# ProveKit v2 Whitepaper (WIP)

No Author Given

No Institute Given

# Table of Contents


1 R1CS with scheduled witnesses. (tbc) . . . 2
2 Preliminaries . . . 2
  2.1 Notation . . . 3
  2.2 Multilinears and univariates . . . 3
  2.3 Interleaved representation . . . 3
  2.4 Liftings . . . 4
  2.5 Properties of interleaved Reed–Solomon codes . . . 4
3 The protocol without zero-knowledge . . . 4
  3.1 Arithmetization. . . . 5
  3.2 Encoding. . . . 5
  3.3 The IOPP . . . 6
  3.4 Soundness. . . . 9
4 WHIR . . . 11
  4.1 Setup . . . 11
  4.2 The IOPP. . . . 12
  4.3 Soundness. . . . 13
5 Adding zero-knowledge . . . 14
  5.1 Randomizing $w_M(\vec{\alpha})$. . . . 14
  5.2 Zero-check masking . . . 17
  5.3 Masked commitments. . . . 20
  5.4 WHIR in zero-knowledge . . . 20
  5.5 Impact on soundness (tbc) . . . 26
6 The non-interactive STARK (tbc) . . . 27
  6.1 Merkle commitments (tbc) . . . 27
  6.2 The . . . 27
  6.3 Fiat–Shamir / BCS transform (tbc) . . . 27
  6.4 Example parameters (tbc) . . . 27
  6.5 Benchmarks (tbc) . . . 27
7 Outlook v3 (tbc) . . . 27
A The complete protocol (tbc) . . . 28
B Correlated agreement (tbc) . . . 28
C Proofs: IOPP soundness (tbc) . . . 29
  C.1 Protocol 1 . . . 29
  C.2 Soundness ZO0K. (tbc) . . . 31
  C.3 The complete protocol (tbc) . . . 31
D Proofs: IOPP zero-knowledge (tbc) . . . 31
  D.1 ZO0K (tbc) . . . 31
  D.2 The complete protocol (tbc) . . . 32
E Alternatives to ZO0K . . . 32

# 1 R1CS with scheduled witnesses. (tbc)

Two-round R1CS, to support lookups into tables, which are baked into the R1CS matrices. The witness $\vec{w}$ consists of

1. Round-1 witness $\vec{w}_1$: These witnesses are independent of the verifier challenge Includes public inputs, constants, and lookup table multiplicities.
2. Round-2 witness $\vec{w}_2$: These witnesses depend on the verifier challenges. These are the verifier challenges themselves, the logUp inverses and their sums.

Ulrich: Let us elaborate on the usefulness of 2-stage R1CS beyond lookups. Ulrich: Need to define $\mathscr{R}_{main}$ and $\mathscr{R}_{aux}$. Target $\vec{w}_1 \in \mathscr{R}_{main} \cap \mathscr{R}_{aux}$ The complete witness for the constraint system is of the form

$$\vec{w} = \vec{w}_1 \| \vec{w}_2,$$

with $\vec{w}_1 \in \mathbb{F}_q^{N_1}$ and $\vec{w}_2 \in \mathbb{F}_q^{N_2}$, Ulrich: Concatenation is never used anywhere. Ulrich: It seems more efficient to treat logup with several basefield challenges. This is for sure the case for the witness builder. Memory constraints in the zerocheck (due to the higher-dimensional hypercube) can be overcome by univariate skip. subject to $M \geq 1$ constraints

$$(A \cdot \vec{w}) \odot (B \cdot \vec{w}) = C \cdot \vec{w},$$

with $A, B, C \in \mathbb{F}^{K \times N}$, and where $\odot$ is the entrywise (Hadamard) product. As for the witness, we split the matrices $M = A, B, C$ into

$$M = \left(M_1 | M_2\right),$$

with $M_1 \in \mathbb{F}_q^{K \times N_1}$ and $M_2 \in \mathbb{F}_q^{K \times N_2}$ corresponding to the $\vec{w}_1$- and $\vec{w}_2$-parts. Typically the system is partially decoupled, with each $A_2, B_2, C_2$ starting with a large block of zero rows, but we currently do not leverage simpler representations than the ones above.

We need:

- Definition of the first stage R1CS with auxiliary lookup relation $\mathscr{R}_{\text{R1CS}} \cap \mathscr{R}_{\text{aux}}$
- Definition of randomized R1CS (with two stages), and its soundness error

**Split-witness builder.** There is a split-witness builder for stepwise computation of the complete witness.

# 2 Preliminaries

Although the entire construction described in the writeup applies to arbitrary prime fields, we confine ourselve to the the following choices. Let $\mathbb{F}_q$ be the Goldilocks prime field [], with the Solinas prime

$$q = 2^{64} - 2^{32} + 1$$

as modulus. We denote by $F \geq \mathbb{F}_q$ any cryptographically large extension, in our case the degree 3 extension $F = \mathbb{F}_{q^5}$, using... Ulrich: Check which irreducible polynomial is best

### 2.1 Notation

Boolean Hypercube $H_n = \{0,1\}^n \subset \mathbb{F}_q^n$, indexing in natural order via $[0, 2^n)$, multivariate and multilinear polynomials, Lagrange polynomial

$$eq(\vec{y}, \vec{\rho}) = \prod_{i=1}^{n} (y_i \cdot \rho_i + (1 - y_i) \cdot (1 - \rho_i)) \tag{1}$$

Inner product notation.

### 2.2 Multilinears and univariates

Given a function $f \in \mathbb{F}^{H_n}$ (a vector of values) we will write

$$f(X_1, \ldots, X_n) \in \mathbb{F}[X_1, \ldots, X_n]^{<2}$$

for its multilinear extension, and associate to it the univariate polynomial

$$u_f(X) = \sum_{i=0}^{2^n-1} f(i_1, \ldots, i_n) \cdot X^i, \tag{2}$$

defined by the values of $f$ over $H_n$, where $i = i_1 + \ldots + 2^{n-1} i_n$ is the decomposition of the index into bits. Under this *values-to-coefficient* regime, we identify the space of multilinears (in $n$ variables) with the univariate space

$$\mathscr{P}_n(\mathbb{F}) = \mathbb{F}[X]^{<2^n} \tag{3}$$

in a way that is particularly efficient for Reed–Solomon encoding.

**Univariate evaluation** Evaluation at a point $z \in \mathbb{F}$ is a linear functional and can be written as inner product $u_m(z) = \langle f, L_z \rangle_{H_n}$ with

$$L_z(X_1, \ldots, X_n) = \prod_{j=1}^{n} \left( (1 - X_j) + X_j \cdot z^{2^{j-1}} \right), \tag{4}$$

and its values over the hypercube are the powers of $z$ in natural order.

