sqrt{\rho}$. In probabilistic proofs, distance preservation at a larger distance typically implies more efficient verification. Can one bound the MCA error for distances γ beyond $1 - \sqrt{\rho}$? [5] conjectures MCA for the univariate powers generator for γ close to the code's capacity $1 - \rho$, and similar conjectures on Reed–Solomon codes are sometimes used in practice for efficiency. Yet [25, 23] shows that this is too much to hope for: there is a family of Reed–Solomon codes for which distance preservation fails for the affine line generator $x \mapsto (1, x)$ when γ is close to the code's capacity $1 - \rho$. Understanding whether MCA holds between the list-decoding and capacity bounds remains open.

1.2 Comparison with prior work

We give a comparison with prior work that studies distance preservation with CA and with MCA.

Table 1: arbitrary linear codes in the unique decoding setting [3] proves that the identity generator has CA error $n \cdot \delta_{\mathcal{C}} \cdot \frac{1}{|\mathbb{F}|}$ within distance $\frac{\delta_{\mathcal{C}}}{3}$. Our results imply that the affine space generator $(x_1, \ldots, x_s) \mapsto (1, x_1, \ldots, x_s)$ has MCA (hence, CA) with the improved error $\frac{n \cdot \delta_{\mathcal{C}}}{3} \cdot \frac{1}{|\mathbb{F}|}$ within distance $\frac{\delta_{\mathcal{C}}}{3}$.

Table 2: arbitrary linear codes in the list-decoding setting [11] shows that the affine line generator $x \mapsto (1, x)$ has CA for every linear code almost up to distance $1 - (1 - \delta_{\mathcal{C}})^{1/3}$, known as the "1.5 Johnson bound". [30] obtains the stronger MCA for the affine space generator $(x_1, \ldots, x_s) \mapsto (1, x_1, \ldots, x_s)$ for every linear code almost up to the 1.5 Johnson bound. [47] improves upon the constant in the error of [30] in the case of the affine line generator $x \mapsto (1, x)$. [8] uses [47] to prove MCA for the multilinear equality generator $\mathbf{x} \mapsto (\widehat{\mathsf{eq}}(\mathbf{v}, \mathbf{x}))_{\mathbf{v} \in \{0,1\}^s}$ almost up to the 1.5 Johnson bound. We recover all of these specific results via our Theorem 3 (MCA for all polynomial generators). Moreover, for the affine space generator $(x_1, \ldots, x_s) \mapsto (1, x_1, \ldots, x_s)$, we improve over prior bounds by removing the dependency on s.

Table 3: Reed–Solomon and Reed–Muller [10] shows that both the affine space generator $(x_1, \ldots, x_s) \mapsto (1, x_1, \ldots, x_s)$ and the univariate powers generator $x \mapsto (1, x, \ldots, x^d)$ have CA for Reed–Solomon codes up to the Johnson bound (i.e., beyond the 1.5 Johnson bound known for arbitrary linear codes). We show that the same bound holds for MCA and for every polynomial generator.

[41] uses [10] to show that the univariate powers generator has CA for Reed–Muller codes up to some distance parameter well below the list-decoding bound. Our results imply a tighter bound on the MCA error for the univariate powers generator for Reed–Muller codes.

Table 1 Results on correlated agreement (CA) and mutual correlated agreement (MCA) based on unique-decoding analyses, for linear/affine space generators and arbitrary linear codes $\mathcal{C} \subseteq \mathbb{F}^n$ of relative distance $\delta_{\mathcal{C}}$.

	Generator	(M)CA	Distance	Error
[3]	$(\mathtt{X}_1, \ldots, \mathtt{X}_s)$	CA	$\delta_{\mathcal{C}}/3$	$n \cdot \delta_{\mathcal{C}} \cdot \frac{1}{\lvert\mathbb{F}\rvert}$
This work	$(1, \mathtt{X}_1, \ldots, \mathtt{X}_s)$	CA	$\delta_{\mathcal{C}}/3$	$\frac{n \cdot \delta_{\mathcal{C}}}{3} \cdot \frac{1}{\lvert\mathbb{F}\rvert}$
This work	$(1, \mathtt{X}_1, \ldots, \mathtt{X}_s)$	MCA	$\delta_{\mathcal{C}}/3$	$\frac{n \cdot \delta_{\mathcal{C}}}{3} \cdot \frac{1}{\lvert\mathbb{F}\rvert}$

Table 2 Comparison of results on correlated agreement (CA) and mutual correlated agreement (MCA) for several generators and arbitrary linear codes $\mathcal{C} \subseteq \mathbb{F}^n$ of relative distance $\delta_{\mathcal{C}}$. The parameter $\eta \in (0, 1)$ is a tradeoff between distance and error. In the table we assume that $d_i \leq n$ for every i. The polynomial $\widehat{\mathsf{eq}}$ is the multilinear equality polynomial.

	Generator	(M)CA	Distance	Error
[11]	$(1, \mathtt{X})$	CA	$1 - (1 - \delta_{\mathcal{C}} + \eta)^{1/3}$	$O\left(\frac{1}{\eta^2}\right) \cdot \frac{1}{\lvert\mathbb{F}\rvert}$
[6]	$(\vec{\mathtt{X}}^{\mathbf{v}})_{\mathbf{v} \in \{0,1\}^s}$	CA	$1 - (1 - \delta_{\mathcal{C}} + \eta)^{1/3}$	$O\left(\frac{2s}{\eta^2}\right) \cdot \frac{1}{\lvert\mathbb{F}\rvert}$
[30]	$(1, \mathtt{X}_1, \ldots, \mathtt{X}_s)$	MCA	$1 - (1 - \delta_{\mathcal{C}} + \eta)^{1/3}$	$O\left(\frac{n \cdot s}{\eta}\right) \cdot \frac{1}{\lvert\mathbb{F}\rvert}$
[47]	$(1, \mathtt{X})$	MCA	$1 - (1 - \delta_{\mathcal{C}} + \eta)^{1/3}$	$O\left(\frac{n}{\eta}\right) \cdot \frac{1}{\lvert\mathbb{F}\rvert}$
[8]	$(\widehat{\mathsf{eq}}(\mathbf{v}, \vec{\mathtt{X}}))_{\mathbf{v} \in \{0,1\}^s}$	MCA	$1 - (1 - \delta_{\mathcal{C}} + \eta)^{1/3} - \eta$	$O\left(\frac{3^s}{s \cdot \eta^2}\right) \cdot \frac{s}{\lvert\mathbb{F}\rvert}$
This work	$(P_j(\vec{\mathtt{X}}))_{j \in [\ell]}$, $\max_{j \in [\ell]} \deg_{\mathtt{X}_i}(P_j) \leq d_i$	MCA	$1 - (1 - \delta_{\mathcal{C}} + \eta)^{\frac{1}{\max_{i \in [s]} d_i + 2}}$	$O\left(\frac{n}{\eta}\right) \cdot \frac{\sum_i d_i}{\lvert\mathbb{F}\rvert}$
	$(1, \mathtt{X}_1, \ldots, \mathtt{X}_s)$	MCA	$1 - (1 - \delta_{\mathcal{C}} + \eta)^{1/3}$	$O\left(\frac{n}{\eta}\right) \cdot \frac{1}{\lvert\mathbb{F}\rvert}$
	$(\widehat{\mathsf{eq}}(\mathbf{v}, \vec{\mathtt{X}}))_{\mathbf{v} \in \{0,1\}^s}$	MCA	$1 - (1 - \delta_{\mathcal{C}} + \eta)^{1/3}$	$O\left(\frac{n \cdot s}{\eta}\right) \cdot \frac{s}{\lvert\mathbb{F}\rvert}$

Table 3 Comparison of results on correlated agreement (CA) and mutual correlated agreement (MCA) for some generators and specific linear codes. $\mathsf{RS}[\mathbb{F}, n, \rho]$ is the Reed–Solomon code of evaluations of univariate polynomials over $\mathbb{F}$ of degree at most $\rho \cdot n - 1$ over a domain of size n. For an extension $\mathbb{B} \subseteq \mathbb{F}$, $\mathsf{RM}[\mathbb{F}, \mathbb{B}^2, k]$ consists of bivariate polynomials $f \colon \mathbb{B}^2 \to \mathbb{F}$ of total degree at most k. The integer $m \geq 3$ is a tradeoff between distance and error. In the table we assume that $d \leq \lvert\mathbb{B}\rvert^2$.

