DEEP Commitments and Their Applications

Alan Szepieniec
alan@neptune.cash

Neptune Cash

Abstract. This note studies a method of committing to a polynomial in a way that allows executions of low degree tests such as FRI to be batched and even deferred. In particular, it achieves (unlimited-depth) aggregation for STARKs.

1 Introduction

In the context of proof-carrying data (PCD) protocols [7] relying on STARKs [3] as the underlying succinct proof system, it is often necessary to prove the correct verification of two or more STARK proofs. In order to minimize the complexity of this task, the FRI [2] low degree test stands out as an obvious candidate for optimization. It represents a conceptually separate, almost detached, step in the STARK workflow, and accounts for roughly half the row count in the recursive execution trace as well as half the proof size. This situation begs the question: *can the FRI steps be merged?*

For context, if the recursive prover *only* has access to succinct proofs then there is little he can do beyond verify each one independently. But there might be reasons why the recursive prover has access to some witness information in addition to the proofs, for instance because he is the one who produced them in the first place; or because he acquired them from the person who did. Either way, the desired end-result is a single proof attesting to the validity of both claims simultaneously. The question is whether the recursive prover can produce it more efficiently when armed with associated witness data, by batching the FRI steps.

The naïve approach involves ignoring all but the initial codeword-commitments in the transcripts of the original FRI steps. After batching the codewords, *a single* execution of FRI suffices to establish that all operands were of low degree. This strategy works in principle but comes with a huge memory cost. In the last step of the single execution of FRI it produces a set of indices where the codewords' consistency is to be tested. The prover must supply the values of *all* codewords in these locations — including the codewords comprising the low-degree-extended algebraic execution trace. In order to answer these queries efficiently, the prover has to store the entire low-degree-extended algebraic execution trace of all proofs that are being merged. The memory cost of this task is prohibitive: the number of columns in the algebraic execution trace is typically on the order of hundreds if not thousands.

Contributions. This note answers the motivating question positively. We circumvent the obstacle posed by the naive approach by decoupling the FRI step from the preceding steps. Our technique reduces an algebraic execution trace to a single polynomial commitment in a way that can be verified independently from a possible follow-up low degree test. The immediate implication is that a single polynomial, along with some supplementary commitment information, suffices as the witness to a polynomial commitment, as opposed to the entire algebraic execution trace. This difference results in a factor 100-1000 reduction in the memory cost of the now-not-so-naïve approach.

A second implication is the capacity to defer low degree tests on committed polynomials. Commitments can be merged — again independently from a follow-up low degree test — and perhaps merged again and again — before finally being subjected to a low degree test. In other words, our construction directly leads to an accumulation scheme [11] (AKA aggregation scheme [8]) but with the caveat that every commitment needs to circulate with an associated witness. To reiterate, this witness is essentially a single polynomial and not the entire low degree extended trace. Furthermore, with techniques already used to achieve zero-knowledge [16], these "witnesses" leak no information about the traces from which they originate, whether immediately or any number of hops in the past.

Technical Overview. The main technique is inspired by the STIR low degree test [1] and uses steps from there to achieve similar effects. Specifically, an *interactive* commitment to a polynomial is given by

- $h : D \to \mathbb{F}$, the (Merkle root of the) polynomial's Reed-Solomon codeword on a domain D;
- $z \in \mathbb{F}$, the out-of-domain indeterminate sampled by the verifier;
- $y \in \mathbb{F}$, the out-of-domain value provided by the prover such that $g(z) = y$;
- $x_0, \ldots, x_{t-1} \in D$, t-many in-domain indeterminates sampled unformly at random from the code's domain D by the verifier; and
- $\gamma \in \mathbb{F}$ a weight for degree correction, sampled by the verifier.

The codewords h represents a random linear combination of terms that can have one of two origins. First, the terms can originate from an algebraic execution trace with quotients, and in this configuration the integrity of the algebraic execution trace is reduced to a polynomial commitment. Second, the terms can originate from the merger of two or more other polynomial commitments, and in this configuration multiple polynomial commitments are reduced to one. The commitments resulting from either of these procedures gives rise to a polynomial (in the honest case)

$$q(X) = \frac{\gamma^{t+1} \cdot X^{t+1} - 1}{\gamma \cdot X - 1} \cdot \frac{h(X) - \mathsf{Ans}(X)}{Z(X)} \tag{1}$$

where $\mathsf{Ans}(X)$ is the interpolant through $\{(x_0, h(x_0)), \ldots, (x_{t-1}, h(x_{t-1})), (z, y)\}$, and $Z(X)$ is the matching vanishing polynomial. The special features of q are two-fold. First, the low degree of $q(X)$ certifies the integrity of all algebraic

execution traces that gave rise to it. Second, the summands of h can be dropped after the correct derivation of the entire commitment is checked. Specifically, a commitment whose derivation-check fails, generates a derived function $q : D \to \mathbb{F}$ that is overwhelmingly likely to be rejected by any follow-up low degree test.

Related Work. The general principle of defferring the expensive cryptographic step to the end of a scheme involving recursive proofs was first proposed by Halo [9]. The concept was later formally studied and encapsulated as *aggregation* [8] and (independently) *accumulation schemes* [11]. While the object of these formal investigations were homomorphic polynomial commitments without witnesses, the former work [8] did explore the possibility of relaxing the homomorphism requirement and requiring knowledge of a witness of some sort.

More recently, Bünz *et al.* [12] focus on achieving accumulation schemes from proof systems whose commitments are not homomorphic — *i.e.*, exactly our setting. Their construction, inspired by ProtoStar [10] and ProtoGalaxy [14], involves tracking an error term and verifying its correct update at each accumulation step. It has the drawback that the quality of an accumulator degrades with the circuit depth, and ultimately they only achieve *bounded-depth PCD*. Our construction is entirely different and does not have this deficiency.

2 Preliminaries

2.1 Proximity Gaps and Correlated Agreement

Definition 1. *A linear code $V \subset \mathbb{F}^n$ of length n and dimension k exhibits a* (ϵ, δ) proximity gap *if for every affine line in the ambient space, either all points are δ-close to the code or at most a fraction ϵ of them are. Symbolically:*

$$\forall u_0, u_1 \in \mathbb{F}^n \,.\, \Pr_{r \xleftarrow{\$} \mathbb{F}} \left[\Delta(r \cdot u_0 + (1-r) \cdot u_1, V) \leq \delta\right] \notin (\epsilon; 1) \ . \tag{2}$$

BCIKS [4] represents the state of the art in terms of proximity gaps for Reed Solomon codes. In summary:

- $0 \leq \delta < \frac{1-\rho}{2}$. Within the unique decoding radius: $\epsilon = \frac{n}{|\mathbb{F}|}$.
- $\frac{1-\rho}{2} \leq \delta < 1 - \sqrt{\rho}$. Up to the Johnson bound: $\epsilon = \frac{(k+1)^2}{2 \cdot \min\left(1 - \sqrt{(\rho)} - \delta, \frac{\sqrt{\rho}}{20}\right)^7 \cdot |\mathbb{F}|}$.
- $1 - \sqrt{\rho} \leq \delta < 1 - \rho$. Beyond the Johnson bound no gaps are proven. However, it is conjectured that constants c_1, c_2 exist such that $\epsilon \leq \frac{1}{((1-\rho-\delta)\cdot\rho)^{c_1}} \cdot \frac{n^{c_2}}{|\mathbb{F}|}$.

