BEEP: IOPPs from List Decoding with Better Elimination of Pretenders

Remco Bloemen
World Foundation
remco@world.org

Albert Garreta
World Foundation[1]
albert@world.org

November 2026

Abstract

We present an improved soundness analysis of the DEEP technique for linear codes [BGK+20] (after a minor modification), which, unlike its predecessor, results in negligible error bounds for parameters typically used in practice. The modification consists simply in performing m DEEP samples, instead of one. The resulting DEEP sampling error decreases exponentially with m, and in practice $m = 2$–5 makes it tiny. The subsequent folding error is of $(\delta \cdot n + 1)/|\mathbb{F}|$, for δ a proximity parameter and n the code length.

We modify typical code-based IOPPs with this technique, such as WHIR, Ligerito, Basefold, etc. [ACF+25, NA25, ZCF23]. As a result, we obtain code-based IOPPs that only require list decodability of the underlying codes, instead of Mutual Correlated Agreement (MCA), with smaller soundness errors than their counterparts.

Since it does not rely on MCA for its soundness, this improved DEEP technique allows to use *arbitrary linear codes* in IOPPs with δ up to the Johnson bound. Additionally, this opens the door to using codes (e.g. perhaps Reed-Solomon codes) with δ beyond the Johnson bound, assuming the code is δ-list decodable, even if it does not have the δ-MCA property.

Finally, we also provide an *affine* version of our improved DEEP analysis, where the folding challenges are affine polynomials instead of field elements, following [CWY+26, FS26]. We provide a simple proof showing that, after the DEEP samples are done, the folding error with affine folding challenges is $2/|\mathbb{F}|$, which is smaller than the affine-MCA error from [CWY+26].

Contents

1 Introduction 2
1.1 Contributions 5
1.1.1 Better Elimination of Pretenders (BEEP), an improvement to DEEP. 5
1.1.2 BEEP-IOR: folding of constrained interleaved codes with BEEP 7
1.1.3 Affine folding version of BEEP 8

[1]The second author was affiliated with Nethermind Research during part of the preparation of this work.

1.1.4 Conclusions . . . 8
1.2 Use of AI and Lean formalization . . . 9
1.3 Acknowledgements . . . 9
2 Preliminaries . . . 9
2.1 General notation and terminology . . . 9
2.2 Linear codes . . . 10
2.3 Zero-evaders . . . 10
2.4 Interactive Oracle Reductions (IOR) . . . 11
3 BEEP: Better Elimination of Pretenders . . . 13
3.1 Pretender codewords and consistency . . . 13
3.2 Most out of domain samples admit few DEEP-MCA-bad challenges . . . 14
4 Folding of constrained interleaved codes with BEEP . . . 19
4.1 The BEEP IOR . . . 19
4.2 Example PCS: BEEP-WHIR and BEEP-Ligerito . . . 21
5 BEEP with affine folding for Reed-Solomon codes . . . 24
5.1 Affine pretender codewords and consistency . . . 24
5.2 The affine version of Theorem 3.4 . . . 25
6 An alternative and more general proof of Theorem 3.4 via linear algebra . . . 30
6.1 Zero-evaders . . . 30
6.2 Few bad seeds via linear algebra . . . 30
Bibliography . . . 34
A Proof of completeness and round-by-round knowledge soundness of the BEEP-IOR . . 37

1 Introduction

Many efficient proof systems have, at their core, a protocol for testing proximity to linear codes [ACF+25, CFW26], called IOPP (or PCS). The soundness of modern such IOPPs relies on the underlying codes satisfying the so-called δ-*Mutual Correlated Agreement (MCA) property* [ACF+25], usually for a proximity parameter δ as large as possible. It is relatively intricate to prove that a code has MCA for parameters δ and error bounds suitable for practical purposes.

Prior to [ACF+25, BCI+20], before good enough MCA results were known to hold, [BGK+20] proposed a technique called DEEP (Domain Extending for Eliminating Pretenders), by which FRI [BBH+18] (the seminal IOPP) achieved soundness without needing MCA (or the weaker plain Correlated Agreement (CA) [BCI+20]) to hold. Instead, [BGK+20] only asked for the underlying code to have good *list decodability* bounds. Arguably, list decodability is a property of codes that is easier to establish than MCA, and for which, in some sense, more and more attractive bounds are known to hold (see below). However, as we will see, the IOPP from [BGK+20] proves an error bound that is too large for practical applications.

List decodability and MCA. A code $\mathcal{C}$ with blocklength n and dimension ℓ is (δ, L)-*list decodable* if for any word $u \in \mathbb{F}^n$ there are at most L codewords from $\mathcal{C}$ at relative Hamming distance $\leq \delta$ from u. The code $\mathcal{C}$ satisfies δ-*MCA* if, for any two words $u_1, u_2 \in \mathbb{F}^n$, for almost all $\alpha \in \mathbb{F}$, if $u_1 + \alpha \cdot u_2$ agrees with a codeword c from $\mathcal{C}$ on a *large* set S (concretely $|S| \geq (1-\delta) \cdot n$), then both u_1 and u_2 agree as well with codewords from $\mathcal{C}$ on S.

It is known that all codes are (δ, L)-list decodable for δ approaching the *Johnson bound*. Precisely, for $\delta = 1 - \sqrt{1 - \Delta(\mathcal{C})} - \eta$, where $\Delta(\mathcal{C})$ is the relative minimum distance of $\mathcal{C}$ and $\eta > 0$ is small, any linear code $\mathcal{C}$ is (δ, L)-list decodable with $L = 1/(2 \cdot \eta \cdot \sqrt{1 - \Delta(\mathcal{C})})$ [Joh62] (see e.g. [BGK+20, Theorem 20]). On the other hand, linear codes in general are only known to have MCA up to the so-called 1.5 Johnson bound $1 - (1 - \Delta(\mathcal{C}))^{1/3}$ [BCG+25, GKL24, Zei26], and this is known to be a tight bound [KTZ26] (see also [Man26] and [BGK+20, Lemma 22]). In contrast, Reed-Solomon (RS) codes achieve MCA up to the Johnson bound [BCH+25, BCI+20, Hab25]. For MCA, all errors we are aware of depend linearly on $n \cdot \mathsf{poly}(\eta^{-1})$, where η is the additive distance to the exact Johnson bound. This is acceptable for practical purposes, but can sometimes become the "bottleneck error" (the largest error) in a proof system, and in some occassions it forces to either use larger fields, more grinding, or a smaller proximity parameter δ than one would hope for. Indeed, some deployed proof systems set δ up to the Unique Decoding Radius (smaller than Johnson's), for which smaller error bounds are known. As we discuss later, recently [CWY+26, FS26] proposed using low degree polynomials as folding challenges, so as to reduce the MCA error.

It was shown recently that in fields of large characteristic and for adequate parameters, RS codes attain MCA above the Johnson bound [DKT26, Jer26]. In fact, the problem of whether RS codes achieve MCA (and list decodability) above Johnson bound, potentially all the way up to capacity, has recently received a great amount of attention and has undergone rapid development due to its practical and theoretical relevance, see the Proximity Prize [Eth25] (and better.codes), the survey [ABF26], and for example [BCH+25, BCP+26, CS25, DG25, DKT+26, DKT26, Gab26, GG25, Jer26, Jo26].

The DEEP technique. We explain the DEEP technique briefly. We focus on Reed-Solomon codes because they allow concretizing some otherwise slightly abstract concepts (namely, that of *zero evader*). We note that the DEEP technique also applies to arbitrary linear codes, and so does our work. For the sake of intuition, we describe the DEEP approach as an interactive protocol between a prover and a verifier, even though the actual relevant results establish a property of the code $\mathcal{C}$, independently of any protocol built around it.

Similarly as above, the DEEP technique also analyses the number of elements $\alpha \in \mathbb{F}$ for which, given $u_1, u_2 \in \mathbb{F}^n$, the *folded word* $u_1 + \alpha \cdot u_2 \in \mathbb{F}^n$ agrees with a codeword $c \in \mathcal{C}$ on a large set S, while u_1, u_2 do not. The element $\alpha \in \mathbb{F}$ is called a *folding challenge*, and in practice it is sent by a verifier. We say α is *MCA-bad* if the previous scenario holds (i.e. $u_1 + \alpha \cdot u_2$ agrees on a large set S but u_1, u_2 not), and in this case we call c an α-*pretender*.

In the DEEP approach [BGK+20], before α is sampled, the verifier sends a so-called *DEEP sample point* $z \in \mathbb{F}$. The prover then replies with $b_1, b_2 \in \mathbb{F}$, which are supposedly the evaluations at z of codewords c_1, c_2 (which we think of as polynomials) that are δ-close to u_1, u_2, respectively. The verifier then responds with the folding challenge α, and the prover and verifier add the constraint (to be proved in further steps of the protocol) that a codeword c close to the folded word $u_1 + \alpha \cdot u_2$ satisfies $c(z) = b_1 + \alpha \cdot b_2$, where $c(z)$ is the evaluation of c at z. The hope is that the number of α-pretenders c such that $c(z) = b_1 + \alpha \cdot b_2$ is much lower and easier to bound than the number of raw α-pretenders.

Recapping, one can show (see our Section 4 or [BGK+20]) that in a typical IOPP including this DEEP sampling step, (round-by-round) knowledge soundness follows from the following property: for most $z \in \mathbb{F}$, for any prover answer $b_1, b_2 \in \mathbb{F}$, there are negligibly many $\alpha \in \mathbb{F}$ such that: 1) α is MCA-bad; and 2) there is an α-pretender $c \in \mathcal{C}$ that, evaluated at z, equals $b_1 + \alpha \cdot b_2$. In [BGK+20] and in the present work, such a bound on these α's is obtained by using only the list decodability of $\mathcal{C}$, and not its MCA or CA (which may in any case not hold).

We are not aware of any other results of this kind, even though plenty of prior work uses evaluations at random points (called *Out Of Domain (OOD)* samples) to pin words to a single close codeword. Even so, these works rely on CA or MCA to obtain soundness of their folding step, and use this pinning phenomena to obtain soundness not of the folding step but of other protocol steps, e.g. code-switching in [ACF+25], see [ACF+24, ACF+25, BCF+25, CFW26, GLH+25] etc.

To the best of our understanding, the reason the above DEEP approach is not used in practice is because the soundness error bound $\varepsilon_{\mathsf{DEEP}}$ obtained in [BGK+20] is too large for practical parameters. Concretely, [BGK+20, Theorem 3] shows the following. Let $0 < \varepsilon < \frac{1}{3}$ be a tunable parameter. Assume that u_1, u_2 do not agree with two codewords of $\mathcal{C}$ on a common set of size $(1 - \delta - \varepsilon) \cdot n$. Then the probability, over both z and α, that some codeword c at relative distance less than δ from $u_1 + \alpha \cdot u_2$ satisfies $c(z) = b_1 + \alpha \cdot b_2$, is less than

$$\varepsilon_{\mathsf{DEEP}} = \max\left(2 \cdot L \cdot \left(\frac{\ell}{|\mathbb{F}|} + \varepsilon\right)^{\frac{1}{3}}, \frac{4}{\varepsilon^2 \cdot |\mathbb{F}|}\right) \tag{1}$$

where L is the list size of the underlying code $\mathcal{C}$ at radius δ. One can see that, for typical ℓ and L, and targeting $\varepsilon_{\mathsf{DEEP}} \leq 2^{-128}$, one needs $|\mathbb{F}|$ to be above 2^{900}, which for practical purposes is far from the standard 128-256 bit fields. Indeed, from $4/(\varepsilon^2 \cdot |\mathbb{F}|) \leq 2^{-128}$ we obtain $|\mathbb{F}| \geq 2^{130}/\varepsilon^2$, and then, taking cubes, we additionally need $\varepsilon + \ell/|\mathbb{F}| \leq \frac{2^{-384}}{8 \cdot L^3}$, which yield in particular that $\varepsilon \leq \frac{2^{-387}}{L^3}$ and so $|\mathbb{F}| \geq 2^{130}/\varepsilon^2 \geq 2^{904} \cdot L^6$.

In Remark 3.11 we discuss why increasing the number of samples and making a naive adaptation of the original DEEP proof [BGK+20, Theorem 3] still yields essentially the same impractical error (still requiring $|\mathbb{F}| \geq 2^{904} \cdot L^6$).

MCA. On the other hand, as we mentioned, today we have practical MCA results for Reed-Solomon codes up to the Johnson bound, and for any linear code up to the 1.5 Johnson bound.

Recently, the error bound of [BCH+25, Theorem 4.6] for RS codes was improved in [DKT26, Theorem 4.4]. In [DKT26] it is shown that, for an RS code of rate ρ, any $\eta > 0$ and $\delta \leq 1 - \sqrt{\rho_-} - \eta$, where $\rho_- := \rho - 1/n$, the fraction of MCA-bad elements $\alpha \in \mathbb{F}$ is less than

$$\varepsilon'_{\mathsf{RS\text{-}MCA}} := \frac{1}{|\mathbb{F}|} \cdot \frac{8 \cdot (r_- + 1/2)^3}{3 \cdot \rho_-} \cdot n, \tag{2}$$

where $r_- := \max(\lceil \sqrt{\rho_-}/(2 \cdot \eta) \rceil, 3)$. For practical rates (ranging from, say $1/2$ to $1/64$) and a small η (say η between 0.01 and 0.001) this is roughly $\ell/(3 \cdot \sqrt{\rho} \cdot \eta^3 \cdot |\mathbb{F}|)$.

Equipped with these bounds, practitioners and researchers turned away from DEEP sampling, using [BCH+25, BCI+20, DKT26] to directly bound the number or raw MCA-bad elements.

1.1 Contributions

We describe the main results of this work.

1.1.1 Better Elimination of Pretenders (BEEP), an improvement to DEEP.

We provide a new soundness analysis of the DEEP technique after modifying the technique slightly. Precisely, the modification consists simply in making m DEEP sample evaluations, rather than only 1, for a tunable parameter m.

We state our main result next, but before we generalize the notion of "an evaluation of a codeword" to arbitrary linear codes $\mathcal{C}$, and not to just RS codes. Throughout, $\mathbb{F}$ is an arbitrary field and $\mathcal{C}$ a code of blocklength n and dimension ℓ.

Remark 1.1 (General linear codes and zero evaders)**.** For a non-RS code $\mathcal{C}$, we generalize the notion of "evaluation of a codeword $c \in \mathcal{C}$ at a point z". A *zero evader* is a family of linear maps $\varphi_z : \mathbb{F}^\ell \to \mathbb{F}^m$, indexed by z in some set $\mathcal{Z}$. Its error is $\max_{\mathbf{w} \in \mathbb{F}^\ell, \mathbf{w} \neq 0} \Pr_{z \in \mathcal{Z}}[\varphi_z(\mathbf{w}) = \mathbf{0}]$. We refer to $z \in \mathcal{Z}$ as a *DEEP sample point*. We extend φ_z to codewords $c = \mathsf{Enc}_{\mathcal{C}}(\mathbf{w}) \in \mathcal{C}$ by letting $\varphi_z(c) = \varphi_z(\mathbf{w})$, and this is what we call the evaluation of c at z.

For example, if $\mathcal{C}$ is an RS code of dimension ℓ, we may take $\mathcal{Z} = \mathbb{F}^m$ and $\varphi_{\mathbf{z}}(c)$ the vector of evaluations of the polynomial corresponding to c, at the points from $\mathbf{z}$. Its error is $(\ell/|\mathbb{F}|)^m$. For general linear codes, MLE or univariate evaluation of the message corresponding to $c \in \mathcal{C}$ lead to similar evader errors, see Section 2.3.

Theorem 1.2 (Theorem 3.4, informal)**.** Let $\mathcal{C}$ be a linear code of blocklength n, dimension ℓ, and relative minimum distance $\Delta(\mathcal{C})$. Let $u_1, u_2 \in \mathbb{F}^n$ be two words, let $\delta \in (0, \Delta(\mathcal{C}))$ be a proximity parameter and let L be such that $\mathcal{C}$ is (δ, L)-list decodable. Let $(\varphi_z)_{z \in \mathcal{Z}}$ be a zero evader with error $\varepsilon_{\mathsf{evader}}$. For every DEEP sample point $z \in \mathcal{Z}$, and $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{F}^m$ (supposedly evaluations under φ of codewords close to u_1, u_2, to be provided by the prover), let $A_{z,\mathbf{b}_1,\mathbf{b}_2} \subset \mathbb{F}$ be the set of MCA-bad folding challenges $\alpha \in \mathbb{F}$ such that, for some α-pretender $c \in \mathcal{C}$, we have $\varphi_z(c) = \mathbf{b}_1 + \alpha \cdot \mathbf{b}_2$. Then

$$\Pr_{z \in \mathcal{Z}}\left[\left(\max_{\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{F}^m} \left|A_{z,\mathbf{b}_1,\mathbf{b}_2}\right|\right) > \varepsilon_{\mathsf{BEEP},2} \cdot |\mathbb{F}|\right] \leq \varepsilon_{\mathsf{BEEP},1},$$

where

$$\varepsilon_{\mathsf{BEEP},1} = \frac{(M \cdot L)^3}{(\delta \cdot n + 1)^2} \cdot \varepsilon_{\mathsf{evader}}, \qquad \varepsilon_{\mathsf{BEEP},2} := \frac{\lfloor \delta \cdot n \rfloor + 1}{|\mathbb{F}|},$$

and M is either $|\mathbb{F}|$ or $\varepsilon_{\mathsf{MCA}} \cdot |\mathbb{F}|$, for the δ-MCA error $\varepsilon_{\mathsf{MCA}}$ of $\mathcal{C}$, if any such nontrivial error is known.

When $\mathcal{C}$ is a RS code of dimension ℓ we have, for some $m \geq 1$,

$$\mathcal{Z} = \mathbb{F}^m, \qquad \varepsilon_{\mathsf{evader}} = \left(\frac{\ell}{|\mathbb{F}|}\right)^m.$$

In Table 1 we compare the errors $\varepsilon_{\mathsf{BEEP},1}, \varepsilon_{\mathsf{BEEP},2}$ with known MCA errors. We also include the affine version of our result (Theorem 5.3), which we discuss later.

Result	Code	Range of δ	Folding error (MCA, or $\varepsilon_{\mathsf{BEEP},2}$)	DEEP error ($\varepsilon_{\mathsf{BEEP},1}$)
MCA [DKT26, Thm 4.4]	RS	$\delta \leq 1 - \sqrt{\rho_-} - \eta$	$\approx \frac{\ell}{3\sqrt{\rho}\,\eta^3\,\lvert\mathbb{F}\rvert}$, see (2)	—
MCA, affine fold [CWY+26, Thm 1.1]	RS	$\delta \leq 1 - \sqrt{\rho} - \eta$, if $n\eta^2 \geq 4/\rho$	$\frac{4}{\eta\,\lvert\mathbb{F}\rvert} + \frac{2^{14}n^2}{\rho^3\eta^{10}\,\lvert\mathbb{F}\rvert^2}$	—
Khatam[2] [Zei26, Thm 3]	linear	$\delta < 1 - \sqrt[3]{1 - \Delta(\mathcal{C}) + \eta} - \gamma$	$\frac{3}{\eta\,\gamma\,\lvert\mathbb{F}\rvert}$	—
BEEP This work, Thm 1.2	linear	$\delta \leq 1 - \sqrt{1 - \Delta(\mathcal{C})} - \eta$	$\frac{\delta n + 1}{\lvert\mathbb{F}\rvert}$	$\frac{(ML)^3\,\varepsilon_{\mathsf{evader}}}{(\delta n+1)^2}$ (e.g. $\varepsilon_{\mathsf{evader}} = (\ell/\lvert\mathbb{F}\rvert)^m$)
	linear	any $\delta \in (0, \Delta(\mathcal{C}^+))$ at which $\mathcal{C}^+$ is (δ, L)-list decodable[3]		
BEEP, affine fold This work, Thm 5.3	RS	any $\delta \in (0, \Delta(\mathcal{C}))$ at which $\mathcal{C}$ is (δ, L)-list decodable	$\frac{2}{\lvert\mathbb{F}\rvert}$	$\frac{(\lvert\mathbb{F}\rvert\ L)^3(\ell+1)^m}{3\ \lvert\mathbb{F}\rvert^m}$

Table 1: Comparison of known MCA error bounds with the errors from Theorem 1.2 and its affine version Theorem 5.3 (discussed below). n is the code length and ℓ its dimension. As long as L is relatively small (e.g. polynomial), the error $\varepsilon_{\mathsf{BEEP},1}$ can always be made arbitrarily small by taking a zero evader with $\varepsilon_{\mathsf{evader}}$ small. For example, when the evader consists in evaluating on m points the polynomial whose coefficients are given by the message entries, then $\varepsilon_{\mathsf{evader}} = (\frac{\ell}{|\mathbb{F}|})^m$, cf. Section 2.3. M is either $|\mathbb{F}|$ or $\varepsilon_{\mathsf{MCA}} \cdot |\mathbb{F}|$, for a δ-MCA error $\varepsilon_{\mathsf{MCA}}$ of the code. In practice m is 4–5 when $M = |\mathbb{F}|$, and 2 when $\varepsilon_{\mathsf{MCA}}$ is the error of [DKT26, Thm 4.4]. In the last row we need $|\mathbb{F}| \geq (\delta \cdot n + 2)^2$.

[2] [Zei26] loses $\gamma \cdot n$ positions of agreement. Without the loss ($\gamma \approx 1/n$), the error is about $3 \cdot n/(\eta \cdot |\mathbb{F}|)$.

[3] $\mathcal{C}^+$ is $\mathcal{C}$ with the dimension increased by one, cf. Section 5.

The statement of Theorem 1.2 is tailored for obtaining round-by-round knowledge soundness when using DEEP sampling in an IOPP. The analogue of our statement in the formulation of [BGK+20] would be that, for any $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{F}^m$,

$$\Pr_{\mathbf{z}\in\mathbb{F}^m, \alpha\in\mathbb{F}}\left[\alpha \in A_{\mathbf{z},\mathbf{b}_1,\mathbf{b}_2}\right] \leq \varepsilon_{\mathsf{BEEP},1} + \varepsilon_{\mathsf{BEEP},2}.$$

The way Theorem 1.2 would be used inside an IOPP is as follows: the result states that for at least a $1 - \varepsilon_{\mathsf{BEEP},1}$ fraction of DEEP samples $z \in \mathcal{Z}$ sent by the verifier, the following occurs: for any prover answer $\mathbf{b}_1, \mathbf{b}_2$, there is at most an $\varepsilon_{\mathsf{BEEP},2}$ fraction of MCA-bad folding challenges $\alpha \in \mathbb{F}$ admitting an α-pretender codeword c with $\varphi_{z(c)} = \mathbf{b}_1 + \alpha \cdot \mathbf{b}_2$. The reader may find apparent that $\varepsilon_{\mathsf{BEEP},1}$ would then be the round-by-round knowledge soundness error of the DEEP sampling round of the IOPP, and that $\varepsilon_{\mathsf{BEEP},2}$ would be the error of the folding round (in our instantiation this latter error has additional additive terms since we simultaneously run a sumcheck round with it, see Construction 4.1 and Theorem 4.3).

