A Simplified Round-by-round Soundness Proof of FRI

Albert Garreta* Nicolas Mohnblatt† Benedikt Wagner‡

August 25, 2026

Abstract

The FRI protocol (ICALP '18) is one of the most influential and widely deployed building blocks at the core of modern SNARK systems. While its concrete security is well understood, existing security proofs are intricate and technically complex.

In this work, we present a novel security analysis of FRI, in particular its round-by-round soundness. This analysis yields additional insights about FRI that previous analysis did not. Equally important, our approach is significantly simpler than previous proofs. This makes it more accessible to a broader audience, lowering the barrier to understanding this fundamental protocol. Furthermore, the relative simplicity of our analysis may pave the way for future formal verification efforts of modern SNARK constructions.

This is the full version of a paper appearing at IACR TCC 2026.

Contents

1 Introduction **2**

2 Preliminaries **3**
2.1 Reed-Solomon Codes . . . 3
2.2 Mutual correlated agreement for proximity generators . . . 3

3 IOPPs and Round-by-round Soundness **5**
3.1 Interactive oracle proof of proximity . . . 5
3.2 Round-by-round soundness . . . 6

4 The FRI IOPP **8**
4.1 Folding univariate functions . . . 8
4.2 Protocol description . . . 8

5 Our Proof **10**
5.1 Intuition and overview . . . 10
5.2 Folding preserves agreement sets . . . 14
5.3 Full proof . . . 16
5.4 Additional properties . . . 20
5.5 Comparison to existing proof strategies . . . 22

A Appendix: Batched FRI **27**

*Nethermind Research, albert@nethermind.io
†zkSecurity, nico@zksecurity.xyz
‡Ethereum Foundation, benedikt.wagner@ethereum.org

1 Introduction

The FRI protocol [BBHR18] is a foundational component of modern cryptographic argument systems, providing an efficient method for proving proximity to a Reed–Solomon (RS) code. It lies at the core of an entire industry of succinct hash-based arguments in production. To highlight this importance, note that this industry will play a critical role in securing billions of dollars on Ethereum in the future[1]. Given its central importance, it is crucial to have a precise and thorough understanding of FRI's security guarantees.

A Novel Analysis of FRI. While several analyses of FRI exist [BKS18, BCI+20, Sta21, BGK+23, ABN23, AFR23, EA23] and its concrete security is well understood, the existing proofs are technically complex and can be challenging to follow. In this work, we present a *novel, simpler and more accessible proof* of round-by-round soundness for FRI, which is the notion needed to deduce security of the non-interactive protocol that is built from it [BCS16, CCH+19, Hol19]. This new analysis offers three main benefits:

- *Accessibility for new researchers*: By lowering the technical entry barrier, we aim to make this area more approachable, encouraging new contributions and fresh ideas that could shape the future of the field.
- *A foundation for formal verification:* Our simplified analysis can serve as a template for formal verification efforts, which are increasingly important in the context of production SNARK systems.
- *Additional insights into FRI:* Our novel proof also shows additional structural properties of FRI, which may be of independent interest. For instance, they may find application when applying FRI in the context of data availability sampling [HASW24] or knowledge soundness.

The third property shows that our proof is not just a simplification of existing proofs, but yields fundamentally new insights. We discuss these in Section 5.4 and compare our novel analysis to previous approaches in Section 5.5.

Application in Formal Verification. One of the ambitious goals of Ethereum's post-quantum roadmap is to achieve formal verification of every component in the stack [Eth25, DHT+26]. We believe that our work contributes to this vision by making the security analysis of a key protocol (namely, FRI) more tractable. Following the publication of a preprint of this document, Hirai [Hir26] released a machine-assisted Lean4 formalization of our proof.

Mutual correlated agreement. We note that our proof relies on the *mutual correlated agreement (MCA) property* [ACY23, ACFY25], whereas standard analyses of FRI can be carried out under the correlated agreement (CA) property. When we initially published this article, MCA was a plausible conjecture believed by most domain experts, but stronger than CA. In subsequent work [Zei24, BCH+25, BCGM25, Hab25], the community made significant progress in understanding MCA, and it is now known to hold in the same regimes as correlated agreement. Interestingly, proofs of MCA use the same tools and proving strategies as those of CA, specifically a notion later coined as *line-decodability* [GG25]. We refer readers to the survey of Arnon, Boneh and Fenzi [ABF26] for further comparison between MCA and related properties.

[1] See for instance https://ethproofs.org/.

2 Preliminaries

Most of the notation and preliminaries in this section are lifted almost verbatim from STIR and WHIR [ACFY24, ACFY25]. As a quick reference, we use the following notation:

- The notation $x \leftarrow X$ means that x is sampled uniformly at random from a finite set X.
- The "hat" symbol over a function (*e.g.*, $\widehat{p}$) indicates that it is a polynomial.
- Given a polynomial $\widehat{p}$, we write $\mathsf{deg}_X(\widehat{p})$ to denote the degree of $\widehat{p}$ in the variable X. We omit the subscript when $\widehat{p}$ is a univariate polynomial.
- For two functions $f, g \colon \mathcal{L} \to \Sigma$, we denote $\Delta(f, g)$ the fractional Hamming distance between f and g, *i.e.*, the fraction of points in which they disagree.
- For a set $\mathcal{S} \subseteq \Sigma^n$, $\Delta(f, \mathcal{S}) := \min_{h \in \mathcal{S}} \Delta(f, h)$.
- For a function $f \colon \mathcal{L} \to \Sigma$ and a set $S \subseteq \mathcal{L}$, we write $f[S]$ to denote the restriction of f to the set S. In particular, for two such functions f, g, we have

$$f[S] = g[S] \iff \forall s \in S : \ f(s) = g(s).$$

- For a ternary relation $\mathcal{R} = \{(\mathtt{x}, \mathtt{y}, \mathtt{w})\}$, let $L(\mathcal{R}) := \{(\mathtt{x}, \mathtt{y}) \mid \exists \mathtt{w}, (\mathtt{x}, \mathtt{y}, \mathtt{w}) \in \mathcal{R}\}$ be the language induced by $\mathcal{R}$.
- For a set $\mathcal{L} \subseteq \mathbb{F}$ and $m \in \mathbb{N}$, we define $\mathcal{L}^m := \{x^m : x \in \mathcal{L}\}$.
- We assume that all sets in this work admit some canonical ordering of their elements.

2.1 Reed-Solomon Codes

FRI allows to prove proximity to a Reed-Solomon code. We recall Reed-Solomon codes here. In general, a *linear code* of length n over a field $\mathbb{F}$ is a linear subspace $\mathcal{C} \subseteq \mathbb{F}^n$. In the context of FRI and Reed-Solomon codes, it is more convenient to interpret a codeword $f \in \mathcal{C}$ as the evaluations of a univariate polynomial $\widehat{f} \in \mathbb{F}^{<d}[X]$ over a fixed domain of size n. We write codewords $f \in \mathcal{C}$ as functions $f : \mathcal{L} \to \mathbb{F}$, where we assume that $\mathcal{L}$ has some implicit canonical ordering.

Definition 2.1 (Reed-Solomon code)**.** The *Reed-Solomon code* over field $\mathbb{F}$, evaluation domain $\mathcal{L} \subseteq \mathbb{F}$, and degree $d \in \mathbb{N}$ is the set of evaluations over $\mathcal{L}$ of univariate polynomials (over $\mathbb{F}$) of degree less than d:

$$\mathsf{RS}[\mathbb{F}, \mathcal{L}, d] := \left\{ f \colon \mathcal{L} \to \mathbb{F} \quad : \quad \exists \widehat{f} \in \mathbb{F}^{<d}[X], \ \forall x \in \mathcal{L}, \ f(x) = \widehat{f}(x) \right\}.$$

The *rate* of $\mathsf{RS}[\mathbb{F}, \mathcal{L}, d]$ is $\rho := d/|\mathcal{L}|$. Its *dimension* is d. The *minimum distance* is $1 - \rho$.

2.2 Mutual correlated agreement for proximity generators

Intuitively, mutual correlated agreement states that if the linear combination $\sum_j r_j f_j$ for some functions f_j and random coefficients r_j agrees with a codeword on a sufficiently large domain S, then each of the f_j agrees with a codeword on *the very same* domain S – all of that with high probability over the r_j. The definition below assumes that r_j are sampled by an algorithm Gen, which allows to model different distributions of r_j's. We note that mutual correlated agreement can be defined for general codes, but we restrict ourselves to Reed-Solomon codes as this is sufficient for this work.

Definition 2.2 (Proximity generator, mutual correlated agreement)**.** Let $\mathcal{C} := \mathsf{RS}[\mathbb{F}, \mathcal{L}, d]$ be a Reed-Solomon code with rate ρ. We say that Gen is a proximity generator for $\mathcal{C}$ with mutual

correlated agreement with error $\varepsilon^{\mathsf{mca}}$ if for every $f_1, \ldots, f_m \colon \mathcal{L} \to \mathbb{F}$ and every $\gamma \in [0,1]$ the following holds:

$$\Pr_{(r_1,\ldots,r_m)\leftarrow\mathsf{Gen}(m)}\left[\begin{array}{l} |S| \geq (1-\gamma)\cdot n \\ \exists S \subseteq \mathcal{L}\ s.t.\ \wedge\ \exists u \in \mathcal{C}, u[S] = \sum_{j\in[m]} r_j \cdot f_j[S] \\ \qquad\qquad \wedge\ \exists i \in [m], \forall u' \in \mathcal{C}, u'[S] \neq f_i[S] \end{array}\right] \leq \varepsilon^{\mathsf{mca}}(\mathcal{C}, m, \gamma).$$

Definition 2.3 (Powers proximity generator)**.** Let $m \in \mathbb{N}$ be a length parameter and $r \in \mathbb{F}$ be a field element. We define the Powers proximity generator as the following randomized algorithm, taking a uniform $r \in \mathbb{F}$ as input:

$$\mathsf{Powers}(m; r) := (1, r, \ldots, r^{m-1}) \in \mathbb{F}^m.$$

Lemma 2.4 (from [BCH+25])**.** *Let $\mathcal{C} := \mathsf{RS}[\mathbb{F}, \mathcal{L}, d]$ be a Reed-Solomon code with rate ρ. The function* Powers *with length m defined in Definition 2.3 is a proximity generator with mutual correlated agreement with error $\varepsilon^{\mathsf{mca}}$ such that for every $\gamma \in [0,1]$:*

$$\varepsilon^{\mathsf{mca}}(\mathcal{C}, m, \gamma) := \begin{cases} \frac{(m-1)}{|\mathbb{F}|} \cdot \left(\frac{2(\eta+1/2)^5 + 3(\eta+1/2)\cdot\gamma\cdot\rho}{3\rho^{3/2}} \cdot |\mathcal{L}| + \frac{\eta+1/2}{\sqrt{\rho}} \right), & \text{if } \gamma < 1 - \sqrt{\rho}, \\ 1, & \text{otherwise,} \end{cases}$$

where $\eta := \max\left(\left\lceil \frac{\sqrt{\rho}}{1-\sqrt{\rho}-\gamma} \right\rceil, 3 \right)$.

3 IOPPs and Round-by-round Soundness

Before we move our attention to FRI and its new analysis, we recall the notion of IOPPs and round-by-round soundness.

3.1 Interactive oracle proof of proximity

In an *interactive oracle proof* for a relation $\mathcal{R}$, we consider a prover and a verifier, where the prover wants to convince the verifier of the validity of some statement. To this end, the prover can provide oracles to the verifier and the verifier can respond with challenges. The verifier can additionally query the oracles provided by the prover. For convenience, we split the statement into $\mathtt{x}$, to which the verifier has full access, and into an oracle part $\mathtt{y}$, to which the verifier has oracle access. The prover also knows some witness $\mathtt{w}$. Completeness states that the honest prover convinces the verifier if $(\mathtt{x}, \mathtt{y}, \mathtt{w}) \in \mathcal{R}$. Soundness instead bounds the probability of the prover succeeding if $(\mathtt{x}, \mathtt{y})$ is a no-instance. For an *interactive oracle proof of proximity* (IOPP), it is convenient to assume that $\mathtt{x}$ describes the parameters of a code, and $\mathtt{y}$ describes an alleged codeword, and the soundness bound should depends only on the (Hamming) distance of the provided $\mathtt{y}$ to the code specified by $\mathtt{x}$. For this work, it is sufficient to consider *public-coin* IOPPs, where the verifier's messages are all uniformly sampled from prescribed sets (that are statistically independent from the execution of the protocol). In this case, we denote $\mathbf{V}^{\mathtt{y},\pi_1,\ldots,\pi_{\mathsf{k}}}(\mathtt{x}, \alpha_1, \ldots, \alpha_{\mathsf{k}+1})$ to be the deterministic function that has oracle access to $\mathtt{y}$ and the prover messages, takes as input the verifier messages and outputs the verifier's decision bit. We now give the precise definition.

Definition 3.1 (Interactive oracle proof of proximity)**.** We say that $\mathsf{IOPP} = (\mathbf{P}, \mathbf{V})$ is a public-coin IOP of proximity for a relation $\mathcal{R}$ with k rounds, perfect completeness and soundness β if the following holds:

- **Perfect completeness.** For every $(\mathtt{x}, \mathtt{y}, \mathtt{w}) \in \mathcal{R}$,

$$\Pr_{\alpha_1,\ldots,\alpha_{\mathsf{k}}}\left[\begin{array}{l|r} \mathbf{V}^{\mathtt{y},\pi_1,\ldots,\pi_{\mathsf{k}}}(\mathtt{x}, \alpha_1, \ldots, \alpha_{\mathsf{k}}) = 1 & \pi_1 \leftarrow \mathbf{P}(\mathtt{x}, \mathtt{y}, \mathtt{w}, \alpha_1) \\ & \vdots \\ & \pi_{\mathsf{k}} \leftarrow \mathbf{P}(\mathtt{x}, \mathtt{y}, \mathtt{w}, \alpha_1, \ldots, \alpha_{\mathsf{k}}) \end{array}\right] = 1.$$

- **Soundness.** For every $(\mathtt{x}, \mathtt{y}) \notin L(\mathcal{R})$ and unbounded malicious prover $\tilde{\mathbf{P}}$,

$$\Pr_{\alpha_1,\ldots,\alpha_{\mathsf{k}}}\left[\begin{array}{l|r} \mathbf{V}^{\mathtt{y},\pi_1,\ldots,\pi_{\mathsf{k}}}(\mathtt{x}, \alpha_1, \ldots, \alpha_{\mathsf{k}}) = 1 & \pi_1 \leftarrow \tilde{\mathbf{P}}(\alpha_1) \\ & \vdots \\ & \pi_{\mathsf{k}} \leftarrow \tilde{\mathbf{P}}(\alpha_1, \ldots, \alpha_{\mathsf{k}}) \end{array}\right] \leq \beta(\mathtt{x}, \mathtt{y}).$$

  When the soundness error depends only on the lengths of the inputs and on the distance δ of $\mathtt{y}$ from the language $L_{\mathtt{x}} := \{\mathtt{y}' : \exists \mathtt{w}, (\mathtt{x}, \mathtt{y}', \mathtt{w}) \in \mathcal{R}\}$ (for an adequately defined notion of distance), we write $\beta(|\mathtt{x}|, |\mathtt{y}|, \delta)$; and sometimes leave out $|\mathtt{x}|$ and $|\mathtt{y}|$, writing $\beta(\delta)$, when the lengths are clear from context.