The following lemma captures the soundness error of a recurring technique, that of reducing a batch of words to one random linear combination. It follows immediately from a (δ, ϵ) proximity gap.

Lemma 1 (linear batching). *Let $u_0, \ldots, u_{t-1} \in \mathbb{F}^n$ be vectors in the ambient space of a code $V \subseteq \mathbb{F}^n$ that exhibits a (δ, ϵ) proximity gap. Then*

$$\Pr_{r_1, \ldots, r_{t-1} \xleftarrow{\$} \mathbb{F}} \left[\Delta\left(u_0 + \sum_{i=1}^{t-1} r_i \cdot u_i, V\right) \leq \delta\right] \notin (\epsilon; 1) \ . \tag{3}$$

Another technique is to use the powers of just one random challenge. The next lemma captures the soundness error of this technique.

Lemma 2 (univariate batching). *Let $u_0, \ldots, u_{t-1} \in \mathbb{F}^n$ be vectors in the ambient space of a code $V \subseteq \mathbb{F}^n$ that exhibits a (δ, ϵ) proximity gap. Then*

$$\Pr_{r \xleftarrow{\$} \mathbb{F}} \left[\Delta \left(\sum_{i=0}^{t-1} r^i \cdot u_i, V \right) \leq \delta \right] \notin (t \cdot \epsilon; 1) \ . \tag{4}$$

In addition to linear batching and univariate batching, *multilinear batching* [13] has been studied. This alternative induces less soundness degradation relative to univariate batching and requires less randomness than linear batching.

Definition 2 (correlated agreement). *A linear code $V \subset \mathbb{F}^n$ of length n and dimension k exhibits* (ϵ, δ) correlated agreement *if for all affine lines of which the proportion that is δ-close to the code is larger than ϵ, there is a common index set $\mathcal{I} \subset \{0, \ldots, n-1\}$ with $\frac{|\mathcal{I}|}{n} = 1 - \delta$ where all points on the line agree with some codeword.*

Correlated agreement is a stronger notion which implies proximity gaps. The converse is not known to hold. BCIKS [4] proves the existence of proximity gaps for Reed-Solomon by proving correlated agreement. The conjecture beyond the Johnson bound is technically two conjectures, since *a priori* it is possible that proximity gaps do exist in this regime while correlated agreement does not.

2.2 Out-of-Domain Evaluation

Lemma 3 ([1], Lemma 4.5). *Let $g : D \to \mathbb{F}$ be a function, $d \in \mathbb{N}$ be a degree parameter, $s \in \mathbb{N}$ a repetition parameter, and $\delta \in [0; 1]$ be a distance parameter. If $\mathsf{RS}[\mathbb{F}, D, d]$ is (δ, ℓ)-list decodable then*

$$\Pr_{\boldsymbol{r} \in (\mathbb{F} \setminus D)^s} [\exists u \neq u' \in \mathsf{List}(f, d, \delta), \forall i \in [0 : s), \hat{u}(r_i) = \hat{u}'(r_i)] \leq \frac{\ell^2}{2} \cdot \left(\frac{d}{|\mathbb{F}| - |D|} \right)^s \ . \tag{5}$$

The value of ℓ depends on the distance parameter δ above which "far" vectors are to be rejected. The optimal δ minimizes the soundness error. Distinguish three cases:

- $0 \leq \delta < \frac{1-\rho}{2}$. Within the *unique decoding radius* there is always at most one codeword so $\ell = 1$.
- $\frac{1-\rho}{2} \leq \delta < 1 - \sqrt{\rho}$. Up to and arbitrarily close to the Johnson bound there are list decoding algorithms such that $\ell < \frac{1}{2 \cdot (1 - \sqrt{\rho} - \delta) \cdot \sqrt{\rho}}$.
- $1 - \sqrt{\rho} \leq \delta < 1 - \rho$. Beyond the Johnson bound and up to the covering radius ℓ is bounded in the general case only by conjecture. For instance, DEEP-FRI conjectures [6, 2.3] $\ell \leq \left(\frac{n}{1-\rho-\delta} \right)^{C_\rho}$ for some C_ρ that depends on ρ, where n is the code length.

2.3 Quotienting

Lemma 4 ([1], Lemma 4.4). *Let $g : D \to \mathbb{F}$ be a function; let $S \in \mathbb{F}^t$ be a list of x-coordinates and $T \in \mathbb{F}^t$ be a list of matching y coordinates of t points (x, y) satisfying $x \in D \Rightarrow y = g(x)$; let $\mathsf{Ans}(X)$ be the minimal-degree interpolant through (S, T); let $Z(X)$ be the vanishing polynomial for S; let $\mathsf{Fill} : S \to \mathbb{F}$ be any function; let $\mathsf{RS}[\mathbb{F}, D, \rho{\cdot}|D|]$ a RS code; and let $\delta \in [0; 1-\rho)$ a distance parameter. If for all polynomials $p(X)$ of degree less than $\rho \cdot |D|$ from the radius-δ Hamming ball centered at g, $p(X)$ disagrees with (S, T) somewhere (*i.e., $\exists i \,.\, p(z_i) \neq y_i$*) then the function*

$$q \,:\, D \to \mathbb{F}\,,\; X \mapsto \begin{cases} \frac{g(X) - \mathsf{Ans}(X)}{Z(X)} & \Leftarrow X \notin S \\ \mathsf{Fill}(X) & \Leftarrow X \in S \end{cases} \tag{6}$$

is δ-far from $\mathsf{RS}[\mathbb{F}, D, \rho \cdot |D| - t]$.

In the special case where $|S| = |T| = 1$ and $S \cap D = \varnothing$ we refer to the operation of computing q from g as a *DEEP update*.

2.4 Degree Correction

The drawback with Lemma 4 is that the distance is established relative to the wrong code. Degree correction fixes this problem with randomness supplied by the verifier. The next lemma is a specialization of the one proved in STIR [1, Lemma 4.13] which applies to a batch of functions as opposed to just one. Below we reformulate the proof to establish just this, in our terms and symbols.