In Section 6 we provide a generalization and alternative proof of Theorem 1.2 which relies solely on linear algebra.

1.1.2 BEEP-IOR: folding of constrained interleaved codes with BEEP

We describe a BEEP variant of typical code-based IOPPs for constrained interleaved codes, such as WHIR [ACF+25], Ligerito [NA25], Basefold [ZCF23], etc. At its core is an Interactive Oracle Reduction (IOR) (Construction 4.1) from $\mathcal{R}_{\mathcal{C}^{\equiv 2}}$ to $\mathcal{R}_{\mathcal{C}}$, which we call BEEP-IOR. Here $\mathcal{C}^{\equiv 2}$ is the 2-wise interleaving of $\mathcal{C}$, and for a code $\mathcal{C}'$ of dimension ℓ' and blocklength n' we define

$$\mathcal{R}_{\mathcal{C}'} := \left\{ \begin{pmatrix} \mathbb{x} = (\mu, \mathbf{v}) \in \mathbb{F} \times \mathbb{F}^{\ell'}, \\ \mathbb{y} = f \in \mathbb{F}^{n'}, \\ \mathbb{w} = \mathbf{f} \in \mathbb{F}^{\ell'} \end{pmatrix} \;\middle|\; \begin{matrix} f = \mathsf{Enc}_{\mathcal{C}'}(\mathbf{f}), \\ \langle \mathbf{f}, \mathbf{v} \rangle = \mu \end{matrix} \right\}.$$

The BEEP-IOR (Construction 4.1) consists of three rounds:

1. DEEP sampling: the verifier sends a DEEP sample challenge $z \in \mathcal{Z}$ and the prover replies with, supposedly, the zero evader values $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{F}^m$ of the two rows $\mathbf{f}_{1*}, \mathbf{f}_{2*} \in \mathbb{F}^\ell$ of $\mathbf{f}$.
2. Folding: the verifier sends a folding challenge $\alpha \in \mathbb{F}$. The oracle and witness are replaced by $g := f_{1*} + \alpha \cdot f_{2*} \in \mathbb{F}^n$ and $\mathbf{g} := \mathbf{f}_{1*} + \alpha \cdot \mathbf{f}_{2*} \in \mathbb{F}^\ell$, respectively (where f_{i*} are the rows of f), and a sumcheck round is performed to reduce the claim $\langle \mathbf{f}, \mathbf{v} \rangle = \mu$ to a linear claim about $\mathbf{g}$, say $\langle \mathbf{g}, \mathbf{v}' \rangle = \mu'$. The sumcheck challenge used is α itself.
3. The prover and verifier batch the linear constraints $\langle \mathbf{g}, \mathbf{v}' \rangle = \mu', \varphi_z(\mathbf{g}) = \mathbf{b}_1 + \alpha \cdot \mathbf{b}_2$, and output the claim $(\mathbb{x}'', g, \mathbf{g}) \in \mathcal{R}_{\mathcal{C}}$ for a certain new $\mathbb{x}''$.

We show that the BEEP-IOR is round-by-round knowledge sound with errors:

$$\varepsilon_1 = \frac{(M \cdot L)^3}{(\delta \cdot n + 1)^2} \cdot \varepsilon_{\mathsf{evader}}, \qquad \varepsilon_2 = \frac{\delta \cdot n + 1}{|\mathbb{F}|} + \frac{2 \cdot L_{\mathcal{C}^{\equiv 2}}}{|\mathbb{F}|}, \qquad \varepsilon_3 = \frac{L}{|\mathbb{F}|}$$

where $L_{\mathcal{C}^{\equiv 2}}$ is a list size bound at radius δ for $\mathcal{C}^{\equiv 2}$, and as before L is a list size for $\mathcal{C}$.

This IOR can then be composed with itself several times so as to obtain an IOR from $\mathcal{R}_{\mathcal{C}^{\equiv k}}$ to $\mathcal{R}_{\mathcal{C}}$, where the output oracle is a linear combination of k oracles. One can then compose this latter IOR with a code-switching IOR (as in WHIR [ACF+25], Ligerito [NA25], Blaze [BCF+24], etc.) or a standard spot check step as in Basefold, FRI-Binius [DP24, ZCF23], so as to replace the linear combination of oracles with a single oracle to a codeword in a new code. This can be repeated in order to obtain a "BEEP variant" of the cited IOPPs. See Section 4.2 for details.

We note that, following [BCF+25], the codes used in this IOPP need not have an efficient decoding algorithm.

1.1.3 Affine folding version of BEEP

Restricting to RS codes, in [FS26], the authors propose to use, as folding challenges α, polynomials of small degree s, instead of field elements. The soundness error from [FS26] is of about $1/|\mathbb{F}|$, with up to s positions of agreement lost after the fold. It requires however s to grow linearly with the block length, e.g. in [FS26, Remark 2] $s = 74$ when $n = 25724$, over a 128-bit field. In [CWY+26] (concurrent to [FS26]), the authors show that affine challenges, i.e. $s = 1$, suffice. They obtain an MCA-type error bound of $4/(\eta \cdot |\mathbb{F}|) + 2^{14} \cdot n^2/(\rho^3 \cdot \eta^{10} \cdot |\mathbb{F}|^2)$ for RS codes of rate ρ and $\delta = 1 - \sqrt{\rho} - \eta$, assuming $n \cdot \eta^2 \geq 4/\rho$ [CWY+26, Theorem 1.1]. As in [FS26], one position of agreement may be lost after the folding.

In Section 5 we provide a version of our results (concretely of Theorem 1.2) where α is an affine polynomial, for RS codes. In this setting, the errors in Theorem 1.2 are as follows: the DEEP sampling error

$$\varepsilon_{\mathsf{affine\text{-}BEEP},1} = \frac{(|\mathbb{F}| \cdot L')^3}{3} \cdot \left(\frac{\ell + 1}{|\mathbb{F}|}\right)^m$$

where L is a list size for the Reed-Solomon code of dimension $\ell + 1$), while the folding error $\varepsilon_{\mathsf{BEEP},2}$ becomes

$$\varepsilon_{\mathsf{affine\text{-}BEEP},2} = \frac{2}{|\mathbb{F}|}.$$

As in [CWY+26, FS26], one position of agreement may be lost. We require $|\mathbb{F}| \geq (\delta \cdot n + 2)^2$.

In contrast to [CWY+26], our analysis in the affine setting is elementary, amounting to about five pages, with half of them being a simple rephrasing for the affine setting of definitions and arguments already used in Section 3. We note that ours is a different result than [CWY+26, FS26] though. Roughly, [CWY+26] proves that RS codes have MCA with a smaller error when using affine folding (up to the Johnson bound, and with one position of agreement loss). Our result, on the other hand, is the affine-variant of the DEEP technique, and requires only list-decodability of the code.

1.1.4 Conclusions

We list some implications of our results:

Code-agnostic IOPPs up to the Johnson bound and beyond. Our work allows to

run typical code-based IOPPs using arbitrary codes (e.g. [BCF+24, BFM+26, GNR26, LLX+26]), not necessarily RS codes, and still use a proximity parameter δ approaching the Johnson bound, over fields of practical size. It suffices to add the DEEP sampling step discussed above and in Section 4.

One can also envision IOPPs using codes with good list decodability for δ beyond the Johnson bound, even if the code does not have MCA at that distance. RS codes are a possible candidate.

Folding error bounds which improve upon those obtained solely with MCA. When using DEEP sampling in an IOPP (Construction 4.1), the error $\varepsilon_{\mathsf{BEEP},2}$ (or the affine error $\varepsilon_{\mathsf{affine\text{-}BEEP},2}$ if using affine folds [CWY+26]), effectively replaces the MCA error bound from (2). As displayed in Table 1, this error is smaller than known plain MCA errors. When using affine folding, our folding error is $2/|\mathbb{F}|$ which again is smaller than the affine-MCA error from [CWY+26]. Smaller errors can lead to a plethora of benefits: such as being able to use smaller fields, less grinding work, or a parameter δ closer to some bound (which decreases proof size).

1.2 Use of AI and Lean formalization

This document was initially written without the use of AI, except for sections 4.2 and 5, which were initially drafted with AI and were substantially edited by us afterward. In the final stages, we used AI to edit and polish the text. We chose this approach with the intention of creating a text which is as amenable to human readers as reasonably possible.

The novel results in this work were obtained through an interactive process with an AI. We had an AI formalize our main results (Theorem 3.4 and Theorem 5.3) in Lean (see https://github.com/worldfnd/BEEP), and the corresponding Lean statements were reviewed by us subsequently.

1.3 Acknowledgements

We thank Ulrich Haböck for insightful discussions that helped improve this work.

2 Preliminaries

2.1 General notation and terminology

A *relation* $\mathcal{R}$ consists of triples $(\mathbb{x}, \mathbb{y}, \mathbb{w})$, where $\mathbb{x}$ is the explicit instance, $\mathbb{y}$ the implicit instance or oracle, and $\mathbb{w}$ the witness.

Given an integer $n \geq 1$, we write $[n]$ to denote the set $\{1, ..., n\}$. We often denote vectors with boldface letters, e.g. $\mathbf{u}, \mathbf{v}$. By $\mathbf{u}_i$ we denote the i-th position of $\mathbf{u}$. When a vector $\mathbf{u}$ is indexed by $\{0,1\}^k$ we denote its multilinear extension with $\mathsf{MLE}[\mathbf{u}]$.

Denote with $\Pr_{s \leftarrow \mathcal{S}}$ the probability under uniform random sampling of $s \in \mathcal{S}$.

2.2 Linear codes

Let $\mathcal{C} \subset \mathbb{F}^n$ be a linear code over $\mathbb{F}$ of dimension ℓ and blocklength n. We fix an injective linear encoding map $\mathsf{Enc}_{\mathcal{C}} : \mathbb{F}^\ell \to \mathcal{C}$. Given two words $w_1, w_2 \in \mathbb{F}^n$, the relative Hamming distance between w_1 and w_2 (the fraction of positions in which they differ) is denoted $\Delta(w_1, w_2)$. The *relative minimum distance* of $\mathcal{C}$, denoted $\Delta(\mathcal{C})$, is the minimum of $\Delta(c_1, c_2)$ over distinct codewords $c_1, c_2 \in \mathcal{C}$.

Given $\delta \in (0, 1)$ and an integer $L \geq 1$, we say $\mathcal{C}$ is (δ, L)-*list-decodable* if for every word $w \in \mathbb{F}^n$ there are at most L codewords that are δ-*close* to w, i.e. such that $\Delta(w, c) \leq \delta$.

We denote codewords and words with non-boldface letters such as w, c, f, etc. We do not use boldface for these, even though they are vectors. We do use boldface letters for messages. For example, we often let $f \in \mathbb{F}^n$ denote the encoding of a message $\mathbf{f} \in \mathbb{F}^\ell$ with a linear code $\mathcal{C}$.

Given a word $u \in \mathbb{F}^n$ and $x \in [n]$, we let $u(x)$ denote the x-th entry of u. Given $S \subset [n]$, we write u_S for the restriction of u to the positions in S. We say that two words *agree on* S if their restrictions to S are equal, and that u *agrees with* $\mathcal{C}$ *on* S if u agrees on S with some codeword of $\mathcal{C}$.

Interleaved codes Given $k \geq 1$, the k-*interleaving* of $\mathcal{C}$ is the code $\mathcal{C}^{\equiv k} \subset (\mathbb{F}^k)^n$ whose codewords are tuples $(c_1, ..., c_k)$ of codewords of $\mathcal{C}$, seen as words of length n over the alphabet $\mathbb{F}^k$. The minimum distance of $\mathcal{C}^{\equiv k}$ is the same as that of $\mathcal{C}$. Moreover, for $\delta < 1 - \sqrt{1 - \Delta(\mathcal{C})} - \eta$, $\mathcal{C}^{\equiv k}$ admits the Johnson decodability bound

$$L = \left\lfloor 1/\left(2 \cdot \eta \cdot \sqrt{1 - \Delta(\mathcal{C})}\right) \right\rfloor,$$

the same as $\mathcal{C}$. For δ at or above the Johnson bound, if $\mathcal{C}$ is (δ, L)-list decodable, then $\mathcal{C}^{\equiv k}$ is (δ, L^k)-list decodable. The following lemma improves such bound for larger values of k.

Lemma 2.1 ([GGR11], see [CFW26, Lemma 3.18])**.** Let $\delta \in [0, \Delta(\mathcal{C}))$ and assume $\mathcal{C}$ is (δ, L) -list decodable. Let $b := \lceil \frac{\delta}{\Delta(\mathcal{C}) - \delta} \rceil$ and $r := \lceil \log_2 \frac{\Delta(\mathcal{C})}{\Delta(\mathcal{C}) - \delta} \rceil$. Then, for every $k \geq 1$, the code $\mathcal{C}^{\equiv k}$ is $(\delta, L_{\equiv})$-list decodable with $L_{\equiv} := \binom{b+r}{r} \cdot L^r$.

2.3 Zero-evaders

We use the notion of a zero evader from [BCF+25], expressed as an indexed family of linear maps with vector-valued outputs.

Definition 2.2 (Zero evader)**.** A *zero evader* is a family of linear maps $\{\varphi_z : \mathbb{F}^e \to \mathbb{F}^m \mid z \in \mathcal{Z}\}$ indexed by a finite non-empty set $\mathcal{Z}$. We call $\varepsilon \in [0, 1]$ an *error bound* of the evader if

$$\max_{\substack{\mathbf{v} \in \mathbb{F}^e \\ \mathbf{v} \neq \mathbf{0}}} \Pr_{z \leftarrow \mathcal{Z}} [\varphi_z \mathbf{v} = \mathbf{0}] \leq \varepsilon.$$

Remark 2.3. In a given basis φ_z is an $m \times e$ matrix. When $m = 1$ we call it a *scalar zero evader*.

Definition 2.4 (Univariate evader)**.** Define the m-sample *univariate evader* with index family $\mathcal{Z} := \mathbb{F}^m$ and $\varphi_{\mathbf{z}}$ for $\mathbf{z} = (z_1, ..., z_m)$ as the map given by the $m \times e$ Vandermonde matrix on $\mathbf{z}$.

Lemma 2.5. The m-sample univariate zero evader has error bound $\varepsilon = \min(1, (\frac{e-1}{|\mathbb{F}|})^m)$.

Proof. Fix a nonzero $\mathbf{v} \in \mathbb{F}^e$. The polynomial $P(X) = \sum_{j \in [e]} v_j \cdot X^{j-1}$ is nonzero and has degree at most $e-1$, so it has at most $e-1$ roots in $\mathbb{F}$. By construction, $\varphi_{\mathbf{z}}(\mathbf{v}) = (P(z_1), ..., P(z_m))$. Since the coordinates of $\mathbf{z}$ are independent and uniform,

$$\Pr_{\mathbf{z} \leftarrow \mathbb{F}^m}[\varphi_{\mathbf{z}}(\mathbf{v}) = \mathbf{0}] = \prod_{i \in [m]} \Pr_{z_i \leftarrow \mathbb{F}}[P(z_i) = 0] \leq \left(\frac{e-1}{|\mathbb{F}|}\right)^m.$$

□

Remark 2.6. We do not require the z_i to be distinct. When applied to message coordinates, the univariate zero evader matches the m-repeated out of domain samples in WHIR [ACF+25], and the codeword queries in the Reed-Solomon instantiation of Ligerito [NA25], etc.

Definition 2.7 (Multilinear evader)**.** Let $e_0 := \lceil \log_2 e \rceil$ and fix a basis. We define the scalar *multilinear evader* with index family $\mathcal{Z} := \mathbb{F}^{e_0}$ and $\varphi_{\mathbf{z}}$ for $\mathbf{z} = (z_1, ..., z_{e_0})$ as the first e values of the MLE tensor product: $\varphi_{\mathbf{z}} = (\bigodot_{i \in [e_0]} \binom{1-z_i}{z_i})_{[e]}$.

Lemma 2.8. The *multilinear evader* has error bound $\varepsilon = \min(1, e_0/|\mathbb{F}|)$.

Proof. Fix a nonzero $\mathbf{v} \in \mathbb{F}^e$ and pad its coordinates with zeros to obtain a vector of length 2^{e_0}. Its multilinear extension $\mathtt{MLE}[\mathbf{v}]$ has total degree at most e_0, and $\varphi_{\mathbf{z}}\mathbf{v} = \mathtt{MLE}[\mathbf{v}](\mathbf{z})$. The result follows from the Schwartz–Zippel bound. □

2.4 Interactive Oracle Reductions (IOR)

An *Interactive Oracle Reduction (IOR) of proximity* $\Pi = (\mathsf{P}, \mathsf{V})$ from a relation $\mathcal{R}$ to a relation $\mathcal{R}'$ is specified as follows. The prover P receives $(\mathtt{x}, \mathtt{y}, \mathtt{w}) \in \mathcal{R}$. The verifier V receives $\mathtt{x}$ and query access to $\mathtt{y}$. They interact for k rounds. In round $j \in [k]$, the prover sends an oracle m_j and the verifier sends a uniformly random message $\rho_j \in \{0,1\}^{r_j}$. After the k rounds, the verifier makes queries to $\mathtt{y}$ and to $m_1, ..., m_k$. Then:

1. V either rejects or outputs a new pair $(\mathtt{x}', \mathtt{y}')$, where $\mathtt{y}'$ is specified implicitly in terms of the oracles $\mathtt{y}$ and $m_1, ..., m_k$.
2. P outputs a witness $\mathtt{w}'$ such that $(\mathtt{x}', \mathtt{y}', \mathtt{w}') \in \mathcal{R}'$.

We write $\langle \mathsf{P}, \mathsf{V} \rangle (\mathtt{x}, \mathtt{y}, \mathtt{w})$ for the interaction of prover and verifier during the IOR.

Definition 2.9 (Completeness)**.** Π has *(perfect) completeness* if for every $(\mathtt{x}, \mathtt{y}, \mathtt{w}) \in \mathcal{R}$,

$$\Pr_{\rho_1,\ldots,\rho_k}[(\mathbb{x}',\mathbb{y}';\mathbb{w}') \in \mathcal{R}' \mid (\mathbb{x}',\mathbb{y}';\mathbb{w}') \leftarrow \langle \mathsf{P},\mathsf{V}\rangle(\mathbb{x},\mathbb{y},\mathbb{w})] = 1.$$

Remark 2.10. An IOP for a relation $\mathcal{R}$ is an IOR from $\mathcal{R}$ to the trivial relation $\mathcal{R}_{\mathsf{triv}} = \{(\mathbb{x} = \bot, \mathbb{y} = \bot, \mathbb{w} = \bot)\}$.

We proceed to define round-by-round knowledge soundness of an IOR, following [BCF+25] in the formulation with relaxed relations of [CFW26].

Definition 2.11 (Knowledge state function with relaxation)**.** Let $\Pi = (\mathsf{P},\mathsf{V})$ be an IOR from a relation $\mathcal{R}$ to $\mathcal{R}'$. A *knowledge state function with relaxation* $(\tilde{\mathcal{R}},\widetilde{\mathcal{R}'})$ for Π is a (possibly inefficient) function State that, on input $(\mathbb{x},\mathbb{y})$, interaction transcript tr, and knowledge state witness $\mathbf{w}$, outputs a bit and has the following properties.

1. **Empty transcript.** If $\mathsf{tr} = \varnothing$ is the empty transcript, then $\mathsf{State}(\mathbb{x},\mathbb{y},\mathsf{tr},\mathbf{w}) = 1$ if and only if $(\mathbb{x},\mathbb{y};\mathbf{w}) \in \tilde{\mathcal{R}}$.
2. **Prover moves.** If tr is a transcript where the prover is about to move and $\mathsf{State}(\mathbb{x},\mathbb{y},\mathsf{tr},\mathbf{w}) = 0$, then $\mathsf{State}(\mathbb{x},\mathbb{y},\mathsf{tr},m,\mathbf{w}) = 0$ for every prover message m.
3. **Full transcript.** If $\mathsf{tr} = (m_1,\rho_1,\ldots,m_k,\rho_k)$ is a full transcript, then $\mathsf{State}(\mathbb{x},\mathbb{y},\mathsf{tr},\mathbf{w}) = 1$ if and only if V on tr outputs $(\mathbb{x}',\mathbb{y}')$ such that $(\mathbb{x}',\mathbb{y}';\mathbf{w}) \in \widetilde{\mathcal{R}'}$.

Definition 2.12 (Round-by-round knowledge soundness)**.** A k-round IOR Π has *round-by-round knowledge soundness with relaxation* $(\tilde{\mathcal{R}},\widetilde{\mathcal{R}'})$ with errors $(\kappa_1,\ldots,\kappa_k)$ if there exists a knowledge state function State as in Definition 2.11 and a deterministic extractor Ext such that for every $(\mathbb{x},\mathbb{y})$, every $j \in [k]$, and every transcript tr where the verifier is about to send ρ_j,

$$\Pr_{\rho_j}\left[\exists \mathbf{w} \text{ such that } \begin{array}{l} \wedge\ \mathsf{State}\big(\mathbb{x},\mathbb{y},\mathsf{tr},\mathsf{Ext}\big(\mathbb{x},\mathbb{y},\mathsf{tr},\rho_j,\mathbf{w}\big)\big) = 0 \\ \wedge\ \mathsf{State}\big(\mathbb{x},\mathbb{y},\mathsf{tr},\rho_j,\mathbf{w}\big) = 1 \end{array}\right] \leq \kappa_j(\mathbb{x},\mathbb{y}).$$

If $(\tilde{\mathcal{R}},\widetilde{\mathcal{R}'}) = (\mathcal{R},\mathcal{R}')$ then we say the round-by-round knowledge soundness of Π is *without relaxation*.

Lemma 2.13 (Composition of IORs [CFW26, Theorem 4.7])**.** Let Π_1 be a perfectly complete k_1-round IOR from $\mathcal{R}_1$ to $\mathcal{R}_2$ with round-by-round knowledge soundness with relaxation $(\widetilde{\mathcal{R}_1},\widetilde{\mathcal{R}_2})$ and errors $\kappa_{11},\ldots,\kappa_{1k_1}$, and let Π_2 be a perfectly complete k_2-round IOR from $\mathcal{R}_2$ to $\mathcal{R}_3$ with round-by-round knowledge soundness with relaxation $(\widetilde{\mathcal{R}_2},\widetilde{\mathcal{R}_3})$ and errors $\kappa_{21},\ldots,\kappa_{2k_2}$. Then the IOR from $\mathcal{R}_1$ to $\mathcal{R}_3$ that runs Π_1 and then Π_2 on the output of Π_1 is perfectly complete, has $k_1 + k_2$ rounds, and has round-by-round knowledge soundness with relaxation $(\widetilde{\mathcal{R}_1},\widetilde{\mathcal{R}_3})$ and errors $\kappa_{11},\ldots,\kappa_{1k_1},\kappa_{21},\ldots,\kappa_{2k_2}$.