### 2.3 Interleaved representation

In alignment with the sumcheck protocol, we will often think of a multilinear $f(X_1, \ldots, X_n)$ decomposed into its $k$-prefix functions

$$f(X_1, \ldots, X_n) = \sum_{i=0}^{2^k-1} eq_i(X_1, \ldots, X_k) \cdot f_i(X_{k+1}, \ldots, X_n) \cdot \tag{5}$$

Here we again used the identification $H_k = [0, 2^k)$, by each index $i$ specifying the prefix $(i_1, \ldots, i_k) \in H_k$ via its bits. We call $(f_0, \ldots, f_{2^{n-k}-1})$ the *interleaved representation* of $f$, with interleaving depth $e = 2^k$. In terms of their univariate polynomials,

$$u_f(X) = \sum_{i=0}^{2^k-1} X^i \cdot u_{f_i}(X^{2^k}), \tag{6}$$

which is the FFT decomposition of $u_f(X)$.

**Remark 1.** The current version of ProveKit instead employs a postfix decomposition of multilinear polynomials, which corresponds, in the univariate representation, to partitioning the polynomial into contiguous blocks of monomial powers. Since our construction relies very little on properties specific to Reed–Solomon codes, the choice of decomposition is largely a matter of convenience. We adopt a prefix decomposition instead, as it leads to notation that is more consistent with our treatment of liftings.

### 2.4 Liftings

Given a multilinear $f$ in $k$ variables, and $n > k$, we call

$$f^*(X_1, \ldots, X_n) = (1 - X_1) \cdot \ldots \cdot (1 - X_{n-k}) \cdot f(X_{n-k+1}, \ldots, X_n) \tag{7}$$

its *lifting* to $n$ variables. In terms of the values over the hypercube, lifting simply places the original vector (the values over $H_k$) on the prefix cube

$$\{0^{n-k}\} \times H_k \subset H_n,$$

and pads it with zero on the remainder of $H_n$. In terms univariate representations,

$$u_{f^*}(X) = u_f(X^{2^{n-k}}). \tag{8}$$

We use lifting to uniformize our R1CS across the two stages at no additional cost.

### 2.5 Properties of interleaved Reed–Solomon codes

Are RS codes over an extension field. Same rate, same distance.

## 3 The protocol without zero-knowledge

In this section we describe the main construction for proving a two-stage R1CS, yet without the additional measures for zero-knowledge. The protocol is a combination Spartan [Set20] and WHIR [ACFY24] as a general inner-product argument. We skip the proving the values of the public circuit polynomials, which in [Set20] is handled by SPARK, and assume that the verifier does compute their values by itself. In this way we obtain succinct proof sizes, yet the verifier costs are of the same order as the witness size.

### 3.1 Arithmetization.

We consider the R1CS witness vectors padded to their next power of two, i.e. as functions

$$w_i : \{0,1\}^{m_i} \longrightarrow \mathbb{F}_q, \quad i = 1, 2,$$

and we denote their multilinear extensions again by the same letters. Likewise, the number of overall constraints are padded to a power of two, say $2^m$, and we view the R1CS matrices as functions

$$A_i, B_i, C_i : \{0,1\}^m \times \{0,1\}^{m_i} \longrightarrow \mathbb{F}_q, \quad i = 1, 2,$$

with the same notation for their multilinear extensions.

Throughout the following we assume that $m_1 \geq m_2$. We lift $w_2$ to the same number of variables as $w_1$, and likewise we do for the second stage matrix polynomials $A_2, B_2, C_2$. The resulting system has

$$w(X_0, X_1, \ldots, X_{m_1}) = \\ (1 - X_0) \cdot w_1(X_1, \ldots, X_{m_1}) + X_0 \cdot w_2^*(X_1, \ldots, X_{m_1}), \quad (9)$$

as the combined witness polynomial, and

$$M(\vec{Y}, X_0, X_1, \ldots, X_{m_1}) = \\ (1 - X_0) \cdot M_1(\vec{Y}, X_1, \ldots, X_{m_1}) + X_0 \cdot M_2^*(\vec{Y}, X_1, \ldots, X_{m_1}), \quad (10)$$

for $M = A, B, C$, as the combined matrix polynomials.

Ulrich: placing of inputs

### 3.2 Encoding.

We encode $w_1$ and $w_2^*$ in interleaved manner, using the same basecode

$$C = \mathsf{RS}[F, D, 2^k], \quad (11)$$

which corresponds to the first basecode of WHIR. (We will discuss the setup of WHIR later.) The multilinears $w_1$ and $w_2^*$ are decomposed into $(m_1 - k)$-prefix components, which are then encoded non-systematically, using their univariate representation from $\mathscr{P}_k$. The resulting words

$$f_1, f_2^* \in C^e,$$

the interleaved Reed–Solomon code of interleaving depth $e = 2^{m_1 - k_1}$, where $f_2^*$ has only $d = 2^{\max\{m_2 - k_1, 0\}}$ non-zero components. (These are the prefix embeddings of the $d$ components of $w_2$.) Thus committing to the lifted $w_2^*$ causes no overhead compared to $w_2$, given that $m_2 \geq k$.

**Lifting small commitments.** Whenever $m_2 < k$, we commit the univariate representation $u_2(X)$ of $w_2$ over a projection of $D$, the image of the multiplicative homomorphism

$$\pi : D \longrightarrow \pi(D), \quad x \mapsto x^{2^{k_1 - m_2}}. \tag{12}$$

Since $D$ is an FFT domain, the map is $2^{k-m_2}$-to-1 and onto. The committed word $f_2 \in \mathbb{F}^{D'}$ is from the smaller code

$$C' := \mathsf{RS}[F, D', 2^{m_2}], \tag{13}$$

which has the same relative rate as $C$. The lifted word

$$f_2^*(x) = f_2(\pi(x)) \tag{14}$$

belongs to $C$, since $u_2^*(X) = u_2(X^{2^{k-m_2}})$ is of degree less than $2^k$. By means of the projection $\pi$ we simulate $f_2^* \in C$ without paying the price of committing over the larger domain.

**Remark 2.** As noted in Remark 1, the current implementation employs a postfix decomposition. However, liftings of the corresponding univariate polynomials are naturally expressed in terms of prefix embeddings, resulting in a somewhat inelegant mixture of prefix and postfix terminology.

### 3.3 The IOPP

Given the commitment of $w_1$ as interleaved word $f_1$ on $D$, and proximity parameter $\theta \in (0, 1)$, the target of the protocol is to show that $f_1$ is $\theta$-close to a solution of our augmented constraint system, characterized by $\mathscr{R}_{main}$ and $\mathscr{R}_{aux}$ and the inputs $\vec{in} = (in_1, \ldots, in_{t_1})$. That is, we want to prove that the committed function belongs to

$$\mathcal{R}_\theta = \left\{ f \in \left(\mathbb{F}_q^D\right)^e \; : \; \begin{array}{c} \exists p(X) = \sum_{i=0}^{2^{m_1}-1} c_i X^i, d(p, f) \leq \theta, \\ \text{with} \\ (c_i) \in \mathscr{R}_{main} \cap \mathscr{R}_{aux}, \\ (c_1, \ldots, c_{t_1}) = \vec{in} \end{array} \right\}, \tag{15}$$

where $d(p, f)$ refers to the interleaved representation of $p(X)$ as $e = 2^{m_1 - k}$ component polynomials of degree $< 2^k$. The protocol is divided into three phases.