Lemma 5. *Let $g : D \to F$ be a function; and let $\eta \xleftarrow{\$} \mathbb{F}$. If $\mathsf{RS}[\mathbb{F}, D, \rho \cdot |D|]$ exhibits (ϵ, δ) correlated agremeent with proximity parameter $\delta < 1 - \rho$; and if g is δ-far from $\mathsf{RS}[\mathbb{F}, D, \rho \cdot |D| - t]$; then*

$$\Pr\left[\Delta\left(\frac{\eta^{t+1} \cdot X^{t+1} - 1}{\eta \cdot X - 1} \cdot g(X), \mathsf{RS}[\mathbb{F}, D, \rho \cdot |D|]\right) < \delta\right] < (t+1) \cdot \epsilon \;\; . \tag{7}$$

Proof. Assume for contradiction that the probability in Equation (7) is greater than or equal to $(t+1) \cdot \epsilon$. Then since $\frac{\eta^{t+1} \cdot X^{t+1} - 1}{\eta \cdot X - 1} \cdot g(X) = \sum_{i=0}^{t} \eta^i \cdot X^i \cdot g(X)$ is a univariate batching with weight η it follows from Lemma 2 that all members of the batch $\{X^i \cdot g(X)\}_{i=0}^{t}$ are δ-close to $\mathsf{RS}[\mathbb{F}, D, \rho \cdot |D|]$. From the correlated agreement assumption it then follows that there is a subdomain $\mathcal{D} \subset D$ of relative density $\frac{|\mathcal{D}|}{|D|} = 1 - \delta$ where all members of the batch agree with some polynomial of degree less than $\rho \cdot |D|$. Let $\{\hat{g_i}(X)\}_{i=0}^{t}$ be such a set of polynomials such that for all $i \in \{0, \ldots, t\}$, $\Delta(\hat{g_i}(X), X^i g(X)) < \delta$.

Proceed with induction on i. In the base case, $\hat{g_0} \in \mathsf{RS}[\mathbb{F}, D, \rho \cdot |D| - i]$. In the inductive case spanning $i \in \{0, \ldots, t-1\}$, assume that $\hat{g_0} \in \mathsf{RS}[\mathbb{F}, D, \rho \cdot |D| - i]$. Then the polynomial $X^{i+1} \cdot \hat{g_0}$ agrees with $X^{i+1} \cdot g(X)$ and with $\hat{g_{i+1}}(X)$ on $\mathcal{D}$. Since $|\mathcal{D}| > \rho \cdot |D| > \deg(\hat{g_{i+1}}(X))$ and $|\mathcal{D}| > \deg(X^{i+1} \cdot \hat{g_0}(X))$ it follows that

$X^{i+1} \cdot g(X)$ and $\hat{g_{i+1}}(X)$ must be identical polynomials, so $\deg(X^{i+1} \cdot \hat{g_0}(X)) = \deg(\hat{g_{i+1}}(X)) < \rho \cdot |D|$ and $\deg(\hat{g_0}(X)) < \rho \cdot |D| - i - 1$ and $\hat{g_0} \in \mathsf{RS}[\mathbb{F}, D, \rho \cdot |D| - i - 1]$. Completing the induction gives $\hat{g_0} \in \mathsf{RS}[\mathbb{F}, D, \rho \cdot |D| - t]$, and as $\hat{g_0}(X)$ agrees with $g(X)$ on $\mathcal{D}$ it follows that g is δ-close to $\mathsf{RS}[\mathbb{F}, D, \rho \cdot |D| - t]$. $\lightning$ $\square$

3 DEEP-ALI

This section reviews the DEEP-ALI construction [6] for reducing a low-degree extended algebraic execution trace to a single Reed-Solomon codeword from a high level perspective. For a more formal treatment of both the protocol and the language it decides we refer to to [6, § 6].

Let $\boldsymbol{f}$ be w-many low degree polynomials representing the execution trace, and to whose Reed-Solomon codewords on $D \subset \mathbb{F}$ the verifier has oracle access. A set of constraints apply to $\boldsymbol{f}$ and generate new polynomials $\boldsymbol{g}$ which are of low degree if *a)* all polynomials $\boldsymbol{f}$ are of low degree, and *b)* $\boldsymbol{f}$ satisfies the constraints.

The constraint set $\mathcal{C}$ is a list of tuples $(\boldsymbol{M}, P, Q)$ where

- $\boldsymbol{M}$ is a list of scaling factors implicitly defining scaled variants of the trace polynomials $\boldsymbol{f}(M_0 X), \boldsymbol{f}(M_1 X), \ldots$;
- P is a circuit over $\mathbb{F}$ whose inputs are $\boldsymbol{f}(M_0 X), \boldsymbol{f}(M_1 X), \ldots$;
- Q is a univariate polynomial that evaluates to zero in the points where the constraint is active and no-where else.

Evaluating the constraints gives rise to $\boldsymbol{g}$, which is then reduced to a single random linear combination g in order to apply FRI. DEEP-ALI tests the consistency between $\boldsymbol{f}$ and g as follows. Assume throughout that $\mathbb{F}$ is large enough to securely sample randomness from.

DEEP-ALI

- The prover sends the oracle $\boldsymbol{f} : (D \to \mathbb{F})^{\mathsf{w}}$.
- The verifier sends $\boldsymbol{\alpha} \xleftarrow{\$} \mathbb{F}^{|\mathcal{C}|}$.
- The prover computes for all $(\boldsymbol{M}, P, Q) \in \mathcal{C}$, $g_i \leftarrow \frac{P(\boldsymbol{f}(M_0 X), \boldsymbol{f}(M_1 X), \ldots)}{Q(X)}$ followed by $g(X) = \boldsymbol{\alpha} \cdot \boldsymbol{g}(X)$ and sends $g : D \to \mathbb{F}$.
- The verifier sends a single random out-of-domain indeterminate z.
- Once per each M across all constraints, the prover sends $\boldsymbol{f}(Mz)$.
- The verifier computes for all $(\boldsymbol{M}, P, Q) \in \mathcal{C}$, $g_i \leftarrow \frac{P(\boldsymbol{f}(M_0 z), \boldsymbol{f}(M_1 z), \ldots)}{Q(z)}$ followed by $g(z) = \boldsymbol{\alpha} \cdot \boldsymbol{g}(z)$.
- The prover explicitly performs the *DEEP update*[1] $h^{(2)}(X) \leftarrow \frac{g(X) - g(z)}{X - z}$. The verifier can simulate $h^{(2)} : D \to \mathbb{F}$ by applying the DEEP update to g in all points where $h^{(2)}$ is queried. Similarly, the prover computes (and the verifier simulates) $\boldsymbol{h^{(1)}}(X) \leftarrow \frac{\boldsymbol{f}(X) - \sum_M \boldsymbol{f}(Mz) \prod_{M' \neq M} \frac{X - Mz}{M'z - Mz}}{\prod_M (X - Mz)}$.
- The prover and verifier run a low degree test (with batching) such as FRI on $\boldsymbol{h^{(1)}}$ and $h^{(2)}$.