3 BEEP: Better Elimination of Pretenders

In this section we prove Theorem 1.2. In Section 5 we prove an analogous version when folding challenges are affine polynomials.

3.1 Pretender codewords and consistency

We start by establishing some concepts and terminology. Let $\mathbb{F}$ be a finite field and let $\mathcal{C} \subset \mathbb{F}^n$ be a linear code of blocklength n and dimension ℓ, with n a power of two. Let $\delta \in (0,1)$ be a proximity parameter.

We fix two linear maps $\theta_1, \theta_2 : \mathbb{F} \to \mathbb{F}$, so that $\theta_i(\alpha) = a_i + \alpha \cdot d_i$ for some $a_i, d_i \in \mathbb{F}$, and we assume that $a_1 \cdot d_2 \neq a_2 \cdot d_1$. We use $\theta_1(\alpha)$ and $\theta_2(\alpha)$ as the weights when folding two words with a challenge $\alpha \in \mathbb{F}$. For example, $(\theta_1(\alpha), \theta_2(\alpha))$ can be $(1, \alpha)$, or $(1-\alpha, \alpha)$ as in Section 4. The assumption $a_1 \cdot d_2 \neq a_2 \cdot d_1$ guarantees that the vectors $(\theta_1(\alpha), \theta_2(\alpha))$ and $(\theta_1(\alpha'), \theta_2(\alpha'))$ are linearly independent for all distinct $\alpha, \alpha' \in \mathbb{F}$.

Let $u_1, u_2 \in \mathbb{F}^n$ be two words. Given $S \subset [n]$, we say (u_1, u_2) has *Correlated Agreement with $\mathcal{C}$ over S* if there exist codewords $c_1, c_2 \in \mathcal{C}$ such that u_1 agrees with c_1 on S and u_2 agrees with c_2 on S.

Let $u_1, u_2 \in \mathbb{F}^n$ be two words. We say that a field element $\alpha \in \mathbb{F}$ satisfies the *δ-Mutual Correlated Agreement (MCA) property* with respect to u_1, u_2, if for every subset $S \subset [n]$ with $|S| \geq (1-\delta) \cdot n$ such that $\theta_1(\alpha) \cdot u_1 + \theta_2(\alpha) \cdot u_2$ agrees with $\mathcal{C}$ on S, the pair (u_1, u_2) has Correlated Agreement (CA) with $\mathcal{C}$ over S.

Definition 3.1 (MCA-bad element)**.** We say α is *MCA-bad* if it does not satisfy the δ-MCA property.

The code $\mathcal{C}$ is said to have *δ-Mutual Correlated Agreement (MCA)* for the maps θ_1, θ_2, with error ε, if for any two words $u_1, u_2 \in \mathbb{F}^n$, there are at most $\varepsilon \cdot |\mathbb{F}|$ MCA-bad field elements.

Definition 3.2 (Pretender codeword)**.** Fix two words $u_1, u_2 \in \mathbb{F}^n$, and let $\alpha \in \mathbb{F}$ be an MCA-bad field element. We say that a codeword $c \in \mathcal{C}$ is an *α-pretender* if, intuitively speaking, it is a witness to the failing of the MCA property for α. Formally, c is an α-pretender if $\theta_1(\alpha) \cdot u_1 + \theta_2(\alpha) \cdot u_2$ and c agree on a set $S \subset [n]$ of size $|S| \geq (1-\delta) \cdot n$ and (u_1, u_2) does not have CA with $\mathcal{C}$ over S.

Fix a zero-evader $(\varphi_z)_{z \in \mathcal{Z}}$, i.e. a family of linear maps $\varphi_z : \mathbb{F}^\ell \to \mathbb{F}^m$, with error bound ε. Abusing the notation, we also define φ_z on $\mathcal{C}$ by letting $\varphi_z(\mathsf{Enc}_{\mathcal{C}}(\mathbf{w})) := \varphi_z(\mathbf{w})$. Since $\mathsf{Enc}_{\mathcal{C}}$ is linear and injective, φ_z is linear on $\mathcal{C}$ and $\Pr_{z \leftarrow Z}[\varphi_z(c) = 0] \leq \varepsilon$ for all nonzero codeword c.

Let $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{F}^m$. For intuition, for an honest prover, $\mathbf{b}_1, \mathbf{b}_2$ will be the values $\varphi_z(c_1), \varphi_z(c_2)$, where $c_1, c_2 \in \mathcal{C}$ are two codewords that are δ-close to u_1 and u_2.

Definition 3.3 (Consistency of MCA-bad elements $\alpha \in \mathbb{F}$ with $z \in \mathcal{Z}_{\mathsf{evader}}$)**.** Let $\alpha \in \mathbb{F}$ and $c \in \mathcal{C}$. Given $z \in \mathcal{Z}$ and $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{F}^m$, we say that (α, c) is *consistent* with $(z, \mathbf{b}_1, \mathbf{b}_2)$ if

$$\varphi_z(c) = \theta_1(\alpha) \cdot \mathbf{b}_1 + \theta_2(\alpha) \cdot \mathbf{b}_2.$$

When α is an MCA-bad element and there exists an α-pretender $c \in \mathcal{C}$ such that (α, c) is consistent with $(z, \mathbf{b}_1, \mathbf{b}_2)$, then we say that α is a *DEEP-MCA-bad element* with respect to $z, \mathbf{b}_1, \mathbf{b}_2$ and u_1, u_2.

When clear from the context, we omit referring to $z, \mathbf{b}_1, \mathbf{b}_2, u_1, u_2$ when speaking of DEEP-MCA-bad elements.

3.2 Most out of domain samples admit few DEEP-MCA-bad challenges

We are now ready to state and prove the rigorous version of Theorem 1.2.

Theorem 3.4 (For most $z \in \mathcal{Z}$, there are few DEEP-MCA-bad elements $\alpha \in \mathbb{F}$)**.** Let $\delta \in (0, 1)$ and let L be such that $\mathcal{C}$ is (δ, L)-list decodable. Let $u_1, u_2 \in \mathbb{F}^n$ be two words, and let $(\varphi_z)_{z \in \mathcal{Z}}$ be a zero evader with error $\varepsilon_{\mathsf{evader}}$.

For each $z \in \mathcal{Z}_{\mathsf{evader}}$ and $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{F}^m$, let $A_{z,\mathbf{b}_1,\mathbf{b}_2} \subset \mathbb{F}$ be the set of DEEP-MCA-bad field elements $\alpha \in \mathbb{F}$ with respect to $z, \mathbf{b}_1, \mathbf{b}_2, u_1, u_2$. Let $\mathbf{b}_1^*, \mathbf{b}_2^*$ be such that $|A_{z,\mathbf{b}_1^*,\mathbf{b}_2^*}|$ is maximal for fixed z, u_1, u_2, and denote $A_z := A_{z,\mathbf{b}_1^*,\mathbf{b}_2^*}$.

Then for any $t \geq \delta \cdot n + 2$,

$$\Pr_{z \leftarrow \mathcal{Z}_{\mathsf{evader}}} [|A_z| \geq t] \leq \frac{(|\mathbb{F}| \cdot L)^3 \cdot \varepsilon_{\mathsf{evader}}}{t \cdot (t-1) \cdot (t - \delta \cdot n - 1)} \tag{3}$$

Moreover, if $\mathcal{C}$ has δ-MCA error $\varepsilon_{\mathsf{MCA}}$ then the factor $(|\mathbb{F}| \cdot L)^3$ in (3) can be replaced by

$$(\varepsilon_{\mathsf{MCA}} \cdot |\mathbb{F}| \cdot L)^3$$

Remark 3.5 (Using Theorem 3.4 to replace the MCA error with the list decodability size in code-based PCS)**.** Looking ahead, we will use this result as follows: before performing the fold of two words during a code-based PCS, we have the verifier sample an element $z \in \mathcal{Z}$, and the prover replies with $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{F}^m$, supposedly the evaluations on φ_z of the two words (which are supposedly codewords). Now, due to Theorem 3.4, with high probability over z, there are very few folding challenges $\alpha \in \mathbb{F}$ that result in a failure of the MCA property and which are consistent with $(z, \mathbf{b}_1, \mathbf{b}_2)$. Hence, at this point we can fold as usual, and obtain knowledge soundness even if the code $\mathcal{C}$ is not known to possess (or is known to not possess) MCA at the proximity parameter δ used by the PCS. We formalize this intuition in Section 4.

We proceed to prove Theorem 3.4. We assume the notation of its hypothesis.

Definition 3.6 (Parameterized lines)**.** We let $\mathcal{L}^1[\mathbb{F} \to \mathcal{C}]$ be the set of all linear maps $\theta : \mathbb{F} \to \mathcal{C}$, so that $\theta(\alpha) = c_{\theta,1} + \alpha \cdot c_{\theta,2}$ for some codewords $c_{\theta,1}, c_{\theta,2} \in \mathcal{C}$. Given $\theta \in \mathcal{L}^1[\mathbb{F} \to \mathcal{C}]$, we let $\mathsf{Graph}(\theta) = \{(\alpha, \theta(\alpha)) \mid \alpha \in \mathbb{F}\} \subset \mathbb{F} \times \mathcal{C}$ and we call $\mathsf{Graph}(\theta)$ a *parameterized line in* $\mathcal{C}$.

The following result is folklore.

Lemma 3.7 (A parameterized line contains few pretenders)**.** Let $u_1, u_2 \in \mathbb{F}^n$ be two words, and let $\theta \in \mathcal{L}^1[\mathbb{F} \to \mathcal{C}]$. Let $A_\theta^{\mathsf{pret}} \subset \mathbb{F}$ be the set of all field elements $\alpha \in \mathbb{F}$ that are MCA-bad and such that the codeword $\theta(\alpha) = c_{\theta,1} + \alpha \cdot c_{\theta,2} \in \mathcal{C}$ is an α-pretender with respect to u_1, u_2. Then $|A_\theta^{\mathsf{pret}}| \leq \delta \cdot n + 1$.

Proof. Since $a_1 \cdot d_2 \neq a_2 \cdot d_1$, there exist codewords $c_1, c_2 \in \mathcal{C}$ such that $\theta(\alpha) = \theta_1(\alpha) \cdot c_1 + \theta_2(\alpha) \cdot c_2$ for all $\alpha \in \mathbb{F}$ (they are the solution to the system $a_1 \cdot c_1 + a_2 \cdot c_2 = c_{\theta,1}$, $d_1 \cdot c_1 + d_2 \cdot c_2 = c_{\theta,2}$). Let $D \subset [n]$ be the set

$$D \coloneqq \{x \in [n] \mid u_1(x) \neq c_1(x) \text{ or } u_2(x) \neq c_2(x)\}.$$

Let $\alpha \in A_\theta^{\mathsf{pret}}$. Since $\theta(\alpha)$ is an α-pretender, there is a set $S_\alpha \subset [n]$ where $\theta(\alpha)$ and $\theta_1(\alpha) \cdot u_1 + \theta_2(\alpha) \cdot u_2$ agree, with $|S_\alpha| \geq (1-\delta) \cdot n$ and such that (u_1, u_2) do not have correlated agreement with $\mathcal{C}$ over S_α. Hence $S_\alpha \cap D \neq \varnothing$.

We claim that for every distinct $\alpha, \alpha' \in A_\theta^{\mathsf{pret}}$ we have $(S_\alpha \cap D) \cap (S_{\alpha'} \cap D) = \varnothing$. Indeed, assume to the contrary that $x \in [n]$ belongs to such set. Then $\theta_1(\alpha) \cdot u_1(x) + \theta_2(\alpha) \cdot u_2(x) = \theta_1(\alpha) \cdot c_1(x) + \theta_2(\alpha) \cdot c_2(x)$ and $\theta_1(\alpha') \cdot u_1(x) + \theta_2(\alpha') \cdot u_2(x) = \theta_1(\alpha') \cdot c_1(x) + \theta_2(\alpha') \cdot c_2(x)$. Since $\alpha \neq \alpha'$, the vectors $(\theta_1(\alpha), \theta_2(\alpha))$ and $(\theta_1(\alpha'), \theta_2(\alpha'))$ are linearly independent, and so we obtain $u_1(x) = c_1(x)$ and $u_2(x) = c_2(x)$, contradicting the fact that $x \in D$, and thus proving the claim.

This immediately implies that $|A_\theta^{\mathsf{pret}}| \leq |D|$, since each set S_α intersects D non-trivially, and the sets $S_\alpha \cap D$ are pairwise disjoint. Now if $|D| \leq \delta \cdot n + 1$ we are done. Hence assume this is not the case.

Since $|S_\alpha| \geq (1-\delta) \cdot n$ and $S_\alpha = (S_\alpha \cap D) \cup (S_\alpha \setminus D)$, we obtain $|S_\alpha \cap D| \geq (1-\delta) \cdot n - |S_\alpha \setminus D| \geq (1-\delta) \cdot n - |[n] \setminus D| = |D| - \delta \cdot n$. Now since the sets $S_\alpha \cap D$ are pairwise disjoint and contained in D,

$$|A_\theta^{\mathsf{pret}}| \cdot (|D| - \delta \cdot n) \leq \left| \bigcup_{\alpha \in A_\theta^{\mathsf{pret}}} (S_\alpha \cap D) \right| \leq |D|.$$

Since we assumed $|D| > \delta \cdot n + 1$, the factor $(|D| - \delta \cdot n)$ is positive. Dividing by it and using the inequality on $|D|$ again:

$$|A_\theta^{\mathsf{pret}}| \leq \frac{|D|}{|D| - \delta \cdot n} = 1 + \frac{\delta \cdot n}{|D| - \delta \cdot n} < 1 + \delta \cdot n,$$

as required. □

Given two codewords $c_1, c_2 \in \mathcal{C}$, we let $\theta_{c_1,c_2} : \mathbb{F} \to \mathcal{C}$ be the linear map from $\mathcal{L}^1[\mathbb{F} \to \mathcal{C}]$ defined as $\theta_{c_1,c_2}(\alpha) = c_1 + \alpha \cdot c_2$.

The following is an elementary characterization of colinearity:

Lemma 3.8 (Colinearity of codewords)**.** Let $(\alpha_i, v_i)_{i\in[3]} \in (\mathbb{F} \times \mathcal{C})^3$ with $\alpha_1, \alpha_2, \alpha_3$ pairwise distinct. Then the three tuples $(\alpha_i, v_i)_{i\in[3]}$ belong to the same parameterized line (i.e. there exists $\theta \in \mathcal{L}^1[\mathbb{F} \to \mathcal{C}]$ such that $(\alpha_i, v_i) \in \mathsf{Graph}(\theta)$ for all $i \in [3]$) if and only if

$$U\Big((\alpha_i, v_i)_{i\in[3]}\Big) := (\alpha_3 - \alpha_1) \cdot (v_2 - v_1) - (\alpha_2 - \alpha_1) \cdot (v_3 - v_1) = 0,$$

Proof. Assume $(\alpha_i, v_i) \in \mathsf{Graph}(\theta_{c_1,c_2})$ for some $c_1, c_2 \in \mathcal{C}$. Then $v_i = c_1 + \alpha_i \cdot c_2$ for all $i = 1, 2, 3$. Now $v_2 - v_1 = (\alpha_2 - \alpha_1) \cdot c_2$ and $v_3 - v_1 = (\alpha_3 - \alpha_1) \cdot c_2$. Plugging this into the definition of $U((\alpha_i, v_i)_{i\in[3]})$ yields $U((\alpha_i, v_i)_{i\in[3]}) = 0$.

Conversely, assume $U((\alpha_i, v_i)_{i\in[3]}) = 0$, so that $\frac{v_2-v_1}{\alpha_2-\alpha_1} = \frac{v_3-v_1}{\alpha_3-\alpha_1}$. Then taking $c_2 = \frac{v_2-v_1}{\alpha_2-\alpha_1}$ and $c_1 = v_1 - \alpha_1 \cdot c_2$ we have $v_1 = c_1 + \alpha_1 \cdot c_2$, $v_2 = v_1 + (\alpha_2 - \alpha_1) \cdot c_2 = c_1 + \alpha_2 \cdot c_2$ and $v_3 = v_1 + (\alpha_3 - \alpha_1) \cdot c_2 = c_1 + \alpha_3 \cdot c_2$, and so $(\alpha_i, v_i) \in \mathsf{Graph}(\theta_{c_1,c_2})$ for all $i \in [3]$. □

Note that, for any $(\alpha_i, v_i)_{i\in[3]} \in (\mathbb{F} \times \mathcal{C})^3$, the element $U((\alpha_i, v_i)_{i\in[3]})$ is a codeword from $\mathcal{C}$.

Lemma 3.9 ($U((\alpha_i, v_i)_{i\in[3]})$ vanishes under φ_z for consistent $\boldsymbol{\alpha}$)**.** Let $z \in \mathcal{Z}$, let $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{F}^m$, and let $(\alpha_i, v_i)_{i\in[3]} \in (\mathbb{F} \times \mathcal{C})^3$ be such that (α_i, v_i) is consistent with $(z, \mathbf{b}_1, \mathbf{b}_2)$ for each $i = 1, 2, 3$. Then

$$\varphi_z\Big(U\Big((\alpha_i, v_i)_{i\in[3]}\Big)\Big) = \mathbf{0}$$

Proof. It follows by direct evaluation, using the linearity of φ_z and the assumption that $\varphi_z(v_i) = \theta_1(\alpha_i) \cdot \mathbf{b}_1 + \theta_2(\alpha_i) \cdot \mathbf{b}_2$ for $i = 1, 2, 3$. Writing $\theta_j(\alpha) = a_j + \alpha \cdot d_j$ and $\mathbf{b}' := d_1 \cdot \mathbf{b}_1 + d_2 \cdot \mathbf{b}_2$, this gives $\varphi_z(v_i) - \varphi_z(v_1) = (\alpha_i - \alpha_1) \cdot \mathbf{b}'$ for $i = 2, 3$, and so:

$$\begin{aligned}\varphi_z\Big(U\Big((\alpha_i, v_i)_{i\in[3]}\Big)\Big) &= (\alpha_3 - \alpha_1) \cdot (\varphi_z(v_2) - \varphi_z(v_1)) - (\alpha_2 - \alpha_1) \cdot (\varphi_z(v_3) - \varphi_z(v_1)),\\ &= (\alpha_3 - \alpha_1) \cdot (\alpha_2 - \alpha_1) \cdot \mathbf{b}' - (\alpha_2 - \alpha_1) \cdot (\alpha_3 - \alpha_1) \cdot \mathbf{b}' = \mathbf{0},\end{aligned}$$

as required. □

Let $z \in \mathcal{Z}$, and let T_z be the set of triples $(\alpha_i, v_i)_{i\in[3]} \in (\mathbb{F} \times \mathcal{C})^3$ such that: 1) (α_i, v_i) is consistent with $(z, \mathbf{b}_1^*, \mathbf{b}_2^*)$ for all $i \in [3]$; 2) $U((\alpha_i, v_i)_{i\in[3]}) \neq 0$; 3) α_i is an MCA-bad point and v_i is an α_i-pretender, for all $i \in [3]$; and 4) $\alpha_i \neq \alpha_j$ for all $1 \leq i < j \leq 3$.

Lemma 3.7 states that a parameterized line in $\mathcal{C}$ contains few points $(\alpha, c) \in \mathbb{F} \times \mathcal{C}$ with c being an α-pretender. On the other hand, for every $\alpha \in A_z$, there exists an α-pretender $c \in \mathcal{C}$. Hence, intuitively, if $|A_z|$ is large, then there are many triples $(\alpha_i, c_i)_i$ that are not colinear, and so we expect $|T_z|$ to be large too. We formalize this intuition in the next result.

Lemma 3.10 (Bounding $|A_z|^3$ with $|T_z|$)**.** Let $z \in \mathcal{Z}$ and let T_z be defined as above. Then

$$|A_z| \cdot (|A_z| - 1) \cdot (|A_z| - \delta \cdot n - 1) \leq |T_z|.$$

Proof. For each $\alpha \in A_z$, fix an α-pretender $c_\alpha \in \mathcal{C}$ such that (α, c_α) is consistent with $(z, \mathbf{b}_1^*, \mathbf{b}_2^*)$. Such c_α exists by the definition of A_z. Let $Y = \{(\alpha, c_\alpha) \mid \alpha \in A_z\}$ and note that $|Y| = |A_z|$. Every triple $(\alpha_i, c_{\alpha_i})_{i\in[3]} \in Y^3$ with $\alpha_i \neq \alpha_j$ for all $i \neq j$ satisfies Conditions 1, 3 and 4 in the definition of T_z. Hence it suffices to lower bound the number of such triples for which $U((\alpha_i, c_{\alpha_i})_{i\in[3]}) \neq 0$.

We claim that for every pair $(\alpha, v), (\alpha', v') \in \mathbb{F} \times \mathcal{C}$ with $\alpha \neq \alpha'$, there exists a unique $\theta \in \mathcal{L}^1[\mathbb{F} \to \mathcal{C}]$ such that (α, v) and (α', v') both belong to $\mathsf{Graph}(\theta)$. Indeed, let $c_2 = \frac{v-v'}{\alpha-\alpha'} \in \mathcal{C}$ and $c_1 = v - \alpha \cdot c_2$. Now $v = c_1 + \alpha \cdot c_2$, and $c_1 + \alpha' \cdot c_2 = v - \alpha \cdot c_2 + \alpha' \cdot c_2 = v'$, hence (α, v) and (α', v') both belong to $\mathsf{Graph}(\theta_{c_1,c_2})$. On the other hand, if $\theta_{v_1,v_2}(\alpha) = v$ and $\theta_{v_1,v_2}(\alpha') = v'$ for some $v_1, v_2 \in \mathcal{C}$, then

$$v = v_1 + \alpha \cdot v_2, \qquad v' = v_1 + \alpha' \cdot v_2$$

Subtracting, $\frac{v-v'}{\alpha-\alpha'} = v_2 = c_2$, and then $v_1 = v - \alpha \cdot v_2 = v - \alpha \cdot c_2 = c_1$, hence $\theta_{v_1,v_2} = \theta_{c_1,c_2}$, completing the proof of the claim.