*Phase 1.* The first phase is devoted to the second stage of the R1CS. The prover receives the stage-2 challenge $\vec{\xi} \leftarrow\$ \mathbb{F}_q^{t_2}$ from the verifier, and responds with the commitment of the lifted witness $w_2^*(X_1, \ldots, X_{m_1})$.

*Phase 2.* The Spartan zero-check. We prove constraint satisfaction of the multilinears

$$\begin{aligned} w_M(\vec{Y}) &= \sum_{\vec{x}\in H_{m_1+1}} M(\vec{Y},\vec{x})\cdot w(\vec{x}), \\ &= \sum_{\vec{x}\in H_{m_1}} M_1(\vec{Y},\vec{x})\cdot w_1(\vec{x}) + \sum_{\vec{x}\in H_{m_1}} M_2^*(\vec{Y},\vec{x})\cdot w_2^*(\vec{x}) \\ &= \sum_{\vec{x}\in H_{m_2}} M_1(\vec{Y},\vec{x})\cdot w_1(\vec{x}) + \sum_{\vec{x}\in H_{m_2}} M_2(\vec{Y},\vec{x})\cdot w_2(\vec{x}), \end{aligned} \tag{16}$$

for $M = A, B, C$. For showing that

$$w_A(\vec{y})\cdot w_B(\vec{y}) - w_C(\vec{y}) = 0, \quad \text{for all } \vec{y}\in H_m.$$

we run the sumcheck protocol on its inner product with $eq(\,.\,,\vec{\rho})$, where $\vec{\rho} \leftarrow\$ F^m$ is random. This reduces the above claim to evaluation claims

$$v_a = w_A(\vec{\alpha}), \quad v_b = w_B(\vec{\alpha}), \quad v_c = w_C(\vec{\alpha}), \tag{17}$$

where $\vec{\alpha}\in F^m$ are the verifier challenges drawn in the course of the sumcheck rounds.

*Phase 3.* The third phase of the protocol is devoted to proving the linear claims (17), which is done by WHIR on the combined witness $w$. For public input and challenge consistency, we further add the constraints

$$in_j = \sum_{\vec{x}\in H_{m_1}} eq(j,\vec{x})\cdot w_1(\vec{x}), \quad j = 0,\dots,t_1, \tag{18}$$

$$\xi_j = \sum_{\vec{x}\in H_{m_1}} eq^*(j,\vec{x})\cdot w_2^*(\vec{x}), \quad j = 0,\dots,t_2, \tag{19}$$

where $\vec{\xi} = (\xi_1,\dots,\xi_{t_2})$ are the challenges drawn in the second stage of the R1CS. The WHIR proof of proximity eventually reduces the linear claims on $w$ to claims on the combined matrix polynomials, which need to be checked by the verifier at full expense.

Let us give the formal description of our IOPP. For simplicity we assume that

$$t_1 = t_2 = t.$$

Recall that $m_1 \geq m_2$.

**Protocol 1 (Main IOPP).** *The R1CS matrices $A_i, B_i, C_i \in \mathbb{F}_q^{m\times m_i}$, $i = 1, 2$, and public inputs $\vec{in} = (in_0,\dots,in_t) \in \mathbb{F}_q^t$ are known to both the prover and the verifier. The prover has committed to the first-stage witness $\vec{w}_1 \in \mathbb{F}_q^{m_1}$ as interleaved Reed–Solomon codeword $f_1 \in C^e$ as described above, and provided the verifier oracle access.*

1. *R1CS second stage: The verifier samples the stage-2 challenge*

$$\vec{\xi} \leftarrow\$ \mathbb{F}_q^t$$

*and sends them to the prover. The prover computes* $\vec{w}_2 \in \mathbb{F}_q^{m_2}$ *the second-stage witness of the R1CS, commits to it as codeword* $f_2 \in C^e$*, as described above, and provides the verifier oracle access.*
2. *R1CS zero-check: The verifier sends a random*

$$\vec{\rho} \leftarrow\$ F^m,$$

*and both prover and verifier engage in the sumcheck protocol for*

$$\sum_{\vec{x} \in H_m} (w_A(\vec{x}) \cdot w_B(\vec{x}) - w_C(\vec{x})) \cdot eq(\vec{x}, \vec{\rho}) = 0, \tag{20}$$

*where* $w_A(\vec{x}), w_B(\vec{x}), w_C(\vec{x})$ *are the multilinears as specified in* (16)*. After the last round of the sumcheck, the prover sends*

$$v_A = w_A(\vec{\alpha}), \quad v_B = w_B(\vec{\alpha}), \quad v_C = w_C(\vec{\alpha}),$$

*where* $\vec{\alpha} \in F^m$ *are the verifier challenges drawn in the course of the sumcheck, and the verifier checks if* $q_m(\alpha_m) = (v_A \cdot v_B - v_C) \cdot eq(\alpha, \rho)$*.*
3. *WHIR: Both prover and verifier run WHIR for the following linear constraints on the combined witness polynomial* $w$*:*

$$v_M = \sum_{\vec{x} \in H_{m_1}} M_1(\vec{\alpha}, \vec{x}) \cdot w_1(\vec{x}) + \sum_{\vec{x} \in H_{m_1}} M_2^*(\vec{\alpha}, \vec{x}) \cdot w_2^*(\vec{x}) \tag{21}$$

*with* $M = A, B, C$*, and*

$$in_i = \sum_{\vec{x} \in H_{m_1}} eq(i, \vec{x}) \cdot w_1(\vec{x}), \quad \xi_i = \sum_{\vec{x} \in H_{m_2}} eq^*(i, \vec{x}) \cdot w_2^*(\vec{x}), \tag{22}$$

*for* $i = 0, \ldots, t$*. (This phase reduces the validity of the linear constraints to claims on the combined matrix polynomials*

$$A(\vec{\alpha}, \vec{\beta}), \quad B(\vec{\alpha}, \vec{\beta}), \quad C(\vec{\alpha}, \vec{\beta}), \tag{23}$$

*where* $\vec{\beta} \in F^{m_1}$ *are the folding randomnesses drawn in the course of WHIR.)*

*If all verifier checks in Phase 2 and Phase 3, including that of the final matrix values* (23)*, pass, then the verifier accepts. Otherwise, it rejects.*

Note that computing the matrix values (23) requires $O(n)$ field operations, where $n$ is the maximum number of non-zero entries in the R1CS matrices. While these values can be proven via the SPARK polynomial commitment scheme for sparse multilinears [Set20], our main use case pivots for letting the verifier take the burden of matrix evaluation.

**Remark 3.** If WHIR is configured to consume only $m_1 - k_r$ variables, for some small $k_r$, then the matrix polynomials in the final WHIR claim are specialized to $M(\vec{\alpha}, \vec{\beta} \| \,.\,)$, $M = A, B, C$, which are multilinears in the remaining $k_r$ variables.