	Code	Generator	(M)CA	Distance	Error
[10]	$\mathsf{RS}[\mathbb{F}, n, \rho]$	$(\mathtt{X}_1, \ldots, \mathtt{X}_s)$	CA	$1 - \left(1 + \frac{1}{2m}\right)\sqrt{\rho}$ for $m \geq 3$	$O\left(\frac{m^7 \cdot n^2}{\rho^{3/2}}\right) \cdot \frac{1}{\lvert\mathbb{F}\rvert}$
[10]		$(1, \mathtt{X}, \ldots, \mathtt{X}^d)$	CA		$O\left(\frac{m^7 \cdot n^2}{\rho^{3/2}}\right) \cdot \frac{d}{\lvert\mathbb{F}\rvert}$
[38]		$(1, \mathtt{X})$	MCA		$O\left(\frac{m^7 \cdot n^2}{\rho^{3/2}}\right) \cdot \frac{1}{\lvert\mathbb{F}\rvert}$
This work		$(P_j(\vec{\mathtt{X}}))_{j \in [\ell]}$, $\max_j \deg_{\mathtt{X}_i}(P_j) \leq d_i$	MCA		$O\left(\frac{m^7 \cdot n^2}{\rho^{3/2}}\right) \cdot \frac{\sum_i d_i}{\lvert\mathbb{F}\rvert}$
[41]	$\mathsf{RM}[\mathbb{F}, \mathbb{B}^2, k]$	$(1, \mathtt{X}, \ldots, \mathtt{X}^d)$	CA	$1 - O\left(\frac{d}{\lvert\mathbb{B}\rvert^{\frac{1}{14d+2}}}\right)$	$\frac{\lvert\mathbb{B}\rvert^4}{k^2 \cdot d} \cdot \frac{d}{\lvert\mathbb{F}\rvert}$
This work		$(1, \mathtt{X}, \ldots, \mathtt{X}^d)$	MCA	$1 - \left(\frac{k}{\lvert\mathbb{B}\rvert} + \eta\right)^{\frac{1}{d+2}}$	$O\left(\frac{\lvert\mathbb{B}\rvert^2}{\eta}\right) \cdot \frac{d}{\lvert\mathbb{F}\rvert}$

2 Technical overview

Prior work with bounds for MCA does not offer suitable starting points towards Theorem 3 and Theorem 5, as existing techniques typically target either specific generators or specific codes. To overcome these limitations, we develop a toolbox for MCA for our setting and explain how it helps us prove Theorem 3 and Theorem 5.

In Section 2.1 we compare the MCA and zero-evading errors. In Section 2.2 we reduce proving Theorem 3 to establishing MCA for the univariate powers generator. In Section 2.3 we sketch how to obtain MCA for the univariate powers generator for all linear codes. In Section 2.4 we sketch the proof of Lemma 4. In Section 2.5 we explain the main idea underlying Theorem 5. Lastly, in Section 2.6 we obtain a bound on the MCA error of a non-polynomial generator for all linear codes and summarize our toolbox for MCA.

2.1 On mutual correlated agreement vs. zero-evading error

We outline why the MCA error is lower bounded by the zero-evading error. This justifies our focus on the class of zero-evading generators, and serves as an opportunity to build intuition on the definition of MCA. A simple initial observation is that the MCA error of $G\colon S \to \mathbb{F}^\ell$ is monotonically non-decreasing, almost by definition. Therefore, it suffices to show that the MCA error at 0 is lower bounded by the zero-evading error:

$$\epsilon_{\mathsf{MCA}}(0) \geq \Pr_{\mathbf{x}\in S}\left[\begin{array}{l} G(\mathbf{x})\cdot\mathbf{U}\in\mathcal{C} \\ \wedge\, \exists j\in[\ell],\ \mathbf{u}_j\notin\mathcal{C}\end{array}\right] \geq \max_{\mathbf{v}\in\mathbb{F}^\ell\setminus\{\mathbf{0}\}} \Pr_{\mathbf{x}\in S}\left[G(\mathbf{x})\cdot\mathbf{v}^\intercal=0\right] \ .$$

The right-hand side is mainly about cancellations: it measures the maximum probability that a non-trivial linear combination of $G(\mathbf{x})$ equals zero. Proving the lower bound corresponds to the intuition that if the combined word $G(\mathbf{x})\cdot\mathbf{U}$ falls into the code while some of the individual words $\mathbf{u}_j$ do not, this must be due to similar cancellations. This can be concretized as follows. Without loss of generality the code is not the entire space, so there exists a word $\mathbf{u}\in\mathbb{F}^n\setminus\mathcal{C}$. Fix a non-zero vector $\mathbf{v}\in\mathbb{F}^\ell\setminus\{\mathbf{0}\}$ and define words $\mathbf{u}_1,\ldots,\mathbf{u}_\ell$ such that, for every $j\in[\ell]$, $\mathbf{u}_j := \mathbf{v}_j\cdot\mathbf{u}$. Since $\mathbf{v}$ is not the zero vector, there is j such that $\mathbf{u}_j\notin\mathcal{C}$. Then, for every seed $\mathbf{x}\in S$, the linear combination $G(\mathbf{x})\cdot\mathbf{U}$ is equal to $(G(\mathbf{x})\cdot\mathbf{v}^\intercal)\cdot\mathbf{u}$ and thus belongs to the code if and only if $G(\mathbf{x})\cdot\mathbf{v}^\intercal=0$. Since $\mathbf{v}$ was arbitrary, this establishes the desired lower bound.

With a bit more effort, we can show that the MCA error at 0 in fact *equals* the zero-evading error. This amounts to showing that there exists, for every $\mathbf{u}_1,\ldots,\mathbf{u}_\ell\in\mathbb{F}^n$, a non-zero vector $\mathbf{v}\in\mathbb{F}^\ell\setminus\{\mathbf{0}\}$ such that

$$\Pr_{\mathbf{x}\in S}\left[\begin{array}{l} G(\mathbf{x})\cdot\mathbf{U}\in\mathcal{C} \\ \wedge\, \exists j\in[\ell],\ \mathbf{u}_j\notin\mathcal{C}\end{array}\right] \leq \Pr_{\mathbf{x}\in S}\left[G(\mathbf{x})\cdot\mathbf{v}^\intercal=0\right] \ .$$

This is neatly achieved by introducing parity checks for the code $\mathcal{C}$ and considering the linear combinations of these parity checks that are violated by the words $\mathbf{u}_1,\ldots,\mathbf{u}_\ell$.