There are exactly $|A_z| \cdot (|A_z| - 1)$ pairs $(\alpha_1, c_{\alpha_1}), (\alpha_2, c_{\alpha_2}) \in Y$ with $\alpha_1 \neq \alpha_2$. Fixing such $(\alpha_1, c_{\alpha_1}), (\alpha_2, c_{\alpha_2}) \in Y$, let $\theta \in \mathcal{L}^1[\mathbb{F} \to \mathcal{C}]$ be the unique map such that both tuples belong to $\mathsf{Graph}(\theta)$ (see the claim above). Then, by Lemma 3.7, there are at most $\delta \cdot n + 1$ pairs $(\alpha_3, c_{\alpha_3}) \in Y$ belonging to the parameterized line $\mathsf{Graph}(\theta)$. Hence there are at least $|A_z| - \delta \cdot n - 1$ tuples $(\alpha_3, c_{\alpha_3}) \in Y$ that do not belong to $\mathsf{Graph}(\theta)$. Each of them satisfies $\alpha_3 \neq \alpha_1$ and $\alpha_3 \neq \alpha_2$, because otherwise, by definition of Y, (α_3, c_{α_3}) would be either (α_1, c_{α_1}) or (α_2, c_{α_2}), and these two tuples belong to $\mathsf{Graph}(\theta)$.

Overall, there are at least $|A_z| \cdot (|A_z| - 1) \cdot (|A_z| - \delta \cdot n - 1)$ triples $(\alpha_i, c_{\alpha_i})_{i\in[3]} \in Y^3$ with $\alpha_i \neq \alpha_j$ for all $i \neq j$, such that (α_3, c_{α_3}) does not belong to the unique parameterized line containing (α_1, c_{α_1}) and (α_2, c_{α_2}). By Lemma 3.8, for such triple we have $U((\alpha_i, c_{\alpha_i})_{i\in[3]}) \neq 0$. The lemma then follows, since, as we discussed before, T_z contains all the triples $(\alpha_i, c_{\alpha_i})_{i\in[3]} \in Y^3$ with $\alpha_i \neq \alpha_j$ for all $i \neq j$ such that $U((\alpha_i, c_{\alpha_i})_{i\in[3]}) \neq 0$. □

We are ready to prove Theorem 3.4.

Proof of Theorem 3.4. Let $\mathcal{Z}_t \subset \mathcal{Z}$ be the set of all $z \in \mathcal{Z}$ such that $|A_z| \geq t$ for all $z \in \mathcal{Z}_t$. Then by Lemma 3.10,

$$\begin{aligned} \sum_{z\in\mathcal{Z}_t} |T_z| &\geq \sum_{z\in\mathcal{Z}_t} |A_z| \cdot (|A_z| - 1) \cdot (|A_z| - \delta \cdot n - 1) \\ &\geq t \cdot (t-1) \cdot (t - \delta \cdot n - 1) \cdot |\mathcal{Z}_t| \end{aligned} \tag{4}$$

We now claim that

$$\sum_{z\in\mathcal{Z}} |T_z| \leq (|\mathbb{F}| \cdot L)^3 \cdot \varepsilon_{\mathsf{evader}} \cdot |\mathcal{Z}|, \tag{5}$$

and, additionally that if $\mathcal{C}$ has δ-MCA error $\varepsilon_{\mathsf{MCA}}$ then the factor $(|\mathbb{F}| \cdot L)^3$ in (5) can be replaced by $(\varepsilon_{\mathsf{MCA}} \cdot |\mathbb{F}| \cdot L)^3$. Indeed, let $P \subset \mathbb{F} \times \mathcal{C}$ be the set of all tuples (α, c) such that c is an α-pretender. We have $|P| \leq |\mathbb{F}| \cdot L$, where the list size factor L is due to the fact that an α-pretender is δ-close to $\theta_1(\alpha) \cdot u_1 + \theta_2(\alpha) \cdot u_2$, and since $\mathcal{C}$ is (δ, L)-list decodable, there are at most L codewords in $\mathcal{C}$ that are δ-close to $\theta_1(\alpha) \cdot u_1 + \theta_2(\alpha) \cdot u_2$. Additionally, we also

have $|P| \leq \varepsilon_{\mathsf{MCA}} \cdot |\mathbb{F}| \cdot L$ if $\mathcal{C}$ has δ-MCA error $\varepsilon_{\mathsf{MCA}}$, since for (α, c) to belong to P, we need α to be an MCA-bad element, and there are at most $\varepsilon_{\mathsf{MCA}} \cdot |\mathbb{F}|$ such elements.

Now, let $(\alpha_i, c_i)_{i\in[3]} \in P$ be such that $\alpha_i \neq \alpha_j$ for all $i \neq j$ and $U((\alpha_i, c_i)_{i\in[3]}) \neq 0$. Then if $(\alpha_i, c_i)_{i\in[3]} \in T_z$, by definition (α_i, c_i) is consistent with $(z, \mathbf{b}_1^*, \mathbf{b}_2^*)$ for all $i \in [3]$. By Lemma 3.9,$\varphi_z(U((\alpha_i, c_i)_{i\in[3]})) = 0$. Then, since $0 \neq U((\alpha_i, c_i)_{i\in[3]}) \in \mathcal{C}$, by the definition of zero evader there are at most $\varepsilon_{\mathsf{evader}} \cdot |\mathcal{Z}|$ elements $z \in \mathcal{Z}$ such that $\varphi_z(U((\alpha_i, c_i)_{i\in[3]})) = 0$. In other words, for every triple $(\alpha_i, c_i)_{i\in[3]} \in P^3$ such that $\alpha_i \neq \alpha_j$ for all $i \neq j$ and $U((\alpha_i, c_i)_{i\in[3]}) \neq 0$,

$$\left|\left\{z \in \mathcal{Z} \mid (\alpha_i, c_i)_i \in T_z\right\}\right| \leq \varepsilon_{\mathsf{evader}} \cdot |\mathcal{Z}|.$$

For the rest of triples $(\alpha_i, c_i)_{i\in[3]} \in P^3$ the same inequality holds, since then we actually have $|\{z \in \mathcal{Z} \mid (\alpha_i, c_i)_i \in T_z\}| = 0$. Now we obtain by double counting and using $T_z \subset P^3$,

$$\begin{aligned}\sum_{z\in\mathcal{Z}_t} |T_z| \leq \sum_{z\in\mathcal{Z}} |T_z| &= \left|\left\{\left(z, (\alpha_i, c_i)_{i\in[3]}\right) \in \mathcal{Z} \times P^3 \;\middle|\; (\alpha_i, c_i)_{i\in[3]} \in T_z\right\}\right| \\ &= \sum_{(\alpha_i,c_i)_{i\in[3]}\in P^3} \left|\left\{z \in \mathcal{Z} \;\middle|\; (\alpha_i, c_i)_{i\in[3]} \in T_z\right\}\right| \leq \sum_{(\alpha_i,c_i)_{i\in[3]}\in P^3} \varepsilon \cdot |\mathcal{Z}| \leq (M \cdot L)^3 \cdot \varepsilon \cdot |\mathcal{Z}|, \end{aligned} \tag{6}$$

where $\varepsilon = \varepsilon_{\mathsf{evader}}$ and M is either $|\mathbb{F}|$ or $\varepsilon_{\mathsf{MCA}} \cdot |\mathbb{F}|$, for $\varepsilon_{\mathsf{MCA}}$ the δ-MCA error of $\mathcal{C}$. The first inequality is due to $\mathcal{Z}_t \subset \mathcal{Z}$; the two equalities are the result of double counting, using that $T_z \subset P^3$ for the first one, so that $\{(z, (\alpha_i, c_i)_{i\in[3]}) \in \mathcal{Z} \times (\mathbb{F} \times \mathcal{C})^3 \mid (\alpha_i, c_i)_{i\in[3]} \in T_z\} = \{(z, (\alpha_i, c_i)_{i\in[3]}) \in \mathcal{Z} \times P^3 \mid (\alpha_i, c_i)_{i\in[3]} \in T_z\}$; the second inequality is due to our previous bound; and the last inequality is due to our prior bounds on $|P|$. This proves the claim.

Combining (4) and (6) we now obtain

$$t \cdot (t-1) \cdot (t - \delta \cdot n - 1) \cdot |\mathcal{Z}_t| \leq (M \cdot L)^3 \cdot \varepsilon_{\mathsf{evader}} \cdot |\mathcal{Z}|$$

and, since $t > \delta \cdot n + 1$, both sides of the inequality are positive and then the theorem follows since $\Pr_{z\leftarrow\mathcal{Z}}[|A_z| \geq t] = |\mathcal{Z}_t| \;/\; |\mathcal{Z}|$. □

Remark 3.11 (Multiple DEEP samples in the original DEEP work [BGK+20])**.** We consider the question of what would happen if in the original soundness analysis of [BGK+20] we increase the amount of DEEP samples, or, more generally, use an arbitrary zero evader. Upon inspecting the proofs of [BGK+20, Theorem 3 and Lemma 28][4], it is apparent that the proof can be word-by-word generalized so as to use an abstract evader $(\varphi_z)_{z\in\mathcal{Z}}$. However, the resulting error still requires a field of size $|\mathbb{F}| \geq 2^{904} \cdot L^6$, when targeting 128 bits of security (see our derivation below (1)). Indeed, the only spot in the proofs of [BGK+20, Theorem 3 and Lemma 28] where the zero evader error $\varepsilon_{\mathsf{evader}}$ plays a role is when concluding that, in their notation, if $(x_0, P_{x_0}(z)), (\beta_0, P_{\beta_0}(z)), (\gamma, P_\gamma(z))$ are colinear over $\mathbb{F} \times \mathbb{F}$ for more z's than $\varepsilon_{\mathsf{evader}} \cdot |\mathcal{Z}|$ (for [BGK+20], this is simply ℓ in Theorem 3 and $\frac{\sigma}{N}$ in Lemma 28), then $(x_0, P_{x_0}(X)), (\beta_0, P_{\beta_0}(X)), (\gamma, P_\gamma(X))$ are colinear over $\mathbb{F} \times \mathbb{F}[X]$. With an abstract zero evader, the proofs of [BGK+20, Theorem 3 and Lemma28] would then yield an error of

[4] In the eprint version of [BGK+20], the corresponding results are Theorem 4.1 and Lemma 4.6

$$\max\left(2 \cdot L \cdot (\varepsilon_{\mathsf{evader}} + \varepsilon)^{\frac{1}{3}}, \frac{4}{\varepsilon^2 \cdot |\mathbb{F}|}\right).$$

The derivation below (1) does not use the term $\ell/|\mathbb{F}|$, hence it applies as is.

4 Folding of constrained interleaved codes with BEEP

We now discuss how to use the soundness analysis from Theorem 3.4 so as to modify usual code-based IOPPs. In Section 4.1 we discuss the DEEP sampling and folding step, and then in Section 4.2 we provide a full description of BEEP-based IOPP.

4.1 The BEEP IOR

We assume the notation from Section 3. Throughout this section, we fix the linear maps θ_1, θ_2 as $\theta_1(\alpha) = 1 - \alpha$ and $\theta_2(\alpha) = \alpha$, so that two words u_1, u_2 are folded as $(1-\alpha) \cdot u_1 + \alpha \cdot u_2$. These maps satisfy the assumption $a_1 \cdot d_2 \neq a_2 \cdot d_1$ made in Section 3 (where $\theta_{i(X)} = a_i + d_i \cdot X$).

Define the following relation:

$$\mathcal{R}_{\mathcal{C}} := \left\{ \begin{pmatrix} \mathbb{x} = (\mu, \mathbf{v}) \in \mathbb{F} \times \mathbb{F}^\ell, \\ \mathbb{y} = f \in \mathbb{F}^n, \\ \mathbb{w} = \mathbf{f} \in \mathbb{F}^\ell \end{pmatrix} \;\middle|\; \begin{matrix} f = \mathsf{Enc}_{\mathcal{C}}(\mathbf{f}), \\ \langle \mathbf{f}, \mathbf{v} \rangle = \mu \end{matrix} \right\}.$$

Let $\mathcal{C}^{\equiv k} \subset (\mathbb{F}^k)^n$ be the k-interleaving of the code $\mathcal{C}$. We think of codewords $f \in \mathcal{C}^{\equiv k}$ as matrices of $k \times n$ field elements, where each row of the matrix is a codeword. We write the rows of f as

$$f = (f_{1*}, ..., f_{k*}) \in \mathcal{C}^k \subset (\mathbb{F}^n)^k, \qquad f_{i*} \in \mathcal{C} \text{ for all } i \in [k].$$

Similarly, we think of messages $\mathbf{f} \in \mathbb{F}^{k \cdot \ell}$ to be encoded with $\mathcal{C}^{\equiv k}$ as $k \times \ell$ matrices with rows $\mathbf{f}_{1*}, ..., \mathbf{f}_{k*} \in \mathbb{F}^\ell$.

We define an Interactive Oracle Reduction (IOR) from $\mathcal{R}_{\mathcal{C}^{\equiv 2}}$ to $\mathcal{R}_{\mathcal{C}}$ with a soundness analysis that does not require Mutual Correlated Agreement (MCA).

In Construction 4.1 below, we look at the linear constraint $\langle \mathbf{f}, \mathbf{v} \rangle = \mu$ from $\mathcal{R}_{\mathcal{C}^{\equiv 2}}$ as a sumcheck claim for the product of the MLE's of $\mathbf{f}$ and $\mathbf{v}$, exploiting that

$$\langle \mathbf{f}, \mathbf{v} \rangle = \sum_{\mathbf{b} \in \{0,1\}^{1+\log(\ell)}} \mathsf{MLE}[\mathbf{f}](\mathbf{b}) \cdot \mathsf{MLE}[\mathbf{v}](\mathbf{b}).$$

where here we index the entries of the $2 \times \ell$ matrices $\mathbf{f}$ and $\mathbf{v}$ over the set $\{0,1\}^{1+\log \ell}$, so that the first bit in the index indicates the row of the matrix. This way $\mathsf{MLE}[\mathbf{f}](X, \mathbf{Y}) = (1 - X) \cdot \mathsf{MLE}[\mathbf{f}_{1*}](\mathbf{Y}) + X \cdot \mathsf{MLE}[\mathbf{f}_{2*}](\mathbf{Y})$, and similarly for $\mathtt{MLE}[\mathbf{v}](X, \mathbf{Y})$, where $\mathbf{Y} = (Y_1, ..., Y_{\log(\ell)})$.

Construction 4.1 (BEEP, an IOR from $\mathcal{R}_{\mathcal{C}^{\equiv 2}}$ to $\mathcal{R}_{\mathcal{C}}$)**.** The prover and verifier receive as input $\mathbb{x} = (\mu, \mathbf{v}) \in \mathbb{F} \times \mathbb{F}^{2 \cdot \ell}$. The verifier also receives $\mathbb{y} = f \in (\mathbb{F}^2)^n$, and the prover receives $\mathbb{w} = \mathbf{f} \in \mathbb{F}^{2 \cdot \ell}$ such that $(\mathbb{x}, \mathbb{y}, \mathbb{w}) \in \mathcal{R}_{\mathcal{C}^{\equiv 2}}$.

Round 1 (DEEP sampling). The verifier samples and sends uniformly at random $z \in \mathcal{Z}$. The prover replies with two vectors $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{F}^m$, supposedly

$$\mathbf{b}_1 = \varphi_z(\mathbf{f}_{1*}), \qquad \mathbf{b}_2 = \varphi_z(\mathbf{f}_{2*})$$

Round 2 (Fold and sumcheck round). The prover sends the quadratic univariate polynomial[5]

$$h(X) := \sum_{\mathbf{b}\in\{0,1\}^{\log \ell}} \mathsf{MLE}[\mathbf{f}](X, \mathbf{b}) \cdot \mathsf{MLE}[\mathbf{v}](X, \mathbf{b}) \in \mathbb{F}^{<3}[X]$$

The verifier checks that $h(0) + h(1) = \mu$, and uniformly samples and sends a *folding and sumcheck challenge* $\alpha \in \mathbb{F}$. Define

$$\begin{aligned} g &:= (1-\alpha)\cdot f_{1*} + \alpha \cdot f_{2*} \in \mathbb{F}^n, \qquad & \mathbf{g} &:= (1-\alpha)\cdot \mathbf{f}_{1*} + \alpha \cdot \mathbf{f}_{2*} \in \mathbb{F}^\ell, \\ \mathbf{v}_\alpha &:= (1-\alpha)\cdot \mathbf{v}_{1*} + \alpha \cdot \mathbf{v}_{2*} \in \mathbb{F}^\ell \end{aligned}$$

Note that $\mathsf{MLE}[\mathbf{f}](\alpha, \mathbf{X}) = \mathsf{MLE}[(1-\alpha)\cdot \mathbf{f}_{1*} + \alpha \cdot \mathbf{f}_{2*}](\mathbf{X}) = \mathsf{MLE}[\mathbf{g}](\mathbf{X})$, and similarly $\mathsf{MLE}[\mathbf{v}](\alpha, \mathbf{X}) = \mathsf{MLE}[\mathbf{v}_\alpha](\mathbf{X})$. If the prover is honest, then $\mathbf{g}$ satisfies all of the following linear claims:

$$\begin{aligned} h(\alpha) &= \sum_{\mathbf{b}\in\{0,1\}^{\log \ell}} \mathsf{MLE}[\mathbf{g}](\mathbf{b}) \cdot \mathsf{MLE}[\mathbf{v}_\alpha](\mathbf{b}) = \langle \mathbf{g}, \mathbf{v}_\alpha \rangle, \\ \varphi_z(\mathbf{g}) &= \varphi_z((1-\alpha)\cdot \mathbf{f}_{1*} + \alpha \cdot \mathbf{f}_{2*}) = (1-\alpha)\cdot \mathbf{b}_1 + \alpha \cdot \mathbf{b}_2. \end{aligned} \tag{7}$$

Round 3 (Batching of linear claims). The verifier uniformly samples $\boldsymbol{\gamma} \in \mathbb{F}^{m+1}$ to batch the linear claims from (7), so that the claims are reduced to the single claim

$$\boldsymbol{\gamma} \cdot \begin{pmatrix} \mathbf{v}_\alpha \\ \varphi_z \end{pmatrix} \cdot \mathbf{g} = \boldsymbol{\gamma} \cdot \begin{pmatrix} h(\alpha) \\ (1-\alpha)\cdot \mathbf{b}_1 + \alpha \cdot \mathbf{b}_2 \end{pmatrix}$$

Output. The prover and verifier output

$$\begin{gathered} \mathbb{x}' := (\mu', \mathbf{v}'), \quad \mathbb{y}' := g \in \mathbb{F}^n, \quad \mathbb{w}' := \mathbf{g} \in \mathbb{F}^\ell, \\ \mathbf{v}' := \boldsymbol{\gamma} \cdot \begin{pmatrix} \mathbf{v}_\alpha \\ \varphi_z \end{pmatrix} \in \mathbb{F}^\ell, \quad \mu' := \boldsymbol{\gamma} \cdot \begin{pmatrix} h(\alpha) \\ (1-\alpha)\cdot \mathbf{b}_1 + \alpha \cdot \mathbf{b}_2 \end{pmatrix} \in \mathbb{F} \end{gathered}$$

For an honest prover, it holds that $(\mathbb{x}', \mathbb{y}', \mathbb{w}') \in \mathcal{R}_{\mathcal{C}}$.

Before stating and proving the completeness and knowledge soundness of Construction 4.1, we define relaxed versions of the relations $\mathcal{R}_{\mathcal{C}^{\equiv 2}}$ and $\mathcal{R}_{\mathcal{C}}$:

[5]Some optimizations related to sumcheck may be used in this step, see e.g. [Gru24] and [BDD+25].

Definition 4.2 (The relaxed relation $(\mathcal{R}_{\mathcal{C}'})^{\delta}$)**.** Let $\mathcal{C}' \subset \mathbb{F}^n$ be a linear code of blocklength n and dimension ℓ. Let $\delta \in (0,1)$ be a proximity parameter. We define the δ-relaxation of the relation $\mathcal{R}_{\mathcal{C}'}$ as:

$$\mathcal{R}^{\delta}_{\mathcal{C}'} := \left\{ \begin{pmatrix} \mathbb{x} = (\mu, \mathbf{v}) \in \mathbb{F} \times \mathbb{F}^{\ell}, \\ \mathbb{y} = f \in \mathbb{F}^n, \\ \mathbb{w} = \mathbf{f} \in \mathbb{F}^{\ell} \end{pmatrix} \;\middle|\; \begin{matrix} \Delta(f, \mathsf{Enc}_{\mathcal{C}'}(\mathbf{f})) \leq \delta, \\ \langle \mathbf{f}, \mathbf{v} \rangle = \mu \end{matrix} \right\}.$$

The relation $\mathcal{R}^{\delta}_{\mathcal{C}'}$ is exactly the same as $\mathcal{R}_{\mathcal{C}'}$, but replacing the constraint $f = \mathsf{Enc}_{\mathcal{C}'}(\mathbf{f})$ with $\Delta(f, \mathsf{Enc}_{\mathcal{C}'}(\mathbf{f})) \leq \delta$. The relaxation $\mathcal{R}^{\delta}_{\mathcal{C}^{\equiv 2}}$ of $\mathcal{R}_{\mathcal{C}^{\equiv 2}}$ is defined in the same way.

Theorem 4.3 (Completeness and round-by-round knowledge soundness of Construction 4.1)**.** Let $\mathcal{C} \subset \mathbb{F}^n$ be a linear code of blocklength n and dimension ℓ, and with minimum relative distance $\Delta(\mathcal{C})$. Let $\delta \in (0, \Delta(\mathcal{C}))$ be a proximity parameter such that $\delta \cdot n$ is an integer, and let $L_{\mathcal{C}}, L_{\mathcal{C}^{\equiv 2}}$ be such that $\mathcal{C}$ is $(\delta, L_{\mathcal{C}})$-list decodable and $\mathcal{C}^{\equiv 2}$ is $(\delta, L_{\mathcal{C}^{\equiv 2}})$-list decodable, respectively, (we do not require decodability to be efficient). Let $(\varphi_z)_{z \in \mathcal{Z}_{\mathsf{evader}}}$ be a zero evader with error $\varepsilon_{\mathsf{evader}}$.

Then Construction 4.1 is a perfectly complete three-round IOR from $\mathcal{R}_{\mathcal{C}^{\equiv 2}}$ to $\mathcal{R}_{\mathcal{C}}$ with round-by-round knowledge soundness with relaxation $(\mathcal{R}^{\delta}_{\mathcal{C}^{\equiv 2}}, \mathcal{R}^{\delta}_{\mathcal{C}})$, with errors

$$\kappa_1 := \frac{(M \cdot L_{\mathcal{C}})^3 \cdot \varepsilon_{\mathrm{evader}}}{(\delta \cdot n + 1)^2}, \qquad \kappa_2 := \frac{\delta \cdot n + 1 + 2 \cdot L_{\mathcal{C}^{\equiv 2}}}{|\mathbb{F}|}, \qquad \kappa_3 := \frac{L_{\mathcal{C}}}{|\mathbb{F}|},$$

where M is either $|\mathbb{F}|$ or $\varepsilon_{\mathsf{MCA}} \cdot |\mathbb{F}|$, for the δ-MCA error $\varepsilon_{\mathsf{MCA}}$ of $\mathcal{C}$.