### 3.4 Soundness.

Without specifying the corresponding soundness notions, let us informally state the soundness guarantees of the protocol, for proximity parameters up to the Johnson radius. This means that $\theta$ as in the main relation (15) satisfies

$$\theta \in \left(0, 1 - \sqrt{1 - \delta_1}\right),$$

where $\delta_1$ is the relative minimum distance of the base code $C_1 = C$. The corresponding list size bound is

$$\ell_1(\theta) = \frac{1}{2\sqrt{1 - \delta_1}} \cdot \frac{1}{\eta},$$

the combinatorial Johnson bound, with $\eta = 1 - \sqrt{1 - \delta_1} - \theta$. WHIR with $r$ rounds involves the further Reed–Solomon codes $C_2, \ldots, C_r$, with each round having its own proximity parameter $\theta_i$, again assumed to be within the Johnson radius. (See Section 4.)

**Theorem 1.** *For proximity parameters $\theta_i \in \left(0, 1 - \sqrt{1 - \delta_i}\right)$, $i = 1, 2, \ldots, r$, where $\delta_i$ are the minimum distances of the Reed–Solomon codes $C_1$, …, $C_r$. Protocol 1 is a round-by-round knowledge sound interactive oracle proof for that $f_1 \in \mathcal{R}_{\theta_1}$ as specified in* (15)*, with the following overall soundness error for the three phases:*

$$\varepsilon_{phase\ 1} \leq \ell_1(\theta_1) \cdot \varepsilon_{R1CS} \tag{24}$$

*where $\varepsilon_{R1CS}$ is the soundness error of the randomized AIR,*

$$\varepsilon_{phase\ 2} \leq \ell_1(\theta_1) \cdot \max\left\{\frac{m}{|F|}, \frac{3}{|F|}\right\}, \tag{25}$$

*and $\varepsilon_{phase\ 3} \leq \varepsilon_{WHIR}(\theta_1, \ldots, \theta_r)$ for the overall soundness error of WHIR with $r$ rounds on the inner product claims* (21) *and* (22)*.*

Let us roughly sketch the analysis behind Theorem 1. First of all, the entire protocol aims to prove a proximate solution of the *lifted* R1CS, specified by $A_1, B_1, C_1$ and the prefix-embedded second stage matrices $A_2^*, B_2^*, C_2^*$. Unlike in certain univariate arithmetizations, see [HHM25], lifting by prefix-embedding does not change the circuit logic – it only introduces variables that are never used. The lifted system has the same soundness error as the original one, i.e. $\varepsilon_{R1CS^*} = \varepsilon_{R1CS}$, and from the point of view of $w_1$, it is equivalent to ask for a second stage response satisfying the lifted system, or a smaller witness satisfying the original system.[1] This is the reason why Phase 1 is a randomized reduction

[1] In fact, a solution of the non-lifted system is simply obtained by truncation.

from the main claim $f_1 \in \mathcal{R}_{\theta_1}$ to that the extended prover transcript $(f_1, f_2^*)$ belongs to

$$\mathcal{R}_{\text{phase 1}} = \left\{ \begin{array}{c} f_1 \in \left(\mathbb{F}_q^D\right)^e, \\ f_2^* \in \left(\mathbb{F}_q^D\right)^e \end{array} : \begin{array}{c} \exists (p_1(X), p_2(X)) \in \mathscr{P}_{m_1}^2 \\ d((p_1, p_2), (f_1, f_2^*)) \leq \theta_1, \\ \text{and their coefficient vectors} \\ (\vec{c}_1, \vec{c}_2) \text{ satisfy the lifted R1CS} \\ \text{on inputs } \vec{in}, \vec{\xi} \end{array} \right\}.$$

(Here, the distance refers to the interleaved representation of $p_1$ and $p_2$.) Note, that the proximate $p_2(X)$ is not demanded of lifted form, in other words, it is not asked to stem from a $\theta_1$-proximate of the non-lifted word $f_2$ on $D'$. (In fact, we do not rule out that $f_2$ is not as close to $C'$.) A usual pigeon-hole argument on the number of $\theta_1$-proximates of $(f_1, f_2^*)$ bounds the soundness error of this reduction by

$$\ell_1(\theta_1) \cdot \varepsilon_{R1CS^*} = \ell_1(\theta_1) \cdot \varepsilon_{R1CS},$$

as stated in the theorem.

The same list size bound also scales the soundness errors of Phase 2, which is a reduction from $\mathcal{R}_{\text{phase 1}}$ to that the prover transcript $f_1, f_2^*$, extended by its proposed sumcheck polynomials $q_1(X), \ldots, q_m(X)$ and the final claims $v_A, v_B, v_C$, belongs to

$$\mathcal{R}_{\text{phase 2}} = \left\{ \begin{array}{c} f_1 \in \left(\mathbb{F}_q^D\right)^e, \\ f_2^* \in \left(\mathbb{F}_q^D\right)^{e_2}, \\ q_1(X), \ldots, q_m(X) \in F[X]^{\leq 3} \\ v_A, v_B, v_C \in F \end{array} : \begin{array}{c} \exists (p_1(X), p_2(X)) \in \mathscr{P}_{m_1}^2 \\ d((p_1, p_2), (f_1, f_2^*)) \leq \theta_1, \\ \text{and} \\ 0 = q_1(0) + q_1(1), \\ q_i(\alpha_i) = q_{i+1}(0) + q_{i+1}(1), \\ \text{for } i = 1, \ldots, m-1, \\ q_m(\alpha_m) = (v_A \cdot v_B - v_C) \cdot eq(\vec{\alpha}, \vec{\rho}) \end{array} \right\}.$$

The first term of the soundness error (25),

$$\ell_1(\theta_1) \cdot \frac{m}{|F|}$$

attributes the reduction from the domain identity to the inner product with $eq(\,.\,, \vec{\rho})$. The other term,

$$\ell_1(\theta_1) \cdot \frac{3}{|F|},$$

is the soundness errors of each of the $m$ rounds of the sumcheck protocol, on the cubic zero-check expression.

Finally, and without going into details, Phase 3 is a reduction from $\mathcal{R}_{\text{phase 2}}$ to the final verifier relation $\mathcal{R}_V$, which trades the correlated agreement claim

$$\exists (p_1(X), p_2(X)) \in \mathscr{P}_{m_1}^2, \quad d((p_1, p_2), (f_1, f_2^*)) \leq \theta_1,$$

for consistency checks on the WHIR sumcheck polynomials and the aggregated WHIR claim on the final polynomial revealed in plain. (The latter claim involves

the evaluation claims on the matrix polynomials $M(\vec{\alpha}, \vec{\beta})$.) Its soundness error is $\varepsilon_{\mathrm{WHIR}}(\theta_1, \ldots, \theta_r)$, the overall error of WHIR with $r$ rounds on the five claims (21) and (22). We describe the protocol in the next section.