Towards upper bounds on ϵ_{MCA}. The previous result gives hope that one might, in general, be able to upper bound the MCA error ϵ_{MCA} of a generator $G\colon S\to\mathbb{F}^\ell$ via (some function of) its zero-evading error ϵ_{ZE}. We do not know this to be true. In this paper we establish results for the MCA error ϵ_{MCA} when G is a polynomial generator (or some other special cases). For these generators, we show that the zero-evading error *does* play a useful role in upper bounding ϵ_{MCA}. We explain this connection in Section 2.3.

2.2 All polynomials generators have MCA for all linear codes

Our approach to prove Theorem 3 consists of two main steps:

(i) we develop tools to reduce MCA of polynomial generators to MCA of the univariate powers generator, with minimum parameter loss; and

(ii) we prove that the univariate powers generator has MCA for all linear codes.

Let $G(\mathbf{x}) := (P_j(\mathbf{x}))_{j\in[s]}$ be a polynomial generator. Each polynomial P_j is a linear combination of monomials in the variables $\mathtt{X}_1, \dots, \mathtt{X}_s$, where the individual degree of $\mathtt{X}_i$ is at most $\deg_{\mathtt{X}_i}(P_j)$. Define $d_i := \max_{j\in[\ell]}\{\deg_{\mathtt{X}_i}(P_j)\}$ and $k := \prod_{i\in[s]}(1+d_i)$. There exists a matrix $\mathbf{A} \in \mathbb{F}^{k\times\ell}$ such that

$$(P_j(\mathtt{X}_1, \dots, \mathtt{X}_s))_{j\in[\ell]} = \left(\bigotimes_{i\in[s]}(1, \mathtt{X}_i, \dots, \mathtt{X}_i^{d_i})\right) \cdot \mathbf{A} \ .$$

The expression $\bigotimes_{i\in[s]}(1, \mathtt{X}_i, \dots, \mathtt{X}_i^{d_i})$ represents the tensor product of the univariate powers generators. The polynomials $(P_j)_{j\in[\ell]}$ are linearly independent, so $\mathbf{A}$ has a left inverse (any non-trivial linear dependence on the rows of $\mathbf{A}$ yields a non-trivial linear dependence on $(P_j)_{j\in[\ell]}$). In sum, every polynomial generator can be viewed as a linear transformation of the tensor products of univariate powers generators. In light of this, we prove lemmas to track MCA across these two operations.

▶ **Lemma 6** (tensor products preserve MCA)**.** *Let $G\colon S \to \mathbb{F}^\ell$, $G'\colon S' \to \mathbb{F}^{\ell'}$ be generators that have MCA for a linear code $\mathcal{C} \subseteq \mathbb{F}^n$ with respective errors $\epsilon_{\mathsf{MCA}}, \epsilon'_{\mathsf{MCA}}$. Then the tensor product generator $G \otimes G'$ has MCA for $\mathcal{C}$ with error $\epsilon_{\mathsf{MCA}} + \epsilon'_{\mathsf{MCA}}$.*

▶ **Lemma 7** (linear transformations preserve MCA)**.** *Let $G\colon S \to \mathbb{F}^\ell$ be a generator that has MCA for a linear code $\mathcal{C} \subseteq \mathbb{F}^n$ with error ϵ_{MCA}. For every $\tilde{\ell} \in \mathbb{N}$ and $\mathbf{A} \in \mathbb{F}^{\ell\times\tilde{\ell}}$ with a left inverse, let $\tilde{G}\colon S \to \mathbb{F}^{\tilde{\ell}}$ be the generator defined as $\tilde{G}(\mathbf{x}) := G(\mathbf{x}) \cdot \mathbf{A}$. Then $\tilde{G}$ has MCA for $\mathcal{C}$ with error ϵ_{MCA}.*

From Lemmas 6 and 7 we deduce that the MCA error $\epsilon_{\mathsf{MCA}}(\gamma)$ for $\mathcal{C}$ of the polynomial generator G is

$$\epsilon_{\mathsf{MCA}}(\gamma) = \sum_{i\in[s]} \xi_{\mathcal{C},d_i,|S_i|}(\gamma)$$

where $\xi_{\mathcal{C},d,|S|}$ is the MCA error for $\mathcal{C}$ of the univariate powers generator $G\colon S \to \mathbb{F}^{d+1}$ where $G(x) := (1, x, \dots, x^d)$. Next, we turn our attention to establishing the MCA error of the univariate powers generator.

▶ Remark 8 (MDS generators). The univariate powers generator is a special case of an *MDS generator*. One can associate to each generator G a corresponding linear code $\mathcal{C}_G$; we say G is MDS if $\mathcal{C}_G$ meets the Singleton bound. In fact, we prove a generalization of Theorem 9: every *MDS generator has MCA for arbitrary linear codes*. We believe this generalization is of independent interest as MDS generators are not necessarily polynomial generators.

2.3 The univariate powers generator

Our theorem below establishes MCA of the univariate powers generator for every linear code.

▶ **Theorem 9.** *Let $G\colon S \subseteq \mathbb{F} \to \mathbb{F}^{d+1}$ be the univariate powers generator where $G(x) := (1, x, \ldots, x^d)$. Let $\mathcal{C} \subseteq \mathbb{F}^n$ be a linear code with relative distance $\delta_{\mathcal{C}}$. Then, for every tradeoff parameter $\eta \in (0,1)$, G has MCA for $\mathcal{C}$ with error $\epsilon_{\mathsf{MCA}}(\gamma) := \xi_{\mathcal{C},d,|S|}(\gamma)$ where*

$$\xi_{\mathcal{C},d,|S|}(\gamma) := \begin{cases} n \cdot \gamma \cdot \frac{d}{|S|} & \text{if } \gamma \leq \delta_{\mathcal{C}}/(d+2) \\ O\left(\frac{n}{\eta}\right) \cdot \frac{d}{|S|} & \text{if } \gamma \leq 1 - (1 - \delta_{\mathcal{C}} + \eta)^{\frac{1}{d+2}} \end{cases} .$$

Moreover, we prove that Theorem 9 is *tight* for the regime $\gamma \leq \delta_{\mathcal{C}}/(d+2)$: we show that for every linear code $\mathcal{C} \subseteq \mathbb{F}^n$ with relative minimum distance $\delta_{\mathcal{C}}$ and $\gamma < \delta_{\mathcal{C}}/2$, the MCA error of the univariate powers generator (when $S = \mathbb{F}$) is lower bounded by $n \cdot \gamma \cdot \frac{d}{|\mathbb{F}|}$. In fact, we prove the lower bound for the weakest form of distance-preservation instead of MCA.

Next, we outline the main ideas behind the result: in Section 2.3.1 we discuss the upper bound arising in the unique-decoding setting, and in Section 2.3.2 we discuss the upper bound arising in the list-decoding setting. Our analysis in the list-decoding setting is directly inspired by the combinatorial techniques in [30, 47].