We defer the proof of Theorem 4.3 to Appendix A, and only provide a brief outline of it here. The proof follows arguments similar to those in [BCF+25, CFW26], with the main difference being that Construction 4.1 has one initial extra round (the DEEP sampling round). For this round, it suffices to let the knowledge state function become 1 iff $|A_z| \geq \delta \cdot n + 2$ or if the input triple $(\mathbb{x}, \mathbb{y}, \mathbf{w})$ is valid, i.e. belongs to $\mathcal{R}^{\delta}_{\mathcal{C}^{\equiv 2}}$. The main result Theorem 3.4 from Section 3 then shows that a flip from state 0 to 1 occurs with probability at most κ_1.

After the first round, if the state is 0, we have $|A_z| < \delta \cdot n + 2$. Because of this, at this point, we are in a setting where our two words f_{1*}, f_{2*} effectively meet an MCA-like property. In this scenario, the arguments from [BCF+25, CFW26] apply. In particular, the extractor uses erasure decoding, leveraging the MCA-like property that Theorem 3.4 provides. As with [BCF+25, CFW26], this means that the code $\mathcal{C}$ does not require an efficient decoder.

4.2 Example PCS: BEEP-WHIR and BEEP-Ligerito

For completeness, we now obtain an IOR from $\mathcal{R}_{\mathcal{C}^{\equiv k}}$ to $\mathcal{R}_{\mathcal{C}'}$, where $\mathcal{C}'$ is a new linear code of the same dimension as $\mathcal{C}$. The oracle part $\mathbb{y}'$ output of this IOR consists of a single oracle, which is supposedly an oracle to a codeword from $\mathcal{C}'$ (this is in contrast to Construction 4.1, where the output $\mathbb{y}'$ is a linear combination of other oracles). One can then repeatedly apply this

IOR in order to obtain an IOPP (or, more intuitively, a PCS) for $\mathcal{R}_{\mathcal{C}^{\equiv k}}$. The resulting IOPP is a "BEEP variant" of Ligerito [NA25], or of WHIR [ACF+25] when using Reed-Solomon codes of different rates.

Remark 4.4 (BEEP-Basefold and BEEP-FRI-Binius [DP24, ZCF23])**.** One can equally envision BEEP variants of other code-based PCSs. For example, instead of performing the codeswitch step mentioned above, we could follow a more traditional approach where the spot checks made after repeatedly folding are verified by the verifier, rather than added as new terms to an ongoing sumcheck claim .

We mostly follow the blueprint of the Zook paper [CFW26, Theorem 9.12] (though we do not consider zk in our case). Concretely, we first compose Construction 4.1 with itself $\log k$ times, so as to obtain an IOR from $\mathcal{R}_{\mathcal{C}^{\equiv k}}$ to $\mathcal{R}_{\mathcal{C}}$, and then we compose this IOR with the code-switching IOR from [CFW26], without zero-knowledge.

Besides the fact that our folding step uses the BEEP sampling technique, this construction is well-known [ACF+25, BCF+24, CFW26, NA25]. Because of this we discuss it only in broad terms.

Construction 4.5 (BEEP folding and code switching: an IOR from $\mathcal{R}_{\mathcal{C}^{\equiv k}}$ to $\mathcal{R}_{\mathcal{C}'}$)**. Parameters.** The code $\mathcal{C}$; a power of two k; a linear code $\mathcal{C}'$ with message space $\mathbb{F}^\ell$ (possibly an interleaved code); zero evaders $(\varphi_z^{(j)})_{z\in\mathcal{Z}_j}$ on $\mathbb{F}^{k\cdot\ell/2^j}$ for $j \in [\log k]$, and $(\varphi'_z)_{z\in\mathcal{Z}'}$ with $\varphi'_z : \mathbb{F}^\ell \to \mathbb{F}^{m'}$; a number of queries $Q \geq 1$.
Inputs. The prover and verifier receive as input $\mathbb{x} = (\mu, \mathbf{v}) \in \mathbb{F} \times \mathbb{F}^{k\cdot\ell}$ and $\mathbb{y} = f \in (\mathbb{F}^k)^n$. The prover also receives $\mathbb{w} = \mathbf{f} \in \mathbb{F}^{k\cdot\ell}$ such that $(\mathbb{x}, \mathbb{y}, \mathbb{w}) \in \mathcal{R}_{\mathcal{C}^{\equiv k}}$.

Step 1: Folding. Let $(\mathbb{x}_0, \mathbb{y}_0, \mathbb{w}_0) := (\mathbb{x}, \mathbb{y}, \mathbb{w})$ and $\mathcal{C}_j = \mathcal{C}^{\equiv k/2^j}$ for $j \geq 0$. Note that $\mathcal{C}_{j-1} = \mathcal{C}_j^{\equiv 2}$ for all $j \geq 1$. For $j \in [\log k]$, the prover and verifier engage in Construction 4.1 to reduce the claim that $(\mathbb{x}_{j-1}, \mathbb{y}_{j-1}, \mathbb{w}_{j-1}) \in \mathcal{R}_{\mathcal{C}_j^{\equiv 2}}$ to a claim of the form $(\mathbb{x}_j, \mathbb{y}_j, \mathbb{w}_j) \in \mathcal{R}_{\mathcal{C}_j}$ for $\mathbb{x}_j, \mathbb{y}_j, \mathbb{w}_j$ output by Construction 4.1. Write $((\hat{\mu}, \hat{\mathbf{v}}), g, \mathbf{g}) := (\mathbb{x}_{\log k}, \mathbb{y}_{\log k}, \mathbb{w}_{\log k})$. If the prover is honest, we have $((\hat{\mu}, \hat{\mathbf{v}}), g, \mathbf{g}) \in \mathcal{R}_{\mathcal{C}}$.

An important property of g is that it is a linear combination of the rows of $f \in \mathbb{F}^{k\times n}$. Hence the verifier can obtain $g(x)$ for any $x \in [n]$ by querying f.

Step 2: Code switching. We apply the IOR from [CFW26, Construction 9.9], without zero-knowledge. The prover and verifier start with the claim $((\hat{\mu}, \hat{\mathbf{v}}), g, \mathbf{g}) \in \mathcal{R}_{\mathcal{C}}$ and proceed as follows:

Round 1 (New oracle and out of domain samples). The prover sends a word g', supposedly $g' = \mathsf{Enc}_{\mathcal{C}'}(\mathbf{g})$. The verifier uniformly samples and sends $z' \in \mathcal{Z}'$. The prover replies with $\mathbf{b}' \in \mathbb{F}^{m'}$, supposedly $\mathbf{b}' = \varphi'_{z'}(\mathbf{g})$.

Round 2 (Queries). The verifier uniformly samples and sends $x_1, ..., x_Q \in [n]$. If the prover is honest, then $\mathbf{g}$ satisfies the following linear claims:

$$\langle \mathbf{g}, \hat{\mathbf{v}} \rangle = \hat{\mu}, \qquad \varphi'_{z'}(\mathbf{g}) = \mathbf{b}', \qquad (M_{\mathcal{C}})_{x_i} \cdot \mathbf{g} = g(x_i) \quad \text{for all } i \in [Q], \tag{8}$$

where $(M_{\mathcal{C}})_x$ denotes the x-th row of an encoding matrix $M_{\mathcal{C}}$ of $\mathcal{C}$. The verifier obtains the values $g(x_1), ..., g(x_Q)$ by querying f, as explained previously.

Round 3 (Batching of linear claims). The verifier uniformly samples and sends $\boldsymbol{\gamma}' \in \mathbb{F}^{1+m'+Q}$ to batch the linear claims from (8) into a single claim.

Output. The prover and verifier output $\mathbb{x}' := (\mu', \mathbf{v}')$, $\mathbb{y}' := g'$ and $\mathbb{w}' := \mathbf{g}$, where

$$\mathbf{v}' := \boldsymbol{\gamma}' \cdot \left(\hat{\mathbf{v}}, \varphi'_{z'}, (M_{\mathcal{C}})_{x_1}, ..., (M_{\mathcal{C}})_{x_Q}\right)^{\mathsf{T}}, \quad \mu' := \boldsymbol{\gamma}' \cdot \left(\hat{\mu}, \mathbf{b}', g(x_1), ..., g(x_Q)\right)^{\mathsf{T}}.$$

For an honest prover, it holds that $(\mathbb{x}', \mathbb{y}', \mathbb{w}') \in \mathcal{R}_{\mathcal{C}'}$.

Theorem 4.6 (Completeness and round-by-round knowledge soundness of Construction 4.5)**.** Let $(\varepsilon_j)_{j \in [\log k]}$ and ε' be error bounds of the zero evaders used in Construction 4.5. Let $\delta \in (0, \Delta(\mathcal{C}))$, with $\delta \cdot n$ an integer, and $\delta' \in (0, 1)$. Let $L_0, ..., L_{\log k}$ and L' be such that $\mathcal{C}^{\equiv k/2^j}$ is (δ, L_j)-list decodable for all j, and $\mathcal{C}'$ is (δ', L')-list decodable.

Then Construction 4.5 is a perfectly complete IOR from $\mathcal{R}_{\mathcal{C}^{\equiv k}}$ to $\mathcal{R}_{\mathcal{C}'}$ with $3 \cdot \log k + 3$ rounds and with round-by-round knowledge soundness with relaxation $(\mathcal{R}^{\delta}_{\mathcal{C}^{\equiv k}}, \mathcal{R}^{\delta'}_{\mathcal{C}'})$. Its errors are $(\kappa_{j,1}, \kappa_{j,2}, \kappa_{j,3})_{j \in [\log k]}$ for the three rounds of each execution of Construction 4.1 during the Folding process, and $(\kappa'_1, \kappa'_2, \kappa'_3)$ for the three rounds of Step 2, where

$$\kappa_{j,1} := \frac{\left(M \cdot L_j\right)^3 \cdot \varepsilon_j}{(\delta \cdot n + 1)^2}, \quad \kappa_{j,2} := \frac{\delta \cdot n + 1 + 2 \cdot L_{j-1}}{|\mathbb{F}|}, \quad \kappa_{j,3} := \frac{L_j}{|\mathbb{F}|},$$
$$\kappa'_1 := \frac{(L')^2 \cdot \varepsilon'}{2}, \quad \kappa'_2 := (1 - \delta)^Q, \quad \kappa'_3 := \frac{L'}{|\mathbb{F}|},$$

and M is either $|\mathbb{F}|$ or $\varepsilon_{\mathsf{MCA}} \cdot |\mathbb{F}|$, for the δ-MCA error $\varepsilon_{\mathsf{MCA}}$ of $\mathcal{C}^{\equiv k/2^j}$.

Proof outline. Once the completeness and round-by-round knowledge of the code-switching step are established, the theorem follows from Theorem 4.3 and the lemma for composing IORs (Lemma 2.13). The code switching step is an instantiation of [CFW26, Construction 9.9] without zero-knowledge, for the relations $\mathcal{R}_{\mathcal{C}}$ and $\mathcal{R}_{\mathcal{C}'}$. This IOR is clearly perfectly complete, and, by [CFW26, Lemma 9.11], this IOR has round-by-round knowledge soundness with relaxation $(\mathcal{R}^{\delta}_{\mathcal{C}}, \mathcal{R}^{\delta'}_{\mathcal{C}'})$ and errors $\kappa'_1, \kappa'_2, \kappa'_3$ (in [CFW26] the queries and the batching challenge are sent in the same round, with error $\kappa'_2 + \kappa'_3$; its proof analyses them one after the other, as we do). □

5 BEEP with affine folding for Reed-Solomon codes

As we mentioned, in the context of Reed-Solomon codes, in [FS26] and [CWY+26], the authors propose, instead of sampling challenges α in $\mathbb{F}$, to sample low degree polynomials. [CWY+26] specifically samples α in the set of linear polynomials $\mathbb{F}^{\mathsf{lin}}[X]$, so that $\alpha = \alpha(X) = \sigma + \tau \cdot X$ for some random $\sigma, \tau \in \mathbb{F}$. We refer to Section 1.1 for more details on [CWY+26, FS26].

In this section we adapt the results from Section 3 to the same setting as [CWY+26], i.e. we sample folding challenges α in $\mathbb{F}^{\mathsf{lin}}[X]$ instead of in $\mathbb{F}$. As a result, the fraction of folding challenges that are DEEP-MCA-bad goes from $(\delta \cdot n + 1)/|\mathbb{F}|$ (Theorem 3.4) to $2/|\mathbb{F}|$ (Theorem 5.3).These techniques apply specifically to Reed-Solomon codes, and not to any linear codes as those of Sections 3 and 4.

In Remark 5.6 we briefly outline how an IOPP using Theorem 5.3 would look like, following [CWY+26, Section 5]. We only mention one point to keep in mind when using Theorem 5.3. In particular, we address one technical subtlety of [CWY+26] and our result. This is that both results consider Correlated Agreement with a set of pairs of codewords $\mathcal{C}_{\theta}^{\equiv 2}$, which is slightly larger than the interleaved code $\mathcal{C}^{\equiv 2}$ of the Reed-Solomon code $\mathcal{C}$ at hand. Remark 5.6 discusses how to obtain Correlated Agreement with $\mathcal{C}^{\equiv 2}$, which is a technique presented in [CWY+26, Section 5].

5.1 Affine pretender codewords and consistency

Fix n distinct elements $x_1, ..., x_n \in \mathbb{F}$. For an integer e with $1 \leq e \leq n$, we let $\mathsf{RS}_e \subset \mathbb{F}^n$ be the Reed-Solomon code of dimension e with evaluation domain $\{x_1, ..., x_n\}$, i.e.

$$\mathsf{RS}_e := \{(p(x_1), ..., p(x_n)) \mid p \in \mathbb{F}[X], \deg(p) < e\}.$$

We identify each codeword of RS_e with its polynomial of degree less than e. This way, given a codeword c and $x \in \mathbb{F}$, we write $c(x)$ for the value of this polynomial at x.

We fix a code dimension $\ell \geq 1$ and we let

$$\mathcal{C} := \mathsf{RS}_{\ell}, \qquad \mathcal{C}^{+} := \mathsf{RS}_{\ell+1}.$$

Given a polynomial $p \in \mathbb{F}[X]$ and a word $u \in \mathbb{F}^n$, we write $p \cdot u$ for the word

$$(p(x_1) \cdot u_1, ..., p(x_n) \cdot u_n) \in \mathbb{F}^n \tag{9}$$

Let $\mathbb{F}^{\mathsf{lin}}[X] := \{\sigma + \tau \cdot X \mid \sigma, \tau \in \mathbb{F}\} \subset \mathbb{F}[X]$ be the set of linear polynomials on $\mathbb{F}$. As in Section 3, fix two linear maps $\theta_i(Y) := a_i + d_i \cdot Y$, $i = 1, 2$, with[Section] 3, $a_1 \cdot d_2 \neq a_2 \cdot d_1$. Given two words $u_1, u_2 \in \mathbb{F}^n$ and a folding challenge $\alpha \in \mathbb{F}^{\mathsf{lin}}[X]$, we define the α-fold of u_1, u_2 as the word (recall our notation convention from (9))

$$\theta_1(\alpha) \cdot u_1 + \theta_2(\alpha) \cdot u_2 \in \mathbb{F}^n.$$

Note that if $c_1, c_2 \in \mathcal{C}$, then the folded word $\theta_1(\alpha) \cdot c_1 + \theta_2(\alpha) \cdot c_2$ belongs to $\mathcal{C}^{+}$ for all $\alpha \in \mathbb{F}^{\mathsf{lin}}[X]$.

For technical reasons, we will need to work with pairs of codewords $c_1, c_2 \in \mathcal{C}^{+}$ whose fold belongs to $\mathcal{C}^{+}$ for all $\alpha \in \mathbb{F}^{\mathsf{lin}}[X]$. These are precisely those in

$$\mathcal{C}_{\theta}^{\equiv 2} := \{(c_1, c_2) \in \mathcal{C}^{+} \times \mathcal{C}^{+} \mid d_1 \cdot c_1 + d_2 \cdot c_2 \in \mathcal{C}\}.$$

In [CWY+26], $\mathcal{C}_\theta^{\equiv 2}$ is the set $\mathcal{C}^+ \times \mathcal{C}$, which corresponds to the specific setting $\theta_1(\alpha) = 1$, $\theta_2(\alpha) = \alpha$, which is the one considered in [CWY+26].

Let $u_1, u_2 \in \mathbb{F}^n$ be two words and let $\delta \in (0,1)$. Given $S \subset [n]$, we say that (u_1, u_2) has *Correlated Agreement with $\mathcal{C}_\theta^{\equiv 2}$ over S up to one position* if there exist a pair $(c_1, c_2) \in \mathcal{C}_\theta^{\equiv 2}$ and a set $S' \subset S$ with $|S'| \geq |S| - 1$ such that u_1 agrees with c_1 on S' and u_2 agrees with c_2 on S'.

Definition 5.1 (Pretender codeword)**.** Let $u_1, u_2 \in \mathbb{F}^n$ be two words and let $\alpha \in \mathbb{F}^{\mathsf{lin}}[X]$. We say that a codeword $c \in \mathcal{C}^+$ is an *α-pretender* (with respect to u_1, u_2) if $\theta_1(\alpha) \cdot u_1 + \theta_2(\alpha) \cdot u_2$ and c agree on a set $S \subset [n]$ of size $|S| \geq (1-\delta) \cdot n$ such that (u_1, u_2) does not have Correlated Agreement with $\mathcal{C}_\theta^{\equiv 2}$ over S up to one position. We say that α is *MCA-bad* (with respect to u_1, u_2) if there exists an α-pretender.

Let $m \geq 1$ be a parameter. Given $\mathbf{z} = (z_1, ..., z_m) \in \mathbb{F}^m$, let $\boldsymbol{\varphi}_{\mathbf{z}} : \mathbb{F}[X] \to \mathbb{F}^m$ consist of evaluating a polynomial at the entries of $\mathbf{z}$, i.e.

$$\boldsymbol{\varphi}_{\mathbf{z}}(p) := (p(z_1), ..., p(z_m)) \in \mathbb{F}^m.$$

This will be our zero evader, and is precisely the univariate evader from Section 2.3 applied to the coefficients of p.

Given $\mathbf{b}, \mathbf{b}' \in \mathbb{F}^m$ we write $\mathbf{b} \odot \mathbf{b}' \in \mathbb{F}^m$ for the entrywise product of $\mathbf{b}$ and $\mathbf{b}'$. For any two $p, p' \in \mathbb{F}[X]$ we have $\boldsymbol{\varphi}_{\mathbf{z}}(p \cdot p') = \boldsymbol{\varphi}_{\mathbf{z}}(p) \odot \boldsymbol{\varphi}_{\mathbf{z}}(p')$.

Definition 5.2 (Consistency of MCA-bad elements $\alpha \in \mathbb{F}^{\mathsf{lin}}[X]$ with $\mathbf{z} \in \mathbb{F}^m$)**.** Let $\alpha \in \mathbb{F}^{\mathsf{lin}}[X]$ and $c \in \mathcal{C}^+$. Given $\mathbf{z} \in \mathbb{F}^m$ and $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{F}^m$, we say that (α, c) is *consistent* with $(\mathbf{z}, \mathbf{b}_1, \mathbf{b}_2)$ if

$$\boldsymbol{\varphi}_{\mathbf{z}}(c) = \boldsymbol{\varphi}_{\mathbf{z}}(\theta_1(\alpha)) \odot \mathbf{b}_1 + \boldsymbol{\varphi}_{\mathbf{z}}(\theta_2(\alpha)) \odot \mathbf{b}_2.$$

When α is MCA-bad and there exists an α-pretender $c \in \mathcal{C}^+$ such that (α, c) is consistent with $(\mathbf{z}, \mathbf{b}_1, \mathbf{b}_2)$, we say that α is a *DEEP-MCA-bad challenge* with respect to $\mathbf{z}, \mathbf{b}_1, \mathbf{b}_2$ and u_1, u_2.

For intuition, for an honest prover $\mathbf{b}_1, \mathbf{b}_2$ will be values $\boldsymbol{\varphi}_{\mathbf{z}}(c_1), \boldsymbol{\varphi}_{\mathbf{z}}(c_2)$, where $(c_1, c_2) \in \mathcal{C}^{\equiv 2}$ are δ-close to u_1 and u_2.

5.2 The affine version of Theorem 3.4

We now state and prove the affine version of Theorem 3.4. We refer to Remark 3.5 and to Remark 5.6 below for explanations on how to use it inside a code-based PCS (see also Section 2.4).

Theorem 5.3 (For most $\mathbf{z} \in \mathbb{F}^m$, there are few DEEP-MCA-bad challenges $\alpha \in \mathbb{F}^{\mathsf{lin}}[X]$)**.** Let $\delta \in (0,1)$ and let L be such that $\mathcal{C}^+$ is (δ, L)-list decodable. Let $u_1, u_2 \in \mathbb{F}^n$ be two words.

For each $\mathbf{z} \in \mathbb{F}^m$ and $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{F}^m$, let $A_{\mathbf{z},\mathbf{b}_1,\mathbf{b}_2} \subset \mathbb{F}^{\mathsf{lin}}[X]$ be the set of DEEP-MCA-bad challenges with respect to $\mathbf{z}, \mathbf{b}_1, \mathbf{b}_2, u_1, u_2$. Let $\mathbf{b}_1^*, \mathbf{b}_2^*$ be such that $|A_{\mathbf{z},\mathbf{b}_1^*,\mathbf{b}_2^*}|$ is maximal for fixed $\mathbf{z}, u_1, u_2$, and denote $A_{\mathbf{z}} := A_{\mathbf{z},\mathbf{b}_1^*,\mathbf{b}_2^*}$. Let

$$N := \max\left(|\mathbb{F}|, \frac{(\delta \cdot n + 2) \cdot (\delta \cdot n + 1)}{2}\right).$$

Then for any integer $t \geq N + 1$,

$$\Pr_{\mathbf{z} \leftarrow \mathbb{F}^m}[|A_{\mathbf{z}}| \geq t] \leq \frac{\left(|\mathbb{F}|^2 \cdot L\right)^3 \cdot ((\ell + 1)/\ |\mathbb{F}|)^m}{t \cdot (t-1) \cdot (t-N)}. \tag{10}$$

Assuming $N = |\mathbb{F}|$ and taking $t = 2N$ we obtain

$$\Pr_{\mathbf{z} \leftarrow \mathbb{F}^m}[|A_{\mathbf{z}}| \geq t] \leq \frac{(|\mathbb{F}| \cdot L)^3}{3} \cdot \left(\frac{\ell + 1}{|\mathbb{F}|}\right)^m$$

We proceed to prove Theorem 5.3. The proof is almost a word-by-word adaptation of that of Theorem 3.4 to the new definitions introduced above. The main difference is in Lemma 3.7, which states that a parameterized line contains few pretenders. In its affine version, this is a generalization of [CWY+26, Lemma 4.4] with a slightly better bound, and arbitrary folding weights $\theta_1(\alpha), \theta_2(\alpha)$ and proximity parameter δ. We include a proof for completeness.