Remark 3.2 (Verifier moves last)**.** The above definition assumes that the prover moves last and sends the final message π_{k}. In most IOPPs for codes (*e.g.* FRI [BBHR18], BaseFold [ZCF24], STIR [ACFY24], WHIR [ACFY25]), it is the verifier that moves last. In such cases, we set $\pi_{\mathsf{k}} = \perp$ and omit it from our notation.

Remark 3.3 (IOPP for Reed-Solomon codes)**.** Let $\mathcal{C}$ be a Reed-Solomon code. We say that $(\mathbf{P}, \mathbf{V})$ is an *IOPP for* $\mathcal{C}$ if it is an IOPP for the ternary relation $\mathcal{R}_{\mathcal{C}} = \{(\mathcal{C}, f, \perp) \mid f \in \mathcal{C}\}$. When it is clear from context, we omit $\mathcal{C}$ from our notation, writing for example $\mathsf{State}(f, \emptyset)$ instead of $\mathsf{State}(\mathcal{C}, f, \emptyset)$ or $\mathbf{V}^{f,\pi_1,\ldots,\pi_{\mathsf{k}}}(\alpha_1, \ldots, \alpha_{\mathsf{k}})$ instead of $\mathbf{V}^{\mathbb{y},\pi_1,\ldots,\pi_{\mathsf{k}}}(\mathbb{x}, \alpha_1, \ldots, \alpha_{\mathsf{k}})$.

3.2 Round-by-round soundness

We recall the notion of round-by-round soundness [CCH+19]. Before we can define it, we first define what a state function is. Intuitively, it associates every (partial) transcript with a state that is either 0 or 1. Intuitively, a 0-state indicates that the prover is about to lose, *i.e.*, the verifier is about to reject. A malicious prover should start with state 0, and if in the end of the interaction the state is still 0, then the verifier rejects. On the other hand, a 1-state indicates one of two things: either the protocol execution started from an honest input $(\mathbb{x}, \mathbb{y}) \in \mathcal{R}$, or the execution started from a rejectable input $(\mathbb{x}, \mathbb{y}) \notin \mathcal{R}$ but the prover got lucky in one of the rounds.

Definition 3.4 (State function)**.** Let $(\mathbf{P}, \mathbf{V})$ be an IOPP for a relation $\mathcal{R} = \{(\mathbb{x}, \mathbb{y}, \mathbb{w})\}$. A state function for $(\mathbf{P}, \mathbf{V})$ is a (possibly inefficient) function State that receives as inputs $\mathbb{x}$, $\mathbb{y}$, and a transcript tr and outputs a bit, and has the following properties:

- Empty transcript: if $\mathsf{tr} = \emptyset$ is the empty transcript, then $\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}) = 1$ if and only if $(\mathbb{x}, \mathbb{y}) \in L(\mathcal{R})$.
- Prover moves: if tr is a transcript where the prover is about to move, and $\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}) = 0$, then for every prover message π, $\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}||\pi) = 0$.
- Full transcript: if tr is a full transcript and $\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}) = 0$, then $\mathbf{V}$ rejects given this interaction transcript.
- Default: whenever tr is any transcript and $\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}) = 1$, then every extension tr' of tr satisfies $\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}') = 1$.

With this definition at hand, we now think about a malicious prover that wants to convince the verifier. It starts in state 0 but must manage to end up in state 1, as otherwise the verifier will reject. Thus, in at least one of the rounds, this malicious prover must be lucky in the sense that the state changes from 0 to 1. Round-by-round soundness asserts that changing the state from 0 to 1 happens only with small probability. A proof of proximity relaxes this requirement and allows the probability of changing state to be large when the instance is close to the language, as long as this probability decreases and becomes small when the instance is far from the language.

Definition 3.5 (Round-by-round soundness)**.** A k-round IOPP $(\mathbf{P}, \mathbf{V})$ for a relation $\mathcal{R} = \{(\mathbb{x}, \mathbb{y}, \mathbb{w})\}$ has *round-by-round soundness* with error functions $(\varepsilon_1, \ldots, \varepsilon_{\mathsf{k}})$ if the IOPP has a state function State, such that for every $\mathbb{x}$, $\mathbb{y}$ and transcript $\mathsf{tr} = (\alpha_1, \pi_2, \ldots, \alpha_{i-1}, \pi_i)$,

$$\Pr_{\alpha_i}\left[\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}) = 0 \wedge \mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}||\alpha_i) = 1\right] \leq \varepsilon_i(\mathbb{x}, \mathbb{y}).$$

As with standard soundness, we write ε_i as a function of distance when appropriate.

Clearly, by the above reasoning, round-by-round soundness implies standard soundness by taking a union bound over all errors in the individual rounds. In practice, however, we are interested in the non-interactive BCS-compiled [BCS16] protocol. The soundness of this non-interactive protocol relates directly to round-by-round soundness and not to soundness. To get

an intuition for that, note that a malicious prover in the non-interactive setting can try to query the random oracle that outputs the verifier messages multiple times per round. For instance, it may try a few queries to find a good α_1, then proceed to the next round and so on. With soundness, we cannot directly bound the success probability of such a non-interactive prover, as the interactive prover in the soundness experiment only gets one challenge per round. On the other hand, round-by-round soundness can be used, and we briefly and informally sketch the intuition here:

Assume the malicious prover starts with a non-codeword that is far from the code. Assume that it makes Q_i queries to the random oracle for each round i. Then, with every query it can change the state from 0 to 1 with probability at most ε_i, so the probability that it gets lucky at some point is at most

$$Q_1\varepsilon_1 + \cdots + Q_{\mathsf{k}}\varepsilon_{\mathsf{k}} \leq (Q_1 + \cdots + Q_{\mathsf{k}}) \max_i \varepsilon_i.$$

Note that this prover runs in time at least $Q_1 + \cdots + Q_{\mathsf{k}}$. Therefore, the security level of such a scheme is about[2] $\lfloor -\log \max_i \varepsilon_i \rfloor$.

Remark 3.6. Our definition of a state function is slightly different from other texts in that we require the *default* property. By including this property, we are effectively considering that if the malicious prover gets lucky in any single round, then it has won the round-by-round soundness experiment; regardless of whether the verifier eventually accepts or rejects. We use this stricter definition of the round-by-round soundness experiment as it allows for a simpler analysis. In particular, given inputs $\mathbb{x}, \mathbb{y}$ and a transcript tr such that $\mathsf{State}(\mathbb{x}, \mathbb{y}, \mathsf{tr}) = 0$, the default property allows us to deduce that all prefixes of tr also have a 0-state.

[2]FRI is often used in combination with other building blocks that have their own soundness errors, and the BCS transform itself also introduces additional error terms, e.g., related to the collision-resistance of the hash function in use. So this is only a sketch of how the round-by-round soundness errors relate to the security level.

4 The FRI IOPP

In this section, we recall the FRI IOPP [BBHR18], before we give our new analysis in the next section. We consider the setting in which a prover holds an initial function $f_0 \colon \mathcal{L}_0 \to \mathbb{F}$, to which the verifier has oracle access. Informally, FRI allows the verifier to test that this initial function is close to some initial Reed-Solomon code $\mathcal{C}_0$, of dimension d_0 and rate $\rho = d_0/|\mathcal{L}_0|$. We assume some basic familiarity with the FRI protocol and its associated tools. For a good introduction, we refer to the lectures of Dan Boneh available online [Whi25a, Whi25b].

4.1 Folding univariate functions

We first recall some notation for folding univariate functions following notation of STIR and WHIR [ACFY24, ACFY25]. Folding is one of the key tools in the FRI protocol as it allows to reduce the size of a proximity testing problem by a factor m. Definition 4.2 below is given for an arbitrary folding factor m. Readers may be more familiar with the case $m = 2$ where $\widehat{p}_{f,z}$ is computed from the odd and even coefficients of $\widehat{f}$ (in the honest case).

Definition 4.1 (Block)**.** Let $m \in \mathbb{N}$ be an integer and $\mathcal{L} \subseteq \mathbb{F}$ be a multiplicative coset of $\mathbb{F}^*$ such that $|\mathcal{L}|$ is divisible by m. For each $y \in \mathcal{L}^m$, we define the set

$$\mathsf{Block}(\mathcal{L}, m, y) := \{x \in \mathcal{L} \ : \ x^m = y\} \subseteq \mathcal{L}.$$

When $\mathcal{L}$ and m are clear from context, we write $\mathsf{Block}(y)$.

Definition 4.2 (Fold function)**.** Let $m \in \mathbb{N}$ be an integer and $\mathcal{L} \subseteq \mathbb{F}$ be a multiplicative coset of $\mathbb{F}^*$ such that $|\mathcal{L}|$ is divisible by m. Let $f \colon \mathcal{L} \to \mathbb{F}$ be a function and $\alpha \in \mathbb{F}$. For each $y \in \mathcal{L}^m$, define $\widehat{p}_{f,y}$ to be the unique polynomial of degree less than m with $\widehat{p}_{f,y}(x) = f(x)$ for each $x \in \mathsf{Block}(y)$. Then, we define the function $\mathsf{Fold}(f, m, \alpha) \colon \mathcal{L}^m \to \mathbb{F}$ as

$$\forall y \in \mathcal{L}^m, \ \mathsf{Fold}(f, m, \alpha)(y) := \widehat{p}_{f,y}(\alpha).$$

By definition, $\mathsf{Fold}(f, m, \alpha)(y)$ can be computed by interpolation from the evaluations of f at all points in $\mathsf{Block}(y)$.

Remark 4.3. It is well-known (cf. [BS05, ACFY24, ACFY25]) that if the function f is in $\mathsf{RS}[\mathbb{F}, \mathcal{L}, d]$, then $\mathsf{Fold}(f, m, \alpha)$ is in $\mathsf{RS}[\mathbb{F}, \mathcal{L}^m, d/m]$, for every $\alpha \in \mathbb{F}$. This is important for the completeness of FRI.

4.2 Protocol description

Next, we restate the formal protocol description. Correctness follows by inspection.

Construction 4.4 (FRI protocol [BBHR18])**.** *Consider the following:*

- *Initial setting:*
  - *a field $\mathbb{F}$;*
  - *an evaluation domain $\mathcal{L}_0 \subseteq \mathbb{F}$ that is a multiplicative coset of $\mathbb{F}^*$;*
  - *an initial degree bound $d_0 \in \mathbb{N}$ such that $d_0 \leq |\mathcal{L}_0|/2$;*
  - *an initial code $\mathcal{C}_0 := \mathsf{RS}[\mathbb{F}, \mathcal{L}_0, d_0]$ with rate $\rho := d_0/|\mathcal{L}_0|$;*
- *Protocol parameters:*
  - *a number of folding rounds $k \in \mathbb{N}$ and a total number of rounds $\mathsf{k} := k + 1$;*
  - *a sequence of folding factors $m_1, \ldots, m_k \in \mathbb{N}$ such that the order of $\mathcal{L}_0$ is divisible by $\prod_{i=1}^{k} m_i$;*

– *a query repetition parameter* $t \in \mathbb{N}$.

For convenience, we define for every $i \in [k]$*:*

- *the* i*-th degree bound* $d_i := d_{i-1}/m_i$*;*
- *the* i*-th evaluation domain* $\mathcal{L}_i := \mathcal{L}_{i-1}^{m_i} = \{x^{m_i} : x \in \mathcal{L}_{i-1}\}$ *of order* $|\mathcal{L}_i| = |\mathcal{L}_{i-1}|/m_i$*;*
- *the* i*-th Reed-Solomon code* $\mathcal{C}_i := \mathsf{RS}[\mathbb{F}, \mathcal{L}_i, d_i]$.

Note that by definition, all codes $\mathcal{C}_0, \ldots, \mathcal{C}_k$ *have the same rate* ρ *and the order of the* $(i-1)$*-th domain* $|\mathcal{L}_{i-1}|$ *is divisible by* $\prod_{j=i}^{k} m_j$*. In particular,* $|\mathcal{L}_{i-1}|$ *is divisible by* m_i*. Finally, given an entry* $z \in \mathcal{L}_0$ *and round number* $i \in \{0, \ldots, k\}$*, we define the following shorthand notation consistent with the definitions of the "local fold check" below:*

$$z^{(i)} := \begin{cases} z, & i = 0, \\ \left(z^{(i-1)}\right)^{m_i}, & i \geq 1. \end{cases} \tag{1}$$

Protocol description. The FRI protocol proceeds as follows:

- ***Initial function.*** *Let* $f_0 \colon \mathcal{L}_0 \to \mathbb{F}$ *be an initial function. In the honest case,* $f_0 \in \mathcal{C}_0$.
- ***Folding phase.*** *For every round* $i = 1, \ldots, k$*:*
  1. ***Folding challenge.*** *The verifier sends a challenges* $\alpha_i \leftarrow \mathbb{F}$.
  2. ***Send folded function.*** *The prover sends a function* $f_i \colon \mathcal{L}_i \to \mathbb{F}$*. In the honest case,* $f_i = \mathsf{Fold}(f_{i-1}, m_i, \alpha_i)$.
- ***Query phase.*** *For every repetition* $j = 1, \ldots, t$*, the verifier samples* $s_j \leftarrow \mathcal{L}_0$*. The verifier sends* $\alpha_{k+1} := (s_1, \ldots, s_t)$.
- ***Verifier decision phase:***
  1. ***Final function check.*** *The verifier reads the final function* f_k *in full and asserts that* $f_k \in \mathcal{C}_k$.
  2. ***Local fold checks.*** *For every repetition* $j = 1, \ldots, t$*:*
     - *Set* $s_j^{(0)} = s_j$.
     - *For every folding round* $i = 1, \ldots, k$*:*
       - *Set* $s_j^{(i)} := (s_j^{(i-1)})^{m_i}$.
       - *Query* $\mathsf{Fold}(f_{i-1}, m_i, \alpha_i)(s_j^{(i)})$*. This involves querying* f_{i-1} *at all the points* $x \in \mathsf{Block}(s_j^{(i)})$.
       - *Assert that* $\mathsf{Fold}(f_{i-1}, m_i, \alpha_i)(s_j^{(i)}) = f_i(s_j^{(i)})$.