[1] We use the word "DEEP update" to refer to a single-point quotienting operation.

The DEEP-ALI is shown [6, Thm. 6.2] to have soundness error bounded by

$$\varepsilon + \frac{2L(d \cdot d_{\mathcal{C}} + \deg(Q_{\mathrm{lcm}})}{|\mathbb{F}|} \quad , \tag{8}$$

where ε is the soundness error of the batched low-degree test on $\boldsymbol{h}^{(\mathbf{1})}$ and $h^{(2)}$, where L is the list size when list-decoding vectors with the given distance parameter δ, where d is the trace domain size (or 1 more than $\deg(\boldsymbol{f}(X))$), where $d_{\mathcal{C}} = \max_{(\boldsymbol{M},P,Q)\in\mathcal{C}} \deg(P)$ is the total degree of the constraints (as multivariate polynomials), and where $Q_{\mathrm{lcm}} = \operatorname{lcm}_{(\boldsymbol{M},P,Q)\in\mathcal{C}} Q(X)$.

4 DEEP-ALI with Explicit Batching

The *batch* low degree test is applied to a *list* of functions as opposed to a singleton; and the test starts by reducing this list to a single function. This reduction is achieved by taking a randomized sum with uniformly random weights. This step is sound assuming a proximity gap. The DEEP-ALI (with explicit batching) looks as follows; this will be the starting point for future modifications.

DEEP-ALI with Explicit Batching

- The prover sends the oracle $\boldsymbol{f} : (D \to \mathbb{F})^{\mathsf{w}}$.
- The verifier sends $\boldsymbol{\alpha} \xleftarrow{\$} \mathbb{F}^{|\mathcal{C}|}$.
- The prover computes for all $(\boldsymbol{M}, P, Q) \in \mathcal{C}$, $g_i \leftarrow \frac{P(\boldsymbol{f}(M_0X),\boldsymbol{f}(M_1X),\ldots)}{Q(X)}$ followed by $g(X) = \boldsymbol{\alpha} \cdot \boldsymbol{g}(X)$ and sends $g : D \to \mathbb{F}$.
- The verifier sends a single random out-of-domain indeterminate z.
- Once per each M across all constraints, the prover sends $\boldsymbol{f}(Mz)$.
- The verifier computes for all $(\boldsymbol{M}, P, Q) \in \mathcal{C}$, $g_i \leftarrow \frac{P(\boldsymbol{f}(M_0z),\boldsymbol{f}(M_1z),\ldots)}{Q(z)}$ followed by $g(z) = \boldsymbol{\alpha} \cdot \boldsymbol{g}(z)$.
- The prover explicitly performs the *DEEP update* $h^{(2)}(X) \leftarrow \frac{g(X)-g(z)}{X-z}$. The verifier can simulate $h^{(2)} : D \to \mathbb{F}$ by applying the DEEP update to g in all points where $h^{(2)}$ is queried. Similarly, the prover computes (and the verifier simulates) $\boldsymbol{h}^{(\mathbf{1})}(X) \leftarrow \frac{\boldsymbol{f}(X)-\sum_M \boldsymbol{f}(Mz)\prod_{M'\neq M}\frac{X-Mz}{M'z-Mz}}{\prod_M(X-Mz)}$.
- The verifier sends *batching weights* $\boldsymbol{\beta} \xleftarrow{\$} \mathbb{F}^{\mathsf{w}+1}$.
- The prover computes (and the verifier simulates) $h^{(0)}(X) \leftarrow \left(\sum_{i=0}^{\mathsf{w}-1} \beta_i \cdot h_i^{(1)}(X)\right) + \beta_{\mathsf{w}} \cdot h^{(2)}(X)$.
- The prover and verifier run a low degree test (without batching) such as FRI on $h^{(0)}$.

Note that the above protocol is identical to DEEP-ALI as presented in § 3 and so has an identical soundness error. However, next we will modify the protocol in particular in regards to $h^{(0)}$.

5 Decouple Batch-Verification from Low Degree Test

We modify the DEEP-ALI protocol (with explicit batching) in order to decouple the batch verification from the low degree test. In particular, the batch verification now happens *before* the low-degree test starts, and with an index set independent from it. The modification is presented in two steps; only the second incurs a soundness degradation.

In the first step, the function $h^{(0)} : D \to \mathbb{F}$, representing (a commitment to) the batch polynomial is sent explicitly. In every point X where the verifier needs $h^{(0)}(X)$, he both reads it directly from this oracle and indirectly through simulation, and moreover he verifies that both values agree. Since this step is redundant from the point of view of the verifier, there is no soundness degradation. (There *is* a performance degradation because now the prover sends another Merkle tree and many openings into it.)

The second step is more involved. Following receipt of $h^{(0)}$, the verifier samples a random indeterminate $z \xleftarrow{\$} \mathbb{F}$ and sends it. The prover responds with a value $y \in \mathbb{F}$, which in the honest case equals $h^{(0)}(z)$. Next, the verifier samples and sends $x_0, \ldots, x_{t-1} \xleftarrow{\$} \mathbb{F}$ and tests the equality

$$h^{(0)}(X) \stackrel{?}{=} \left(\sum_{i=0}^{\mathsf{w}-1} \beta_i \cdot h_i^{(1)}(X) \right) + \beta_{\mathsf{w}} \cdot h^{(2)}(X) \tag{9}$$

in these points. The protocol proceeds with the function

$$q(X) = \frac{\gamma^{(t+1)} \cdot X^{(t+1)} - 1}{\gamma \cdot X - 1} \cdot \frac{h^{(0)}(X) - \mathsf{Ans}(X)}{Z(X)} \tag{10}$$

in place of $h^{(0)}(X)$, where $\gamma \xleftarrow{\$} \mathbb{F}$ was sampled by the verifier and sent along with $(x_0, \ldots, x_{t-1})$, where $\mathsf{Ans}(X)$ is the minimal-degree interpolant through the points $(x_i, h^{(0)}(x_i))$ for all $i \in [0 : t)$ as well as (z, y), and where $Z(X)$ is the vanishing polynomial for the x-coordinates of these points. The prover supplies $\mathsf{Fill}(X)$ which is a minimal (in the sense of information content) function that agrees with the factor on the right in Equation (10) when $X \in \{x_0, \ldots, x_{t-1}\}$.

To analyze the soundness degradation of this step, it suffices to analyze the worlds in which q is not δ-far from the code. To this end, distinguish three jointly exhaustive cases.