We start with a new definition of parameterized lines. Below, $\mathbb{F}(X)$ denotes the set of all rational functions over $\mathbb{F}$, on the variable X.

Definition 5.4 (Parameterized lines for the affine fold)**.** We let $\mathcal{L}^1[\mathbb{F}^{\mathsf{lin}}[X] \to \mathbb{F}(X)]$ be the set of all maps $\theta : \mathbb{F}^{\mathsf{lin}}[X] \to \mathbb{F}(X)$ of the form $\theta(\alpha) = F_\theta + \alpha \cdot G_\theta$ for some $F_\theta, G_\theta \in \mathbb{F}(X)$. Given $\theta \in \mathcal{L}^1[\mathbb{F}^{\mathsf{lin}}[X] \to \mathbb{F}(X)]$, we let $\mathsf{Graph}(\theta) = \{(\alpha, \theta(\alpha)) \mid \alpha \in \mathbb{F}^{\mathsf{lin}}[X]\} \subset \mathbb{F}^{\mathsf{lin}}[X] \times \mathbb{F}(X)$, and we call $\mathsf{Graph}(\theta)$ a *parameterized line*.

Lemma 5.5 (A parameterized line contains few pretenders, for the affine fold; analogue of Lemma 3.7; slight generalization of [CWY+26, Lemma 4.4])**.** Let $u_1, u_2 \in \mathbb{F}^n$ be two words, and let $\theta \in \mathcal{L}^1[\mathbb{F}^{\mathsf{lin}}[X] \to \mathbb{F}(X)]$. Let $A_\theta^{\mathsf{pret}} \subset \mathbb{F}^{\mathsf{lin}}[X]$ be the set of all $\alpha \in \mathbb{F}^{\mathsf{lin}}[X]$ such that $\theta(\alpha) \in \mathcal{C}^+$ and $\theta(\alpha)$ is an α-pretender with respect to u_1, u_2. Then

$$|A_\theta^{\mathsf{pret}}| \leq \max\left(|\mathbb{F}|, \frac{(\delta \cdot n + 2) \cdot (\delta \cdot n + 1)}{2}\right) = N.$$

Proof. Fix $\theta = F_\theta + G_\theta \cdot Y \in \mathcal{L}^1[\mathbb{F}^{\mathsf{lin}}[X] \to \mathbb{F}(X)]$. Recall that $\theta_i(Y) := a_i + d_i \cdot Y$, $i = 1, 2$, with $a_1 \cdot d_2 \neq a_2 \cdot d_1$. As in the proof of Lemma 3.7, since $a_1 \cdot d_2 \neq a_2 \cdot d_1$ there exist $F_1, F_2 \in \mathbb{F}(X)$ such that $\theta(Y) = \theta_1(Y) \cdot F_1 + \theta_2(Y) \cdot F_2$ as polynomials on Y (it suffices to solve the system of equations over the field $\mathbb{K} := \mathbb{F}(X)$ obtained by equating the coefficients of the linear polynomials on Y on both sides).

Let $V \subset \mathbb{F}^{\mathsf{lin}}[X]$ be the set of all $\alpha \in \mathbb{F}^{\mathsf{lin}}[X]$ such that $\theta(\alpha) \in \mathcal{C}^+$. Note that $\mathbb{F}^{\mathsf{lin}}[X]$ is a linear space of dimension two over $\mathbb{F}$. We claim that either $V = \mathbb{F}^{\mathsf{lin}}[X]$ or $|V| \leq |\mathbb{F}|$. If V is empty there is nothing to prove, so assume that V contains some α_0. Let $W \subset \mathbb{F}^{\mathsf{lin}}[X]$ be the set of all $\beta \in \mathbb{F}^{\mathsf{lin}}[X]$ such that $\beta \cdot G_\theta \in \mathcal{C}^+$. Since $\mathcal{C}^+$ is an $\mathbb{F}$-vector space, W is an $\mathbb{F}$-vector subspace of $\mathbb{F}^{\mathsf{lin}}[X]$. Now, since $\theta(\alpha_0) \in \mathcal{C}^+$, an element $\alpha \in \mathbb{F}^{\mathsf{lin}}[X]$ belongs to V if and only if $(\alpha - \alpha_0) \cdot G_\theta \in \mathcal{C}^+$, i.e. if and only if $\alpha - \alpha_0 \in W$. It follows that $V = \{\alpha_0 + \beta \mid \beta \in W\}$, and so $|V| = |W|$. But W is an $\mathbb{F}$-vector space of dimension ≤ 2, hence either $|V| \leq |\mathbb{F}|$ or , or $V = \mathbb{F}^{\mathsf{lin}}[X]$.

By definition $A_\theta^{\mathsf{pret}} \subset V$. Hence if $V \neq \mathbb{F}^{\mathsf{lin}}[X]$, we have $|A_\theta^{\mathsf{pret}}| \leq |\mathbb{F}|$ and we are done. Hence assume $V = \mathbb{F}^{\mathsf{lin}}[X]$. We claim that $(F_1, F_2) \in \mathcal{C}_\theta^{\equiv 2}$. Indeed, since $V = \mathbb{F}^{\mathsf{lin}}[X]$, we have $(\theta(0), \theta(1) - \theta(0), \theta(X) - \theta(0)) = (F_\theta, G_\theta, X \cdot G_\theta) \in {\mathcal{C}^+}^3$, meaning $F_\theta \in \mathcal{C}^+$ and $G_\theta \in \mathcal{C}$. We know that the following equality of polynomials holds

$$\theta(Y) = \theta_1(Y) \cdot F_1 + \theta_2(Y) \cdot F_2 = (a_1 \cdot F_1 + a_2 \cdot F_2) + (d_1 \cdot F_1 + d_2 \cdot F_2) \cdot Y = F_\theta + Y \cdot G_\theta$$

and so $a_1 \cdot F_1 + a_2 \cdot F_2 = F_\theta \in \mathcal{C}^+$ and $d_1 \cdot F_1 + d_2 \cdot F_2 = G_\theta \in \mathcal{C}$. Since $a_1 \cdot d_2 \neq a_2 \cdot d_1$, solving this system we obtain that F_1 and F_2 are linear combinations of F_θ and G_θ with coefficients in $\mathbb{F}$, and so $F_1, F_2 \in \mathcal{C}^+$. Hence $(F_1, F_2) \in \mathcal{C}_\theta^{\equiv 2}$, as claimed.

Similarly as in Lemma 3.7, define

$$D := \{i \in [n] \mid u_1(i) \neq F_1(x_i) \text{ or } u_2(i) \neq F_2(x_i)\} \subset [n].$$

Looking ahead, from here on the proof is analogous to that of Lemma 3.7, except that the sets $S_\alpha \cap D$ (for $\alpha \in A_\theta^{\mathsf{pret}}$) contain at least 2 elements instead of 1, and, for $\alpha \neq \alpha'$, these sets pairwise intersect on at most one element, instead of being pairwise disjoint.

Let $\alpha \in A_\theta^{\mathsf{pret}}$. Since $\theta(\alpha)$ is an α-pretender, there is a set $S_\alpha \subset [n]$ where $\theta(\alpha)$ and $\theta_1(\alpha) \cdot u_1 + \theta_2(\alpha) \cdot u_2$ agree, with $|S_\alpha| \geq (1 - \delta) \cdot n$ and such that (u_1, u_2) does not have Correlated Agreement with $\mathcal{C}_\theta^{\equiv 2}$ over S_α up to one position. We argue that $|S_\alpha \cap D| \geq 2$. Indeed, assume towards contradiction that $|S_\alpha \cap D| \leq 1$, and let $S_\alpha{}' := S_\alpha \setminus D$. Then $S_\alpha{}' \subset S_\alpha$ is a set of size $\geq |S_\alpha| - 1$ over which u_1, u_2 agree with F_1, F_2, respectively (by definition of D). Since $(F_1, F_2) \in \mathcal{C}_\theta^{\equiv 2}$, this means that (u_1, u_2) has Correlated Agreement with $\mathcal{C}_\theta^{\equiv 2}$ over S_α up to one position, a contradiction. Hence $|S_\alpha \cap D| \geq 2$ for all $\alpha \in A_\theta^{\mathsf{pret}}$.

We now claim that for every two distinct $\alpha, \alpha' \in A_\theta^{\mathsf{pret}}$ the set $(S_\alpha \cap D) \cap (S_{\alpha'} \cap D)$ has at most one element. Indeed, let $i \in [n]$ be an element of such set. Then

$$\theta_1(\alpha(x_i)) \cdot u_1(i) + \theta_2(\alpha(x_i)) \cdot u_2(i) = \theta_1(\alpha(x_i)) \cdot F_1(x_i) + \theta_2(\alpha(x_i)) \cdot F_2(x_i),$$

and similarly for α'. Since $i \in D$, the vector $(u_1(i) - F_1(x_i), u_2(i) - F_2(x_i))$ is nonzero, and so the vectors $(\theta_1(\alpha(x_i)), \theta_2(\alpha(x_i)))$ and $(\theta_1(\alpha'(x_i)), \theta_2(\alpha'(x_i)))$ are linearly dependent. Hence the determinant of $((a_1 + \alpha(X) \cdot d_1, a_2 + \alpha(X) \cdot d_2), (a_1 + \alpha'(X) \cdot d_1, a_2 + \alpha'(X) \cdot d_2))$ vanishes at x_i. This determinant equals $(\alpha(X) - \alpha'(X)) \cdot (a_2 \cdot d_1 - d_2 \cdot a_1)$. Since we assumed $a_2 \cdot d_1 \neq d_2 \cdot a_1$, we have $\alpha(x_i) = \alpha'(x_i)$. Since $\alpha \neq \alpha'$ are polynomials of degree at most one, there is at most one $i \in D$ satisfying $\alpha(x_i) = \alpha'(x_i)$. This proves the claim.

Because of the claim, every subset of D with two elements is contained in S_α for at most one $\alpha \in A_\theta^{\mathsf{pret}}$. By the same argument as in the proof of Lemma 3.7 we have $|S_\alpha \cap D| \geq |D| - \delta \cdot n$. Let $s := \max(2, |D| - \delta \cdot n)$ so that $|S_\alpha \cap D| \geq s$ for all $\alpha \in A_\theta^{\mathsf{pret}}$. Now, double counting the set $E = \{(\alpha, \{i, j\}) \mid \alpha \in A_\theta^{\mathsf{pret}}, \{i, j\} \subset S_\alpha \cap D, i \neq j\}$,

$$|A_\theta^{\mathsf{pret}}| \cdot \frac{s \cdot (s-1)}{2} \leq |E| \leq \frac{|D| \cdot (|D|-1)}{2}$$

where the second inequality uses that every subset of D with two elements is contained in at most one S_α. Now if $|D| \leq \delta \cdot n + 2$ then $s = 2$ and we are done. Otherwise $s = |D| - \delta \cdot n$ and

$$|A_\theta^{\mathsf{pret}}| \leq \frac{|D|}{|D| - \delta \cdot n} \cdot \frac{|D| - 1}{|D| - \delta \cdot n - 1} < \frac{\delta \cdot n + 2}{2} \cdot (\delta \cdot n + 1),$$

completing the proof. □

Equipped with Lemma 5.5, we now prove Theorem 5.3 by adapting word by word the proof of Theorem 3.4 to the affine folding setting.

Proof of Theorem 5.3. We follow the proof of Theorem 3.4, instantiated with $\mathcal{C}$ a Reed-Solomon code, taking challenges $\alpha \in \mathbb{F}^{\mathsf{lin}}[X]$ instead of in $\mathbb{F}$, and using the parameterized lines $\mathcal{L}^1[\mathbb{F}^{\mathsf{lin}}[X] \to \mathbb{F}(X)]$ instead of $\mathcal{L}^1[\mathbb{F} \to \mathcal{C}]$. We go through the proof and explain in what ways it is the same as that of Theorem 3.4 and, when it is not, we explain how to modify the proof to our setting.

First, let $(\alpha_i, v_i)_{i\in[3]} \in (\mathbb{F}^{\mathsf{lin}}[X] \times \mathcal{C}^+)^3$ with $\alpha_1, \alpha_2, \alpha_3$ pairwise distinct, and let

$$U\Big((\alpha_i, v_i)_{i\in[3]}\Big) := (\alpha_3 - \alpha_1) \cdot (v_2 - v_1) - (\alpha_2 - \alpha_1) \cdot (v_3 - v_1).$$

$U((\alpha_i, v_i)_{i\in[3]})$ is a polynomial of degree at most $\ell + 1$ (unlike in Theorem 3.4 where it has degree $\leq \ell$). The analogue of Lemma 3.8 holds: $(\alpha_i, v_i)_{i\in[3]}$ belongs to the same parameterized line from $\mathcal{L}^1[\mathbb{F}^{\mathsf{lin}}[X] \to \mathbb{F}(X)]$ if and only if $U((\alpha_i, v_i)_{i\in[3]}) = 0$ as a polynomial. The proof goes through exactly as in Lemma 3.8 , where now operations take place in $\mathbb{F}(X)$ rather than in $\mathbb{F}$.

The analogue of Lemma 3.9 holds as well. Concretely, let $\mathbf{z} \in \mathbb{F}^m$ and assume that (α_i, v_i) is consistent with $(\mathbf{z}, \mathbf{b}_1, \mathbf{b}_2)$ for all $i \in [3]$. Then we claim that

$$\varphi_{\mathbf{z}}\Big(U\Big((\alpha_i, v_i)_{i\in[3]}\Big)\Big) = \mathbf{0}.$$

Indeed, following the proof of Lemma 3.9, let $\mathbf{b}' := d_1 \cdot \mathbf{b}_1 + d_2 \cdot \mathbf{b}_2$. By the consistency assumption, $\varphi_z(v_i) - \varphi_z(v_1) = (\varphi_z(\alpha_i) - \varphi_z(\alpha_1)) \odot \mathbf{b}'$ for both $i = 2, 3$. Now the proof follows as in Lemma 3.9, using the linearity of $\varphi_{\mathbf{z}}$ and the definition of $U((\alpha_i, v_i)_{i\in[3]})$.

Next, let $T_{\mathbf{z}}$ be defined in the exact same way as in Section 3, using triples of tuples from $\mathbb{F}^{\mathsf{lin}}[X] \times \mathcal{C}^+$ instead of $\mathbb{F} \times \mathcal{C}$, and, of course, using the affine versions of the definitions of α-pretenders and consistency. Then the same proof as that of Lemma 3.10 yields

$$|A_{\mathbf{z}}| \cdot (|A_{\mathbf{z}}| - 1) \cdot (|A_{\mathbf{z}}| - N) \leq |T_{\mathbf{z}}|.$$

It suffices to use Lemma 5.5 with the bound $|A_\theta^{\mathsf{pret}}| \leq N$ instead of the bound $|A_\theta^{\mathsf{pret}}| \leq \delta \cdot n + 1$ given by Lemma 3.7.

Finally, let $P \subset \mathbb{F}^{\mathsf{lin}}[X] \times \mathcal{C}^+$ be the set of all tuples (α, c) such that c is an α-pretender. Since $\mathcal{C}^+$ is (δ, L)-list decodable, $|P| \leq |\mathbb{F}^{\mathsf{lin}}[X]| \cdot L = |\mathbb{F}|^2 \cdot L$. Let $(\alpha_i, c_i)_{i\in[3]} \in P^3$ be such

that $U((\alpha_i, c_i)_{i\in[3]}) \neq 0$. Then since $U((\alpha_i, c_i)_{i\in[3]})$ is a polynomial of degree at most $\ell+1$ (note that in Theorem 3.4 it would be of degree at most ℓ), there are at most $(\ell+1)^m$ elements $\mathbf{z} \in \mathbb{F}^m$ such that the triple belongs to $T_{\mathbf{z}}$. Using now the same double counting arugment as in the proof of Theorem 3.4,

$$t \cdot (t-1) \cdot (t-N) \cdot |\mathcal{Z}_t| \leq \sum_{\mathbf{z}\in\mathcal{Z}_t} |T_{\mathbf{z}}| \leq \sum_{\mathbf{z}\in\mathbb{F}^m} |T_{\mathbf{z}}| \leq \left(|\mathbb{F}|^2 \cdot L\right)^3 \cdot (\ell+1)^m,$$

where $\mathcal{Z}_t$ is the set of tuples $\mathbf{z} \in \mathbb{F}^m$ such that $|A_{\mathbf{z}}| \geq t$. Again, this completes the proof since $\Pr_{\mathbf{z}\leftarrow\mathbb{F}^m}[|A_{\mathbf{z}}| \geq t|] = |\mathcal{Z}_t|/|\mathbb{F}|^m$. □

Remark 5.6 (Correlated Agreement with $\mathcal{C}^{\equiv 2}$ instead of $\mathcal{C}_\theta^{\equiv 2}$)**.** In Definition 5.1 we ask for Correlated Agreement with $\mathcal{C}_\theta^{\equiv 2}$, and not with $\mathcal{C}^{\equiv 2}$, i.e. with pairs of codewords of $\mathcal{C}$ as in Section 3. The set $\mathcal{C}_\theta^{\equiv 2}$ strictly contains $\mathcal{C}^{\equiv 2}$, and so, when a codeword $c \in \mathcal{C}^+$ agrees with the α-fold of two words u_1, u_2 on a subset of $[n]$ of size at least $(1-\delta)\cdot n$, and c is not an α-pretender, we obtain a pair from $\mathcal{C}_\theta^{\equiv 2}$ that is close to (u_1, u_2), and this pair need not belong to $\mathcal{C}^{\equiv 2}$. Here and below, we say that a pair of codewords (c_1, c_2) is *close* to a pair of words (w_1, w_2) if there is a subset of $[n]$ of size at least $(1-\delta)\cdot n - 1$ where w_1 agrees with c_1 and w_2 agrees with c_2.

In a bigger protocol, for example when aiming to build an IOR from $\mathcal{R}_{\mathcal{C}^{\equiv 2}}$ to $\mathcal{R}_{\mathcal{C}}$ using affine folds, one would instead prefer to obtain a pair from $\mathcal{C}^{\equiv 2}$ instead of $\mathcal{C}_\theta^{\equiv 2}$. This can be ensured using the methods described in [CWY+26, Section 5]. To do so, one can have an initial communication round where the verifier samples an additional challenge $\xi \in \mathbb{F}$, and replaces the words u_1 and u_2 by $u_1' := u_1$ and $u_2' := u_2 + \xi \cdot u_1$, and then proceeds to perform the DEEP sampling and the affine folding with these two words instead of u_1 and u_2. One also performs analogous linear changes on the witness and any linear constraints placed on the witness.

The reason this works is the following. Let Λ be the set of all pairs of codewords of $\mathcal{C}^+$ that are close to the original words (u_1, u_2). Note that Λ does not depend on ξ, since u_1, u_2 are fixed before ξ is sampled. On the other hand, the pairs of codewords of $\mathcal{C}^+$ that are close to the new words (u_1', u_2') are the pairs $(c_1, c_2 + \xi \cdot c_1)$ with $(c_1, c_2) \in \Lambda$.

Now let $(c_1, c_2) \in \Lambda$. If (c_1, c_2) belongs to $\mathcal{C}^{\equiv 2}$, then so does $(c_1, c_2 + \xi \cdot c_1)$, for all $\xi \in \mathbb{F}$. Otherwise we claim that $(c_1, c_2 + \xi \cdot c_1)$ belongs to $\mathcal{C}_\theta^{\equiv 2}$ for at most one $\xi \in \mathbb{F}$. Indeed, let $s_1, s_2 \in \mathbb{F}$ be the coefficients of the monomial X^ℓ of c_1 and c_2, which are not both zero. By definition of $\mathcal{C}_\theta^{\equiv 2}$, the pair $(c_1, c_2 + \xi \cdot c_1)$ belongs to $\mathcal{C}_\theta^{\equiv 2}$ if and only if $d_1 \cdot s_1 + d_2 \cdot (s_2 + \xi \cdot s_1) = 0$. Here we assume that $d_2 \neq 0$ (this is the case for the folding maps $(\theta_1, \theta_2) = (1, \alpha)$ and $(1-\alpha, \alpha)$). Then this equation has at most one solution ξ, proving the claim.

Hence there are at most $|\Lambda|$ elements $\xi \in \mathbb{F}$ for which some pair from $\mathcal{C}_\theta^{\equiv 2}$ that is close to the new words does not belong to $\mathcal{C}^{\equiv 2}$. . For the rest of $\xi \in \mathbb{F}$, every pair (c_1', c_2') from $\mathcal{C}_\theta^{\equiv 2}$ that is close to (u_1', u_2') belongs to $\mathcal{C}^{\equiv 2}$. From such a pair we obtain $(c_1', c_2' - \xi \cdot c_1') \in \mathcal{C}^{\equiv 2}$, which is close to (u_1, u_2). Note that $|\Lambda|$ is bounded by the list size of $(\mathcal{C}^+)^{\equiv 2}$ at distance $\delta + 1/n$.

6 An alternative and more general proof of Theorem 3.4 via linear algebra

In this section we present the proof of Theorem 3.4 in an alternative way where folding is an arbitrary random linear map, structured as a zero-evader.

6.1 Zero-evaders

We use the notion of a zero evader from [BCF+25], expressed as an indexed family of linear maps with vector-valued outputs and error bounds for subspaces. This allows stronger bounds when testing several vectors to be zero simultaneously as they all map to zero exactly when their span does. We bound the probability of this event for any fixed subspace in a given family.

Definition 6.1 (Zero evader)**.** A *zero evader* is a family of linear maps $\varphi_z : \mathbb{F}^n \to \mathbb{F}^m$ indexed by a finite non-empty set $\mathcal{Z}$, called the *seeds*. For a nonempty family $\mathcal{W}$ of linear subspaces of $\mathbb{F}^n$, we call $\varepsilon_{\mathcal{W}} \in [0,1]$ an *error bound* if

$$\max_{W \in \mathcal{W}} \Pr_{z \leftarrow \mathcal{Z}} [W \subseteq \ker \varphi_z] \leq \varepsilon_{\mathcal{W}}.$$

Remark 6.2. In a given basis φ_z is a $m \times n$ matrix such that $\varphi_z(\mathbf{v}) = \varphi_z \cdot \mathbf{v}$. When $m = 1$ we call it a *scalar zero evader*.

Remark 6.3. For $1 \leq d \leq n$ we write ε_d for an error bound where $\mathcal{W}$ consists of all d-dimensional subspaces. By subspace containment, these bounds may be chosen so that $\varepsilon_d \geq \varepsilon_{d+1}$ for $1 \leq d < n$, which we require.