Choosing parameters. In practice, we often choose $\mathbb{F}$ such that the order of $\mathbb{F}^*$ is divisible by a large power of 2 and pick the values $m_1, \ldots, m_k$ to be powers of 2.

5 Our Proof

In this section, we present our new security analysis. We start with an informal overview in Section 5.1. We then state and prove a key lemma about folding univariate functions in Section 5.2. Using this lemma, we give our formal proof in Section 5.3 and prove some additional properties in Section 5.4. Finally, we give a high-level comparison of our proof with exists strategies in Section 5.5. We extend our analysis to the case of batched FRI in Section A.

5.1 Intuition and overview

To prove that FRI has round-by-round soundness, we first define a state function. Then, for each round of interaction, we will bound the probability that a random verifier message changes the state from 0 to 1. Before describing the state function, we introduce the *graph of prover messages*, a useful tool in our reasoning which appears implicitly [BCI+20] or explicitly [BKS18, ABN23] in previous analyses of FRI.

The prover message graph. Consider an execution of the FRI protocol for the initial function f_0 with k rounds of folding. A full transcript for this interaction is of the form

$$\mathsf{tr} = (\alpha_1, f_1, \ldots, \alpha_k, f_k, \alpha_{k+1}),$$

where the values $\alpha_1, \ldots, \alpha_k$ and $f_1, \ldots, f_k$ are respectively the verifier and prover messages during the folding rounds, and α_{k+1} defines the verifier's queries to the oracles. By definition of the FRI protocol, the functions $f_0, \ldots, f_k$ are defined over related domains ($\mathcal{L}_0, \ldots, \mathcal{L}_k$, respectively) and are expected to be consecutive applications of the Fold function to f_0.

We can draw a layered graph in which the i-th layer represents the i-th evaluation domain $\mathcal{L}_i$, and each node of the layer corresponds to a point $y \in \mathcal{L}_i$. We draw an edge between nodes $x \in \mathcal{L}_{i-1}$ and $y \in \mathcal{L}_i$ if and only if $x^{m_i} = y$, or equivalently $x \in \mathsf{Block}(y)$. In this case, we follow typical graph conventions and call y the *parent* node and x the *child* node; children nodes with the same parent node are called *siblings*. An illustration of this graph for $|\mathcal{L}_0| = 16$, $k = 3$ and $m_i = 2$ for all i is given in[3] Figure 1.

Coloring the graph. We will now *color* the graph to reflect the verifier's observations and its eventual output. During the query phase of FRI, the verifier reads the final function f_k in full and runs tests to assert that the rest of the graph is consistent with it. The verifier samples a node $s^{(0)} \in \mathcal{L}_0$ and defines a path of related queries $s^{(1)} \in \mathcal{L}_1, \ldots, s^{(k)} \in \mathcal{L}_k$ that traverses the graph vertically. Roughly, our coloring therefore considers all such paths and marks them as red if any of the verifier's assertions fail, and green otherwise. More concretely, the coloring rules are defined below and illustrated in Figure 2:

1. *Base case.* The bottom layer is always green.
2. *Inductive rule.* Given a node $x \in \mathcal{L}_{i-1}$ at layer $i-1$ with parent $y \in \mathcal{L}_i$:
   (a) if the parent y is red, then x is red. This reflects the fact that any verification path that goes through x will also go through $x^{m_i} = y$.
   (b) if the parent y is green, compute $\mathsf{Fold}(f_{i-1}, m_i, \alpha_i)(y)$ from x and its siblings and
      i. color x green if an only if $\mathsf{Fold}(f_{i-1}, m_i, \alpha_i)(y) = f_i(y)$ (*i.e.*, if the verifier's assertion passes).
      ii. color x red otherwise (*i.e.*, if the verifier's assertion fails).

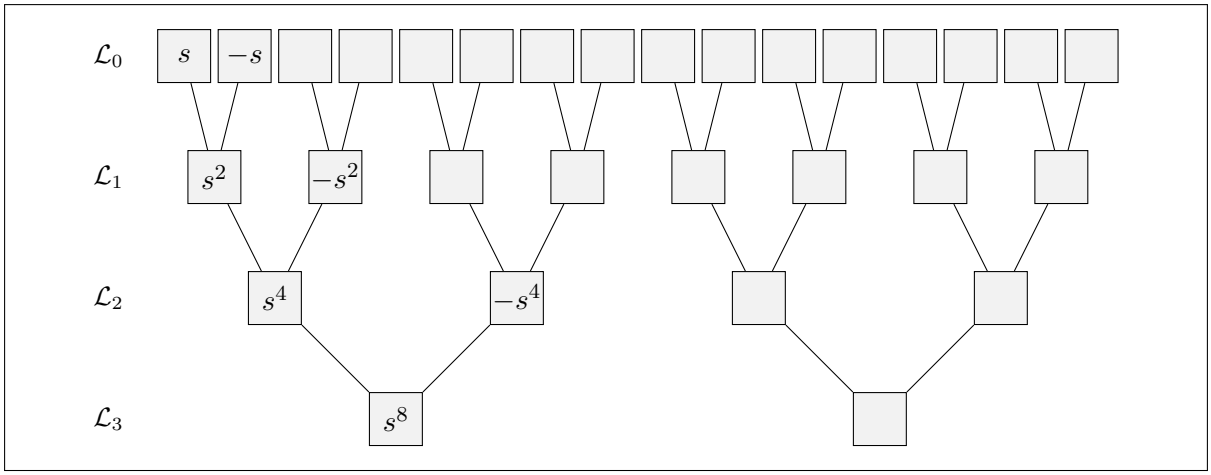

Figure 1: A prover message graph before coloring. We label some values to make explicit the relation between parent and children nodes.

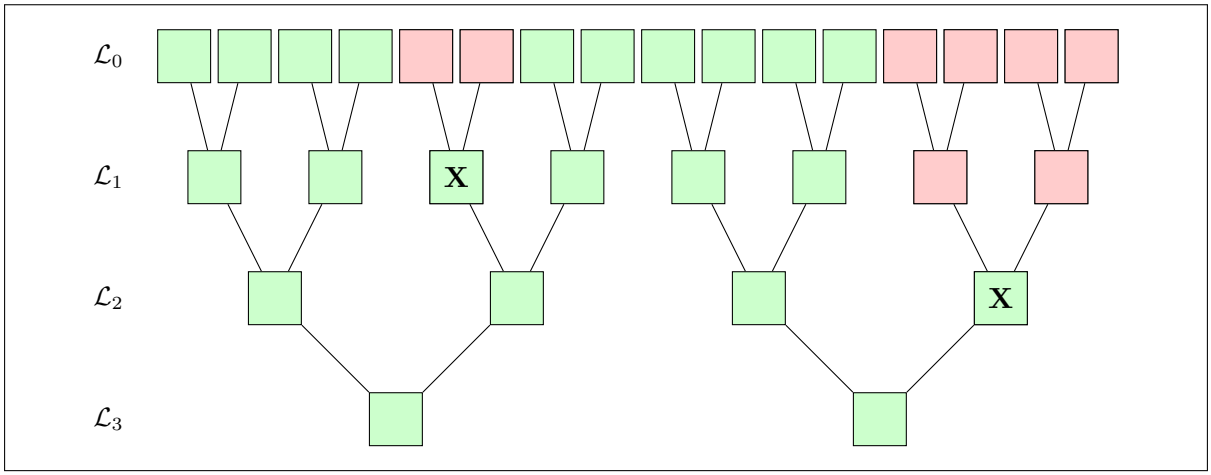

Figure 2: A colored prover message graph. We mark a node $y \in \mathcal{L}_i$ with an **X** if the value $f_i(y)$ is not consistent with $\mathsf{Fold}(f_{i-1}, m_i, \alpha_i)(y)$.

We stress the following important intuition about the prover message graph:

- if the verifier sample $s^{(0)}$ is at a red node in the top layer, the verifier will reject.
- red propagates upwards: if the prover folds incorrectly at any layer (marked by an **X** in Figure 2), the consequences are felt all the way to the top layer.
- the green nodes in the graph correspond to subsets of the evaluation domains $\mathcal{L}_i$ where the Fold function was applied honestly.

Given these observations, we are ready to informally describe and analyze our state function.

State function. By definition, we need to set the initial state $\mathsf{State}(f_0, \emptyset)$ to be 0 if and only if $f_0 \notin \mathcal{C}_0$. We now separate the description of the state function between the folding rounds and the query round.

[3] We sincerely apologize to all computer scientists whose hearts ache at the sight of two upside-down trees — we draw them this way to show the temporal order of prover messages.

- *Folding rounds.* As noted in our intuition for the prover message graph, the verifier will reject all queries that induce a path through red nodes. Therefore, during the folding phase, we can focus on the subgraph that contains paths that exclusively traverse green nodes; trusting that all other paths will be rejected in the query phase. This allows us to reason only in terms of honestly folded functions.

  For convenience, we refer to this accepting subgraph as the green subgraph and denote W_i the set of nodes at the i-th layer of the green subgraph. In Figure 2, this corresponds to all nodes that are green and not marked with an **X**.

  Ideally, we would like to establish that if $\mathsf{Fold}(f_{i-1}, m_i, \alpha_i)$ agrees with a codeword $u_i \in \mathcal{C}_i$ on the green subset $W_i \subseteq \mathcal{L}_i$, then f_{i-1} agrees with a codeword $u_{i-1} \in \mathcal{C}_{i-1}$ on $W_{i-1} \subseteq \mathcal{L}_{i-1}$. In other words, that:

  > *(ideal property)* "agreement propagates upwards through the green subgraph."

  Unfortunately, this property does not always hold true. We can, however, show that a relaxed version of this implication holds with high probability using mutual correlated agreement. Our relaxed ideal property becomes

  > *(relaxed ideal property)* "if all layers of the green subgraph are *large enough*, then agreement propagates upwards through the green subgraph."

  The notion of *large enough* is a parameter of our analysis. We let $\theta \in [0, 1]$ be an arbitrary parameter and say that a layer $W_i \subseteq \mathcal{L}_i$ is large enough if $|W_i| \geq (1 - \theta) \cdot |\mathcal{L}_i|$. As we will see, θ imposes a tradeoff between the analysis of the folding rounds and that of the query round. We can accept a relaxation of the ideal property because if any layer of the green subgraph is small (*i.e.*, not large enough), then the red subset at that layer must be large. These red nodes will propagate upwards and cause the verifier to reject with high probability. We discuss this further in the analysis section for the query phase.

  We define the state function such that it indicates whether the relaxed ideal property holds for a given initial function $f_0 \notin \mathcal{C}_0$ and transcript tr: when all the honest folded functions are safe (as per our relaxed ideal property), we set $\mathsf{State}(f_0, \mathsf{tr}) = 0$; when there is an honest fold that is exploitable (one for which all green subgraph layers are large but agreement does not propagate upward), we set $\mathsf{State}(f_0, \mathsf{tr}) = 1$. The event "the state changed from 0 to 1 at round i" is captured by the following semi-formal description. Given an initial function f_0 and a transcript $\mathsf{tr} := (\alpha_1, f_1, \ldots, \alpha_{i-1}, f_{i-1})$ such that $\mathsf{State}(f_0, \mathsf{tr}) = 0$, and an i-th verifier message α_i, we say that $\mathsf{State}(f_0, \mathsf{tr}||\alpha_i) = 1$ if and only if there exists a set $S \subseteq \mathcal{L}_{i-1}$ such that:

  1. S^{m_i} is large enough. Specifically, $|S^{m_i}| \geq (1 - \theta) \cdot |\mathcal{L}_i|$. Note that this implies that S itself is large enough.
  2. The honest fold of f_{i-1} agrees with a codeword $u \in \mathcal{C}_i$ on the set S^{m_i}. In other words, there is agreement with the code on this large enough set.
  3. There are no codewords in $\mathcal{C}_{i-1}$ that agree with f_{i-1} on the set S. Effectively, that agreement does not propagate upwards through the large enough set of green nodes.

- *Query round.* The query round is the final round of the protocol. Therefore, by the "full transcript" property of the state function (see Definition 3.4), the output of the state function at this round must be equivalent to the verifier's final decision. We therefore set the state function to be 1 if and only if all the verifier checks pass.

Analysis. To complete our proof of round-by-round soundness, we must bound the probability at round i that the state changes from 0 of 1. We separate the analysis between the folding rounds and the query round.

- *Folding rounds.* The analysis of the folding rounds relies on the mutual correlated agreement property. This can be applied because the Fold function is indeed a linear combination that uses the Powers proximity generator. Any challenge α_i that causes an exploitable fold at round i (as defined by the state function) is also a challenge that realizes the bad event of mutual correlated agreement. This intuition is formally stated and proven in Lemma 5.1. Let $\varepsilon^{\mathsf{mca}}$ be the error associated with the Powers proximity generator, we establish that

$$\varepsilon_i^{\mathsf{fold}}(\delta) := \varepsilon^{\mathsf{mca}}(\mathcal{C}_i, m_i, \theta), \tag{2}$$

  where $\delta := \Delta(f_0, \mathcal{C}_0)$. Notice that $\varepsilon_i^{\mathsf{fold}}$ is constant with respect to the distance δ. This error can be adjusted by varying θ, our notion of "large enough". From the definition of $\varepsilon^{\mathsf{mca}}$, it follows that smaller values of θ give smaller values of $\varepsilon_i^{\mathsf{fold}}$. However, we cannot set θ to be arbitrarily small. As we will see, doing so would be detrimental to the error in the query round.

- *Query round.* The analysis of the query round follows directly from our discussion on the prover message graph. Consider an initial function f_0 and a transcript $\mathsf{tr} := (\alpha_1, f_1, \ldots, \alpha_k, f_k)$ such that $\mathsf{State}(f_0, \mathsf{tr}) = 0$, and a final verifier message α_{k+1}. Notice that if $f_k \notin \mathcal{C}_k$, then the verifier rejects immediately. Therefore, we will assume that $f_k \in \mathcal{C}_k$.