1. The batch sum $\left(\sum_{i=0} \beta_i \cdot h_i^{(1)}(X)\right) + \beta_{\mathsf{w}} \cdot h^{(2)}(X)$ is closer than δ to the code whereas at least one of its summands was not. The probability of this event is exactly ϵ from the (δ, ϵ) proximity gap as per Lemma 1.
2. There is no polynomial $p(X)$ that agrees with $h^{(0)}(X)$ in more than a fraction $1 - \delta$ of points and for which $p(z) = y$. Then by Lemmata 4 and 5 $q(X)$ is δ-far from the code except with probability bounded by $(t + 1) \cdot \epsilon$.
3. There is such a polynomial $p(X)$ and the batch sum is δ-far from the code. By Lemma 3, $p(X)$ is unique except with probability bounded by $\frac{\ell^2}{2} \cdot \left(\frac{d}{|\mathbb{F}| - |D|}\right)$.

Given that $p(X)$ is unique, the probability that it *also* satisfies

$$p(X) \stackrel{?}{=} \left(\sum_{i=0}^{\mathsf{w}-1} \beta_i \cdot h_i^{(1)}(X)\right) + \beta_{\mathsf{w}} \cdot h^{(2)}(X) \tag{11}$$

in $X \in \{x_0, \ldots, x_{t-1}\}$ is $(1-\delta)^t$. Consequently, Lemmata 4 and 5, $q(X)$ is δ-far from the code except with probability bounded by $(t+1) \cdot \epsilon$.

In conclusion, the soundness error degrades by a term

$$(2t+3) \cdot \epsilon + \frac{\ell^2}{2} \cdot \left(\frac{d}{|\mathbb{F}| - |D|}\right) \quad . \tag{12}$$

For reference, the complete protocol is presented below.

DEEP-ALI with Decoupled Batch-Verification

- The prover sends the oracle $\boldsymbol{f} : (D \to \mathbb{F})^{\mathsf{w}}$.
- The verifier sends $\boldsymbol{\alpha} \stackrel{\$}{\leftarrow} \mathbb{F}^{|\mathcal{C}|}$.
- The prover computes for all $(\boldsymbol{M}, P, Q) \in \mathcal{C}$, $g_i \leftarrow \frac{P(\boldsymbol{f}(M_0 X), \boldsymbol{f}(M_1 X), \ldots)}{Q(X)}$ followed by $g(X) = \boldsymbol{\alpha} \cdot \boldsymbol{g}(X)$ and sends $g : D \to \mathbb{F}$.
- The verifier sends a single random out-of-domain indeterminate z.
- Once per each M across all constraints, the prover sends $\boldsymbol{f}(Mz)$.
- The verifier computes for all $(\boldsymbol{M}, P, Q) \in \mathcal{C}$, $g_i \leftarrow \frac{P(\boldsymbol{f}(M_0 z), \boldsymbol{f}(M_1 z), \ldots)}{Q(z)}$ followed by $g(z) = \boldsymbol{\alpha} \cdot \boldsymbol{g}(z)$.
- The prover explicitly performs the *DEEP update* $h^{(2)}(X) \leftarrow \frac{g(X) - g(z)}{X - z}$. The verifier can simulate $h^{(2)} : D \to \mathbb{F}$ by applying the DEEP update to g in all points where $h^{(2)}$ is queried. Similarly, the prover computes (and the verifier simulates) $\boldsymbol{h^{(1)}}(X) \leftarrow \frac{\boldsymbol{f}(X) - \sum_M \boldsymbol{f}(Mz) \prod_{M' \neq M} \frac{X - Mz}{M'z - Mz}}{\prod_M (X - Mz)}$.
- The verifier sends *batching weights* $\boldsymbol{\beta} \stackrel{\$}{\leftarrow} \mathbb{F}^{\mathsf{w}+1}$.
- The prover computes and sends $h^{(0)}(X) \leftarrow \left(\sum_{i=0}^{\mathsf{w}-1} \beta_i \cdot h_i^{(1)}(X)\right) + \beta_{\mathsf{w}} \cdot h^{(2)}(X)$.
- The verifier samples and sends $z \stackrel{\$}{\leftarrow} \mathbb{F}$.
- The prover computes and sends $y \leftarrow h^{(0)}(z)$.
- The verifier samples $x_0, \ldots, x_{t-1} \stackrel{\$}{\leftarrow} D$ and checks for all $i \in [0 : t)$ that $h^{(0)}(x_i) = \left(\sum_{j=0}^{t-1} \beta_j \cdot h_j^{(1)}(x_i)\right) + \beta_{\mathsf{w}} h^{(2)}(x_j)$.
- The verifier samples $\gamma \stackrel{\$}{\leftarrow} \mathbb{F}$ and sends $(x_0, \ldots, x_{t-1}, \gamma)$.
- The prover and verifier run a low degree test (without batching) such as FRI on $q(X) = \frac{\gamma^{t+1} \cdot X^{t+1} - 1}{\gamma \cdot X - 1} \cdot \frac{h^{(0)}(X) - \mathsf{Ans}(X)}{Z(X)}$, where $\mathsf{Ans}(X)$ is the minimal-degree polynomial interpolating through $\{(x_0, h^{(0)}(x_0)), \ldots, (x_{t-1}, h^{(0)}(x_{t-1})), (z, y)\}$ and $Z(X)$ is the vanishing polynomial for $\{x_0, \ldots, x_{t-1}, z\}$. The prover supplies $\mathsf{Fill}(X)$ which is a minimal function that agrees with the fraction $\frac{h^{(0)}(X) - \mathsf{Ans}(X)}{Z(X)}$ on $\{x_0, \ldots, x_{t-1}\}$, which the verifier needs in order to simulate $q(X)$.

5.1 DEEP Commitment Scheme

The previous protocol already suffices to address the motivating question of amortizing the cost of verifying low degree tests by decoupling the witnesses from the prior protocols they relate to. However, an even better result lies within reach. It is possible to take the reduction to a single polynomial and make the resulting polynomial commitment an independent certificate of integrity of the computational claim. The polynomial commitment can be batched with other polynomial commitment without noticeable effect on the soundness error, provided that eventually a low degree test is applied to the aggregate.

Consider the tuple $(h^{(0)}, z, y, \{x_0, \ldots, x_{t-1}\}, \gamma)$ as it appears in the transcript. We will refer to this tuple from here on out as a *DEEP Commitment*. This name stretches the traditional denotation of "commitment", referring to something that is produced entirely by the prover, is independent of protocol history, and relies on cryptographic hard problem. However, the common feature justifying overloading the term is that it binds the prover down to a single value — in this case: a polynomial.