In particular ε_1 recovers the error bound from [BCF+25]. Intuitively, ε_d bounds the probability of annihilating any fixed collection of d linearly independent vectors simultaneously.

Lemma 6.4 (Few folding vectors lie in a subspace)**.** Let $(\psi_z)_{z \in \mathcal{Z}}$ be a scalar zero evader on $\mathbb{F}^n$ with error bounds ε_d. For every j-dimensional subspace $H \subset \mathbb{F}^n$ with $0 \leq j < n$, we have $|\{z \mid \psi_z \in H\}| \leq |\mathcal{Z}|\ \varepsilon_{n-j}$.

Proof. The condition $\psi_s \in H$ is equivalent to $H^\perp \subseteq \ker \psi_s$ and $\dim H^\perp = n - j$. The result follows from the definition of ε_{n-j}. □

6.2 Few bad seeds via linear algebra

We start by fixing notation. Let $\mathbb{F}$ be a finite field of size $q = |\mathbb{F}|$. We denote with $(q)_{n+1}$ the falling factorial $q(q-1)\cdots(q-n)$. We let ℓ be the message length, n the code length, b the folding size and m the check size. Let $\mathcal{C} \subseteq \mathbb{F}^n$ be a code of dimension ℓ that is (δ, L)-list decodable and has MCA bound E.

We use two zero evaders, one for folding and one for checking. For folding we use a scalar zero evader $\psi_s \in \mathbb{F}^{1 \times b}$ with seeds $s \in \mathcal{Z}_{\mathsf{fold}}$ and error bounds $\varepsilon_d^{\mathsf{fold}}$. We abbreviate $N = |\mathcal{Z}_{\mathsf{fold}}|$.

For checking we use a zero evader $\varphi_z \in \mathbb{F}^{m\times\ell}$ with seeds $z \in \mathcal{Z}_{\mathsf{check}}$ and error bounds $\varepsilon_d^{\mathsf{check}}$. We define $\hat{\varphi}_z \in \mathbb{F}^{m\times n}$ as $\varphi_z \mathsf{Enc}_{\mathcal{C}}^{-1}$, where $\mathsf{Enc}_{\mathcal{C}}^{-1}$ is a left inverse of $\mathsf{Enc}_{\mathcal{C}}$.

Setup. Abstractly, the protocol we are interested in is as follows:

1. The prover binds, e.g. through a commitment, to b words $U \in \mathbb{F}^{b\times n}$. In the honest case these are codewords $D \in \mathcal{C}^b$.
2. The verifier challenges with $z \leftarrow \mathcal{Z}_{\mathsf{check}}$ and the prover provides $B(z) \in \mathbb{F}^{b\times m}$. In the honest case $B(z) = D\hat{\varphi}_z^\top$.
3. The verifier challenges with $s \leftarrow \mathcal{Z}_{\mathsf{fold}}$ so that the prover is now bound to $\mathbf{u}_s = \psi_s U$. In the honest case $\mathbf{u}_s = \psi_s D$ is a codeword and $\mathbf{u}_s\hat{\varphi}_z^\top = \psi_s B(z)$.
4. The prover and verifier establish, by any protocol, that there exists a codeword $\mathbf{c} \in \mathcal{C}$ and $S \subseteq [n]$ such that proximity and consistency hold

$$|S| > (1-\delta)n, \quad \mathbf{c}_S = (\mathbf{u}_s)_S, \tag{11}$$

$$\mathbf{c}\widehat{\varphi_z}^\top = \psi_s B(z). \tag{12}$$

Our goal is that every witness $(\mathbf{c}, S)$ satisfying step 4 admits codewords $D \in \mathcal{C}^b$ that satisfy $D_S = U_S$. We quantify failures of this property by bounding the number of bad folding seeds for a randomly sampled checking seed.

Ignoring the consistency checking part of the protocol we define pretenders as witness codewords that survive just the proximity check.

Definition 6.5 (Pretender)**.** We call a witness $\mathbf{c} \in \mathcal{C}$ a *pretender* if there exists an $S \subseteq [n]$ satisfying the proximity equation (11) but there exists no codewords $D \in \mathcal{C}^b$ satisfying $D_S = U_S$. Let $\mathcal{P}_s$ be the set of pretenders for s.

Without a consistency check, our only hope is to bound the number of folding seeds that can have pretenders. Such bounds are known as *mutual correlated agreement* (MCA) theorems for code and folding evader.

Definition 6.6 (MCA bound)**.** An integer $0 \le E \le N$ is an *MCA bound* if $|\{s \in \mathcal{Z}_{\mathsf{fold}} : \mathcal{P}_s \neq \varnothing\}| \le E$. The trivial MCA bound is N.

With the consistency check added, we define bad seeds as those that let pretenders slip through.

Definition 6.7 (Bad seeds)**.** A pair of seeds (z, s) is *bad* if it admits a pretender that also satisfies the consistency equation (12). For given z, define the bad set $A_z = \{s \in \mathcal{Z}_{\mathsf{fold}} : (z, s) \text{ is bad}\}$.

The main theorem is that for uniform random checking seed z, the set of bad folding seeds is bounded.

Theorem 6.8 (The bad set is likely small)**.** Let $\beta = \lceil \delta n \rceil N \varepsilon_1^{\mathsf{fold}}$ then $|A_z| \le E$ and for any $\beta < t \le E$,

$$\Pr_{z \leftarrow \mathcal{Z}_{\mathsf{check}}} [|A_z| \ge t] \le \frac{(E)_{b+1} L^{b+1} \varepsilon_1^{\mathsf{check}}}{(t-\beta) \prod_{d \in [b]} (t - N\varepsilon_d^{\mathsf{fold}})}.$$

In particular, for sumcheck folding we obtain a bound cubic in the list size.

Corollary 6.9 (Sumcheck folds)**.** Sumcheck rounds fold using a multilinear evader $\psi_\alpha = (1-\alpha, \alpha)$, so we have $b = 2$, $N = q$, $\varepsilon_1^{\mathsf{fold}} = \frac{1}{q}$ and $\varepsilon_2^{\mathsf{fold}} = 0$ and hence for $t > \lceil \delta n \rceil$,

$$\Pr_{z \leftarrow \mathcal{Z}_{\mathsf{check}}} [|A_z| \ge t] \le \frac{(E)_3 L^3 \varepsilon_1^{\mathsf{check}}}{t(t-1)(t - \lceil \delta n \rceil)}.$$

6.2.1 Proof of Theorem 6.8

We will show that b linearly independent folding vectors, together with their codewords, determine a unique explanation D. An additional codeword can contradict that explanation; any fixed such collection passes checking with probability at most $\varepsilon_1^{\mathsf{check}}$.

Lemma 6.10 (Explanation)**.** Let $\psi_{s_1}, ..., \psi_{s_b}$ form a basis and let $\mathbf{c}_1, ..., \mathbf{c}_b \in \mathcal{C}$. They determine a unique $D \in \mathcal{C}^b$ satisfying $\psi_{s_i} D = \mathbf{c}_i$. If these pairs each satisfy the checking equation (12), then $D\hat{\varphi}_z^\top = B(z)$.

Proof. Since ψ_{s_i} form a basis, the system $\psi_{s_i} D = \mathbf{c}_i$ for $i \in [b]$ has a unique solution D. Its rows are linear combinations of $\mathbf{c}_i$ so $D \in \mathcal{C}^b$. For $i \in [b]$ we have $\psi_{s_i}(D\hat{\varphi}_z^\top - B(z)) = \mathbf{0}$ and since ψ_{s_i} form a basis it follows $D\hat{\varphi}_z^\top - B(z) = \mathbf{0}$. □

Lemma 6.11 (Contradiction)**.** Fix the b pairs from Lemma 6.10 and their explanation D, independently of z. Fix an additional pair $(s, \mathbf{c})$ with $\mathbf{c} \in \mathcal{C}$ and $\mathbf{c} \ne \psi_s D$. If all $b+1$ pairs satisfy (12), then $(\mathbf{c} - \psi_s D)\hat{\varphi}_z^\top = \mathbf{0}$. The probability over $z \leftarrow \mathcal{Z}_{\mathsf{check}}$ of this event is at most $\varepsilon_1^{\mathsf{check}}$.

Proof. If all checking equations hold, the preceding lemma gives $D\hat{\varphi}_z^\top = B(z)$. Therefore

$$(\mathbf{c} - \psi_s D)\hat{\varphi}_z^\top = \mathbf{c}\hat{\varphi}_z^\top - \psi_s B(z) = \mathbf{0}.$$

The difference is a fixed nonzero codeword, so the checking evader bounds this by $\varepsilon_1^{\mathsf{check}}$. □

To turn many bad seeds into many contradictions, we need to bound how many folding seeds can produce pretenders consistent with a single explanation D.

Lemma 6.12 (Codewords have few pretenders)**.** For every $D \in \mathcal{C}^b$,

$$|\{s \in \mathcal{Z}_{\mathsf{fold}} \mid \psi_s D \in \mathcal{P}_s\}| \le \beta, \quad \beta := \lceil \delta n \rceil N \varepsilon_1^{\mathsf{fold}}.$$

Proof. Fix $D \in \mathcal{C}^b$, let $I = \{i \in [n] : U_{*i} \neq D_{*i}\}$ and choose a set $J \subseteq I$ such that $|J| = \min(\lceil \delta n \rceil, |I|)$. Every witnessing agreement set S for a pretender $\psi_s D$ intersects J. If $J = I$, disjointness would mean $D_S = U_S$, contradicting pretenderhood. Otherwise, $|J| = \lceil \delta n \rceil$, while S omits fewer than $\delta n \le \lceil \delta n \rceil$ columns.

At an intersecting column i we have $\psi_s(U_{*i} - D_{*i}) = \mathbf{0}$. Each discrepancy is fixed and nonzero. A union bound then gives $|\{s \in \mathcal{Z}_{\mathsf{fold}} \mid \psi_s D \in \mathcal{P}_s\}| \le |J|\ N\varepsilon_1^{\mathsf{fold}} \le \lceil \delta n \rceil N \varepsilon_1^{\mathsf{fold}}$. □

We now count collections that contradict their explanation while passing all checks. Each fixed collection passes rarely, but many bad seeds produce many such collections.

Definition 6.13. Let $M(z)$ count ordered collections of $b+1$ pairs $(s_i, \mathbf{c}_i)$ with distinct folding seeds s_i and pretender codewords $\mathbf{c}_i \in \mathcal{P}_{s_i}$ that all satisfy consistency check (12) such that $(\psi_{s_i})_{i \in [b]}$ is a basis and $\mathbf{c}_{b+1} \neq \psi_{s_{b+1}} D$, where D is the explanation from Lemma 6.10.

Lemma 6.14 (Upper bound)**.** $\mathop{\mathbb{E}}_{z \leftarrow \mathcal{Z}_{\mathsf{check}}} [M(z)] \le (E)_{b+1} L^{b+1} \varepsilon_1^{\mathsf{check}}$.

Proof. The lists $\mathcal{P}_s$ are fixed independently of z. By (11) and (δ, L)-list decodability of $\mathcal{C}$, each satisfies $|\mathcal{P}_s| \le L$ and is nonempty for at most E folding seeds. There are therefore at most $(E)_{b+1} L^{b+1}$ possible ordered collections with distinct seeds. Each fixed collection whose final pair contradicts the explanation of its first b pairs passes all checks with probability at most $\varepsilon_1^{\mathsf{check}}$, by Lemma 6.11. Summing these probabilities gives the bound. □

Lemma 6.15 (Lower bound)**.** Let $a = |A_z|$, if $a > \beta$ then $M(z) \ge (a - \beta) \prod_{d \in [b]} (a - N\varepsilon_d^{\mathsf{fold}})$.

Proof. For each $s \in A_z$, fix a pretender $\mathbf{c}_s$ such that $\mathbf{c}_s \hat{\varphi}_z^\top = \psi_s B(z)$. Membership in A_z guarantees such a choice.

After choosing $j < b$ independent folding vectors, their span contains at most $N\varepsilon_{b-j}^{\mathsf{fold}}$ seeds. Thus there are at least $\prod_{d \in [b]} (a - N\varepsilon_d^{\mathsf{fold}})$ ordered choices for the first b pairs.

The first b pairs determine the unique $D \in \mathcal{C}^b$ satisfying $\psi_{s_i} D = \mathbf{c}_i$. By Lemma 6.12 at most β seeds satisfy $\mathbf{c}_s = \psi_s D$, since every such seed satisfies $\psi_s D = \mathbf{c}_s \in \mathcal{P}_s$. Hence at least $a - \beta$ choices satisfy $\mathbf{c}_s \neq \psi_s D$. These choices automatically differ from the first b seeds. □

Proof of Theorem 6.8. By definition $A_z \subseteq \{s \in \mathcal{Z}_{\mathsf{fold}} : \mathcal{P}_s \neq \varnothing\}$ so $|A_z| \le E$. On the event $|A_z| \ge t$, the lower bound lemma gives

$$M(z) \ge (t - \beta) \prod_{d \in [b]} \left(t - N\varepsilon_d^{\mathsf{fold}}\right).$$

Taking expectations and applying the upper bound lemma gives

$$\Pr_{z \leftarrow \mathcal{Z}_{\mathsf{check}}} [|A_z| \ge t](t - \beta) \prod_{d \in [b]} \left(t - N\varepsilon_d^{\mathsf{fold}}\right) \le (E)_{b+1} L^{b+1} \varepsilon_1^{\mathsf{check}}.$$

The result follows from dividing by the positive left hand factors. □

Bibliography

[ABF26] G. Arnon, D. Boneh, and G. Fenzi, "Open Problems in List Decoding and Correlated Agreement," Cryptology ePrint Archive, Paper 2026/680, 2026. [Online]. Available: https://eprint.iacr.org/2026/680

[ACF+24] G. Arnon, A. Chiesa, G. Fenzi, and E. Yogev, "STIR: Reed-Solomon Proximity Testing with Fewer Queries," in *Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Proceedings*, L. Reyzin and D. Stebila, Eds., in Lecture Notes in Computer Science, vol. 14929. Springer, 2024, pp. 380–413. doi: 10.1007/978-3-031-68403-6_12.

[ACF+25] G. Arnon, A. Chiesa, G. Fenzi, and E. Yogev, "WHIR: Reed-Solomon Proximity Testing with Super-Fast Verification," in *Advances in Cryptology - EUROCRYPT 2025 - 44th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Madrid, Spain, May 4-8, 2025, Proceedings, Part IV*, S. Fehr and P.-A. Fouque, Eds., in Lecture Notes in Computer Science, vol. 15604. Springer, 2025, pp. 214–243. doi: 10.1007/978-3-031-91134-7_8.

[BBH+18] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev, "Fast Reed-Solomon Interactive Oracle Proofs of Proximity," in *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, in Leibniz International Proceedings in Informatics (LIPIcs), vol. 107. 2018, p. 14:1–14:17.

[BCF+24] M. Brehm, B. Chen, B. Fisch, N. Resch, R. D. Rothblum, and H. Zeilberger, "Blaze: Fast SNARKs from Interleaved RAA Codes," Cryptology ePrint Archive, Paper 2024/1609 (a version appears at EUROCRYPT 2025), 2024. [Online]. Available: https://eprint.iacr.org/2024/1609

[BCF+25] B. Bünz, A. Chiesa, G. Fenzi, and W. Wang, "Linear-Time Accumulation Schemes," Cryptology ePrint Archive, Paper 2025/753 (a version appears at TCC 2025), 2025. [Online]. Available: https://eprint.iacr.org/2025/753

[BCG+25] S. Bordage, A. Chiesa, Z. Guan, and I. Manzur, "All Polynomial Generators Preserve Distance with Mutual Correlated Agreement," Cryptology ePrint Archive, Paper 2025/2051 (a version appears at CCC 2026), 2025. [Online]. Available: https://eprint.iacr.org/2025/2051

[BCH+25] E. Ben-Sasson, D. Carmon, U. Haböck, S. Kopparty, and S. Saraf, "On Proximity Gaps for Reed-Solomon Codes," Cryptology ePrint Archive, Paper 2025/2055 (a version appears at STOC 2026), 2025. [Online]. Available: https://eprint.iacr.org/2025/2055

[BCI+20] E. Ben-Sasson, D. Carmon, Y. Ishai, S. Kopparty, and S. Saraf, "Proximity Gaps for Reed-Solomon Codes," in *61st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2020)*, 2020, pp. 900–909.

[BCP+26] J. Brakensiek, Y. Chen, A. Putterman, Z. Zhang, and K. Z. Zheng, "Algorithmic List Decoding of Reed-Solomon Codes up to Capacity," Electronic Colloquium on Computational Complexity, Report TR26-164, 2026. [Online]. Available: https://eccc.weizmann.ac.il/report/2026/164/

[BDD+25] S. Bagad, Q. Dao, Y. Domb, and J. Thaler, "Speeding Up Sum-Check Proving," Cryptology ePrint Archive, Paper 2025/1117, 2025. [Online]. Available: https://eprint.iacr.org/2025/1117

[BFM+26] A. Baweja, G. Fenzi, P. Mishra, and T. Mopuri, "Field-Agnostic SNARKs with Small Proofs via Encode-Repeat-Accumulate (ERA) Codes," Cryptology ePrint Archive, Paper 2026/864, 2026. [Online]. Available: https://eprint.iacr.org/2026/864

[BGK+20] E. Ben-Sasson, L. Goldberg, S. Kopparty, and S. Saraf, "DEEP-FRI: Sampling Outside the Box Improves Soundness," in *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, T. Vidick, Ed., in Leibniz International Proceedings in Informatics (LIPIcs), vol. 151. Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020, p. 5:1–5:32. doi: 10.4230/LIPIcs.ITCS.2020.5.

[CFW26] A. Chiesa, G. Fenzi, and G. Weissenberg, "Zero-Knowledge IOPPs for Constrained Interleaved Codes," Cryptology ePrint Archive, Paper 2026/391, 2026. [Online]. Available: https://eprint.iacr.org/2026/391

[CS25] E. Crites and A. Stewart, "On Reed-Solomon Proximity Gaps Conjectures," Cryptology ePrint Archive, Paper 2025/2046, 2025. [Online]. Available: https://eprint.iacr.org/2025/2046

[CWY+26] A. Chiesa, G. Weissenberg, E. Yogev, and K. Z. Zheng, "Affine Combiners for Reed-Solomon Proximity Gaps," Cryptology ePrint Archive, Paper 2026/2863, 2026. [Online]. Available: https://eprint.iacr.org/2026/2863

[DG25] B. E. Diamond and A. Gruen, "On the Distribution of the Distances of Random Words," Cryptology ePrint Archive, Paper 2025/2010 (a version appears at CRYPTO 2026), 2025. [Online]. Available: https://eprint.iacr.org/2025/2010

[DKT+26] Q. Dao, S. D. Kominers, J. Thaler, and K. Z. Zheng, "Counterexamples to Beyond-Johnson Proximity Gaps over Binary Fields," Cryptology ePrint Archive, Paper 2026/2424, 2026. [Online]. Available: https://eprint.iacr.org/2026/2424

[DKT26] Q. Dao, S. D. Kominers, and J. Thaler, "Reed-Solomon Codes Beyond Johnson: Efficient Decoding and Smaller Cryptographic Proofs," Cryptology ePrint Archive, Paper 2026/2056, 2026. [Online]. Available: https://eprint.iacr.org/2026/2056

[DP24] B. E. Diamond and J. Posen, "Polylogarithmic Proofs for Multilinears over Binary Towers," Cryptology ePrint Archive, Paper 2024/504 (a version appears at EUROCRYPT 2026), 2024. [Online]. Available: https://eprint.iacr.org/2024/504

[Eth25] Ethereum Foundation, "The Proximity Prize." [Online]. Available: https://proximityprize.org/

[FS26] M. Frigo and abhi shelat, "Polygero: Proximity Testing From Polynomial Challenges," Cryptology ePrint Archive, Paper 2026/2481, 2026. [Online]. Available: https://eprint.iacr.org/2026/2481

[Gab26] A. Gabizon, "Linear List Size Bounds for Reed-Solomon Beyond the Johnson Radius," Cryptology ePrint Archive, Paper 2026/2048, 2026. [Online]. Available: https://eprint.iacr.org/2026/2048

[GG25] R. Goyal and V. Guruswami, "Optimal Proximity Gaps for Subspace-Design Codes and (Random) Reed-Solomon Codes," Cryptology ePrint Archive, Paper 2025/2054, 2025. [Online]. Available: https://eprint.iacr.org/2025/2054

[GGR11] P. Gopalan, V. Guruswami, and P. Raghavendra, "List Decoding Tensor Products and Interleaved Codes," *SIAM Journal on Computing*, vol. 40, no. 5, pp. 1432–1462, 2011.

[GKL24] Y. Gao, H. Kan, and Y. Li, "Linear Proximity Gap for Linear Codes within the 1.5 Johnson Bound," Cryptology ePrint Archive, Paper 2024/1810, 2024. [Online]. Available: https://eprint.iacr.org/2024/1810

[GLH+25] Y. Guo, X. Liu, K. Huang, W. Qu, T. Tao, and J. Zhang, "DeepFold: Efficient Multilinear Polynomial Commitment from Reed-Solomon Code and Its Application to Zero-knowledge Proofs," in *34th USENIX Security Symposium (USENIX Security 25)*, Seattle, WA: USENIX Association, 2025, pp. 3497–3516. [Online]. Available: https://www.usenix.org/conference/usenixsecurity25/presentation/guo-yanpei

[GNR26] K. Gurkan, A. Novakovic, and R. D. Rothblum, "Bolt: Faster SNARKs from Sketched Codes," Cryptology ePrint Archive, Paper 2026/310 (a version appears at CRYPTO 2026), 2026. [Online]. Available: https://eprint.iacr.org/2026/310

[Gru24] A. Gruen, "Some Improvements for the PIOP for ZeroCheck," Cryptology ePrint Archive, Paper 2024/108, 2024. [Online]. Available: https://eprint.iacr.org/2024/108

[Hab25] U. Haböck, "A Note on Mutual Correlated Agreement for Reed-Solomon Codes," Cryptology ePrint Archive, Paper 2025/2110, 2025. [Online]. Available: https://eprint.iacr.org/2025/2110

[Jer26] F. G. Jeronimo, "Algorithmic List Decoding at Capacity and Optimal Proximity Gaps for Reed-Solomon Codes," Electronic Colloquium on Computational Complexity, Report TR26-169, 2026. [Online]. Available: https://eccc.weizmann.ac.il/report/2026/169/

[Jo26] S. Jo, "Reed-Solomon Mutual Correlated Agreement Beyond the Johnson Radius," Cryptology ePrint Archive, Paper 2026/1432, 2026. [Online]. Available: https://eprint.iacr.org/2026/1432

[Joh62] S. M. Johnson, "A New Upper Bound for Error-Correcting Codes," *IRE Transactions on Information Theory*, vol. 8, no. 3, pp. 203–207, 1962.