# 4 WHIR

For completeness we give a concise description of the WHIR [ACFY24] proof of proximity to constrained Reed–Solomon codes. Given a word defined on some set, $g_0 : D \longrightarrow \mathbb{F}_q$, WHIR [ACFY24] is an interactive oracle proof of proximity to a univariate polynomial $u_P(X)$, the multivariate representation of which, say $P(X_1, \ldots, X_{k_0})$, satisfies a linear claim of the form

$$c_0 = \langle P_0, K_0 \rangle_{H_{k_0}}, \tag{26}$$

where $K_0(X_1, \ldots, X_{k_0})$ is a publicly known (and often succinctly evaluable) multilinear. Similar to Basefold [ZCF23] and preceding works, e.g. [BCHO22,CBBZ22] [Ham22,KT23], the construction of WHIR rests on the connection between multilinear sumcheck specialization and the FRI-like folding of their univariate representations.

## 4.1 Setup

The mechanics of WHIR is in alignment with a specified sequence of univariate spaces

$$\mathscr{P}_{k_0} \supset \mathscr{P}_{k_1} \supset \ldots \supset \mathscr{P}_{k_r}, \tag{27}$$

with $k_0 > k_1 > \ldots > k_r \geq 0$. (These will correspond to the number of remaining variables after each round.) As before, we commit polynomials in interleaved representation, using the linear isomorphism

$$\mathscr{P}_{k_i} \simeq \mathscr{P}_{k_{i+1}}^{e_{i+1}}, \quad e_{i+1} = \dim \mathscr{P}_{k_i} / \mathscr{P} k_{i+1},$$

by FFT decomposition (6), for each $i = 0, \ldots, r-1$. The decomposed polynomials, elements from

$$\mathscr{P}_{k_1}^{e_1}, \mathscr{P}_{k_2}^{e_2}, \ldots, \mathscr{P}_{k_r}^{e_r}, \tag{28}$$

are then committed as words from the interleaved codes

$$C_1^{e_1}, C_2^{e_2}, \ldots, C_r^{e_r}, \tag{29}$$

with specified base codes $C_i = \mathsf{RS}\left[F, D_i, 2^{k_i}\right]$ of (typically increasing) minimum distance. Their evaluation domains $D_i$ are chosen as multiplicative cosets[2]in $\mathbb{F}_q$, and are of size $|D_i| = 2^{n_i}$ for some $n_i > k_i$. Each round further specifies $s_i \geq 1$ for the number of samples in the consistency check between adjacent oracles, as described below.

[2] More generally, disjoint unions of multiplicative cosets of size $2^{k_i}$.

### 4.2 The IOPP.

In the following $\mathbb{F}_q$ is an FFT-friendly prime field, and $F$ a finite extension of it. The protocol starts with initial claim (26) and consists of $r$ rounds. Each round $i = 1, \ldots, r$ reduces a claim $c_{i-1} = \langle P_{i-1}, K_{i-1} \rangle_{H_{k_{i-1}}}$ on the previous polynomial $P_{i-1}$ to a claim $c_i = \langle P_i, K_i \rangle_{H_{k_i}}$ on a new polynomial.

**Protocol 2 (WHIR).** *The prover has a multilinear $P_0 \in \mathbb{F}_q[X_1, \ldots, X_{k_0}]$, satisfying the inner-product claim*

$$c_0 = \langle P_0, K_0 \rangle_{H_{k_0}}$$

*with some public multilinear $K_0 \in F[X_1, \ldots, X_{k_0}]$. The verifier has oracle access to $g_0 \in (\mathbb{F}_q^D)^{e_1}$ the commitment of $P_0$'s univariate representation $u_{P_0}(X) \in \mathscr{P}_{k_0}$ as interleaved word from $C_1^{e_1}$.*

*Both prover and verifier engage in the following $r$ rounds, for $i = 1, \ldots, r$, each consisting of three phases.*

1. *Sumcheck for $r_i = k_{i-1} - k_i$ rounds, on the previous inner product claim between $P_{i-1}$ and $K_{i-1}$. In the course of these rounds, the prover has sent the quadratic refinements polynomials $q_{i,1}(X)$, …$q_{i,r_i}(X)$, and the verifier has answered with $\vec{\beta}_i = (\beta_{i,1}, \ldots \beta_{1,r_i}) \in F^{r_i}$. The prover commits to*

   $$P_i = P_{i-1}(\vec{\beta}_i, \ .),$$

   *as the polynomial $u_{P_i}(X) \in \mathscr{P}_{k_i}$ represented by the interleaved word*

   $$g_i \in C_{i+1}^{e_{i+1}},$$

   *and provides the verifier oracle access. (In the case $i = r$, the prover reveals the polynomial $u_{P_r}(X)$ in plain.)*
2. *Query phase.*
   (a) *The verifier draws a DEEP query $x_0 \leftarrow\$ F$, on which the prover answers with the value $v_0$ of the univariate representations $u_{P_i}(X)$ at $x_0$.*
   (b) *The verifier queries $g_{i-1}$ from the previous round at $s_i > 0$ random positions*

   $$x_{i,1}, \ldots, x_{i,s_i} \leftarrow\$ D_i,$$

   *and computes the expected values $v_1, \ldots, v_{s_i}$ for $u_{P_i}(X)$ at these points from the oracle answers, and $\vec{\beta}_i$.*
3. *Aggregation. The verifier samples $\gamma_i \leftarrow\$ F$ for combining the specialized sumcheck claim*

   $$q_{i,r_i}(\beta_{i,r_i}) = \langle P_i, K_{i-1}(\vec{\beta}_i, \ .) \rangle_{H_{k_i}}$$

   *and the evaluation claims*

   $$v_j = \langle P_i, L_{x_i} \rangle_{H_{k_i}}, \quad j = 0, \ldots, s_i,$$

   *into a the new inner product claim $c_i = \langle P_i, K_i \rangle_{H_{k_i}}$ with the new $c_i$ and kernel $K_i$ as clarified below.*

*If the consistency checks in all the sumcheck rounds hold, and if the final inner product claim $c_r = \langle P_r, K_r \rangle$ holds for the multilinear representation of the revealed polynomial $u_{P_r}(X)$, then the verifier accepts.*

In the aggregation step, the combined kernel is

$$K_i = K_{i-1,i,1}(\vec{\beta}_i, X_1, \ldots, X_{k_i}) + \sum_{j=0}^{s_i} \gamma_i^{j+1} \cdot L_{x_j}(X_1, \ldots, X_{k_i}),$$

and $c_i = q_{i,r_i}(\beta_{i,r_i}) + \sum_{j=0}^{s_i} \gamma_i^{j+1} \cdot v_j$.