  From the definition of our state function, we know that $\mathsf{State}(f, \mathsf{tr}||\alpha_{k+1}) = 1$ if and only if the verifier's sample is in the green subset $W_0 \subseteq \mathcal{L}_0$. In other words, the probability that the state changes from a single query is exactly $|W_0|/|\mathcal{L}_0|$. We can now separate our analysis in two mutually exclusive cases:

  1. if any green subset is small, then $|W_0|/|\mathcal{L}_0| \leq 1 - \theta$. This follows from the fact that red propagates upwards when coloring the prover message graph.
  2. if all green subsets are large, then we know that "agreement propagates upwards through green nodes". This follows from the fact that $\mathsf{State}(f, \mathsf{tr}) = 0$ and our definition of the state function for folding rounds. Therefore, there exists a codeword $u_0 \in \mathcal{C}_0$ such that $f_0[W_0] = u_0[W_0]$. Consequently

$$\frac{|W_0|}{|\mathcal{L}_0|} \leq 1 - \Delta(f_0, u_0) \leq 1 - \delta,$$

     where $\delta = \Delta(f_0, \mathcal{C}_0)$ as defined previously.

  Combining both cases gives us our final upper bound:

$$\varepsilon^{\mathsf{qry}}(\delta) := \frac{|W_0|}{|\mathcal{L}_0|} \leq \max\{1 - \theta, 1 - \delta\}.$$

  We now see the effect of the parameter θ on the query round error: it is a constant value that acts as a clamp, making sure that the error is never lower than $1 - \theta$. Therefore, to minimize the query round error, we would want to use large values of θ. This is at odds with the preferred setting of θ in the folding round, and hence why we refer to the variable θ as the *tradeoff parameter*.

  We give all formal details of our proof in Sections 5.2 and 5.3.

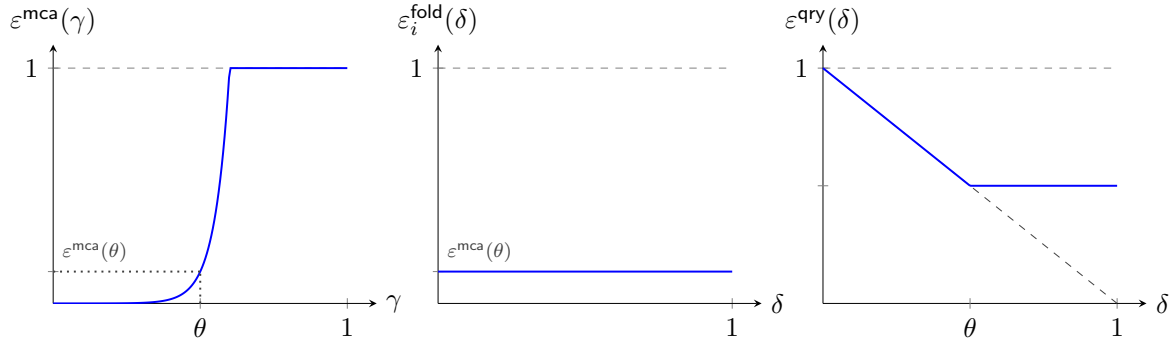

Figure 3: Sketches of the error functions $\varepsilon^{\mathsf{mca}}$, $\varepsilon_i^{\mathsf{fold}}$ and $\varepsilon^{\mathsf{qry}}$ for a single repetition of the FRI query phase (*i.e.*, $t = 1$). We assume the code $\mathcal{C}_i$ and folding factor m_i are fixed and omit them from the arguments of $\varepsilon^{\mathsf{mca}}$.

Interpreting the error functions. It is helpful to visualize the error functions in order to reason about the effect of various parameters on the security of FRI. We illustrate what the functions look like in Figure 3. These sketches are for pedagogical purposes only, in practice these functions take much smaller values, typically around $1/2^{100}$.

The left plot shows the mutual correlated agreement error of the Powers generator. For small values of θ, the value $\varepsilon^{\mathsf{mca}}(\mathcal{C}_i, m_i, \theta)$ is extremely small. For values beyond the list-decoding capacity of the code $\mathcal{C}_i$, the error becomes large [CS25]. In between, the behavior of the function is still unknown. The error function given in Lemma 2.4 is a proven upper bound on the probability of the bad MCA event and shoots steeply up to 1 around the Johnson bound $(1 - \sqrt{\rho})$ as illustrated.

The middle and right plots respectively show the errors $\varepsilon_i^{\mathsf{fold}}$ and $\varepsilon^{\mathsf{qry}}$ as functions of the distance $\delta := \Delta(f_0, \mathcal{C}_0)$. Notice how taking small values of θ will decrease the constant folding round error but cause the query round error to be clamped earlier. Finally, we note that the query round error can be made to decrease faster by adjusting the number of repetitions of the query phase (where additional queries will make the error function more concave).

5.2 Folding preserves agreement sets

Here we state and prove a simple lemma that is crucial to our analysis. Intuitively, we consider folding a function with a random folding challenge r, and show that the distance to the code does not go down. The lemma states even more: except with small probability over r, for all large enough sets S^m on which the folded function agrees with the code, the original function agrees on S with its code. We show that this follows from mutual correlated agreement. The proof of the lemma is a generalization of Lemma 4.9 in STIR [ACFY24].

Lemma 5.1 (Folding preserves agreement set)**.** *Let $m \in \mathbb{N}$ be a folding factor, $\mathcal{L} \subseteq \mathbb{F}$ be a multiplicative coset of $\mathbb{F}^*$ such that $|\mathcal{L}|$ is divisible by m and $d \in \mathbb{N}$ be a degree bound with $d \geq m$. Consider the Reed-Solomon codes $\mathcal{C}_0 := \mathsf{RS}[\mathbb{F}, \mathcal{L}, d]$ and $\mathcal{C}_1 := \mathsf{RS}[\mathbb{F}, \mathcal{L}^m, d/m]$, both with rate $\rho = d/|\mathcal{L}|$.*

If Powers *is a proximity generator with mutual correlated agreement (cf. Definition 2.2) with error $\varepsilon^{\mathsf{mca}}$, then for every function $f \colon \mathcal{L} \to \mathbb{F}$ and distance parameter $\gamma \in [0, 1]$,*

$$\Pr_{r \leftarrow \mathbb{F}} \left[\begin{array}{l} |S^m| \geq (1-\gamma) \cdot |\mathcal{L}^m| \\ \exists S \subseteq \mathcal{L} \text{ s.t. } \wedge\, \exists u' \in \mathcal{C}_1, \mathsf{Fold}(f, m, r)[S^m] = u'[S^m] \\ \wedge\, \forall u \in \mathcal{C}_0, u[S] \neq f[S] \end{array} \right] \leq \varepsilon^{\mathsf{mca}}(\mathcal{C}_1, m, \gamma).$$

Proof. Let $f\colon \mathcal{L} \to \mathbb{F}$ be a function, and define functions $f_1, \ldots, f_m\colon \mathcal{L}^m \to \mathbb{F}$ so that

$$\mathsf{Fold}(f, m, r) = \sum_{i=1}^{m} r^{i-1} f_i \ : \mathcal{L}^m \to \mathbb{F}, \tag{3}$$

as follows. Recall the polynomials $\widehat{p}_{f,z}$ as defined in Definition 4.2, *i.e.*, for every $z \in \mathcal{L}^m$, $\widehat{p}_{f,z}$ is the unique polynomial of degree less than m that agrees with f on $\mathsf{Block}(z)$. We let $f_i\colon \mathcal{L}^m \to \mathbb{F}$ be the function that maps z to the i-th coefficient of $\widehat{p}_{f,z}$, *i.e.*, so that $\widehat{p}_{f,z}(Y) = \sum_{i=1}^{m} f_i(z) Y^{i-1}$ for every $z \in \mathcal{L}^m$. One can see that these functions f_i satisfy (3), by definition of Fold (cf. Definition 4.2).

Now, using the equality (3) to substitute $\mathsf{Fold}(f, m, r)$ in the probability we want to bound, we have

$$\begin{aligned}
&\Pr_r \left[\begin{array}{l} |S^m| \geq (1-\gamma) \cdot |\mathcal{L}^m| \\ \exists S \subseteq \mathcal{L} \ s.t. \ \ \wedge \exists u' \in \mathcal{C}_1, \mathsf{Fold}(f, m, r)[S^m] = u'[S^m] \\ \qquad\qquad\ \ \wedge \forall u \in \mathcal{C}_0, u[S] \neq f[S] \end{array} \right] \\
\leq &\Pr_r \left[\begin{array}{l} |S^m| \geq (1-\gamma) \cdot |\mathcal{L}^m| \\ \exists S \subseteq \mathcal{L} \ s.t. \ \ \wedge \exists u' \in \mathcal{C}_1, \sum_{i=1}^{m} r^{i-1} f_i[S^m] = u'[S^m] \\ \qquad\qquad\ \ \wedge \forall u \in \mathcal{C}_0, u[S] \neq f[S] \end{array} \right].
\end{aligned}$$

The latter is almost the event whose probability is bounded in the definition of mutual correlated agreement (cf. Definition 2.2) for the code $\mathcal{C}_1 = \mathsf{RS}[\mathbb{F}, \mathcal{L}^m, d/m]$, except for the last condition, which reads

$$\forall u \in \mathcal{C}_0, u[S] \neq f[S], \tag{4}$$

while in the aforementioned event, the corresponding condition is

$$\exists i \in [m], \forall u' \in \mathcal{C}_1, u'[S^m] \neq f_i[S^m]. \tag{5}$$

Thus, to prove the present lemma, it suffices to prove that (4) implies (5).

To this end, assume towards contradiction that (5) does not hold, *i.e.,* for every $i \in [m]$, there is a $u_i \in \mathcal{C}_1$ such that f_i and u_i agree on S^m. We will show that then f agrees with a codeword in $\mathcal{C}_0$ on S. To do so, we will first define a bivariate polynomial $\widehat{Q}(X, Y) \in \mathbb{F}[X, Y]$ and will show that f agrees with $\widehat{Q}(X^m, X)$ on S. We will also show that $\widehat{Q}(X^m, X)$ has degree less than d, which implies our desired result, namely that f agrees on S with a codeword from $\mathcal{C}_0$.

Precisely, we define the bivariate polynomial $\widehat{Q} \in \mathbb{F}[X, Y]$ as follows: Fix any subset $\tilde{S} \subseteq S^m$ of size $\min\{|S^m|, d/m\}$. For every $x \in \tilde{S}$, let $I_x \in \mathbb{F}[X]$ denote the polynomial of degree less than $|\tilde{S}|$ such that

$$I_x(z) = \begin{cases} 1 & \text{if } z = x, \\ 0 & \text{if } z \in \tilde{S} \setminus \{x\} \end{cases}.$$

Note that the degree of I_x is less than d/m since $|\tilde{S}| \leq d/m$. Now, define

$$\widehat{Q}(X, Y) := \sum_{x \in \tilde{S}} I_x(X) \cdot \widehat{p}_{f,x}(Y) \in \mathbb{F}[X, Y].$$

Clearly, $\widehat{Q}(X, Y)$ has degree less than d/m on the variable X, and degree less than m on the variable Y.

Next, we show that for every $x \in S^m$, we have $\widehat{Q}(x, Y) = \widehat{p}_{f,x}(Y)$ as polynomials. Indeed, we have

$$\widehat{Q}(\tilde{x}, Y) = \sum_{x \in \tilde{S}} I_x(\tilde{x}) \cdot \widehat{p}_{f,x}(Y) = \widehat{p}_{f,\tilde{x}}(Y) = \sum_{i=1}^{m} f_i(\tilde{x}) Y^{i-1} = \sum_{i=1}^{m} u_i(\tilde{x}) Y^{i-1} \quad \text{for all } \tilde{x} \in \tilde{S}, \tag{6}$$

where the first equality is by definition of $\widehat{Q}$; the second equality follows from the definition of the polynomials $I_x(X)$; the third from the definition of the functions f_i; and the fourth from the assumption that f_i agrees with the codeword u_i on $S^m \supseteq \tilde{S}$, for all $i \in [m]$.

Assume now that $|\tilde{S}| = d/m$, and recall that $\widehat{Q}(X,Y)$ has degree less than d/m on the variable X. In particular, $\widehat{Q}(X,Y)$ can be seen as a univariate polynomial on the variable X, of degree less than d/m, with coefficients on the field of rational functions $\mathbb{F}(Y)$. Similarly, $\sum_{i=1}^{m} u_i(X)Y^{i-1}$ can be seen as a univariate polynomial on the variable X of degree less than d/m, with coefficients in $\mathbb{F}(Y)$. Since we assumed $|\tilde{S}| = d/m$, because of (6), these two polynomials agree on d/m distinct points, and so they must be equal as polynomials. Hence, in this case,

$$\widehat{Q}(X,Y) = \sum_{i=1}^{m} u_i(X)Y^{i-1},$$

and in particular

$$\widehat{Q}(x,Y) = \sum_{i=1}^{m} u_i(x)Y^{i-1} = \widehat{p}_{f,x}(Y) \quad \text{for all } x \in S^m, \tag{7}$$

as claimed. On the other hand, if $|\tilde{S}| < d/m$, then $\tilde{S} = S^m$, and then (7) holds for all $x \in S^m$ because of Equation (6).

Finally, define the polynomial $\widehat{f}(X) := \widehat{Q}(X^m, X)$, which has degree less than d because $\widehat{Q}(X^m, Y)$ has degree on X at most $m(d/m - 1)$, and so $\widehat{Q}(X^m, X)$ has degree on X at most

$$m(d/m - 1) + (m - 1) = d - m + m - 1 = d - 1.$$

Now, note that $\widehat{f}$ and f agree on S, because for every $x \in S$:

$$\widehat{f}(x) = \widehat{Q}(x^m, x) = \widehat{p}_{f,x^m}(x) = f(x),$$

where the first equality follows by definition of $\widehat{f}$, the second from Equation (7), and the third by definition of $\widehat{p}_{f,x^m}$, since $\widehat{p}_{f,x^m}$ is unique polynomial of degree less than m agreeing with f on $\mathsf{Block}(x^m)$, and $x \in \mathsf{Block}(x^m)$. Hence, f agrees with the codeword $\widehat{f} \in \mathsf{RS}[\mathbb{F}, \mathcal{L}, d]$ on S, as needed. This completes the proofs that (4) implies (5), and therefore completes the proof of the desired lemma. □

5.3 Full proof

We now state the theorem for round-by-round soundness and give its full proof.