2.3.1 Unique-decoding setting

Fix $\mathbf{u}_1, \ldots, \mathbf{u}_{d+1} \in \mathbb{F}^n$. We define the set of "bad seeds":

$$\mathsf{B} := \left\{ x \in S \;\middle|\; \exists\, T_x \subseteq [n] : \begin{array}{l} |T_x| \geq n \cdot (1-\gamma) \\ \wedge\ (\sum_{j \in [d+1]} G(x)_j \cdot \mathbf{u}_j)|_{T_x} \in \mathcal{C}|_{T_x} \\ \wedge\ \vee_{j \in [d+1]}\, \mathbf{u}_j|_{T_x} \notin \mathcal{C}|_{T_x} \end{array} \right\} .$$

The goal is to show $|\mathsf{B}| \leq n \cdot \gamma \cdot d$. Assume without loss of generality that $|\mathsf{B}| \geq d+1$ (else the claim holds trivially). For every $x \in \mathsf{B}$, define $\mathbf{u}_x := \sum_{j \in [d+1]} G(x)_j \cdot \mathbf{u}_j$ and let T_x be a set specified in the definition of B (break ties arbitrarily); as we are within the unique decoding bound, there is a unique codeword $\mathbf{c}_x \in \mathcal{C}$ such that $\mathbf{u}_x|_{T_x} = \mathbf{c}_x|_{T_x}$. We first argue that it suffices to find codewords $\mathbf{c}_1^*, \ldots, \mathbf{c}_{d+1}^* \in \mathcal{C}$ such that

$$\forall\, x \in \mathsf{B},\ \mathbf{c}_x = \sum_{j \in [d+1]} G(x)_j \cdot \mathbf{c}_j^* \ . \tag{1}$$

Consider the agreement set T defined as

$$T := \{ i \in [n] : \forall\, j \in [d+1], \mathbf{u}_j[i] = \mathbf{c}_j^*[i] \} \ .$$

By definition of B, for every $x \in \mathsf{B}$ there exists $i \in T_x$ with $i \notin T$. Moreover, note that $T \subseteq T_x$ because

$$\mathbf{u}_x|_T = \sum_{j \in [d+1]} G(x)_j \cdot \mathbf{u}_j|_T = \sum_{j \in [d+1]} G(x)_j \cdot \mathbf{c}_j^*|_T = \mathbf{c}_x|_T \ .$$

We deduce that, $T \subsetneq T_x$.

Set $\mathbf{v}_i := (\mathbf{u}_1[i] - \mathbf{c}_1^*[i], \ldots, \mathbf{u}_{d+1}[i] - \mathbf{c}_{d+1}^*[i])$ for every $i \in [n]$. The vector $\mathbf{v}_i$ is non-zero precisely when $i \notin T$. For every $i \in T_x$, $\mathbf{u}_x[i] = \mathbf{c}_x[i]$ by definition, so $G(x) \cdot \mathbf{v}_i^{\mathsf{T}} = \mathbf{u}_x[i] - \mathbf{c}_x[i] = 0$. Hence, for every $i \in T_x \setminus T$, it holds that

(i) $\mathbf{v}_i$ is non-zero and

(ii) $G(x) \cdot \mathbf{v}_i^{\mathsf{T}} = 0$.

This yields the following equivalent characterization of the set of bad seeds:

$$\mathsf{B} = \{x \in \mathsf{B} : \exists\, i \in T_x \setminus T,\ G(x) \cdot \mathbf{v}_i^\intercal = 0\} \ .$$

We use this characterization to bound $|\mathsf{B}|$ as follows:

$$\begin{aligned} |\mathsf{B}| &= |\{x \in \mathsf{B} : \exists\, i \in T_x \setminus T,\ G(x) \cdot \mathbf{v}_i^\intercal = 0\}| \\ &\leq \sum_{i \in [n] \setminus T} |\{x \in S : G(x) \cdot \mathbf{v}_i^\intercal = 0\}| \\ &\leq n \cdot d \ , \end{aligned} \tag{2}$$

where the last inequality follows from the zero-evading error of G. In fact, with more effort in the technical sections we can lower bound the size of T by $n \cdot (1-\gamma)$, which leads to the claimed upper bound $|\mathsf{B}| \leq n \cdot \gamma \cdot d$.

In sum, we reduce the problem of bounding the size of B to bounding the size of $T_x \setminus T$ and the zero-evading error of G, once there is an agreement set T such that $T \subsetneq T_x$ for all $x \in \mathsf{B}$. We use this idea also to bound the MCA error in the list-decoding regime in Section 2.3.2 and to show Theorem 5 in Section 2.5.

We are left to show the existence of $\mathbf{c}_1^*, \ldots, \mathbf{c}_{d+1}^*$ satisfying Equation (1). Fix distinct $x_1, \ldots, x_{d+1} \in B$. Let $\mathbf{M} \in \mathbb{F}^{(d+1)\times(d+1)}$ be the matrix whose i-th row is $(1, x_i, \ldots, x_i^d)$; $\mathbf{M}$ is invertible because it is a Vandermonde matrix. For every $i \in [d+1]$, let $\mathbf{c}_i$ be the (unique) closest codeword to $\mathbf{u}_{x_i}$. We construct $\mathbf{c}_1^*, \ldots, \mathbf{c}_{d+1}^*$ as follows:

$$\begin{pmatrix} \mathbf{c}_1^* \\ \vdots \\ \mathbf{c}_{d+1}^* \end{pmatrix} := \mathbf{M}^{-1} \cdot \begin{pmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_{d+1} \end{pmatrix} \ .$$

For every $x \in \mathsf{B}$, a repeated application of the triangle inequality shows that

$$\delta\left(\mathbf{c}_x, G(x) \cdot \begin{pmatrix} \mathbf{c}_1^* \\ \vdots \\ \mathbf{c}_{d+1}^* \end{pmatrix}\right) < (d+2) \cdot \gamma \leq \delta_{\mathcal{C}} \ .$$

The last inequality uses the fact that $\gamma \leq \delta_{\mathcal{C}}/(d+2)$. Since $\delta_{\mathcal{C}}$ is $\mathcal{C}$'s minimum distance, we deduce the desired property:

$$\mathbf{c}_x = G(x) \cdot \begin{pmatrix} \mathbf{c}_1^* \\ \vdots \\ \mathbf{c}_{d+1}^* \end{pmatrix} \ .$$

2.3.2 List-decoding setting

We sketch our proof that, when $\gamma \leq 1 - (1 - \delta_{\mathcal{C}} + \eta)^{\frac{1}{d+2}}$, the set of bad seeds can be upper bounded as

$$|\mathsf{B}| \leq O\left(\frac{n}{\eta}\right) \cdot d \ . \tag{3}$$

The main challenge is that, as we are not guaranteed to be in the unique decoding setting, there is no notion of a unique closest codeword. We cannot find fixed $\mathbf{c}_1^*, \ldots, \mathbf{c}_{d+1}^* \in \mathcal{C}$ such that $\mathbf{c}_x = \sum_{j \in [d+1]} G(x) \cdot \mathbf{c}_j^*$ for every $x \in \mathsf{B}$. Therefore, we cannot define an agreement set T that is included in T_x for every bad seed x in B.