### 4.3 Soundness.

Every round of WHIR is a randomized reduction $\mathcal{R}_{i-1} \longrightarrow \mathcal{R}_i$ in the verifier randomnesses $\beta_{i,1}, \ldots, \beta_{i,r_i}$, $x_{i,0}, x_{i,1}, \ldots, x_{i,s_i}$, and $\gamma_i$, where

$$\mathcal{R}_i = \left\{ g_i \in (\mathbb{F}_q^{D_i})^{e_i} \ : \ \begin{array}{c} \exists p_i(X) \in \mathscr{P}_{k_i}, \\ d(g_i, p_i) \leq \theta_i, \\ \langle P_i, K_i \rangle_{H_{k_i}} = c_i \end{array} \right\}. \tag{30}$$

(Again, $P_i$ is the multilinear representation of $p_i$, and the distance refers to the interleaved representation of $p_i$.) We use somewhat lazy notation here, hiding the dependency of $\mathcal{R}_i$ on the previous messages: The inner product kernel $K_i$ incorporates the specialized sumcheck claim on the previous oracle $g_{i-1}$, its values at the $s_i$ in-domain queries, and the claimed value for $p_i$ at the single out-of-domain point $x_{i,0}$. We note that in the case $i = r$, since $p_r(X) \in \mathscr{P}_{k_r}$ is revealed in plain, $g_r = p_r$, the distance condition in (30) becomes vacuous.

The soundness analysis of the sumcheck phase, in parallel with the multilinear combinations of the interleaved components, is the most technically involved part. Its proof relies on (degree-2) curve decodability of Reed–Solomon codes, yielding correlated agreement under the additional constraints imposed by the inner products. Without specifying the intermediate relations for each step of a WHIR round, we state its soundness properties as an (interactive oracle) reduction. For a detailed analysis we refer to [Hab24, Section 6.4], which generalizes in a straight-forward manner to arbitrary inner product constraints.

**Theorem 2 (WHIR Soundness).** *For $\theta_i \in (0, 1 - \sqrt{1 - \delta_i})$, where $\delta_i$ is the minimum distance of the Reed–Solomon code $C_i$. Then round $i$ in Protocol 2 is round-by-round (knowledge) sound oracle reduction $\mathcal{R}_{i-1} \longrightarrow R_i$, with the following soundness errors for its steps.*

1. *Sumcheck phase: For each of the $r_i$ steps,*

$$\varepsilon_{S,i} = \varepsilon_{CA}(C_{i-1}, 2, \theta_{i-1}),$$

*where $\varepsilon_{CA}(C_{i-1}, 2, \theta_{i-1})$ is the soundness error of the constrained correlated agreement theorem (Theorem 3) for $C_{i-1}$, under degree-2 constraints, using proximity parameter $\theta_{i-1}$.*

2. *Query phase:*

$$\varepsilon_{Q,i} = \frac{\ell_i(\theta_i)^2}{2} \cdot \frac{2^{n-k_i}}{|F \setminus D_i|} + (1-\theta_{i-1})^{s_i},$$

*where $\ell_i(\theta_i)$ is the list size bound for $C_i$ at distance $\theta_i$.*

3. *Aggregation step:*

$$\varepsilon_{A,i} = \ell_i(\theta_i) \cdot \frac{s_i}{|F|},$$

*again with the same list size bound as above.*

**Remark 4.** For the soundness analysis, one may alternatively use the high-level notion of mutual correlated agreement [ACFY24], a consequence of curve-decodability [Hab25,GG25].

## 5 Adding zero-knowledge

### 5.1 Randomizing $w_M(\vec{\alpha})$.

Recall that in the end of Phase 2 of Protocol 1, the prover reveals the values $w_A(\vec{\alpha})$, $w_B(\vec{\alpha})$, $w_C(\vec{\alpha})$, which are subject to

$$q_m(\alpha_m) = (w_A(\vec{\alpha}) \cdot w_B(\vec{\alpha}) - w_C(\vec{\alpha})) \cdot eq(\vec{\alpha}, \vec{\rho}), \tag{31}$$

where $\alpha \in F^m$ are the random challenges collected over the rounds of the sum-check. In this section we discuss two different strategies to prevent these values from leaking private witness data. Both extend the witness (either $w_1$ or $w_2$) by random values, and add trivial gates to assure their propagation through the R1CS matrices.

**Randomization by extension field elements.** This is the canonical solution, where the openings over $F$ are randomized by random witnesses over $F$. Note that it is sufficient to randomize $w_A(\vec{\alpha})$ and $w_B(\vec{\alpha})$; the value of $w_C(\vec{\alpha})$ is then uniquely determined by (31), as long as $eq(\vec{\alpha}, \vec{\rho}) \neq 0$.

We allocate two private extension field variables $R_A, R_B$ in the second stage R1CS, and set their values to $r_A, r_B \leftarrow\$ F$. Their gates are

$$R_A \cdot 1 = R_A, \quad 1 \cdot R_B = R_B,$$

located at arbitrary row indices $\vec{x}_A, \vec{x}_B \in H_m$. Then

$$w_A(\vec{Y}) = w'_A(\vec{Y}) + r_A \cdot eq(\vec{x}_A, \vec{Y}), \tag{32}$$

$$w_B(\vec{Y}) = w'_B(\vec{Y}) + r_B \cdot eq(\vec{x}_B, \vec{Y}) \tag{33}$$

with $w'_A$, $w'_B$, not depending on the two random witnesses $r_A$, $r_B$.

**Lemma 1.** *For any $\vec{\rho} \in F^m$. Except for a fraction of $\varepsilon \leq \frac{3m}{|F|}$ points $\vec{\alpha} \in F^m$, the distribution of $(w_A(\vec{\alpha}), w_B(\vec{\alpha}), w_C(\vec{\alpha}))$ is uniform on the set*

$$\left\{(v_A, v_B, v_C) \in F^3 \;:\; q_m(\alpha_m) = (v_A \cdot v_B - v_C) \cdot eq(\vec{\alpha}, \vec{\rho})\right\}.$$

*Proof.* The exceptional set is the union of the roots of the multilinears $eq(\vec{x}_A, \,.\,)$, $eq(\vec{x}_B, \,.\,)$ and $eq(\,.\,, \vec{\rho})$. For all other $\vec{\alpha} \in F^m$, the vector $(w_A(\vec{\alpha}), w_B(\vec{\alpha}))$ is uniformly distributed over $F^2$, since the coefficients of $r_A$ and $r_B$ do not vanish in (32) and (33), and they uniquely determine $w_C(\vec{\alpha})$ via the claim (31). □

The bound in Lemma 1 is essentially tight, as multivariate Lagrangians have exactly a fraction of $m/|F|$ zeros in $F^m$. In other words, local randomization of the witness propagates only to *almost all* points outside the witness domain.[3] However, the fraction of exceptional points is of order $1/|F|$, and excluding them does not conflict our soundness goals.