To see why equivocation is hard, consider the following argument. The low degree test ensures that the committer can only open a committed word to a polynomial from the radius-δ Hamming ball centered at that word, except with some negligible probability expressed in terms of δ. Any Hamming ball containing more than one codeword is likely to contain polynomials that agree *somewhere*. The point is that the indeterminate z is chosen at random, by the verifier. The probability of having multiple candidate polynomials that agree somewhat with a function $h^{(0)}$ and also have the same value in a random point is bounded in exact terms by Lemma 3.

In fact, with the same denotational allowance, it is a polynomial commitment scheme. The prover can establish that $h^{(0)}(X)$ evaluates to given values in given indeterminates by incorporating them into the interpolant $\mathsf{Ans}(X)$, vanishing polynomial $Z(X)$, and hole filler $\mathsf{Fill}(X)$. The quotient $q(X)$ will be low degree if and only if the extra points indeed match with the polynomial $h^{(0)}(X)$.

One of the reasons why this tuple receives special focus is that it allows us to define the following notion, which helps with proving. Informally, δ-well-formedness helps to distinguish which commitments have a matching witness.

Definition 3 (δ-well-formed DEEP commitment). *Let $(h, z, y, S, \gamma) : (D \to \mathbb{F}) \times \mathbb{F} \times \mathbb{F} \times \binom{D}{t} \times \mathbb{F}$ be a DEEP commitment, and let ρ be the rate of the associated code and $\delta \in [0; 1-\rho)$ a proximity parameter. The DEEP commitment is* δ-well-formed *if there is a polynomial $p(X)$ of degree less than $\rho \cdot |D|$ such that $\Delta(h, p) < \delta$ and $p(z) = y$ and $p(S) = h(S)$.*

An obvious but important feature of δ-well-formedness is that the function q implicitly defined by a DEEP commitment will be rejected by a low degree test (except with high probability quantifiable in terms of δ) if the the commitment is not δ-well-formed; and accepted otherwise.

6 Application: Batched Low-Degree Test

We are now in a position to address explicitly the motivating question articulated in the introduction: to split the protocol into two parts, in order to batch the second halves across multiple executions. To show that this construction retains soundness, we prove soundness for the two components separately, before proving that their composition is indeed still sound. The two components are separated at the DEEP commitment $(h^{(0)}, z, y, \{x_0, \ldots, x_{t-1}\}, \gamma)$, which is the output of the first component (*Reduction to a Single Polynomial*) and the input of the second (*DEEP Low Degree Test*).

Definition 4 (soundness for reduction to single polynomials). *Let $\mathcal{P}$ be a protocol between prover and verifier that takes a APR intsance (*i.e.*, the problem instance that DEEP-ALI takes) and outputs a DEEP commitment (or early reject). $\mathcal{P}$ is sound with soundness error ε if for every APR instance $\mathcal{C}$, and every witness $\boldsymbol{f}$,*

$$\Pr\left[\begin{array}{l} \mathcal{C} \text{ is not satisfied by } \boldsymbol{f} \\ \wedge\, \mathcal{P}(\mathcal{C}, \boldsymbol{f}) \to (h^{(0)}, z, y, \{x_0, \ldots, x_{t-1}\}, \gamma) \\ \wedge\, (h^{(0)}, z, y, \{x_0, \ldots, x_{t-1}\}, \gamma) \text{ is } \delta\text{-well-formed} \end{array}\right] \leq \varepsilon \ . \tag{13}$$

Definition 5 (soundness for DEEP low degree tests). *Let $\mathcal{P}$ be a protocol between a prover and a verifier that takes a DEEP commitment (and implicitly, a witness for the prover) and produces no output (other than the verifier's verdict, accept or reject). $\mathcal{P}$ is sound with soundness error ε if for any DEEP commitment $(h^{(0)}, z, y, \{x_0, \ldots, x_{t-1}\}, \gamma)$ and for any witness $\boldsymbol{f}$,*

$$\Pr\left[\begin{array}{l} \mathcal{P}((h^{(0)}, z, y, \{x_0, \ldots, x_{t-1}\}, \gamma)) \text{ accepts} \\ \wedge\, (h^{(0)}, z, y, \{x_0, \ldots, x_{t-1}\}, \gamma) \text{ is not } \delta\text{-well-formed} \end{array}\right] \leq \varepsilon \ . \tag{14}$$

Theorem 1. *The* Reduce to Single Polynomial *protocol is sound with soundness error bounded by that of* DEEP-ALI with Decoupled Batch-Verification*.*

Proof. Suppose A is an adversary that, when fed a false witness $\boldsymbol{f}$, succeeds with probability ε_{A} to reduce it to a δ-well-formed DEEP commitment $(h^{(0)}, z, y, \{x_0, \ldots, x_{t-1}\}, \gamma)$. Note that the *Reduction to a Single Polynomial* protocol is a prefix of the *DEEP-ALI with Decoupled Batch-Verification* protocol. If the commitment is δ-well-formed, continuing with the rest of the *DEEP-ALI with Decoupled Batch-Verification* will result in an accepting verifier. It follows that ϵ_{A} must be bounded by the soundness error of that protocol. □

Theorem 2. *The* DEEP Low Degree Test *is sound with soundness error bounded by that of the low degree test.*

Proof. Because (h, y) is not δ-well-formed, the quotient $q(X) = \frac{\gamma^{t+1} \cdot X^{t+1} - 1}{\gamma \cdot X - 1} \cdot \frac{h(X) - \mathsf{Ans}(X)}{Z(X)}$ is δ-far from the code as per Lemmata 4 and 5. It follows that it will be accepted with probability at most the soundness error of the low degree test for proximity parameter δ. □

The obvious corollary is that composing the *Reduction to a Single Polynomial* with a *DEEP Low Degree Test* is a sound protocol for APR with soundness error bounded by the the sum of soundness errors of *DEEP-ALI with Decoupled Batch-Verification* and the low degree test. The soundness error of the low degree test is counted twice in this quantity; this duplicate count is an artifact of the black box reduction.

The less obvious corollary is that composing multiple *Reductions to Single Polynomial* with a single *batched DEEP Low Degree Test* is sound as well. The soundness error comprises a linear number of terms originating from the *Reductions to Single Polynomials*, one term originating from the (unbatched) low degree test, and one term capturing the batching step which involves the ϵ from the (δ, ϵ) proximity gap.

7 Application: Aggregation

The above decomposition views the DEEP commitment as an ephemeral object with a short life span in between the *Reduction to Single Polynomial* and the *DEEP Low Degree Test*. What if we extend the life span and view a DEEP commitment as its own standalone object? The unlocked use cases is the second reason for the particular focus on the tuple $(h^{(0)}, z, y, \{x_0, \ldots, x_{t-1}\}, \gamma)$.