Our solution relies on a "cutoff" parameter $\gamma' := 1 - (1 - \delta_{\mathcal{C}} + \eta)^{\frac{1}{d+1}}$ $(\delta_{\mathcal{C}} > \gamma' > \gamma)$, adapted from [30]. We partition B into two disjoint sets B_1 and B_2:

(1) B_1 contains all seeds $x \in \mathsf{B}$ for which there exist $\tilde{\mathbf{c}}_1, \ldots, \tilde{\mathbf{c}}_{d+1} \in \mathcal{C}$ such that their agreement set $\tilde{T} := \{i \in [n] : \wedge_{j\in[d+1]} \mathbf{u}_j[i] = \tilde{\mathbf{c}}_j[i]\}$ satisfies $\tilde{T} \subsetneq T_x$ and $|\tilde{T}| \geq n \cdot (1-\gamma')$; and
(2) $\mathsf{B}_2 := \mathsf{B} \setminus \mathsf{B}_1$.

We then separately upper bound the sizes of B_1 and B_2.

Bounding the size of B_1. The definition of B_1 allows us to apply our counting argument similarly as in Section 2.3.1. For every $\tilde{T}$, define $\mathsf{B}_{1,\tilde{T}} := \{x \in \mathsf{B}_1 : \tilde{T} \subsetneq T_x\}$. Note that $\mathsf{B}_1 \subseteq \bigcup_{\tilde{T}} \mathsf{B}_{1,\tilde{T}}$, which implies that

$$|\mathsf{B}_1| \leq |\{\tilde{T} \subseteq [n] : \exists\, \tilde{\mathbf{c}}_1, \ldots, \tilde{\mathbf{c}}_{d+1} \in \mathcal{C},\ \forall\, j \in [d+1],\ \mathbf{u}_j|_{\tilde{T}} = \tilde{\mathbf{c}}_j|_{\tilde{T}}\}| \cdot \max_{\tilde{T}}\{|\mathsf{B}_{1,\tilde{T}}|\} \ .$$

The number of $\tilde{T}$ is upper bounded by $1/\eta$ by a combinatorial argument based on the Corrádi lemma. For every fixed $\tilde{T}$, we bound the size of $\mathsf{B}_{1,\tilde{T}}$ via a similar argument as in Equation (2), namely, to bound the size of $T_x \setminus \tilde{T}$ and the zero-evading error of G. Together these yield

$$|\mathsf{B}_1| \leq \frac{1}{\eta} \cdot n \cdot d \ .$$

In fact, because we have a lower bound $|\tilde{T}| \geq n \cdot (1-\gamma')$, the upper bound on $|\mathsf{B}_1|$ can be improved to $\frac{1}{\eta} \cdot n \cdot \gamma' \cdot d$ as in Section 2.3.1.

Bounding the size of B_2. Since $\mathsf{B}_2 = \mathsf{B} \setminus \mathsf{B}_1$, we do not have large agreement sets $\tilde{T}$. Nevertheless, we can bound the size of B_2 using a similar counting argument as before (Equation (2)).

Assume for contradiction that B_2 is large. Using a probabilistic argument, one can show that there exists a set $\hat{T}$ where

(i) $n \cdot (1-\delta_{\mathcal{C}}) < |\hat{T}| < n \cdot (1-\gamma')$;
(ii) there exist $\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_{d+1} \in \mathcal{C}$ with $\mathbf{u}_j|_{\hat{T}} = \hat{\mathbf{c}}_j|_{\hat{T}}$; and
(iii) $|\{x \in \mathsf{B}_2 : \hat{T} \subsetneq T_x\}| \geq C(d, \eta, \delta_{\mathcal{C}})$ for some quantity $C(d, \eta, \delta_{\mathcal{C}})$.

Note that properties of $\hat{T}$ described in Items i and ii allow us to adapt our usual counting argument (Equation (2)) to obtain an upper bound on the possible number of T_x such that $\hat{T} \subsetneq T_x$. Specifically,

$$|\{x \in \mathsf{B}_2 : \hat{T} \subsetneq T_x\}| < C(d, \eta, \delta_{\mathcal{C}}) \ ,$$

which contradicts Item iii above. Hence, B_2 cannot be large.

2.4 The affine space generator

The affine space generator is $G\colon \mathbb{F}^s \to \mathbb{F}^{s+1}$ where $G(x_1, \ldots, x_s) := (1, x_1, \ldots, x_s)$. We show that, for any linear code $\mathcal{C} \subseteq \mathbb{F}^n$, the affine space generator has the same MCA error as the affine line generator $x \mapsto (1, x)$, up to a small constant multiplicative overhead. This implies our Lemma 4.

Let $\mathbf{u}_0, \ldots, \mathbf{u}_s \in \mathbb{F}^n$ be words, and define $U := \mathsf{span}\{\mathbf{u}_1, \ldots, \mathbf{u}_s\}$ and $V := \mathbf{u}_0 + U$. The intuition for the proof is fairly simple: if the "bad seed" set for the affine space generator is large, then we can find a line of the form $\mathbf{u}' + t \cdot \tilde{\mathbf{u}}$, with $\mathbf{u}' \in V$ and $\tilde{\mathbf{u}} \in U$ that contradicts the MCA error of the affine line generator. More precisely, there are many $t \in \mathbb{F}$ such that there exists $T \subseteq [n]$ with $(\mathbf{u}' + t \cdot \tilde{\mathbf{u}})|_T \in \mathcal{C}|_T$ and $\tilde{\mathbf{u}}|_T \notin \mathcal{C}|_T$.

Here we outline this reduction. The set of bad seeds for the affine space generator is

$$\mathsf{B} := \left\{ \mathbf{x} = (x_1, \ldots, x_s) \in \mathbb{F}^s \ : \begin{array}{l} \exists\, T_{\mathbf{x}} \subseteq [n] : \\ \quad |T_{\mathbf{x}}| \geq n \cdot (1-\gamma) \\ \quad \wedge\, (\mathbf{u}_0 + \sum_{j\in[s]} x_j \cdot \mathbf{u}_j)|_{T_{\mathbf{x}}} \in \mathcal{C}|_{T_{\mathbf{x}}} \\ \quad \wedge\, \exists j \in \{0,\ldots,s\},\ \mathbf{u}_j|_{T_{\mathbf{x}}} \notin \mathcal{C}|_{T_{\mathbf{x}}} \end{array} \right\} .$$

For any $\mathbf{x} \in \mathsf{B}$, fix $T_{\mathbf{x}}$ as in the definition of B. The first step is to find $(\lambda_1^\star, \ldots, \lambda_s^\star) \in \mathbb{F}^s$ such that if $\tilde{\mathbf{u}} := \sum_{j\in[s]} \lambda_j^\star \cdot \mathbf{u}_j \in U$ then $\mathsf{B}' := \{\mathbf{x} \in \mathsf{B} : \tilde{\mathbf{u}}|_{T_{\mathbf{x}}} \notin \mathcal{C}|_{T_{\mathbf{x}}}\} \subseteq \mathsf{B}$ is large, which we do via a linear algebraic observation. Consider the quotient linear space $\mathbb{F}^{|T_{\mathbf{x}}|}/\mathcal{C}|_{T_{\mathbf{x}}}$ and the linear map

$$\mathcal{L} : \mathbb{F}^s \to \mathbb{F}^{|T_{\mathbf{x}}|}/\mathcal{C}|_{T_{\mathbf{x}}}, \quad (\lambda_1, \ldots, \lambda_s) \mapsto \sum_{j=1}^{s} \lambda_j \cdot [\mathbf{u}_j] \ ,$$

where $[\mathbf{u}]$ denotes the class of $\mathbf{u}|_{T_{\mathbf{x}}}$ in $\mathbb{F}^{|T_{\mathbf{x}}|}/\mathcal{C}|_{T_{\mathbf{x}}}$. For every $\mathbf{x} \in \mathsf{B}$, the kernel of $\mathcal{L}$ is a proper subspace of $\mathbb{F}^s$. By an averaging argument over the choice of $(\lambda_1, \ldots, \lambda_s)$, we deduce that there exists $(\lambda_1^\star, \ldots, \lambda_s^\star) \in \mathbb{F}^s$ such that B' has size at least $|\mathsf{B}|\,(1 - 1/|\mathbb{F}|)$.