For completeness we also cite an extension to multiple evaluation points:

**Lemma 2.** *Choose $\Omega = \{\vec{x}_1, \ldots, \vec{x}_k\}$ a set of $k \geq 1$ distinct points on the Boolean hypercube $H_n$. Then, except with probability of at most $\varepsilon = \frac{k \cdot n}{|F|}$ in $\vec{\alpha}_1, \ldots, \vec{\alpha}_k \in F^n$, the multi-point evaluation matrix*

$$E_{\vec{\alpha}_1,\ldots,\vec{\alpha}_k} = \begin{pmatrix} eq(\vec{x}_1, \vec{\alpha}_1) & eq(\vec{x}_1, \vec{\alpha}_2) & \ldots & eq(\vec{x}_1, \vec{\alpha}_k) \\ eq(\vec{x}_2, \vec{\alpha}_1) & eq(\vec{x}_2, \vec{\alpha}_2) & \ldots & eq(\vec{x}_2, \vec{\alpha}_k) \\ \vdots & \vdots & & \vdots \\ eq(\vec{x}_k, \vec{\alpha}_1) & eq(\vec{x}_k, \vec{\alpha}_2) & \ldots & eq(\vec{x}_k, \vec{\alpha}_k) \end{pmatrix}$$

*is of full rank.*

The proof shows that the determinant is a non-zero multilinear in the coordinates of $\vec{\alpha}_1, \ldots, \vec{\alpha}_k$. We postpone it to the appendix.

**Randomization by basefield elements.** It is tempting to believe that the univariate strategy from [HK24], which randomizes an extension field opening by extension-field-degree many basefield values, carries over to the multilinear world in a straight-forward manner. However, the best generic result we found is the following one, which has a way too large bound for the exceptional set.

**Lemma 3.** *Let $F$ be a degree $e$ extension of a prime field $\mathbb{F}_q$, and $\Omega = \{\vec{x}_1, \ldots, \vec{x}_e\}$ a set of $e$ distinct points on the Boolean hypercube $H_m \subset \mathbb{F}_q^m$. Then, except for a fraction of at most $\varepsilon = \frac{m}{q-1}$ points $\vec{\alpha} \in F^m$, the elements*

$$eq(\vec{x}_1, \vec{\alpha}), \ldots, eq(\vec{x}_e, \vec{\alpha}) \in F$$

*are $\mathbb{F}_q$-linear independent.*

[3] This diverges from univariate IOPs, where local randomization propagates to *all* points outside the witness domain.

The proof of the lemma relies on the well-known Moore determinant as a distinguisher for $\mathbb{F}_q$-linear independence. However, we require the property also over $K$ the field of rational functions in several variables, and hence state it in full generality.

**Lemma 4.** *Let $K$ be a (possibly infinite) field of characteristic $q < \infty$, and let $f_1, \ldots, f_k \in K$, $k \geq 1$. Then $f_1, \ldots, f_k$ are $\mathbb{F}_q$-linear independent if and only if the determinant of the Moore matrix*

$$M(f_1, \ldots, f_k) = \begin{pmatrix} f_1 & f_2 & \cdots & f_k \\ f_1^q & f_2^q & \cdots & f_k^q \\ \vdots & \vdots & \ddots & \vdots \\ f_q^{q^{k-1}} & f_2^{q^{k-1}} & \cdots & f_k^{q^{k-1}} \end{pmatrix}$$

*is a non-zero element in $K$.*

For a proof of Lemma 4, we refer the reader to [Gos96, Lemma 1.3.3].

*Proof (of Lemma 3).* Since the points $\vec{x}_1, \ldots, \vec{x}_n$ are distinct, the Lagrangians $eq(\vec{x}_i, \vec{Y})$, $i = 1, \ldots, e$, are $\mathbb{F}_q$-linear independent elements of $K = \mathbb{F}_p(\vec{Y})$, the rational function field in $m$ variables. By Lemma 4, their Moore determinant

$$\det M(eq(\vec{x}_1, \vec{Y}), \ldots, eq(\vec{x}_e, \vec{Y})) \neq 0,$$

as an element of $K$. Moreover, since all matrix entries are polynomials, we conclude that the Moore determinant is a non-zero *polynomial*. The $(i, j)$-th entry in the Moore matrix is has total degree at most $m \cdot q^{i-1}$, and the total degree the determinant is at most

$$m + mq + \ldots + mq^{e-1} = m \cdot \frac{q^e - 1}{q - 1}.$$

By Schwartz-Zippel, the fraction of $\vec{\alpha} \in F^m$ for which the determinant evaluates zero is at most

$$\frac{m}{q-1} \cdot \frac{q^e - 1}{q^e} < \frac{m}{q-1}.$$

For all other $\vec{\alpha}$ the determinant is non-zero, and another application of Lemma 4 yields that $eq(\vec{x}_1, \vec{\alpha}), \ldots, eq(\vec{x}_e, \vec{\alpha})$ are $\mathbb{F}_p$-linear independent elements of $F$. $\square$

It is hard to tell whether the Schwartz–Zippel bound in Lemma 3 is tight. A high polynomial degree is not a guarantee for a large set of zeros. We tried shortly with GPT-5.6, but did not achieve significant improvements. However, it proposed the following work around, which explicitly uses the multiplicative structure of the Lagrangians.

*Blowing up soundness.* Instead of a single set $\Omega$ of size $e$, we choose several sets

$$\Omega_1, \dots \Omega_s \subset H_m,$$

in a very particular manner, so that their exceptional sets are statistically independent. (The latter will be possible because of the tensor structure of the Lagrangians.) Each $\Omega_j$ is a set of $e$ points contained in a sub-cube of small dimension

$$t = \lceil \log_2 e \rceil,$$

with its free part touching the variables $Y_{i_j(1)}, \dots, Y_{i_j(t)}$ only, meaning that its points do not vary outside these $t$ coordinates. We further require that the free variables are disjoint between the sets, i.e.

$$\left\{Y_{i_j(1)}, \dots, Y_{i_j(t)}\right\} \cap \left\{Y_{i_{j'}(1)}, \dots, Y_{i_{j'}(t)}\right\} = \varnothing,$$

whenever $j \neq j'$. The crux is that for each such set $\Omega_j$, we can drop its fixed variables in the proof of Lemma 3, as long as their *eq*-terms are not zero, since a common non-zero factor does not change $\mathbb{F}_q$-linear independence. The exceptional set for $\Omega_j$ is only in the variables $Y_{i_j(1)}, \dots, Y_{i_j(t)}$, independent of the other variables, and we arrive at the following statement.

**Lemma 5.** *With $\Omega_1, \dots, \Omega_s \subset H_m$ as above. Then except for an $\varepsilon$-fraction of points $\vec{\alpha} \in F^m$, where*

$$\varepsilon < s \cdot \frac{m-t}{|F|} + \left(\frac{t}{q-1}\right)^s, \tag{34}$$

*at least one of the sets $\Omega = \Omega_j$ is such that its values $eq(\vec{x}, \vec{\alpha})$, $\vec{x} \in \Omega$, are $\mathbb{F}_q$-linearly independent.*

*Proof.* The first term bounds the the probability that $\vec{\alpha} \in F^m$ hits one of the $m - t$ fixed variables of $\Omega_j$, times $e$ for the union bound over all sets. For all other $\vec{\alpha}$, the exceptional sets in Lemma 3 are statistically independent, and of relative size $t/(q-1)$ each. □

In our application we would use $s = e$, to make the second term in (34) of order $O(1/|F|)$. That is, we would require $e^2$ random witnesses over $\mathbb{F}_q$ to randomize $w_A(\alpha)$, and another $e^2$ random witnesses for $w_B(\alpha)$. They can be placed at arbitrary positions in the witness, yet their constraints (as before of the form $R_j \cdot 1 = R_j$, or $1 \cdot R_j = R_j$) have to respect the positioning described above.