Theorem 5.2. *Consider FRI with parameters as in Construction 4.4 and assume* Powers *is a proximity generator with mutual correlated agreement (cf. Definition 2.2) with error* $\varepsilon^{\mathsf{mca}}$. *For every tradeoff parameter* $\theta \in [0,1]$, *FRI has round-by-round soundness with error functions* $\varepsilon_1^{\mathsf{fold}}, \ldots, \varepsilon_k^{\mathsf{fold}}, \varepsilon^{\mathsf{qry}} \colon (0,1] \to [0,1]$ *such that for every initial function* $f_0 \notin \mathcal{C}_0$:

- *for all* $i \in [k]$, $\varepsilon_i^{\mathsf{fold}}(\delta) = \varepsilon^{\mathsf{mca}}(\mathcal{C}_i, m_i, \theta)$,
- $\varepsilon^{\mathsf{qry}}(\delta) = (\max\{1-\theta, 1-\delta\})^t = (1 - \min\{\theta, \delta\})^t$,

where $\delta := \Delta(f_0, \mathcal{C}_0)$.

Proof. We describe the state function State_θ for an arbitrary θ and for each round give an upper bound on the probability that the state changes from 0 to 1. By default, the state function never changes from 1 back to 0, and the state function does not change if the prover moves. We describe below how the state function may change from 0 to 1 when the verifier moves, and bound the probability of such a change happening.

0. **Empty transcript.** For an empty transcript, we say that $\mathsf{State}_\theta(f_0, \emptyset) = 1$ if and only if $f_0 \in \mathcal{C}_0$.

1. **Folding rounds** $(\varepsilon_1^{\mathsf{fold}}, \ldots, \varepsilon_k^{\mathsf{fold}})$**.** For rounds $i \in \{1, \ldots, k\}$, the transcript tr has the form

$$\mathsf{tr} = \begin{cases} \emptyset, & i = 1, \\ (\alpha_1, f_1, \ldots, \alpha_{i-1}, f_{i-1}), & 1 < i \leq k, \end{cases} \tag{8}$$

and the verifier sends a random message $\alpha_i \in \mathbb{F}$.

- *State function.* Suppose that $\mathsf{State}_\theta(f_0, \mathsf{tr}) = 0$. We say that $\mathsf{State}_\theta(f_0, \mathsf{tr}||\alpha_i) = 1$ if and only if there exists a set $S \subseteq \mathcal{L}_{i-1}$ such that:
  (a) $|S^{m_i}| \geq (1 - \theta) \cdot |\mathcal{L}_i|$; and
  (b) there exists $u' \in \mathcal{C}_i$ such that $\mathsf{Fold}(f_{i-1}, m_i, \alpha_i)[S^{m_i}] = u'[S^{m_i}]$; and
  (c) for all $u \in \mathcal{C}_{i-1}$, there exists $x \in S$ such that $u(x) \neq f_{i-1}(x)$.
- *Bounding the error.* By Lemma 5.1:

$$\Pr_{\alpha_i}[\mathsf{State}_\theta(f_0, \mathsf{tr}) = 0 \ \wedge \ \mathsf{State}_\theta(f_0, \mathsf{tr}||\alpha_i) = 1] \leq \varepsilon^{\mathsf{mca}}(\mathcal{C}_i, m_i, \theta).$$

  Setting $\varepsilon_i^{\mathsf{fold}}(\delta) := \varepsilon^{\mathsf{mca}}(\mathcal{C}_i, m_i, \theta)$ for all $\delta \in [0, 1]$ completes the proof for all folding rounds.

2. **Query round** $\varepsilon^{\mathsf{qry}}$**.** A final round partial transcript tr has the form $\mathsf{tr} = (\alpha_1, f_1, \ldots, \alpha_k, f_k)$. The final verifier message α_{k+1} has the form $\alpha_{k+1} = (s_1, \ldots, s_t)$ with $s_1, \ldots, s_t \in \mathcal{L}_0$.

- *State function.* Suppose that $\mathsf{State}_\theta(f_0, \mathsf{tr}) = 0$. We say that $\mathsf{State}_\theta(f_0, \mathsf{tr}||\alpha_{k+1}) = 1$ if and only if both of the following conditions are satisfied:
  (a) the final prover message f_k is in the corresponding code $\mathcal{C}_k$, and
  (b) for all $j \in [t]$, for all $i \in [k]$, $\mathsf{Fold}(f_{i-1}, m_i, \alpha_i)(s_j^{(i)}) = f_i(s_j^{(i)})$, where $s_j^{(i)}$ is defined as in the verifier decision phase (and Equation (1)).
- *Bounding the error.* Consider an initial function f_0 and transcript tr such that $\mathsf{State}_\theta(f_0, \mathsf{tr}) = 0$. We can assume that Item 2a (final prover message is in the code) is true, otherwise the state will remain 0. Throughout this section, we will use the notation $z^{(i)}$ as defined in Equation (1).
  Defining $\mathsf{AccQueries}$**.** Let $\mathsf{AccQueries} \subseteq \mathcal{L}_0$ denote the (potentially empty) set

$$\mathsf{AccQueries} := \left\{ z \in \mathcal{L}_0 \ : \ \forall i \in [k] \ \mathsf{Fold}(f_{i-1}, m_i, \alpha_i)(z^{(i)}) = f_i(z^{(i)}) \right\}.$$

  In other words, $\mathsf{AccQueries}$ is the set of queries in $\mathcal{L}_0$ for which the verifier's "local fold checks" are satisfied for every prover message.
  By definition of $\mathsf{AccQueries}$ and the state function, it holds that $\mathsf{State}_\theta(f_0, \mathsf{tr}||\alpha_{k+1}) = 1$ if and only if for every repetition $j \in [t]$ of the query phase, $s_j \in \mathsf{AccQueries}$. Therefore, it holds that:

$$\begin{aligned} \Pr_{\alpha_{k+1}} \left[\begin{array}{c} \mathsf{State}_\theta(f_0, \mathsf{tr}) = 0 \\ \wedge \\ \mathsf{State}_\theta(f_0, \mathsf{tr}||\alpha_{k+1}) = 1 \end{array} \right] &= \Pr_{s_1, \ldots, s_t \leftarrow \mathcal{L}_0}[\forall j \in [t], s_j \in \mathsf{AccQueries}] \\ &= \left(\frac{|\mathsf{AccQueries}|}{|\mathcal{L}_0|} \right)^t. \end{aligned} \tag{9}$$

Upper bounding $|\mathsf{AccQueries}|$**.** Let $\delta := \Delta(f_0, \mathcal{C}_0)$. The rest of the proof aims to show that

$$|\mathsf{AccQueries}|/|\mathcal{L}_0| \leq \max\{1-\theta, 1-\delta\}.$$

We start by defining the sets $W_0, \ldots, W_k$ inductively as follows:

- base case: $W_0 := \mathsf{AccQueries}$,
- induction: $W_{i+1} := \{x^{m_{i+1}} : x \in W_i\} = W_i^{m_{i+1}}$.

We show in Claim 5.3 that the sets $W_0, \ldots W_k$ are subsets of $\mathcal{L}_0, \ldots, \mathcal{L}_k$ for which all the nodes are green and all their immediate children are green (*i.e.*, that these are the layers of the green subgraph from Section 5.1). We then split our analysis into two cases: either there exists a set W_i that is smaller than $(1-\theta) \cdot |\mathcal{L}_i|$ or all sets W_i are large. The former case is solved using Claim 5.4; the latter using Claim 5.5 and bounding the size of AccQueries with the size of the maximal agreement set between f_0 and a codeword in $\mathcal{C}_0$.

We state and prove all claims immediately, with the case analysis and application of the claims following thereafter.

Claim 5.3. *For every* $i \in [k]$, $f_i[W_i] = \mathsf{Fold}(f_{i-1}, m_i, \alpha_i)[W_i]$.

Proof. By definition of the sets $W_0, \ldots, W_k$, it holds that for all $i \in [k]$, $W_i = \{z^{(i)} : z \in W_0\}$. For each $i \in [k]$, let A_i be the subset of elements a from $\mathcal{L}_i$ such that $f_i(a) = \mathsf{Fold}(f_{i-1}, m_i, \alpha_i)(a)$. To prove Claim 5.3, it suffices to show that $W_i \subseteq A_i$.

To this end, fix any $i \in [k]$, and let $w \in W_i$ be arbitrary. As noted above, we can write $w = z^{(i)}$ for some $z \in W_0 = \mathsf{AccQueries}$. By definition of the set AccQueries, we have

$$\mathsf{Fold}(f_{i-1}, m_i, \alpha_i)(w) = \mathsf{Fold}(f_{i-1}, m_i, \alpha_i)(z^{(i)}) = f_i(z^{(i)}) = f_i(w),$$

where we have used the definition of AccQueries in the second equality. The above proves that $w \in A_i$, by definition of A_i, and thus $W_i \subseteq A_i$ as w was arbitrary. This shows that the claim is true. □

Claim 5.4. *For every* $\gamma \in [0,1]$, *if there exists* $i^* \in \{0, \ldots, k\}$ *such that* $|W_{i^*}| < (1-\gamma) \cdot |\mathcal{L}_{i^*}|$, *then* $|W_0| < (1-\gamma) \cdot |\mathcal{L}_0|$.

Proof. The statement holds trivially for $i^* = 0$; we therefore focus on the case $i^* \in [k]$. First, note that W_{i-1} is contained in the set $U = \bigcup_{x \in W_i} \mathsf{Block}(x)$. Then, since the sets $\mathsf{Block}(x)$ are disjoint, we have $|U| = m_i \cdot |W_i|$, and so

$$|W_{i-1}| \leq m_i \cdot |W_i|.$$

Recalling that $|\mathcal{L}_{i-1}| = m_i \cdot |\mathcal{L}_i|$, it holds that

$$\frac{|W_{i-1}|}{|\mathcal{L}_{i-1}|} \leq \frac{|W_i|}{|\mathcal{L}_i|}.$$

Therefore, if there exists $i^* \in [k]$ such that $|W_{i^*}| < (1-\gamma) \cdot |\mathcal{L}_{i^*}|$, then

$$\frac{|W_0|}{|\mathcal{L}_0|} \leq \frac{|W_{i^*}|}{|\mathcal{L}_{i^*}|} < 1-\gamma,$$

proving the claim. □

Claim 5.5. *If* $f_k \in \mathcal{C}_k$ *and for all* $i \in \{0, \ldots, k\}$, $|W_i| \geq (1-\theta) \cdot |\mathcal{L}_i|$, *then there exists* $u_0 \in \mathcal{C}_0$ *such that* $u_0[W_0] = f_0[W_0]$.

Proof. We claim that for all $i \in [k]$, if there exists $u_i \in \mathcal{C}_i$ such that $u_i[W_i] = f_i[W_i]$ then there exists $u_{i-1} \in \mathcal{C}_{i-1}$, $u_{i-1}[W_{i-1}] = f_{i-1}[W_{i-1}]$. Given that $W_k \subseteq \mathcal{L}_k$ and that $f_k \in \mathcal{C}_k$, we can apply this implication for all $i \in [k]$ and conclude that there exists $u_0 \in \mathcal{C}_0$ such that $u_0[W_0] = f_0[W_0]$.
It remains to prove this claimed implication. Assume for contradiction that there exists $i^* \in [k]$ such that both of the following conditions are satisfied:

- there exists $u_{i^*} \in \mathcal{C}_{i^*}$ such that $u_{i^*}[W_{i^*}] = f_{i^*}[W_{i^*}]$; and
- for all $u \in \mathcal{C}_{i^*-1}$, $u[W_{i^*-1}] \neq f_{i^*-1}[W_{i^*-1}]$.

We can rewrite the first bullet point using Claim 5.3:

$$u_{i^*}[W_{i^*}] = f_{i^*}[W_{i^*}] = \mathsf{Fold}(f_{i^*-1}, m_{i^*}, \alpha_{i^*})[W_{i^*}].$$

Given that $W_{i^*} = (W_{i^*-1})^{m_i^*}$ by definition, it holds that Items 1b and 1c are satisfied for the set W_{i^*-1}, as shown by the equation above and the second bullet point respectively. Since $\mathsf{State}_\theta(f_0, \mathsf{tr}) = 0$, this implies that $|W_{i^*}| < (1-\theta) \cdot |\mathcal{L}_{i^*}|$ (Item 1a is false). However, the premise of Claim 5.5 states that for all $i \in \{0, \ldots, k\}$, $|W_i| \geq (1-\theta) \cdot |\mathcal{L}_i|$. This is a contradiction. So, we have proven the implication, and thus the claim. □

Given Claims 5.4 and 5.5 above, we can separate our analysis into two mutually exclusive cases:

- Case 1: $\exists i \in \{0, \ldots, k\}, |W_i| < (1-\theta) \cdot |\mathcal{L}_i|$. Here, it follows by applying Claim 5.4 for $\gamma = \theta$ that
$$|W_0| < (1-\theta) \cdot |\mathcal{L}_0|.$$
Therefore, substituting $\mathsf{AccQueries}$ for W_0 and dividing both sides by $|\mathcal{L}_0|$ yields
$$\frac{|\mathsf{AccQueries}|}{|\mathcal{L}_0|} < 1 - \theta \leq \max\{1-\theta, 1-\delta\},$$
where the second inequality follows by definition of the max function.
- Case 2: $\forall i \in \{0, \ldots, k\}, |W_i| \geq (1-\theta) \cdot |\mathcal{L}_i|$. Here, it follows from Claim 5.5 that there exist $u_0 \in \mathcal{C}_0$ such that $u_0[W_0] = f_0[W_0]$. Therefore, the relative agreement between f_0 and u_0, denoted $1 - \Delta(f_0, u_0)$, is at least $|\mathsf{AccQueries}|/|\mathcal{L}_0|$:
$$1 - \Delta(f_0, u_0) \geq |\mathsf{AccQueries}|/|\mathcal{L}_0|.$$
By definition of the relative distance to a code, it holds that $\Delta(f_0, \mathcal{C}_0) \leq \Delta(f_0, u_0)$. Substituting δ for $\Delta(f_0, \mathcal{C}_0)$:
$$1 - \delta \geq 1 - \Delta(f_0, u_0).$$
Chaining these inequalities, we find that
$$\frac{|\mathsf{AccQueries}|}{|\mathcal{L}_0|} \leq 1 - \Delta(f_0, u_0) \leq 1 - \delta \leq \max\{1-\theta, 1-\delta\},$$
where the final inequality follows by definition of the max function.

In both cases, we have shown that $|\mathsf{AccQueries}|/|\mathcal{L}_0| \leq \max\{1-\theta, 1-\delta\}$. Plugging this into Equation (9), we find that:

$$\Pr_{\alpha_{k+1}}\left[\begin{array}{c} \mathsf{State}_\theta(f_0, \mathsf{tr}) = 0 \\ \wedge \\ \mathsf{State}_\theta(f_0, \mathsf{tr} || \alpha_{k+1}) = 1 \end{array}\right] \leq (\max\{1-\theta, 1-\delta\})^t.$$

Setting $\varepsilon^{\mathsf{qry}}(\delta) := (\max\{1-\theta, 1-\delta\})^t$ and noting that $\max\{1-\theta, 1-\delta\} = 1 - \min\{\theta, \delta\}$ completes the proof for the query rounds.