The next protocol merges two DEEP commitments, $(h_1, z_1, y_1, S_1, \gamma_1)$ and $(h_2, z_2, y_2, S_2, \gamma_2)$ (with respective interpolants $\mathsf{Ans}_1(X)$, $\mathsf{Ans}_2(X)$, vanishing polynomials $Z_1(X)$, $Z_2(X)$, and hole fillers $\mathsf{Fill}_1(X)$, $\mathsf{Fill}_2(X)$). It allows a user to choose between running the *DEEP Low Degree Test*, or merge the DEEP commitment with other DEEP commitments first. The choice can be delayed indefinitely, as long as a *DEEP Low Degree Test* is run eventually.

DEEP Merge

- The verifier samples and sends $r \xleftarrow{\$} \mathbb{F}$.
- The prover computes and sends $h_3(X) = \frac{\gamma_1^{t+1} \cdot X^{t+1} - 1}{\gamma_2 \cdot X - 1} \cdot \frac{h_1(X) - \mathsf{Ans}_1(X)}{Z_1(X)} + r \cdot \frac{\gamma_2^{t+1} \cdot X^{t+1} - 1}{\gamma_2 \cdot X - 1} \cdot \frac{h_2(X) - \mathsf{Ans}_2(X)}{Z_2(X)}$.
- The verifier samples and sends $z \xleftarrow{\$} \mathbb{F}$.
- The prover computes and sends $y \leftarrow h_3(z)$.
- The verifier samples $S = (x_0, \ldots, x_{t-1}) \xleftarrow{\$} D^t$ and checks for all $i \in [0 : t)$ that $h_3(x_i) = \frac{h_1(x_i) - \mathsf{Ans}_1(x_i)}{Z_1(x_i)} + r \cdot \frac{h_2(x_i) - \mathsf{Ans}_2(x_i)}{Z_2(x_i)}$.
- The verifier samples $\gamma_3 \xleftarrow{\$} \mathbb{F}$.

The protocol's completeness follows from construction. Its soundness, in the configuration of a circuit whose inputs are *Reductions to Single Polynomials*, whose internal nodes are *DEEP Merges*, and whose outputs are (potentially batched) *DEEP Low Degree Tests*, follows inductively from the following lemma.

Lemma 6. *Let $(h_3, z_3, y_3, S_3, \gamma_3)$ be a DEEP commitment resulting from merging DEEP commitments $(h_1, z_1, y_1, S_1, \gamma_1)$ and $(h_2, z_2, y_2, S_2, \gamma_2)$. Let*

$(h_1, z_1, y_1, S_1, \gamma_1)$ or $(h_2, z_2, y_2, S_2, \gamma_2)$ (or both) be not δ-well-formed. Then the probability that $(h_3, z_3, y_3, S_3, \gamma_3)$ is δ-well-formed is bounded by

$$(3t+4) \cdot \epsilon + \frac{\ell^2}{2} \cdot \left(\frac{d}{|\mathbb{F}| - |D|} \right) + (1-\delta)^t \tag{15}$$

where ϵ is the false witness probability from the (δ, ϵ) proximity gap.

Proof. Since $(h_1, z_1, y_1, S_1, \gamma_1)$ or $(h_2, z_2, y_2, S_2, \gamma_2)$ is not δ-well-formed (or neither are), then by Lemmata 4 and 5, one of the functions $\frac{\gamma_1^{t+1} \cdot X^{t+1} - 1}{\gamma_2 \cdot X - 1} \cdot \frac{h_1(X) - \mathsf{Ans}_1(X)}{Z_1(X)}$ or $\frac{\gamma_2^{t+1} \cdot X^{t+1} - 1}{\gamma_2 \cdot X - 1} \cdot \frac{h_2(X) - \mathsf{Ans}_2(X)}{Z_2(X)}$ is δ-far from the code (or both are) except with probability $(t+1) \cdot \epsilon$ each. Assuming one of the summands is δ-far, h_3 (when computed honestly) is δ-far from the code except with probability ϵ from the (δ, ϵ) proximity gap. In summary, except with probability $(2t+3) \cdot \epsilon$, the radius-δ Hamming ball centered at (honestly computed) h_3 contains no codewords. Consequently, in this case, no low-degree polynomial agrees with h_3, and so h_3 is incapable of being well-formed for any indeterminate z, including z_3.

What remains is the case wherein $h_3(X)$ is not honestly computed, specifically: $h_3(X) \neq \frac{\gamma_1^{t+1} \cdot X^{t+1} - 1}{\gamma_2 \cdot X - 1} \cdot \frac{h_1(X) - \mathsf{Ans}_1(X)}{Z_1(X)} + r \cdot \frac{\gamma_2^{t+1} \cdot X^{t+1} - 1}{\gamma_2 \cdot X - 1} \cdot \frac{h_2(X) - \mathsf{Ans}_2(X)}{Z_2(X)}$. Distinguish two cases.

1. There is no polynomial $p(X)$ that agrees with $h_3(X)$ in more than a fraction of $1 - \delta$ of points and for which $p(z) = y$. Then by Lemmata 4 and 5, $q(X)$ is δ-far from the code except with probability bounded by $(t+1) \cdot \epsilon$.
2. Such a polynomial $p(X)$ does exist. By Lemma 3, except with probability bounded by $\frac{\ell^2}{2} \cdot \left(\frac{d}{|\mathbb{F}| - |D|} \right)$, $p(X)$ is unique. The probability that $p(X)$ *also* satisfies

$$p(X) \stackrel{?}{=} \frac{\gamma_1^{t+1} \cdot X^{t+1} - 1}{\gamma_2 \cdot X - 1} \cdot \frac{h_1(X) - \mathsf{Ans}_1(X)}{Z_1(X)} + r \cdot \frac{\gamma_2^{t+1} \cdot X^{t+1} - 1}{\gamma_2 \cdot X - 1} \cdot \frac{h_2(X) - \mathsf{Ans}_2(X)}{Z_2(X)} \tag{16}$$

   on all $X \in S_3$ is $(1-\delta)^t$.

By the union bound, the probability of any event in which $(h_3, z_3, y_3, S_3, \gamma_3)$ is δ-well-formed is bounded by

$$(3t+4) \cdot \epsilon + \frac{\ell^2}{2} \cdot \left(\frac{d}{|\mathbb{F}| - |D|} \right) + (1-\delta)^t \ . \tag{17}$$

□

8 Conclusion

We close with a few remarks that do not fit elsewhere.

Abstraction level. The IOPs presented here represent non-interactive proofs in the standard model. The following transformations are implicit: the BCS [5]

transform, which turns an IOP into a succinct interactive proof; the Fiat-Shamir transform [15], which turns an interactive proof into a non-interactive proof in the random oracle model; and lastly the random oracle is replaced with a concrete hash function.

How possible? The construction and its security proofs offer little intuition as to *why* it works. More specifically, why does decoupling the batch-verification from the low degree test, and the same applications that follow from it, fail with plain DEEP-ALI?