We are left to construct $\mathbf{u}' \in V$ such that for many $t \in \mathbb{F}$, $\mathbf{u}' + t \cdot \tilde{\mathbf{u}}$ breaks the MCA condition. The key idea is to consider the following line *in seed space* for $\mathbf{v} \in \mathbb{F}^s$:

$$\mathbf{x}_{\mathbf{v}}(t) := \mathbf{v} + t \cdot (\lambda_1^\star, \ldots, \lambda_s^\star) \ .$$

While there is no guarantee that $\mathbf{x}_{\mathbf{v}}(t) \in \mathsf{B}'$, by using another averaging argument, there exists $\mathbf{v}^\star \in \mathbb{F}^s$ such that:

$$\Pr_{t\in\mathbb{F}}[\mathbf{x}_{\mathbf{v}^\star}(t) \in \mathsf{B}'] \geq \frac{|\mathsf{B}'|}{|\mathbb{F}^s|} \ .$$

In other words, the density of $t \in \mathbb{F}$ such that $\mathbf{x}_{\mathbf{v}^\star}(t) \in \mathsf{B}'$ is at least the density of B' in $\mathbb{F}^s$. Set $\mathbf{u}' := \mathbf{u}_0 + \sum_{j\in[s]} \mathbf{v}_j^\star \cdot \mathbf{u}_j$. By the definition of B', for every $t \in \mathbb{F}$ such that $\mathbf{x}_{\mathbf{v}^\star}(t) = \mathbf{x} \in \mathsf{B}'$, we have:

$$(\mathbf{u}_0 + \sum_{j\in[s]} x_j \cdot \mathbf{u}_j)|_{T_{\mathbf{x}}} \in \mathcal{C}|_{T_{\mathbf{x}}} \quad \text{and} \quad \tilde{\mathbf{u}}|_{T_{\mathbf{x}}} \notin \mathcal{C}|_{T_{\mathbf{x}}} \ .$$

But here $\mathbf{u}_0 + \sum_{j\in[s]} x_j \cdot \mathbf{u}_j = \mathbf{u}' + t \cdot \tilde{\mathbf{u}}$. Let ϵ_{MCA} be the error of the affine line generator for the code $\mathcal{C}$. By our discussion,

$$\epsilon_{\mathsf{MCA}}(\gamma) \geq \Pr_{t\in\mathbb{F}} \left[\begin{array}{l} \exists\, T_t \subseteq [n] : \\ \quad |T_t| \geq n \cdot (1-\gamma) \\ \quad \wedge\, (\mathbf{u}' + t \cdot \tilde{\mathbf{u}})|_{T_t} \in \mathcal{C}|_{T_t} \\ \quad \wedge\, \tilde{\mathbf{u}}|_{T_t} \notin \mathcal{C}|_{T_t} \end{array} \right] \geq \Pr_{t\in\mathbb{F}}[\mathbf{x}_{\mathbf{v}^\star}(t) \in \mathsf{B}'] \geq \frac{|\mathsf{B}'|}{|\mathbb{F}^s|} \geq \frac{|\mathsf{B}|\,(1 - 1/|\mathbb{F}|)}{|\mathbb{F}^s|} \ .$$

In turn, the size of B is at most $\epsilon_{\mathsf{MCA}}(\gamma) \cdot |\mathbb{F}|^s \cdot (1 - 1/|\mathbb{F}|)^{-1}$, which concludes the proof.

2.5 Polynomial generators and Reed–Solomon codes

We outline our proof of Theorem 5. Similarly to Theorem 3, first we establish that the univariate powers generator has MCA for Reed–Solomon codes up to the list-decoding bound and thereafter rely on our MCA-preserving transformations to establish MCA for all polynomial generators. We discussed the second (generic) step in Section 2.2 (see Lemmas 6 and 7); below we focus on the first step.

We prove that the univariate powers generator $G(x) := (1, x, \ldots, x^d)$ has MCA for $\mathcal{C} := \mathsf{RS}[\mathbb{F}, D, k]$ with error

$$\epsilon_{\mathsf{MCA}}(\gamma) := O\left(\frac{m^7 \cdot n^2}{\rho^{3/2}}\right) \cdot \frac{d}{|\mathbb{F}|} \quad \text{if } \gamma \leq 1 - \left(1 + \frac{1}{2m}\right) \cdot \sqrt{\rho} \ .$$

We bound the size of the set of bad seeds B defined as follows:

$$\mathsf{B} := \left\{ x \in \mathbb{F} \;\middle|\; \exists\, T_x \subseteq D : \begin{array}{l} |T_x| \geq n \cdot (1 + 1/2m) \cdot \sqrt{\rho} \\ \wedge\ \left(\sum_{i=0}^{d} z^i \cdot \mathbf{u}_i\right)\Big|_{T_x} \in \mathcal{C}|_{T_x} \\ \wedge\, \exists\, \ell \in \{0, \ldots, d\}, \mathbf{u}_\ell|_{T_x} \notin \mathcal{C}|_{T_x} \end{array} \right\} \ .$$

Adapting techniques from [10, 38], B can be partitioned into sets $(\mathsf{B}_i)_{i \in [\frac{m+1/2}{\sqrt{\rho}}]}$ where, for each i, if $|\mathsf{B}_i| > \frac{(m+1/2)^6 \cdot d \cdot n^2}{3\rho}$ then there exist codewords $\mathbf{c}_1, \ldots, \mathbf{c}_{d+1}$ such that

$$\mathsf{B}_i = \left\{ x \in \mathbb{F} \;\middle|\; \exists\, T_x \subseteq D : \begin{array}{l} |T_x| \geq n \cdot (1 + 1/2m) \cdot \sqrt{\rho} \\ \wedge\ \left(\sum_{j=0}^{d} x^j \cdot \mathbf{u}_j\right)\Big|_{T_x} = \left(\sum_{j=0}^{d} x^j \cdot \mathbf{c}_j\right)\Big|_{T_x} \\ \wedge\, \exists\, \ell \in \{0, \ldots, d\}, \mathbf{u}_\ell|_{T_x} \notin \mathcal{C}|_{T_x} \end{array} \right\} \ .$$

Letting $\tilde{T} := \{i \in [n] : \forall j \in [d+1],\ \mathbf{u}_j[i] = \mathbf{c}_j[i]\}$, we deduce that $\tilde{T} \subsetneq T_x$ for every $x \in \mathsf{B}_i$. Hence, using the idea explained in Section 2.3 (specifically the counting argument captured in Equation (2)) we show that the number of $x \in \mathbb{F}$ with $\tilde{T} \subsetneq T_x$ is at most $n \cdot d$, which implies that $|\mathsf{B}_i| \leq n \cdot d$. Since $|\mathsf{B}_i| > \frac{(m+1/2)^6 \cdot d \cdot n^2}{3\rho} > n \cdot d$, we reach a contradiction. Therefore,

$$|\mathsf{B}| = \sum_{i \in [\frac{m+1/2}{\sqrt{\rho}}]} |\mathsf{B}_i| \leq \frac{m + 1/2}{\sqrt{\rho}} \cdot \frac{(m+1/2)^6 \cdot d \cdot n^2}{3\rho} = \frac{(m+1/2)^7 \cdot d \cdot n^2}{3\rho^{3/2}} \ .$$

2.6 An MCA toolbox, and going beyond polynomial generators

Our results rely, in particular, on several transformations that we prove preserve MCA. In fact, via such transformations, we prove that a zero-evading generator in [1], which is *not* a polynomial generator, has MCA for arbitrary linear codes; no prior work shows any distance preservation property for generators that are not polynomial generators. This shows that the "MCA toolbox" that we develop enables going beyond polynomial generators. Below we briefly discuss the generator in [1] and our MCA toolbox.