3. **Verifier decision.** A full transcript is of the form $\mathsf{tr} = (\alpha_1, f_1, \ldots, \alpha_k, f_k, \alpha_{k+1})$. If we have $\mathsf{State}_\theta(f_0, \mathsf{tr}) = 0$, then either Item 2a or Item 2b is false. These conditions are identical to the checks performed by the verifier (the *final function check* and *local fold check* respectively.) Therefore, we have
$$\mathbf{V}^{f_0, f_1, \ldots, f_k}(\alpha_1, \ldots, \alpha_{k+1}) = 0$$
as desired.

□

5.4 Additional properties

Usually, the property that FRI is supposed to provide is the following: if a verifier accepts a FRI interaction, then the initial function was somewhat close to a Reed–Solomon codeword (except with negligible probability). Our round-by-round soundness analysis allows to surface this property. As we will show here, our analysis also shows additional properties. These properties may be used in the future as a stepping stone, e.g., towards proving round-by-round knowledge soundness [CMS19, BCFW25] or towards constructing efficient code commitments and data availability sampling schemes [HASW24].

More concretely, we show in Corollary 5.6 that, under certain conditions, a FRI transcript guarantees that the prover has *committed* to a Reed-Solomon codeword. This codeword agrees with the initial function on a large subset of its domain; hence proving proximity. Furthermore, the commitment is binding in the sense that any attempt to open it at a point where the initial function disagrees with the committed codeword results in rejection by the verifier. We note that morally, this property is similar to opening-consistency in [HASW24], but in contrast to their work, our analysis is not restricted to the unique decoding regime[4]. Consequently, we believe that this can be a starting point for future work towards data availability sampling from FRI beyond the unique decoding radius.

Corollary 5.6. *Consider FRI with parameters as in Construction 4.4, a fixed tradeoff parameter* $\theta^* \in [0, 1]$ *and let* $\mathsf{State}_{\theta^*}$ *be the state function defined in the proof of Theorem 5.2.*

For every distance parameter $\delta^* \in [0, 1 - \rho] \cap [0, \theta^*]$*, if* f_0 *is an initial function and* $\mathsf{tr} := (\alpha_1, f_1, \ldots, \alpha_k, f_k)$ *is a transcript such that*
$$\mathsf{State}_{\theta^*}(f_0, \mathsf{tr}) = 0 \ \wedge \ \Pr_{\alpha_{k+1}}[\mathsf{State}_{\theta^*}(f_0, \mathsf{tr} || \alpha_{k+1}) = 1] \geq (1 - \delta^*)^t$$
then the following properties are true:

1. ***Proximity**: f_0 is at least δ^*-close to the code $\mathcal{C}_0$, i.e., $\Delta(f_0, \mathcal{C}_0) \leq \delta^*$.*
2. ***Binding**: there exists a unique codeword $u_0 \in \mathcal{C}_0$ such that $u_0[\mathsf{AccQueries}] = f_0[\mathsf{AccQueries}]$, where $\mathsf{AccQueries} \subseteq \mathcal{L}_0$ is the set of accepting queries for the pair (f_0, tr).*
3. ***Extractability**: there exists a deterministic polynomial-time extractor[5] $\mathbf{E}$ that, given (f_0, tr) as input, returns u_0 as defined in the binding property above.*
4. ***Erasure detection**: let u_0 be defined as in the binding property and $\alpha_{k+1} := (s_1, \ldots, s_t)$ be a final round verifier message. If there exists $i \in [t]$ such that $f_0(s_i) \neq u_0(s_i)$, then $\mathbf{V}^{f_0, \ldots, f_k}(\alpha_1, \ldots, \alpha_{k+1}) = 0$. In other words, the verifier detects that s_i is a position where f_0 disagrees with the codeword u_0 (i.e., s_i is an erasure).*

[4] The notion of opening-consistency itself is already tied to the unique decoding radius.

[5] Note that this extractor is *straightline* by definition: there is no algorithm involved that it could rewind. It just gets an input and produces an output.

Proof. We first establish a number of useful facts and use them to prove the properties individually. Recall that Equation (9) relates the probability of the state changing in the query round to the size of AccQueries. Therefore, by assumption

$$\left(\frac{|\mathsf{AccQueries}|}{|\mathcal{L}_0|}\right)^t \geq (1-\delta^*)^t.$$

Since t is a positive integer, it holds that:

$$|\mathsf{AccQueries}| \geq (1-\delta^*) \cdot |\mathcal{L}_0|. \tag{10}$$

By assumption, we have that $\delta^* \leq 1-\rho$. Therefore, $1-\delta^* \geq \rho$. Substituting into Equation (10):

$$|\mathsf{AccQueries}| \geq (1-\delta^*) \cdot |\mathcal{L}_0| \geq \rho \cdot |\mathcal{L}_0|. \tag{11}$$

Similarly, it holds by assumption that $\delta^* \leq \theta^*$. Therefore,

$$|\mathsf{AccQueries}| \geq (1-\delta^*) \cdot |\mathcal{L}_0| \geq (1-\theta^*) \cdot |\mathcal{L}_0|. \tag{12}$$

For all $i \in [k]$, we let W_i be defined as in the proof of Theorem 5.2. We now show that,

$$\forall i \in [k], |W_i| \geq (1-\theta^*) \cdot |\mathcal{L}_i|. \tag{13}$$

The above follows from the fact that $W_0 = \mathsf{AccQueries}$ by definition, $|W_0| \geq (1-\theta^*) \cdot |\mathcal{L}_0|$ (as per Equation (12)) and by applying the contrapositive to Claim 5.4. In other words, since W_0 is large enough, it must be the case that all other W_i were large enough, otherwise red nodes would have propagated upwards. Finally, since Equation (13) is true, we can apply Claim 5.5. Therefore,

$$\exists u_0 \in \mathcal{C}_0, f_0[\mathsf{AccQueries}] = u_0[\mathsf{AccQueries}]. \tag{14}$$

Given all the numbered facts above, we prove the properties individually. In all of the analysis below, we let $u_0 \in \mathcal{C}_0$ be a codeword as in Equation (14).

- *Proximity.* By definition of the distance to a code, $\Delta(f_0, \mathcal{C}_0) \leq \Delta(f_0, u_0)$. Since f_0 and u_0 agree on the set AccQueries, it holds that $\Delta(f_0, u_0) \leq 1 - |\mathsf{AccQueries}|/|\mathcal{L}_0|$. From Equation (10), it holds that $1 - |\mathsf{AccQueries}|/|\mathcal{L}_0| \leq \delta^*$. Chaining these inequalities yields $\Delta(f_0, \mathcal{C}_0) \leq \delta^*$.

- *Binding.* To prove the binding property, it suffices to show that u_0 is unique. Consider $u' \in \mathcal{C}_0$ such that $f_0[\mathsf{AccQueries}] = u'[\mathsf{AccQueries}]$. We will show that $u_0 = u'$. Let $\widehat{u_0}, \widehat{u'} \in \mathbb{F}^{<d}[X]$ denote the polynomials of degree less than d that interpolate u_0 and u' respectively. It holds by definition of u_0 and u' that

  $$\forall x \in \mathsf{AccQueries}, \widehat{u_0}(x) = f_0(x) = \widehat{u'}(x).$$

  Furthermore, from Equation (11), it holds that $|\mathsf{AccQueries}|$ is at least $\rho \cdot |\mathcal{L}_0| = d$. Therefore, since $\widehat{u_0}$ and $\widehat{u'}$ agree on AccQueries, it holds that $\widehat{u_0} = \widehat{u'}$ as polynomials. We conclude that $u' = u_0$.

- *Extractability.* Given the above, extractability is simply proven by considering the following extractor $\mathbf{E}$: pick any d points in AccQueries and interpolate the unique polynomial of degree less than d that agrees with f_0 on those values. The codeword u_0 is obtained by evaluating this polynomial on $\mathcal{L}_0$. Notice that $\mathbf{E}$ does not run an error- or list-decoding algorithm.

- *Erasure detection.* Let $\alpha_{k+1} := (s_1, \ldots, s_t)$ be a final round verifier message. Recall that by definition of AccQueries, it follows that $\mathbf{V}^{f_0,\ldots,f_k}(\alpha_1, \ldots, \alpha_{k+1}) = 1$ if and only if for all $i \in [t]$ $s_i \in$ AccQueries. Furthermore, it holds that if $s_i \in$ AccQueries, then $u_0(s_i) = f_0(s_i)$. Therefore, if there exists s_i such that $u_0(s_i) \neq f_0(s_i)$, then $s_i \notin$ AccQueries. It follows that in such a case, $\mathbf{V}^{f_0,\ldots,f_k}(\alpha_1, \ldots, \alpha_{k+1}) = 0$.

□

5.5 Comparison to existing proof strategies

Let us put our proof into context by comparing it with other analyses.

Proofs using CA. The soundness of FRI is first shown in the eponymous paper [BBHR18]. The error bounds were later improved by Ben-Sasson, Kopparty and Saraf [BKS18] and Ben-Sasson *et al.* [BCI+20]. Our proof follows the ideas of Ben-Sasson *et al.* [BCI+20] and Augot, Bordage and Nardi [ABN23] but uses mutual correlated agreement (MCA) to analyze folding rounds. In contrast, our two references use a weaker notion known as *correlated agreement* (CA), or plain correlated agreement to avoid confusion. Their proofs achieve the same bounds as ours, however we consider their reasoning to be conceptually harder to grasp. The main difference lies in the guarantee that folding provides in the case when challenges are safe (*i.e.*, state remains 0):

- MCA: if all green subsets are large enough, then agreement propagates upwards through green nodes.
- CA: if all green subsets are large enough, then agreement propagates upwards through layers at unknown nodes.

The latter loses information about where the agreement is located. Furthermore, agreement may not be aligned with green nodes. Overcoming these hurdles requires the introduction of additional tools, notably a notion of *weighted* distance (rather than standard Hamming distance) and a correlated agreement theorem for this weighted distance. The ethSTARK documentation [Sta21] and the work by Block *et al.* [BGK+23] show that the analysis given by Ben-Sasson *et al.* [BCI+20] is in fact a proof of the round-by-round soundness of FRI.

Do we gain anything? One may wonder whether using MCA rather than CA allows us to prove that FRI has more bits of security per query. In our analysis, the answer is no. As stated above, our upper bounds on the round errors is identical to previous analyses. Based on toy problem analyses, it is unlikely that these bounds will be improved. As discussed in the ethSTARK documentation [Sta21], there exists an attack strategy that convinces the verifier with probability $(\max\{\rho, 1 - \delta\})^t$. The same term appears in our upper bound by setting $\theta \in [0, 1 - \rho]$ and assuming that the MCA error is 0 for $\theta \in [0, 1 - \rho]$. More recently, Arnon, Boneh and Fenzi [ABF26] present a toy protocol implicit in folding rounds and show that the soundness error for this toy protocol is lower bounded by the CA error. Therefore, unless there exists a gap between the CA error and MCA error, our analysis of the folding rounds matches this lower bound.

Despite not improving the soundness error, we prove that FRI satisfies stronger properties in Corollary 5.6. Our analysis is the first to show that FRI's folding phase (also known as the "commit" phase) is indeed a binding commitment to some Reed-Solomon codeword. A restricted version of this fact is implicitly shown in FRIDA [HASW24] to construct a code commitment from FRI.

Other proofs. The security of FRI is also discussed in works that introduce new frameworks as a means of illustrating the power of those frameworks. Attema, Fehr and Resch [AFR23] prove that an interactive proof derived from FRI (using Merkle commitments to replace the prover's oracles) meets their notion of generalized special soundness. Evans and Angeris [EA23] prove the security of a variant of FRI using only the tools of linear algebra. In both cases, the analyses provide looser bounds and are restricted to a smaller set of FRI parameters.

Acknowledgements

We thank Mark Simkin and Ignacio Manzur for early discussions on the topic. We also thank the 2025 cohort of the Ethereum Foundation's PQ Workshop for motivating the publication of this manuscript. Thank you to Ariel Gabizon and Yoichi Hirai for individually examining our proof and catching mistakes in an earlier draft of this work.

Use of AI Tools

We recognize and disclose the use of AI tools to generate diagrams and for occasional grammar checking. The rest of the work is entirely our own.

References

[ABF26] Gal Arnon, Dan Boneh, and Giacomo Fenzi. Open problems in list decoding and correlated agreement. Cryptology ePrint Archive, Paper 2026/680, 2026. URL: https://eprint.iacr.org/2026/680.

[ABN23] Daniel Augot, Sarah Bordage, and Jade Nardi. Efficient multivariate low-degree tests via interactive oracle proofs of proximity for polynomial codes. *DCC*, 91(3):1111–1151, 2023. doi:10.1007/s10623-022-01134-z.

[ACFY24] Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. STIR: Reed-solomon proximity testing with fewer queries. In Reyzin and Stebila [RS24], pages 380–413. doi:10.1007/978-3-031-68403-6_12.

[ACFY25] Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. WHIR: Reed-solomon proximity testing with super-fast verification. In Serge Fehr and Pierre-Alain Fouque, editors, *EUROCRYPT 2025, Part IV*, volume 15604 of *LNCS*, pages 214–243. Springer, Cham, May 2025. doi:10.1007/978-3-031-91134-7_8.

[ACY23] Gal Arnon, Alessandro Chiesa, and Eylon Yogev. IOPs with inverse polynomial soundness error. In *64th FOCS*, pages 752–761. IEEE Computer Society Press, November 2023. doi:10.1109/FOCS57990.2023.00050.

[AFR23] Thomas Attema, Serge Fehr, and Nicolas Resch. Generalized special-sound interactive proofs and their knowledge soundness. In Guy N. Rothblum and Hoeteck Wee, editors, *TCC 2023, Part III*, volume 14371 of *LNCS*, pages 424–454. Springer, Cham, November / December 2023. doi:10.1007/978-3-031-48621-0_15.

[BBHR18] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast reed-solomon interactive oracle proofs of proximity. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *ICALP 2018*, volume 107 of *LIPIcs*, pages 14:1–14:17. Schloss Dagstuhl, July 2018. doi:10.4230/LIPIcs.ICALP.2018.14.

[BCFW25] Benedikt Bünz, Alessandro Chiesa, Giacomo Fenzi, and William Wang. Linear-time accumulation schemes. In Benny Applebaum and Huijia (Rachel) Lin, editors, *TCC 2025, Part I*, volume 16268 of *LNCS*, pages 369–399. Springer, Cham, December 2025. doi:10.1007/978-3-032-12287-2_13.