[KTZ26] S. D. Kominers, J. Thaler, and K. Z. Zheng, "The One-and-a-Half Johnson Bound Is Tight for Proximity Gaps of General Linear Codes," Cryptology ePrint Archive, Paper 2026/1894, 2026. [Online]. Available: https://eprint.iacr.org/2026/1894

[LLX+26] Z. Li, H. Liu, C. Xing, Y. Yao, and C. Yuan, "More Efficient SNARKs via Quasi-Abelian Codes: Faster, Smaller, and Field-Agnostic," Cryptology ePrint Archive, Paper 2026/939, 2026. [Online]. Available: https://eprint.iacr.org/2026/939

[Man26] I. Manzur, "Mutual Correlated Agreement and List-Decoding: Improvements and Separations," Cryptology ePrint Archive, Paper 2026/2617, 2026. [Online]. Available: https://eprint.iacr.org/2026/2617

[NA25] A. Novakovic and G. Angeris, "Ligerito: A Small and Concretely Fast Polynomial Commitment Scheme," Cryptology ePrint Archive, Paper 2025/1187, 2025. [Online]. Available: https://eprint.iacr.org/2025/1187

[ZCF23] H. Zeilberger, B. Chen, and B. Fisch, "BaseFold: Efficient Field-Agnostic Polynomial Commitment Schemes from Foldable Codes," Cryptology ePrint Archive, Paper 2023/1705 (a version appears at CRYPTO 2024), 2023. [Online]. Available: https://eprint.iacr.org/2023/1705

[Zei26] H. Zeilberger, "Khatam: Proximity Gaps for Multilinear Evaluation for All Linear Codes," Cryptology ePrint Archive, Paper 2024/1843 (a version appears at CRYPTO 2026), 2026. [Online]. Available: https://eprint.iacr.org/2024/1843

A Proof of completeness and round-by-round knowledge soundness of the BEEP-IOR

Proof. We prove Theorem 4.3. Let Π denote Construction 4.1. In our formalization of IORs (Section 2.4), every round starts with the prover sending a message, and then the verifier replies with a challenge. Since, as described, Rounds 1 and 3 of Construction 4.1 start off with a verifier challenge, we modify Π so that the prover sends empty messages at the beginning of those rounds.

It is clear that Π is perfectly complete. We prove soundness of Π. Let $\mathbb{x} = (\mu, \mathbf{v}) \in \mathbb{F} \times \mathbb{F}^{2\times\ell}$ and $\mathbb{y} = f \in \mathbb{F}^{2\times n}$, with $f = (f_{1*}, f_{2*})$. Let tr be a partial interaction transcript for Π.

We progressively describe a knowledge state function State for Π. For all $\mathbf{w}$, if $\mathsf{tr} = \varnothing$ is the empty transcript, then we set $\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \mathbf{w}) = 1$ if and only if $(\mathbb{x}, \mathbb{y}; \mathbf{w}) \in \mathcal{R}^{\delta}_{\mathcal{C}^{=2}}$. If tr is a transcript where the prover is about to move, then we define $\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, m, \mathbf{w}) = \mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \mathbf{w})$ for every prover message m. If $\mathsf{tr} = (m_1, \rho_1, m_2, \rho_2, m_3, \rho_3)$ is a full transcript, then we define $\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \mathbf{w}) = 1$ if and only if V on tr outputs $(\mathbb{x}', \mathbb{y}')$ such that $(\mathbb{x}', \mathbb{y}'; \mathbf{w}) \in \mathcal{R}^{\delta}_{\mathcal{C}}$.

So far State satisfies the conditions of Knowledge State function from Definition 2.11. We now complete our definition of State on the rest of partial transcripts tr for which $\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \mathbf{w})$ has not yet been defined. These are precisely the nonempty transcripts where the prover is about to move.

Round 1. Let $m_1 = \varnothing$ be an empty first prover message. For every first verifier challenge $z \in \mathcal{Z}$, we let

$$\mathsf{State}(\mathbb{x}, \mathbb{y}, m_1, z, \mathbf{w}) = 1 \iff |A_z| \geq \delta \cdot n + 2, \text{ or } (\mathbb{x}, \mathbb{y}, \mathbf{w}) \in \mathcal{R}^{\delta}_{\mathcal{C}^{\equiv 2}},$$

where, A_z is the set of elements $\alpha \in \mathbb{F}$ that are DEEP-MCA-bad with respect to $(z, \mathbf{b}_1^*, \mathbf{b}_2^*)$ and the words $f_{1*}, f_{2*} \in \mathbb{F}^n$.

We define an extractor Ext for transcripts of the form $\mathsf{tr} = (m_1, z)$ as $\mathsf{Ext}(\mathbb{x}, \mathbb{y}, m_1, z, \mathbf{w}) = \mathbf{w}$. We now bound the following probability:

$$s_1 := \Pr_{z \leftarrow \mathcal{Z}}\left[\exists \mathbf{w} \text{ such that } \begin{array}{l} \wedge\, \mathsf{State}(\mathbb{x}, \mathbb{y}, m_1, \mathsf{Ext}(\mathbb{x}, \mathbb{y}, m_1, z, \mathbf{w})) = 0 \\ \wedge\, \mathsf{State}(\mathbb{x}, \mathbb{y}, m_1, z, \mathbf{w}) = 1 \end{array}\right] \tag{13}$$

We show that (13) is at most κ_1 as defined in the statement of the present Theorem 4.3. Indeed, assume $\mathsf{State}(\mathbb{x}, \mathbb{y}, m_1, \mathbf{w}) = 0$. Then by definition we have $(\mathbb{x}, \mathbb{y}, \mathbf{w}) \notin \mathcal{R}^{\delta}_{\mathcal{C}^{\equiv 2}}$. The only way we can have $\mathsf{State}(\mathbb{x}, \mathbb{y}, m_1, z, \mathbf{w}) = 1$ then is if $|A_z| \geq \delta \cdot n + 2$. By Theorem 3.4 taking $t = \delta \cdot n + 2$, we have $\Pr_{z \leftarrow \mathcal{Z}}[|A_z| \geq \delta \cdot n + 2] \leq (M \cdot L_{\mathcal{C}})^3 \cdot \varepsilon_{\mathsf{evader}}/((\delta \cdot n + 2) \cdot (\delta \cdot n + 1)) \leq \kappa_1$, where M is as in the statement of Theorem 4.3. It follows that $s_1 \leq \kappa_1$.

Round 2. Let $m_1 = \varnothing$ be an empty first prover message; $z \in \mathcal{Z}$ a verifier first challenge; and $m_2 = (\mathbf{b}_1, \mathbf{b}_2, h(X)) \in \mathbb{F}^m \times \mathbb{F}^m \times \mathbb{F}^{<3}[X]$ a second prover message. Let $\mathsf{tr} = (m_1, z, m_2)$. For every second verifier challenge $\alpha \in \mathbb{F}$, we let

$$\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \alpha, \mathbf{w}) = 1 \iff |A_z| \geq \delta \cdot n + 2 \text{ or } \left(\begin{array}{l} \mathbf{w} \in \mathbb{F}^{\ell} \\ \wedge\, h(0) + h(1) = \mu \\ \wedge\, \langle \mathbf{w}, (1-\alpha) \cdot \mathbf{v}_{1*} + \alpha \cdot \mathbf{v}_{2*} \rangle = h(\alpha) \\ \wedge\, \varphi_z \mathbf{w} = (1-\alpha) \cdot \mathbf{b}_1 + \alpha \cdot \mathbf{b}_2 \\ \wedge\, \Delta((1-\alpha) \cdot f_{1*} + \alpha \cdot f_{2*}, \mathsf{Enc}_{\mathcal{C}}(\mathbf{w})) \leq \delta \end{array}\right)$$

We define an extractor Ext for transcripts of the form (tr, α) as follows. The extractor proceeds under the assumption that $\Delta((1-\alpha) \cdot f_{1*} + \alpha \cdot f_{2*}, \mathsf{Enc}_{\mathcal{C}}(\mathbf{w})) \leq \delta$. If that is not the case, the extractor may fail and output $\perp$. The extractor computes $\mathsf{Enc}_{\mathcal{C}}(\mathbf{w})$ and finds the maximal set $S_\alpha \subset [n]$ where $\mathsf{Enc}_{\mathcal{C}}(\mathbf{w})$ and $(1-\alpha) \cdot f_{1*} + \alpha \cdot f_{2*}$ agree. If $|S_\alpha| < (1-\delta) \cdot n$, the extractor outputs $\perp$.

Next, the extractor computes $\mathbf{w}_{1*}, \mathbf{w}_{2*} \in \mathbb{F}^{\ell}$ such that $\mathsf{Enc}_{\mathcal{C}}(\mathbf{w}_{i*})$ agrees with f_{i*} on S_α for both $i = 1, 2$, with the following procedure: let $M_{\mathcal{C}} \in \mathbb{F}^{n \times \ell}$ be an encoding matrix for $\mathcal{C}$. The extractor sets up the following system of linear equations on variables $(\mathbf{w}_{1*}, \mathbf{w}_{2*})$:

$$\begin{aligned} (M_{\mathcal{C}} \cdot \mathbf{w}_{i*})_{S_\alpha} &= (f_{i*})_{S_\alpha} \quad \text{for } \ i = 1, 2, \\ (1-\alpha) \cdot \mathbf{w}_{1*} + \alpha \cdot \mathbf{w}_{2*} &= \mathbf{w} \end{aligned} \tag{14}$$

and finds a solution $(\mathbf{w}_{1*}, \mathbf{w}_{2*})$ to it. If the system does not have a solution, the extractor outputs $\perp$, and otherwise it outputs $\mathbf{w}' := (\mathbf{w}_{1*}, \mathbf{w}_{2*}) \in \mathbb{F}^{2 \times \ell}$.

We claim that (14) has a solution if and only if f_{1*}, f_{2*} have correlated agreement with $\mathcal{C}$ over S_α, and in that case $\mathbf{w} = (1-\alpha)\cdot\mathbf{w}_{1*} + \alpha\cdot\mathbf{w}_{2*}$. Indeed, the direct implication is clear. Conversely, assume f_{1*} and f_{2*} agree with codewords $c_1, c_2 \in \mathcal{C}$ over S_α with $c_i = \mathsf{Enc}_{\mathcal{C}}(\mathbf{w}_{i*})$, $i = 1, 2$ for some $\mathbf{w}_{i*} \in \mathbb{F}^\ell$. Now $\mathbf{w} = (1-\alpha)\cdot\mathbf{w}_{1*} + \alpha\cdot\mathbf{w}_{2*}$ because $\mathsf{Enc}_{\mathcal{C}}(\mathbf{w})$ agrees with $(1-\alpha)\cdot f_{1*} + \alpha\cdot f_{2*}$ on S_α, and so $\mathsf{Enc}_{\mathcal{C}}(\mathbf{w} - ((1-\alpha)\cdot\mathbf{w}_{1*} + \alpha\cdot\mathbf{w}_{2*}))$ is zero on S_α and $|S_\alpha| \geq (1-\delta)\cdot n > (1-\Delta(\mathcal{C}))\cdot n$ by assumption, forcing $\mathbf{w} = (1-\alpha)\cdot\mathbf{w}_{1*} + \alpha\cdot\mathbf{w}_{2*}$ by definition of minimum distance of the code $\mathcal{C}$. This proves the claim.

We now bound the following probability:

$$s_2 := \Pr_{\alpha\leftarrow\mathbb{F}}\left[\exists \mathbf{w} \text{ such that } \begin{array}{l} \wedge\, \mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \mathsf{Ext}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \alpha, \mathbf{w})) = 0 \\ \wedge\, \mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \alpha, \mathbf{w}) = 1 \end{array}\right] \tag{15}$$

We show that s_2 is at most κ_2 as defined in the statement of the present Theorem 4.3.

First, if z is such that $|A_z| \geq \delta\cdot n + 2$, then $\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \mathbf{w}) = 1$ for all $\mathbf{w}$, and so there is nothing to prove since $s_2 = 0$ in that case. Hence we assume $|A_z| \leq \delta\cdot n + 1$ from now on. Second, if $h(0) + h(1) \neq \mu$, then $\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \alpha, \mathbf{w}) = 0$ for all $\alpha, \mathbf{w}$ and there is nothing to prove. Hence we also assume $h(0) + h(1) = \mu$.

Let $B_1 \subset \mathbb{F}$ be the set of DEEP-MCA-bad elements with respect to $(z, \mathbf{b}_1, \mathbf{b}_2, f_{1*}, f_{2*})$. This is precisely the set $A_{z,\mathbf{b}_1,\mathbf{b}_2}$. Its cardinality is at most $|A_z| \leq \delta\cdot n + 1$.

We further define a subset of challenges $B_2 \subset \mathbb{F}$ as follows. Let $W \subset \mathbb{F}^{2\times\ell}$ be the set of messages $\mathbf{w}'$ such that $\Delta_{\mathcal{C}^{\equiv 2}}(\mathsf{Enc}_{\mathcal{C}^{\equiv 2}}(\mathbf{w}'), f) \leq \delta$. Since $\mathcal{C}^{\equiv 2}$ is $(\delta, L_{\mathcal{C}^{\equiv 2}})$-list decodable, $|W| \leq L_{\mathcal{C}^{\equiv 2}}$. Then we let $B_2 \subset \mathbb{F}$ be the set of all $\alpha \in \mathbb{F}$ such that for some $\mathbf{w}' = (\mathbf{w}_{1*}, \mathbf{w}_{2*}) \in W$, we have

$$h_{\mathbf{w}'}(X) \neq h(X) \quad \text{and} \quad h_{\mathbf{w}'}(\alpha) = h(\alpha)$$

where $h_{\mathbf{w}'}(X)$ is the quadratic polynomial $h_{\mathbf{w}'}(X) := \langle(1-X)\cdot\mathbf{w}_{1*} + X\cdot\mathbf{w}_{2*}, (1-X)\cdot\mathbf{v}_{1*} + X\cdot\mathbf{v}_{2*}\rangle$. If $h_{\mathbf{w}'}(X) \neq h(X)$, then there are at most 2 field elements α such that $h_{\mathbf{w}'}(\alpha) = h(\alpha)$, and so $|B_2| \leq 2\cdot|W| \leq 2\cdot L_{\mathcal{C}^{\equiv 2}}$.

We show that if $\alpha \notin B_1 \cup B_2$, then $\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \alpha, \mathbf{w}) = 1$ implies that $\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \mathsf{Ext}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \alpha, \mathbf{w})) = 1$. Once this claim is proved, it follows that

$$s_2 \leq \frac{|B_1| + |B_2|}{|\mathbb{F}|} \leq \frac{\delta\cdot n + 1 + 2\cdot L_{\mathcal{C}^{\equiv 2}}}{|\mathbb{F}|} = \kappa_2$$

To prove the claim, assume $\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \alpha, \mathbf{w}) = 1$. Then $(1-\alpha)\cdot f_{1*} + \alpha\cdot f_{2*}$ and $\mathsf{Enc}_{\mathcal{C}}(\mathbf{w})$ agree on the set $S_\alpha \subset [n]$ computed by the extractor, with $|S_\alpha| \geq (1-\delta)\cdot n$. Since $\alpha \notin B_1$, α is not DEEP-MCA-bad with respect to $(z, \mathbf{b}_1, \mathbf{b}_2, f_{1*}, f_{2*})$. Now, $(\alpha, \mathsf{Enc}_{\mathcal{C}}(\mathbf{w}))$ must be consistent with $(z, \mathbf{b}_1, \mathbf{b}_2)$ because $\varphi_z(\mathsf{Enc}_{\mathcal{C}}(\mathbf{w})) = \varphi_z(\mathbf{w}) = (1-\alpha)\cdot\mathbf{b}_1 + \alpha\cdot\mathbf{b}_2$ and $\Delta((1-\alpha)\cdot f_{1*} + \alpha\cdot f_{2*}, \mathsf{Enc}_{\mathcal{C}}(\mathbf{w})) \leq \delta$ due to $\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \alpha, \mathbf{w}) = 1$. Hence, $\mathsf{Enc}_{\mathcal{C}}(\mathbf{w})$ is not an α-pretender. This means that f_{1*}, f_{2*} have correlated agreement with $\mathcal{C}$ over S_α. By our previous claim, the system of equations (14) has a solution $\mathbf{w}' = (\mathbf{w}_{1*}, \mathbf{w}_{2*}) \in \mathbb{F}^{2\times\ell}$, which is output by the extractor. Now, $\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \mathbf{w}') = 1$ if and only if $|A_z| \geq \delta\cdot n + 2$ or $(\mathbb{x}, \mathbb{y}, \mathbf{w}') \in \mathcal{R}^{\delta}_{\mathcal{C}^{\equiv 2}}$. We assumed the former does not occur, and so we now have to show that $(\mathbb{x}, \mathbb{y}, \mathbf{w}') \in \mathcal{R}^{\delta}_{\mathcal{C}^{\equiv 2}}$, i.e. we must prove that $\langle\mathbf{w}', \mathbf{v}\rangle = \mu$ and $\Delta_{\mathcal{C}^{\equiv 2}}(f, \mathsf{Enc}_{\mathcal{C}^{\equiv 2}}(\mathbf{w}')) \leq \delta$. The second holds because f_{1*}, f_{2*} agree with $\mathsf{Enc}_{\mathcal{C}}(\mathbf{w}_{1*}), \mathsf{Enc}_{\mathcal{C}}(\mathbf{w}_{2*})$ on S_α. We now prove the

first. We know that $\langle \mathbf{w}, (1-\alpha)\cdot \mathbf{v}_{1*} + \alpha \cdot \mathbf{v}_{2*}\rangle = h(\alpha)$ and that $\mathbf{w} = (1-\alpha)\cdot \mathbf{w}_{1*} + \alpha \cdot \mathbf{w}_{2*}$, and so $h_{\mathbf{w}'}(\alpha) = h(\alpha)$. Since $\mathbf{w}' \in W$ and $\alpha \notin B_2$, we must have that $h_{\mathbf{w}'}(X) = h(X)$. Since $h(0) + h(1) = \mu$, we obtain

$$\mu = h_{\mathbf{w}'}(0) + h_{\mathbf{w}'}(1) = \langle \mathbf{w}_{1*}, \mathbf{v}_{1*}\rangle + \langle \mathbf{w}_{2*}, \mathbf{v}_{2*}\rangle = \langle \mathbf{w}', \mathbf{v}\rangle,$$

as needed. This completes the proof of our claim.

Round 3. Let m_1, m_2 be a first and second message; m_3 an empty third prover message; and $z \in \mathcal{Z}$ and $\alpha \in \mathbb{F}$ verifier first and second challenges. Let $\mathsf{tr} = (m_1, z, m_2, \alpha, m_3)$. By definition, for every third verifier challenge $\boldsymbol{\gamma} \in \mathbb{F}^{1+m}$, we have

$$\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \boldsymbol{\gamma}, \mathbf{w}) = 1 \iff (\mathbb{x}', \mathbb{y}', \mathbf{w}) \in \mathcal{R}_{\mathcal{C}}^{\delta} \text{ and } \mathsf{V}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \boldsymbol{\gamma}) = 1$$

where

$$\mathbb{x}' := (\mu', \mathbf{v}'), \quad \mathbb{y}' := g = (1-\alpha)\cdot f_{1*} + \alpha \cdot f_{2*}$$

$$\mathbf{v}' := \boldsymbol{\gamma} \cdot \begin{pmatrix} \mathbf{v}_\alpha \\ \varphi_z \end{pmatrix} \in \mathbb{F}^\ell, \quad \mu' := \boldsymbol{\gamma} \cdot \begin{pmatrix} h(\alpha) \\ (1-\alpha)\cdot \mathbf{b}_1 + \alpha \cdot \mathbf{b}_2 \end{pmatrix} \in \mathbb{F}.$$

We define an extractor Ext for transcripts of the form $(\mathsf{tr}, \boldsymbol{\gamma})$ as $\mathsf{Ext}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \boldsymbol{\gamma}, \mathbf{w}) = \mathbf{w}$. We now bound the following probability:

$$s_3 := \Pr_{\boldsymbol{\gamma} \leftarrow \mathbb{F}^{1+m}}\left[\exists \mathbf{w} \text{ such that } \begin{array}{l} \wedge\, \mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \mathsf{Ext}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \boldsymbol{\gamma}, \mathbf{w})) = 0 \\ \wedge\, \mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \boldsymbol{\gamma}, \mathbf{w}) = 1 \end{array}\right] \tag{16}$$

We show that $s_3 \le \kappa_3 = L_{\mathcal{C}}/\,|\mathbb{F}|$. Indeed, assume $\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \mathbf{w}) = 0$ and $\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \boldsymbol{\gamma}, \mathbf{w}) = 1$ for some $\mathbf{w}$. Since $\mathsf{V}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \boldsymbol{\gamma}) = 1$, we have $h(0) + h(1) = \mu$. Moreover since $(\mathbb{x}', \mathbb{y}', \mathbf{w}) \in \mathcal{R}_{\mathcal{C}}^{\delta}$, we have $\mathbf{w} \in \mathbb{F}^\ell$ and $\Delta(g, \mathsf{Enc}_{\mathcal{C}}(\mathbf{w})) = \Delta((1-\alpha)\cdot f_{1*} + \alpha \cdot f_{2*}, \mathsf{Enc}_{\mathcal{C}}(\mathbf{w})) \le \delta$. Hence, to have $\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}, \mathbf{w}) = 0$ either $\varphi_z \mathbf{w} \ne (1-\alpha)\cdot \mathbf{b}_1 + \alpha \cdot \mathbf{b}_2$, or $\langle \mathbf{w}, \mathbf{v}_\alpha\rangle \ne h(\alpha)$. On the other hand, since $(\mathbb{x}', \mathbb{y}', \mathbf{w}) \in \mathcal{R}_{\mathcal{C}}^{\delta}$, we have $\langle \mathbf{w}, \mathbf{v}'\rangle = \mu'$, i.e.

$$\boldsymbol{\gamma} \cdot (\mathbf{v}_\alpha, \varphi_z)^{\mathsf{T}} \cdot \mathbf{w} = \boldsymbol{\gamma} \cdot (h(\alpha), (1-\alpha)\cdot \mathbf{b}_1 + \alpha \cdot \mathbf{b}_2)^{\mathsf{T}}.$$

By Schwartz-Zippel lemma the probability over $\boldsymbol{\gamma}$ that the above equality holds, for fixed $\mathbf{w}$ for which one of these claims fails, is $\le \frac{1}{|\mathbb{F}|}$. Finally, there are at most $L_{\mathcal{C}}$ tuples $\mathbf{w} \in \mathbb{F}^\ell$ such that $(\mathbb{x}', \mathbb{y}', \mathbf{w}) \in \mathcal{R}_{\mathcal{C}}^{\delta}$, and so, by a union bound, $s_3 \le L_{\mathcal{C}}/\,|\mathbb{F}|$. □