MCA for a non-polynomial generator. Fix a finite field $\mathbb{F}_q$ with q elements and let $\mathbb{F}_{q^k}$ be its extension of degree k. Fix an isomorphism ϕ of $\mathbb{F}$-vector spaces that maps $\mathbb{F}_{q^k}$ to $\mathbb{F}_q^k$. The generator $G \colon \mathbb{F}_{q^k} \times \mathbb{F}_q^k \to \mathbb{F}_q^\ell$ in [1] is defined as

$$G(x, \mathbf{v}) := (\phi(x^j) \cdot \mathbf{v}^\intercal)_{j \in [\ell]} \ .$$

The zero-evading error of this generator is $\epsilon_{\mathsf{ZE}} = \frac{\ell - 1}{q^k} + \frac{1}{q}$.

While G is not a polynomial generator, it can be viewed as the *matrix-vector composition* of two polynomial generators in the following sense. Consider $G_{\mathsf{in}} \colon \mathbb{F}_q^k \to \mathbb{F}_q^k$ and $G_{\mathsf{out}} \colon \mathbb{F}_{q^k} \to (\mathbb{F}_{q^k})^\ell$ defined as

$$\forall\, \mathbf{v} \in \mathbb{F}_q^k, G_{\mathsf{in}}(\mathbf{v}) := \mathbf{v} \ ;$$
$$\forall\, x \in \mathbb{F}_{q^k}, G_{\mathsf{out}}(x) := (x, x^2, \ldots, x^\ell) \ .$$

Then

$$\forall\, j \in [\ell], G(x, \mathbf{v})[j] = \langle \phi(G_{\mathsf{out}}(x)[j]), G_{\mathsf{in}}(\mathbf{v}) \rangle \ .$$

In other words, $G(x, \mathbf{v}) = \mathbf{A}_x \cdot G_{\mathsf{in}}(\mathbf{v})^\intercal$ where $\mathbf{A}_x \in \mathbb{F}^{\ell \times k}$ is the matrix whose j-th row equals $\phi(G_{\mathsf{out}}(x)[j])$.

We show that the matrix-vector composition of two generators as above results in a generator whose MCA error is the sum of their two MCA errors. Moreover, we show that removing some coordinates in the output of a generator (to construct a sub-generator) does not change its MCA error. Since G_{in} is a sub-generator of the affine space generator and G_{out} is a sub-generator of the univariate powers generator, we can bound the MCA error of G using Theorem 3.

Toolbox for MCA. We summarize the transformations that we prove preserve MCA. Formal statements and proofs for these are provided in the full version.

- A sub-generator of a generator has the same MCA error for any code as the generator.
- The linear transformation of a generator has the same MCA error for any code as the generator.
- The tensor product of two generators has MCA error that is the sum of the MCA errors of the original generators.
- If the affine line generator $x \mapsto (1, x)$ has MCA error for a code, then the affine space generator $(x_1, \ldots, x_s) \mapsto (1, x_1, \ldots, x_s)$ has the same MCA error for the code.
- The matrix-vector composition of two generators has MCA error that is the sum of the MCA errors of the original generators.

These transformations enable us to capture all polynomial generators, and even beyond that as [1]'s generator illustrates. The MCA properties of these transformations are likely to be useful beyond this paper.

References

1 Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple construction of almost k-wise independent random variables. *Random Structures and Algorithms*, 3(3):289–304, 1992. doi:10.1002/RSA.3240030308.

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

3 Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In *Proceedings of the 24th ACM Conference on Computer and Communications Security*, CCS '17, pages 2087–2104, 2017. doi:10.1145/3133956.3134104.

4 Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. STIR: Reed–solomon proximity testing with fewer queries. In *Proceedings of the 44th Annual International Cryptology Conference*, CRYPTO '24, pages 380–413, 2024. doi:10.1007/978-3-031-68403-6_12.

5 Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. WHIR: Reed–solomon proximity testing with super-fast verification. In *Proceedings of the 44th Annual International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT '25, pages 214–243, 2025. doi:10.1007/978-3-031-91134-7_8.

6 Daniel Augot, Sarah Bordage, and Jade Nardi. Efficient multivariate low-degree tests via interactive oracle proofs of proximity for polynomial codes. *Des. Codes Cryptography*, 91(3):1111–1151, November 2022. doi:10.1007/S10623-022-01134-Z.

7 Anubhav Baweja, Pratyush Mishra, Tushar Mopuri, and Matan Shtepel. FICS and FACS: Fast iopps and accumulation via code-switching. Cryptology ePrint Archive, Report 2025/737, 2025.

8 Anubhav Baweja, Pratyush Mishra, Tushar Mopuri, and Matan Shtepel. Query-optimal IOPPs for linear-time encodable codes. IACR Cryptology ePrint Archive, Report 2025/1588, 2025.

9 Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast Reed–Solomon interactive oracle proofs of proximity. In *Proceedings of the 45th International Colloquium on Automata, Languages and Programming*, ICALP '18, pages 14:1–14:17, 2018. doi:10.4230/LIPIcs.ICALP.2018.14.

10 Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity gaps for Reed–Solomon codes. In *Proceedings of the 61st Annual IEEE Symposium on Foundations of Computer Science*, FOCS '20, pages 900–909, 2020. doi:10.1109/FOCS46700.2020.00088.

11 Eli Ben-Sasson, Lior Goldberg, Swastik Kopparty, and Shubhangi Saraf. DEEP-FRI: Sampling outside the box improves soundness. In *Proceedings of the 11th Innovations in Theoretical Computer Science Conference*, ITCS '20, pages 5:1–5:32, 2020. doi:10.4230/LIPIcs.ITCS.2020.5.

12 Eli Ben-Sasson, Swastik Kopparty, and Shubhangi Saraf. Worst-case to average case reductions for the distance to a code. In *Proceedings of the 33rd ACM Conference on Computer and Communications Security*, CCS '18, pages 24:1–24:23, 2018. doi:10.4230/LIPIcs.CCC.2018.24.

13 Rishabh Bhadauria, Zhiyong Fang, Carmit Hazay, Muthuramakrishnan Venkitasubramaniam, Tiancheng Xie, and Yupeng Zhang. Ligero++: A new optimized sublinear iop. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, pages 2025–2038, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3372297.3417893.

14 Jonathan Bootle, Alessandro Chiesa, Ziyi Guan, and Siqi Liu. Linear-time probabilistic proofs with sublinear verification for algebraic automata over every field. Cryptology ePrint Archive, Paper 2022/1056, 2022.