[BCGM25] Sarah Bordage, Alessandro Chiesa, Ziyi Guan, and Ignacio Manzur. All polynomial generators preserve distance with mutual correlated agreement. Cryptology ePrint Archive, Report 2025/2051, 2025. URL: https://eprint.iacr.org/2025/2051.

[BCH+25] Eli Ben-Sasson, Dan Carmon, Ulrich Haböck, Swastik Kopparty, and Shubhangi Saraf. On proximity gaps for reed–solomon codes. Cryptology ePrint Archive, Report 2025/2055, 2025. URL: https://eprint.iacr.org/2025/2055.

[BCI+20] Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity gaps for reed-solomon codes. In *61st FOCS*, pages 900–909. IEEE Computer Society Press, November 2020. doi:10.1109/FOCS46700.2020.00088.

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In Martin Hirt and Adam D. Smith, editors, *TCC 2016-B, Part II*, volume 9986 of *LNCS*, pages 31–60. Springer, Berlin, Heidelberg, October / November 2016. doi:10.1007/978-3-662-53644-5_2.

[BGK+23] Alexander R. Block, Albert Garreta, Jonathan Katz, Justin Thaler, Pratyush Ranjan Tiwari, and Michal Zajac. Fiat-Shamir security of FRI and related SNARKs. In Jian Guo and Ron Steinfeld, editors, *ASIACRYPT 2023, Part II*, volume 14439 of *LNCS*, pages 3–40. Springer, Singapore, December 2023. doi:10.1007/978-981-99-8724-5_1.

[BKS18] Eli Ben-Sasson, Swastik Kopparty, and Shubhangi Saraf. Worst-case to average case reductions for the distance to a code. In Rocco A. Servedio, editor, *33rd Computational Complexity Conference, CCC 2018, San Diego, CA, USA, June 22-24, 2018*, volume 102 of *LIPIcs*, pages 24:1–24:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. URL: https://doi.org/10.4230/LIPIcs.CCC.2018.24, doi:10.4230/LIPICS.CCC.2018.24.

[BS05] Eli Ben-Sasson and Madhu Sudan. Simple PCPs with poly-log rate and query complexity. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 266–275. ACM Press, May 2005. doi:10.1145/1060590.1060631.

[CCH+19] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-Shamir: from practice to theory. In Moses Charikar and Edith Cohen, editors, *51st ACM STOC*, pages 1082–1090. ACM Press, June 2019. doi:10.1145/3313276.3316380.

[CMS19] Alessandro Chiesa, Peter Manohar, and Nicholas Spooner. Succinct arguments in the quantum random oracle model. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part II*, volume 11892 of *LNCS*, pages 1–29. Springer, Cham, December 2019. doi:10.1007/978-3-030-36033-7_1.

[CS25] Elizabeth Crites and Alistair Stewart. On reed–solomon proximity gaps conjectures. Cryptology ePrint Archive, Report 2025/2046, 2025. URL: https://eprint.iacr.org/2025/2046.

[DHT+26] Quang Dao, Alexander Hicks, Devon Tuma, Julian Sutherland, Katerina Hristova, Frantisek Silvasi, Ilia Vlasov, and Chung Thai Nguyen. Compositional formal verification of SNARKs with ArkLib. Real World Crypto 2026. To appear., 2026.

[EA23] Alex Evans and Guillermo Angeris. Succinct proofs and linear algebra. Cryptology ePrint Archive, Report 2023/1478, 2023. URL: https://eprint.iacr.org/2023/1478.

[Eth25] Ethereum Foundation. zkEVM formal verification project. Online, https://verified-zkevm.org/, 2025.

[GG25] Rohan Goyal and Venkatesan Guruswami. Optimal proximity gaps for subspace-design codes and (random) reed-solomon codes. Cryptology ePrint Archive, Report 2025/2054, 2025. URL: https://eprint.iacr.org/2025/2054.

[Hab25] Ulrich Haböck. A note on mutual correlated agreement for reed-solomon codes. Cryptology ePrint Archive, Report 2025/2110, 2025. URL: https://eprint.iacr.org/2025/2110.

[HASW24] Mathias Hall-Andersen, Mark Simkin, and Benedikt Wagner. FRIDA: Data availability sampling from FRI. In Leonid Reyzin and Douglas Stebila, editors, *CRYPTO 2024, Part VI*, volume 14925 of *LNCS*, pages 289–324. Springer, Cham, August 2024. doi:10.1007/978-3-031-68391-6_9.

[Hir26] Yoichi Hirai. Lean formalization of "a simplified round-by-round soundness proof of fri". Github https://github.com/zksecurity/simple-rbr-fri, 2026.

[Hol19] Justin Holmgren. On round-by-round soundness and state restoration attacks. Cryptology ePrint Archive, Report 2019/1261, 2019. URL: https://eprint.iacr.org/2019/1261.

[RS24] Leonid Reyzin and Douglas Stebila, editors. *CRYPTO 2024, Part X*, volume 14929 of *LNCS*. Springer, Cham, August 2024.

[Sta21] StarkWare. ethSTARK documentation. Cryptology ePrint Archive, Report 2021/582, 2021. URL: https://eprint.iacr.org/2021/582.

[Whi25a] Whiteboard Sessions and Dan Boneh. Zk whiteboard sessions – S2M7: FRI and proximity proofs (part 1). https://www.youtube.com/watch?v=MBDBrEr2XQg, 2025. YouTube video, accessed 10 October 2025.

[Whi25b] Whiteboard Sessions and Dan Boneh. Zk whiteboard sessions – S2M8: FRI and proximity proofs (part 2). https://www.youtube.com/watch?v=CWbx_rnj7LI, 2025. YouTube video, accessed 10 October 2025.

[ZCF24] Hadas Zeilberger, Binyi Chen, and Ben Fisch. BaseFold: Efficient field-agnostic polynomial commitment schemes from foldable codes. In Reyzin and Stebila [RS24], pages 138–169. doi:10.1007/978-3-031-68403-6_5.

[Zei24] Hadas Zeilberger. Khatam: Reducing the communication complexity of code-based SNARKs. Cryptology ePrint Archive, Report 2024/1843, 2024. URL: https://eprint.iacr.org/2024/1843.

A Appendix: Batched FRI

In practice, we are often interested in simultaneously testing the proximity of multiple initial functions $g_1, \ldots, g_b \colon \mathcal{L}_0 \to \mathbb{F}$ to the RS code $\mathcal{C}_0$. FRI can be extended to this effect using an additional round of batching [BCI+20]. The overall protocol, which we refer to as *batched FRI*, is in fact an IOPP for the *interleaved Reed-Solomon code* $\mathcal{C}_0^{\equiv b}$ as defined below.

Interleaved codes. Consider a Reed-Solomon code $\mathsf{RS}[\mathbb{F}, \mathcal{L}, d]$ as in Definition 2.1 and an interleaving factor $b \in \mathbb{N}$. We define the b-interleaved Reed-Solomon code $\mathsf{RS}[\mathbb{F}, \mathcal{L}, d]^{\equiv b}$ as

$$\mathsf{RS}[\mathbb{F}, \mathcal{L}, d]^{\equiv b} := \{(g_1, \ldots, g_b) : \forall i \in [b], g_i \in \mathsf{RS}[\mathbb{F}, \mathcal{L}, d]\}.$$

We write $\mathbf{g} := (g_1, \ldots, g_b)$ and treat the tuple as a function $\mathbf{g} : \mathcal{L} \to \mathbb{F}^b$, where for all $x \in \mathcal{L}$, $\mathbf{g}(x) := (g_1(x), \ldots, g_b(x))$. The fractional Hamming distance between tuples $\mathbf{g}$ and $\mathbf{h}$ and between the tuple $\mathbf{g}$ and a set $S \subseteq \mathbb{F}^b$ are defined with respect to this tuple-as-a-function view. The rate ρ of the interleaved code is identical to the rate of the underlying RS code and is computed as $d/|\mathcal{L}|$. Likewise, its minimum distance is also $1 - \rho$.

Construction A.1 (Batched FRI [BCI+20])**.** *Consider the same initial setting and protocol parameters as in Construction 4.4, and let* $\mathbf{P}_{\mathsf{FRI}}$, $\mathbf{V}_{\mathsf{FRI}}$ *respectively denote the FRI prover and FRI verifier. Additionally, consider a batch size* $b \in \mathbb{N}$ *and proximity generator* Gen *that takes as input a length parameter* $n \in \mathbb{N}$ *and randomness* $\mathsf{s} \in \mathsf{S}$*, for some set* S*, and outputs* $(\beta_1, \ldots, \beta_n) \in \mathbb{F}^n$*. The batched FRI protocol proceeds as follows:*

- ***Initial functions.*** *Let* $g_1, \ldots, g_b \colon \mathcal{L}_0 \to \mathbb{F}$ *be initial functions and write* $\mathbf{g} := (g_1, \ldots, g_b)$*. In the honest case,* $\mathbf{g} \in \mathcal{C}_0^{\equiv b}$*.*
- ***Batching phase:***
  1. ***Batching challenge.*** *The verifier sends a batching challenge* $\mathsf{s} \leftarrow \mathsf{S}$ *and computes* $(\beta_1, \ldots, \beta_b) := \mathsf{Gen}(b; \mathsf{s})$*.*
  2. ***Send batched function.*** *The prover sends a function* $f_0 : \mathcal{L}_0 \to \mathbb{F}$*. In the honest case,* $f_0 := \sum_{i=1}^{b} \beta_i \cdot g_i$*.*
- ***FRI interactive phase:***
  1. ***Folding phase.*** *The prover and verifier run the FRI folding phase with initial function* f_0*, producing challenges* $\alpha_1, \ldots, \alpha_k$ *and functions* $f_1, \ldots, f_k$*. In the honest case, the prover runs the honest FRI prover* $\mathbf{P}_{\mathsf{FRI}}$*.*
  2. ***Query phase.*** *The prover and verifier run the FRI query phase, producing a query challenge* $\alpha_{k+1} := (s_1, \ldots, s_t)$*.*
- ***Verifier decision phase:***
  1. ***FRI decision phase.*** *Assert that* $\mathbf{V}_{\mathsf{FRI}}^{f_0, \ldots, f_k}(\alpha_1, \ldots, \alpha_{k+1}) = 0$*.*
  2. ***Local batching check.*** *For every repetition* $j = 1, \ldots, t$ *done in the verifier call* $\mathbf{V}_{\mathsf{FRI}}$ *above, the verifier additionally queries* $\mathbf{g}(s_j)$ *and asserts that* $f_0(s_j) = \sum_{i=1}^{b} \beta_i \cdot g_i(s_j)$*.*

Round-by-round soundness. We treat batched FRI as an IOPP for the interleaved RS code and show that it has round-by-round soundness.

Theorem A.2. *Consider batched FRI with parameters as in Construction A.1, and assume* Gen *is a proximity generator with mutual correlated agreement with error* $\varepsilon^{\mathsf{mca}}_{\mathsf{Gen}}$ *and* Powers *is a proximity generator with mutual correlated agreement with error* $\varepsilon^{\mathsf{mca}}_{\mathsf{Powers}}$*. For every tradeoff parameter* $\theta \in [0,1]$*, batched FRI has round-by-round soundness with error functions* $\varepsilon^{\mathsf{batch}}, \varepsilon^{\mathsf{fold}}_1, \ldots, \varepsilon^{\mathsf{fold}}_k$*,* $\varepsilon^{\mathsf{qry}} \colon (0,1] \to [0,1]$ *such that for every initial input* $\mathbf{g} \notin {\mathcal{C}_0}^{\equiv b}$*:*

- $\varepsilon^{\mathsf{batch}}(\delta) = \varepsilon^{\mathsf{mca}}_{\mathsf{Gen}}(\mathcal{C}_0, b, \theta)$,
- *for all* $i \in [k]$, $\varepsilon^{\mathsf{fold}}_i(\delta) = \varepsilon^{\mathsf{mca}}_{\mathsf{Powers}}(\mathcal{C}_i, m_i, \theta)$,
- $\varepsilon^{\mathsf{qry}}(\delta) = (\max\{1-\theta, 1-\delta\})^t = (1 - \min\{\theta, \delta\})^t$,

where $\delta := \Delta\left(\mathbf{g}, {\mathcal{C}_0}^{\equiv b}\right)$.

Proof. The proof is almost identical to the proof of Theorem 5.2. To extend the proof, we must define the state function for the batching phase and use the *local batching check* to argue that the conclusions established on f_0 translate to meaningful conclusions about $g_1, \ldots, g_b$.

0. **Empty transcript.** For an empty transcript, we say that $\mathsf{State}_\theta(\mathbf{g}, \emptyset) = 1$ if and only if $\mathbf{g} \in {\mathcal{C}_0}^{\equiv b}$.

1. **Batching phase.** At this stage, the transcript tr is empty and the verifier sends a batching challenge s. As in the protocol description, we let $(\beta_1, \ldots, \beta_b) := \mathsf{Gen}(\mathsf{s})$.

   - *State function.* Suppose that $\mathsf{State}_\theta(\mathbf{g}, \emptyset) = 0$. We say that $\mathsf{State}_\theta(\mathbf{g}, \mathsf{s}) = 1$ if and only if there exists a set $S \subseteq \mathcal{L}_0$ such that:
     (a) $|S| \geq (1-\theta) \cdot |\mathcal{L}_0|$; and
     (b) there exists $u' \in \mathcal{C}_0$ such that $\sum_{i=1}^{b} \beta_i g_i[S] = u'[S]$; and
     (c) there exists $i \in [b]$ such that for all $u \in \mathcal{C}_0$, $u[S] \neq g_i[S]$.
   - *Bounding the error.* By definition of mutual correlated agreement:

     $$\Pr_{\mathsf{s}}[\mathsf{State}_\theta(\mathbf{g}, \emptyset) = 0 \ \wedge \ \mathsf{State}_\theta(\mathbf{g}, \mathsf{s}) = 1] \leq \varepsilon^{\mathsf{mca}}_{\mathsf{Gen}}(\mathcal{C}_0, b, \theta).$$

     Setting $\varepsilon^{\mathsf{batch}}(\delta) := \varepsilon^{\mathsf{mca}}_{\mathsf{Gen}}(\mathcal{C}_0, b, \theta)$ for all $\delta \in [0,1]$ completes the proof for the batching phase.