The issue seems to be related to the capacity of malicious provers to commit to functions that are halfway in between the far-from-code batch sum and some codeword, say with relative distances $\delta_1 < \delta$ and $\delta_2 < \delta$ respectively, and note that $\delta_1 + \delta_2 \geq \delta$. The discrepancy between the batch sum and the committed codeword is not caught with probability $(1 - \delta_1)^t$, at which point the prover can continue with a committed function whose distance from the code is less than δ.

The feature of the present proposal (and that of its inspiration, STIR [1]) that thwarts this attack vector is that the queried in-domain indeterminates and matching responses are being quotiented out in the process deriving a new committed function $q(X)$ whose low degree testifies to the computational integrity claim. This quotienting step effectively shifts focus from the concrete function $h^{(0)}$, first to the radius-δ Hamming ball centered at $h^{(0)}$, and then through the out-of-domain evaluation at z to the (with high probability) unique polynomial $p(X)$. As a result, the question is not whether the function $h^{(0)}$ agrees with the batch sum in all of the in-domain samples; instead, the question is whether the polynomial $p(X)$ agrees with the batch sum in all the in-domain samples.

Non-interactive commitments. It is possible to transform the commitment scheme into a (standard) non-interactive one, by applying the Fiat-Shamir transform to the necessary rounds in a way that ignores the prior protocol transcript. The result is a non-interactive commitment scheme that is computationally binding in the random oracle model. The drawback of this transform is a rather large (but quantifiable) soundness error degradation resulting from a black box reduction. Since two steps of interaction are being made non-interactive, the soundness error degrades by a *factor* Q^2, where Q is the number of queries the computationally bounded adversary is allowed to make to the random oracle. The soundness error of the remaining protocol should be small enough to compensate for this degradation, resulting in a significant performance downgrade.

No extraction. PCD [7] and its refinements [11,8,12] are typically defined in terms of *knowledge soundness*, which requires the existence of an extractor capable of producing the prover's witness in a slightly unrealistic model of reality such as possessing the capacity to rewind the prover, or having access to an idealized oracle. We opt here against knowledge soundness because regular soundness suffices. In particular, typically when the commitment is a group element then *every* group element is a commitment to *something*; and in this context, soundness becomes moot and knowledge-soundness necessary to qual-

ify that something. By contrast, in the present context, not all commitments are δ-well-formed, and so soundness is not moot.

References

1. Arnon, G., Chiesa, A., Fenzi, G., Yogev, E.: STIR: reed-solomon proximity testing with fewer queries. In: Reyzin, L., Stebila, D. (eds.) CRYPTO 2024, Part X. LNCS, vol. 14929, pp. 380–413. Springer (2024). https://doi.org/10.1007/978-3-031-68403-6_12
2. Ben-Sasson, E., Bentov, I., Horesh, Y., Riabzev, M.: Fast reed-solomon interactive oracle proofs of proximity. In: Chatzigiannakis, I., Kaklamanis, C., Marx, D., Sannella, D. (eds.) ICALP 2018. LIPIcs, vol. 107, pp. 14:1–14:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2018). https://doi.org/10.4230/LIPICS.ICALP.2018.14
3. Ben-Sasson, E., Bentov, I., Horesh, Y., Riabzev, M.: Scalable zero knowledge with no trusted setup. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11694, pp. 701–732. Springer (2019). https://doi.org/10.1007/978-3-030-26954-8_23
4. Ben-Sasson, E., Carmon, D., Ishai, Y., Kopparty, S., Saraf, S.: Proximity gaps for reed-solomon codes. In: Irani, S. (ed.) FOCS 2020. pp. 900–909. IEEE (2020). https://doi.org/10.1109/FOCS46700.2020.00088
5. Ben-Sasson, E., Chiesa, A., Spooner, N.: Interactive oracle proofs. In: Hirt, M., Smith, A.D. (eds.) TCC 2016. LNCS, vol. 9986, pp. 31–60 (2016). https://doi.org/10.1007/978-3-662-53644-5_2
6. Ben-Sasson, E., Goldberg, L., Kopparty, S., Saraf, S.: DEEP-FRI: sampling outside the box improves soundness. In: Vidick, T. (ed.) ITCS 2020, Seattle, Washington, USA. LIPIcs, vol. 151, pp. 5:1–5:32. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020). https://doi.org/10.4230/LIPICS.ITCS.2020.5
7. Bitansky, N., Canetti, R., Chiesa, A., Tromer, E.: Recursive composition and bootstrapping for SNARKS and proof-carrying data. In: Boneh, D., Roughgarden, T., Feigenbaum, J. (eds.) STOC 2013. pp. 111–120. ACM (2013). https://doi.org/10.1145/2488608.2488623
8. Boneh, D., Drake, J., Fisch, B., Gabizon, A.: Halo infinite: Proof-carrying data from additive polynomial commitments. In: Malkin, T., Peikert, C. (eds.) CRYPTO 2021. LNCS, vol. 12825, pp. 649–680. Springer (2021). https://doi.org/10.1007/978-3-030-84242-0_23
9. Bowe, S., Grigg, J., Hopwood, D.: Halo: Recursive proof composition without a trusted setup. IACR ePrint p. 1021 (2019), https://eprint.iacr.org/2019/1021
10. Bünz, B., Chen, B.: Protostar: Generic efficient accumulation/folding for special-sound protocols. In: Guo, J., Steinfeld, R. (eds.) ASIACRYPT 2023. Lecture Notes in Computer Science, vol. 14439, pp. 77–110. Springer (2023). https://doi.org/10.1007/978-981-99-8724-5_3
11. Bünz, B., Chiesa, A., Mishra, P., Spooner, N.: Recursive proof composition from accumulation schemes. In: Pass, R., Pietrzak, K. (eds.) TCC 2020. LNCS, vol. 12551, pp. 1–18. Springer (2020). https://doi.org/10.1007/978-3-030-64378-2_1
12. Bünz, B., Mishra, P., Nguyen, W., Wang, W.: Accumulation without homomorphism. IACR ePrint p. 474 (2024), https://eprint.iacr.org/2024/474
13. Diamond, B.E., Posen, J.: Proximity testing with logarithmic randomness. IACR Cryptol. ePrint Arch. p. 630 (2023), https://eprint.iacr.org/2023/630

14. Eagen, L., Gabizon, A.: Protogalaxy: Efficient protostar-style folding of multiple instances. IACR ePrint p. 1106 (2023), https://eprint.iacr.org/2023/1106
15. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer (1986). https://doi.org/10.1007/3-540-47721-7_12
16. Haböck, U., Kindi, A.: A note on adding zero-knowledge to starks. IACR Cryptol. ePrint Arch. p. 1037 (2024), https://eprint.iacr.org/2024/1037