15 Jonathan Bootle, Alessandro Chiesa, and Siqi Liu. Zero-knowledge succinct arguments with a linear-time prover. In *Proceedings of the 42nd Annual International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT '22, 2022.

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

17 Martijn Brehm, Binyi Chen, Ben Fisch, Nicolas Resch, Ron D. Rothblum, and Hadas Zeilberger. Blaze: Fast SNARKs from interleaved RAA codes. In *Proceedings of the 44th Annual International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT '25, 2025.

18 Benedikt Bünz, Alessandro Chiesa, Giacomo Fenzi, and William Wang. Linear-Time Accumulation Schemes. In *Proceedings of the 16th Theory of Cryptography Conference*, TCC '25, 2025.

19 Benedikt Bünz, Pratyush Mishra, Wilson Nguyen, and William Wang. Accumulation without homomorphism. In *Proceedings of the 16th Innovations in Theoretical Computer Science Conference*, ITCS '25, pages 23:1–23:25, 2025. doi:10.4230/LIPIcs.ITCS.2025.23.

20 Benedikt Bünz, Pratyush Mishra, Wilson Nguyen, and William Wang. Arc: Accumulation for Reed–Solomon codes. In *Proceedings of the 45th Annual International Cryptology Conference*, CRYPTO '25, 2025.

21 Yeyuan Chen and Zihan Zhang. Explicit folded Reed–Solomon and multiplicity codes achieve relaxed generalized singleton bounds. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, pages 1–12, 2025. doi:10.1145/3717823.3718114.

22 Mahdi Cheraghchi, Venkatesan Guruswami, and Ameya Velingker. Restricted isometry of fourier matrices and list decodability of random linear codes. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 1764–1778, 2014.

23 Elizabeth Crites and Alistair Stewart. On reed–solomon proximity gaps conjectures. Cryptology ePrint Archive, Paper 2025/2046, 2025.

24 Benjamin Diamond and Jim Posen. Polylogarithmic proofs for multilinears over binary towers. Cryptology ePrint Archive, Report 2024/504, 2024.

25 Benjamin E. Diamond and Angus Gruen. On the distribution of the distances of random words. Cryptology ePrint Archive, Paper 2025/2010, 2025.

26 Benjamin E. Diamond and Jim Posen. Proximity Testing with Logarithmic Randomness. *IACR Communications in Cryptology*, 1(1), 2024. doi:10.62056/AKSDKP10.

27 Ethereum Foundation. The Proximity Prize, 2025. URL: https://proximityprize.org/.

28 Asaf Ferber, Matthew Kwan, and Lisa Sauermann. List-decodability with large radius for Reed–Solomon codes. In *Proceedings of the 63rd Annual IEEE Symposium on Foundations of Computer Science*, FOCS '22, pages 720–726, 2022.

29 Yiwen Gao, Dongliang Cai, Yang Xu, and Haibin Kan. From list-decodability to proximity gaps. Cryptology ePrint Archive, Paper 2025/870, 2025.

30 Yiwen Gao, Haibin Kan, and Yuan Li. Linear proximity gap for linear codes within the 1.5 johnson bound. IACR Cryptology ePrint Archive, Report 2024/1810, 2024.

31 Albert Garreta, Nicolas Mohnblatt, and Benedikt Wagner. A simplified round-by-round soundness proof of FRI. Cryptology ePrint Archive, Paper 2025/1993, 2025.

32 Eitan Goldberg, Chong Shangguan, and Itzhak Tamo. List-decoding and list-recovery of Reed–Solomon codes beyond the Johnson radius for every rate. *IEEE Transactions on Information Theory*, 69(4):2261–2268, 2023. doi:10.1109/TIT.2022.3222877.

33 Rohan Goyal and Venkatesan Guruswami. Optimal proximity gaps for subspace-design codes and (random) Reed–Solomon codes. Cryptology ePrint Archive, Paper 2025/2054, 2025.

34 Zeyu Guo, Ray Li, Chong Shangguan, Itzhak Tamo, and Mary Wootters. Improved list-decodability and list-recoverability of Reed–Solomon codes via tree packings. In *Proceedings of the 62nd Annual IEEE Symposium on Foundations of Computer Science*, FOCS '21, pages 708–719, 2021.

35 Zeyu Guo and Zihan Zhang. Randomly punctured Reed–Solomon codes achieve the list decoding capacity over polynomial-size alphabets. In *Proceedings of the 64th IEEE Symposium on Foundations of Computer Science*, FOCS '23, pages 164–176, 2023. doi:10.1109/FOCS57990.2023.00019.

36 Venkatesan Guruswami, Johan Håstad, and Swastik Kopparty. On the list-decodability of random linear codes. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 176–186, 2011.

37 Venkatesan Guruswami, Johan Håstad, Madhu Sudan, and David Zuckerman. Combinatorial bounds for list decoding. *IEEE Transactions on Information Theory*, 48(5):1021–1034, 2002. doi:10.1109/18.995539.

38 Ulrich Haböck. A note on mutual correlated agreement for Reed–Solomon codes. Cryptology ePrint Archive, Report 2025/2110, 2025.

39 Matan Levi, Jonathan Mosheiff, and Nikhil Shagrithaya. Random Reed–Solomon codes and random linear codes are locally equivalent. In *Proceedings of the 66th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '25, 2025.

40 Ray Li and Mary Wootters. Improved list-decodability of random linear binary codes. In *Proceedings of the 22nd International Conference on Randomization and Computation*, RANDOM '18, pages 50:1–50:19, 2018. doi:10.4230/LIPIcs.APPROX-RANDOM.2018.50.

41 Dor Minzer and Kai Zhe Zheng. Improved Round-by-round Soundness IOPs via Reed-Muller Codes. In *Proceedings of the 64th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '25, 2025.

42 Andrija Novakovic and Guillermo Angeris. Ligerito: A small and concretely fast polynomial commitment scheme. Cryptology ePrint Archive, Paper 2025/1187, 2025.

43 Guy N. Rothblum, Salil Vadhan, and Avi Wigderson. Interactive proofs of proximity: Delegating computation in sublinear time. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, STOC '13, pages 793–802, 2013. doi:10.1145/2488608.2488709.

44 Atri Rudra and Mary Wootters. Every list-decodable code for high noise has abundant near-optimal rate puncturings. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 764–773, 2014. doi:10.1145/2591796.2591797.

45 Chong Shangguan and Itzhak Tamo. Combinatorial list-decoding of Reed–Solomon codes beyond the Johnson radius. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC '20, pages 538–551, 2020. doi:10.1145/3357713.3384295.

46 Ruihan Wang, Carmit Hazay, and Muthuramakrishnan Venkitasubramaniam. Ligetron: Lightweight scalable end-to-end zero-knowledge proofs post-quantum zk-snarks on a browser. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 1760–1776, 2024. doi:10.1109/SP54263.2024.00086.

47 Hadas Zeilberger. Khatam: Reducing the communication complexity of code-based SNARKs. IACR Cryptology ePrint Archive, Paper 2024/1843, 2025.

48 Hadas Zeilberger, Binyi Chen, and Ben Fisch. Basefold: Efficient field-agnostic polynomial commitment schemes from foldable codes. In *Proceedings of the 44th Annual International Cryptology Conference*, volume 14929 of *CRYPTO '24*, pages 138–169, 2024. doi:10.1007/978-3-031-68403-6_5.