2. **FRI folding.** Let $\mathsf{State}^{\mathsf{FRI}}_\theta$ denote the FRI state function as in Theorem 5.2 and define f_0 as in Construction A.1. For every folding round $i \in [k]$, the batched FRI transcript has the form

   $$\mathsf{tr} := (\mathsf{s}, f_0, \mathsf{tr}'),$$

   where tr' is a FRI transcript for the i-th folding round as in Equation (8). The verifier then sends a message $\alpha_i \in \mathbb{F}$.

   - *State function.* For every folding round $i \in [k]$, consider an initial input $\mathbf{g}$ and transcript $\mathsf{tr} := (\mathsf{s}, f_0, \mathsf{tr}')$ such that $\mathsf{State}_\theta(\mathbf{g}, \mathsf{tr}) = 0$. Since we are running FRI as a subroutine, we say that $\mathsf{State}_\theta(\mathbf{g}, \mathsf{tr}||\alpha_i) = 1$ if and only if both of the following conditions are satisfied:
     (a) $\mathsf{State}^{\mathsf{FRI}}_\theta(f_0, \mathsf{tr}') = 0$; and
     (b) $\mathsf{State}^{\mathsf{FRI}}_\theta(f_0, \mathsf{tr}'||\alpha_i) = 1$.
   - *Bounding the error.* For every folding round $i \in [k]$, consider an initial input $\mathbf{g}$ and transcript $\mathsf{tr} := (\mathsf{s}, f_0, \mathsf{tr}')$ such that $\mathsf{State}_\theta(\mathbf{g}, \mathsf{tr}) = 0$. By definition of the state function, it holds that

     $$\Pr_{\alpha_i}\left[\begin{array}{c} \mathsf{State}_\theta(\mathbf{g}, \mathsf{tr}) = 0 \\ \wedge \\ \mathsf{State}_\theta(\mathbf{g}, \mathsf{tr}||\alpha_i) = 1 \end{array}\right] = \Pr_{\alpha_i}\left[\begin{array}{c} \mathsf{State}^{\mathsf{FRI}}_\theta(f_0, \mathsf{tr}') = 0 \\ \wedge \\ \mathsf{State}^{\mathsf{FRI}}_\theta(f_0, \mathsf{tr}'||\alpha_i) = 1 \end{array}\right].$$

By our analysis of the folding rounds of FRI, it follows that for all $i \in [k]$, the batched FRI error for folding rounds is $\varepsilon_i^{\mathsf{fold}}(\delta) = \varepsilon_{\mathsf{Powers}}^{\mathsf{mca}}(\mathcal{C}_i, m_i, \theta)$.

3. **FRI query.** As above, let $\mathsf{State}_\theta^{\mathsf{FRI}}$ denote the FRI state function as in Theorem 5.2. A final round partial transcript for batched FRI has the form

$$\mathsf{tr} := (\mathsf{s}, f_0, \mathsf{tr}'),$$

where tr' is a final round partial transcript for the FRI protocol. The final verifier message, denoted α_{k+1} to remain consistent with our FRI analysis, has the form $\alpha_{k+1} = (s_1, \ldots, s_t)$ with $s_1, \ldots, s_t \in \mathcal{L}_0$.

- *State function.* Suppose that $\mathsf{State}_\theta(\mathbf{g}, \mathsf{tr}) = 0$. We say that $\mathsf{State}_\theta(\mathbf{g}, \mathsf{tr}||\alpha_{k+1}) = 1$ if and only if both of the following conditions are satisfied:
  (a) the FRI state function is satisfied $\mathsf{State}_\theta^{\mathsf{FRI}}(f_0, \mathsf{tr}'||\alpha_{k+1}) = 1$, and
  (b) for all $j \in [t]$, $f_0(s_j) = \sum_{i=1}^{b} \beta_i \cdot g_i(s_j)$.
- *Bounding the error.* Suppose that $\mathsf{State}_\theta(\mathbf{g}, \mathsf{tr}) = 0$. The argument to bound the query round error is a simple extension of the argument for non-batched FRI. Recall that we define the set of accepting queries $\mathsf{AccQueries}$ and show that the probability of changing states is exactly $(|\mathsf{AccQueries}|/|\mathcal{L}_0|)^t$ (see Equation (9)). In batched FRI, the verifier also performs a *local batching check* (as reflect by Item 3b above). Therefore, we additionally note that

$$f_0[\mathsf{AccQueries}] = \sum_{i=1}^{b} \beta_i \cdot g_i[\mathsf{AccQueries}]. \tag{15}$$

  The remainder of the proof gives an upper bound for $|\mathsf{AccQueries}|/|\mathcal{L}_0|$. Recall that we define the set $W_0 := \mathsf{AccQueries}$ and then inductively define W_i for all $i \in [k]$. In our regular FRI analysis, we distinguish two cases:
  - in the first case, there exists $i \in \{0, \ldots, k\}$ such that $|W_i| < (1-\theta) \cdot |\mathcal{L}_i|$. By Claim 5.4, we can upper bound $|\mathsf{AccQueries}|/|\mathcal{L}_0|$ by $1 - \theta$. All this analysis carries over to the batched setting identically.
  - in the second case, all sets are "large enough". Claim 5.5 establishes that there exists a codeword $u_0 \in \mathcal{C}_0$ such that $u_0[W_0] = f_0[W_0]$. We now extend this claim to $\mathbf{g}$ in Claim A.3. Given this claim, we can apply the same reasoning as in case 2 of the regular FRI analysis to establish that $|\mathsf{AccQueries}|/|\mathcal{L}_0| \leq \Delta\left(\mathbf{g}, \mathcal{C}_0^{\equiv b}\right)$.

  Claim A.3. *If $f_k \in \mathcal{C}_k$ and for all $i \in \{0, \ldots, k\}$, $|W_i| \geq (1-\theta) \cdot |\mathcal{L}_i|$, then there exists $\mathbf{v} \in \mathcal{C}_0^{\equiv b}$ such that $\mathbf{v}[W_0] = \mathbf{g}[W_0]$.*

  Proof. Note first that the premise for the above claim is identical to the premise of Claim 5.5. Therefore, by that same claim, it holds that there exists $u_0 \in \mathcal{C}_0$ such that

$$u_0[W_0] = f_0[W_0].$$

  We will now show that this observation extends to $\mathbf{g}$. First, note that $|W_0| \geq (1-\theta) \cdot |\mathcal{L}_0|$ and therefore Item 1a of the batching phase state function is satisfied for the set W_0. Secondly, it holds by Equation (15) and the definition of W_0 that

$$u_0[W_0] = f_0[W_0] = \sum_{i=1}^{b} \beta_i \cdot g_i[W_0].$$

Therefore Item 1b of the batching phase state function is satisfied for the set W_0.

Since $\mathsf{State}_\theta(\mathbf{g}, \mathsf{tr}) = 0$, it holds by the *Default* property of state functions (Definition 3.4) that $\mathsf{State}_\theta(\mathbf{g}, \mathsf{s}) = 0$. Therefore, it must be the case that Item 1c is false (otherwise all three items would be satisfied and $\mathsf{State}_\theta(\mathbf{g}, \mathsf{s}) = 1$, a contradiction). We therefore conclude that for all $i \in [b]$, there exists a codeword $v_i \in \mathcal{C}_0$ such that $v_i[W_0] = g_i[W_0]$. Let $\mathbf{v} = (v_1, \ldots, v_b)$. By definition of interleaved RS codes we have that $\mathbf{v}[W_0] = \mathbf{g}[W_0]$, which concludes our proof of Claim A.3. □

To conclude our proof for the query phase, we upper bound the query round error by taking an upper bound on $|\mathsf{AccQueries}|/|\mathcal{L}_0|$ that is consistent with both of the cases presented above

$$\Pr_{\alpha_{k+1}}\left[\begin{array}{c}\mathsf{State}_\theta(\mathbf{g}, \mathsf{tr}) = 0 \\ \wedge \\ \mathsf{State}_\theta(\mathbf{g}, \mathsf{tr}||\alpha_{k+1}) = 1\end{array}\right] \leq (\max\{1-\theta, 1-\delta\})^t,$$

where $\delta := \Delta\left(\mathbf{g}, \mathcal{C}_0^{\equiv b}\right)$.

4. **Verifier decision.** A full transcript is of the form $\mathsf{tr} = (\mathsf{s}, f_0, \mathsf{tr}')$, where $\mathsf{tr}' := (\alpha_1, f_1, \ldots, \alpha_k, f_k, \alpha_{k+1})$ is a complete FRI transcript. If we have $\mathsf{State}_\theta(f_0, \mathsf{tr}) = 0$, then either Item 3a or Item 3b is false. These conditions are identical to the checks performed by the verifier, namely running the FRI decision phase for (f_0, tr') and the *local batching check*. Therefore, we have

$$\mathbf{V}^{\mathbf{g}, f_0, \ldots, f_k}(\mathsf{s}, \alpha_1, \ldots, \alpha_{k+1}) = 0$$

as desired.

□

Additional properties. As was the case for FRI, we can prove that batched FRI upholds additional properties.

Corollary A.4. *Consider batched FRI with parameters as in Construction A.1, a fixed tradeoff parameter $\theta^* \in [0, 1]$ and let $\mathsf{State}_{\theta^*}$ be the state function defined in the proof of Theorem A.2.*

For every distance parameter $\delta^* \in [0, 1-\rho] \cap [0, \theta^*]$, if $\mathbf{g} := (g_1, \ldots, g_b)$ is an initial function and $\mathsf{tr} := (\mathsf{s}, f_0, \alpha_1, f_1, \ldots, \alpha_k, f_k)$ is a transcript such that

$$\mathsf{State}_{\theta^*}(\mathbf{g}, \mathsf{tr}) = 0 \ \wedge \ \Pr_{\alpha_{k+1}}[\mathsf{State}_{\theta^*}(\mathbf{g}, \mathsf{tr}||\alpha_{k+1}) = 1] \geq (1-\delta^*)^t$$

then the following properties are true:

1. ***Proximity**: $\mathbf{g}$ is at least δ^*-close to the code $\mathcal{C}_0^{\equiv b}$, i.e., $\Delta(\mathbf{g}, \mathcal{C}_0^{\equiv b}) \leq \delta^*$.*
2. ***Binding**: there exists a unique codeword $\mathbf{v} \in \mathcal{C}_0^{\equiv b}$ such that $\mathbf{v}[\mathsf{AccQueries}] = \mathbf{g}[\mathsf{AccQueries}]$, where $\mathsf{AccQueries} \subseteq \mathcal{L}_0$ is the set of accepting queries for the pair $(\mathbf{g}, \mathsf{tr})$.*
3. ***Extractability**: there exists a deterministic polynomial-time extractor[6] $\mathbf{E}$ that, given $(\mathbf{g}, \mathsf{tr})$ as input, returns $\mathbf{v}$ as defined in the binding property above.*
4. ***Erasure detection**: let $\mathbf{v}$ be defined as in the binding property and $\alpha_{k+1} := (s_1, \ldots, s_t)$ be a final round verifier message. If there exists $i \in [t]$ such that $\mathbf{g}(s_i) \neq \mathbf{v}(s_i)$, then $\mathbf{V}^{\mathbf{g}, f_1, \ldots, f_k}(\mathsf{s}, \alpha_1, \ldots, \alpha_{k+1}) = 0$. In other words, the verifier detects that s_i is a position where $\mathbf{g}$ disagrees with the codeword $\mathbf{v}$ (i.e., s_i is an erasure).*

[6]Note that this extractor is *straightline* by definition: there is no algorithm involved that it could rewind. It just gets an input and produces an output.

Proof. Corollary A.4 follows from Theorem A.2 in the same way Corollary 5.6 follows from Theorem 5.2, with the exception that we will also need to use Claim A.3.

Firstly, note that the results of Equations (10) to (13) all carry over identically in the interleaved setting. Similarly to Equation (14), we use Equation (13) and Claim A.3 to establish that

$$\exists \mathbf{v} \in {\mathcal{C}_0}^{\equiv b}, \mathbf{g}[\mathsf{AccQueries}] = \mathbf{v}[\mathsf{AccQueries}]. \tag{16}$$

Given these facts, the additional properties follow naturally:

- *Proximity.* Proximity in the interleaved setting follows from Equation (10) identically to the proof of Corollary 5.6.
- *Binding.* The reasoning here extends the reasoning in Corollary 5.6 to account for the fact that an interleaved codeword is defined by b polynomials. We copy the proof almost verbatim.

  > To prove the binding property, it suffices to show that $\mathbf{v}$ from Equation (16) is unique. Consider $\mathbf{v}' \in {\mathcal{C}_0}^{\equiv b}$ such that $\mathbf{g}[\mathsf{AccQueries}] = \mathbf{v}'[\mathsf{AccQueries}]$. We will show that $\mathbf{v} = \mathbf{v}'$. Let $v_1, \ldots, v_b$ and $v'_1, \ldots, v'_b$ be the individual constituents of $\mathbf{v}$ and $\mathbf{v}'$ respectively. Fix an arbitrary $i \in [b]$ and let $\widehat{v_i}, \widehat{v'_i} \in \mathbb{F}^{<d}[X]$ denote the polynomials of degree less than d that interpolate v_i and v'_i respectively. It holds by definition of v_i and v'_i that
  >
  > $$\forall x \in \mathsf{AccQueries}, \widehat{v_i}(x) = g_i(x) = \widehat{v'_i}(x).$$
  >
  > Furthermore, from Equation (11), it holds that $|\mathsf{AccQueries}|$ is at least $\rho \cdot |\mathcal{L}_0| = d$. Therefore, since $\widehat{v_i}$ and $\widehat{v'_i}$ agree on $\mathsf{AccQueries}$, it holds that $\widehat{v_i} = \widehat{v'_i}$ as polynomials and $v_i = v'_i$. Since i was arbitrary, the above applies for all $i \in [b]$; we therefore conclude that $\mathbf{v} = \mathbf{v}'$.

- *Extractability.* Extractability follows by interpolation, as in Corollary 5.6. Specifically, $\mathbf{E}$ picks any d points in $\mathsf{AccQueries}$ and for all $i \in [b]$ interpolates the unique polynomial $\widehat{v_i}$ of degree less than d that agrees with g_i on those values. The codeword $\mathbf{v} := (v_1, \ldots, v_b)$ is obtained by evaluating the polynomials $\widehat{v_1}, \ldots, \widehat{v_b}$ on $\mathcal{L}_0$. Notice that $\mathbf{E}$ does not run an error- or list-decoding algorithm.
- *Erasure detection.* Erasure detection follows as in Corollary 5.6, by noting that if there is a query s_i such that $\mathbf{v}(s_i) \neq \mathbf{g}(s_i)$, then $s_i \notin \mathsf{AccQueries}$ and therefore the verifier rejects.

□