### 5.2 Zero-check masking

The zero-check, Phase 2 in Protocol 1, is turned into zero-knowledge by using the sumcheck randomization from Libra [XZZ+19]. The prover commits to the univariate polynomials

$$h_0 \leftarrow\$ F, \quad h_1(X), \dots, h_m(X) \leftarrow\$ F[X]^{<d}, \tag{35}$$

where $d = 3$ is the degree of the sumcheck (20). These polynomials define the sparse mask

$$h(X_1, \ldots, X_n) = h_0 + X_1 \cdot h_1(X_1) + \ldots + X_m \cdot h_n(X_m), \tag{36}$$

which is overlaid with the hypercube sum (20). The prover claims the sum $c' = \sum_{\vec{x} \in H_m} h_m(X_m)$, the computation of which is almost for free via the succinct representation

$$\frac{1}{2^m} \sum_{\vec{x} \in H_m} h(\vec{x}) = h_0 + \frac{1}{2} \left( h_1(1) + \ldots + h_m(1) \right).$$

Then both run the sumcheck on the combined claim

$$c_0 = \alpha \cdot c + c' = \sum_{\vec{x} \in H_n} \alpha \cdot G(\vec{x}) + h, \tag{37}$$

where $\alpha \leftarrow\$ F$ is random, and $c = \sum_{x \in H_n} G(x_1, \ldots, x_n)$ is the target claim. (In our case $G$ is be the multivariate zero-check term, and $c = 0$.)

The sparse mask carries sufficient entropy to perfectly hide the witness in the sumcheck polynomials

$$\hat{q}_k(X) = \sum_{\vec{x} \in H_{n-k}} \alpha \cdot G(\alpha_1, \ldots, \alpha_{k-1}, X, \vec{x}) + h(\alpha_1, \ldots, \alpha_{k-1}, X, \vec{x}),$$

for $k = 1, \ldots, n$, as the following lemma shows.

**Lemma 6.** *For any $\alpha \in F$, $(\alpha_1, \ldots, \alpha_n) \in F^n$. The distribution of the sumcheck prover messages $c_0, \hat{q}_1(X), \ldots, \hat{q}_n(X)$ is uniform on the set*

$$\mathcal{R} = \left\{ c_0 \in F, \hat{q}_1, \ldots, \hat{q}_n \in F[X]^{\leq d} \; : \; \begin{array}{c} c_0 = \hat{q}_1(0) + \hat{q}_1(1), \\ \hat{q}_k(\alpha_k) = \hat{q}_{k+1}(0) + \hat{q}_{k+1}(1), \\ k = 1, \ldots n-1 \end{array} \right\},$$

*independent of $G$.*

For the sake of completeness, we give a proof the lemma.

*Proof.* Fix $G$, $\alpha \in F$ and $(\alpha_1, \ldots, \alpha_n) \in F^n$, and choose any set $S \subset F$ of $d$ distinct points, containing the element 1, but not 0. Using the consistency equations it is easily seen that the linear mapping, which sends $(c_0, \hat{q}_1(X), \ldots, \hat{q}_n(X))$ to the values

$$(\vec{v}_1, \ldots, \vec{v}_n) = (\hat{q}_1|_{S \cup \{0\}}, \hat{q}_1|_S, \ldots, \hat{q}_n|_S) \tag{38}$$

is a bijection between $\mathcal{R}$ and $F^{dn+1}$. We leave this to the reader.[4] To prove the lemma, it suffices to show that the values (38) are in one-to-one correspondence

[4] The values $\vec{v}_1$ uniquely determine the polynomial $\hat{q}_1(X) \in F[X]^{\leq d}$ by degree, and with it $c_0 = \hat{q}_1(0) + \hat{q}_1(1)$. Likewise, $\hat{q}_1(\alpha_1)$ together with the values $\vec{v}_2$ uniquely determine $\hat{q}_2(X) \in F[X]^{\leq d}$ satisfying $\hat{q}_1(\alpha_1) = \hat{q}_2(0) + \hat{q}_2(1)$, since the letter determines the value of $\hat{q}_2(X)$ at the additional point 0. And so on.

with $h_0 \in F, h_1, \ldots, h_n \in F[X]^{<d}$. But this is evident from the concrete form

$$\begin{aligned}\hat{q}_k(X) = \sum_{\vec{x} \in H_{n-k}} & \alpha \cdot G(\alpha_1, \ldots, \alpha_{k-1}, X, \vec{x}) \\ & + 2^{n-k}(h_0 + \alpha_1 h_1(\alpha_1) + \ldots + \alpha_{k-1} h_{k-1}(\alpha_{k-1}) + X h_k(X)) \\ & \qquad + \sum_{\vec{x} \in H_{n-k}} x_{k+1} h_{k+1}(x_{k+1}) + \ldots + x_n h_n(x_n),\end{aligned}$$

in which the other $h_0, \ldots, h_{k-1}, h_{k+1}, \ldots, h_n$ impact only the constant part of the polynomial. (The representation partially collapses in the two edge cases. For $k = 1$ there are no terms except $h_0$ preceding $X h_1(X)$, and for $k = n$, there are no terms following $X h_n(X)$.) □

The sumcheck reduces the claim (37) to that

$$\hat{q}_m(\alpha_m) = G(\alpha_1, \ldots, \alpha_m) + h_1(\alpha_1) + \ldots + h_m(\alpha_m),$$

where

$$G(\vec{\alpha}) = (w_A(\vec{\alpha}) \cdot w_B(\vec{\alpha}) - w_C(\vec{\alpha})) \cdot eq(\vec{\alpha}, \vec{\rho}),$$

for which the prover sends the values $w_A(\vec{\alpha})$, $w_B(\vec{\alpha})$, $w_C(\vec{\alpha})$. The complementary term

$$h_1(\alpha_1) + \ldots + h_m(\alpha_m) = \hat{q}_m(\alpha_m) - G(\vec{\alpha}) \tag{39}$$

can be proven in zero-knowledge by a separate protocol.

**Remark 5.** In our current implementation, we append the $dm+1$ coefficients[5] for $h_0, h_1, \ldots, h_m$ to the first stage witness $\vec{w}_1$, and use zk-WHIR with the additional constraint

$$\hat{q}_m(\alpha_m) - G(\vec{\alpha}) = \sum_{\vec{x} \in H_{m_1}} I_\alpha(\vec{x}) \cdot w_1(\vec{x}), \tag{40}$$

where $I_\alpha$ is the inner product kernel for $h_1(\alpha_1) + \ldots + h_m(\alpha_m)$. This requires the witness to be sufficiently below a power two. See the complete protocol in Section A.

Let us quickly sketch the transcript simulator for the zero-check. The prover chooses arbitrary random polynomials $w_1(X_1, \ldots, X_{m_1})$ and $w_2(X_1, \ldots, X_{m_2})$, with their image polynomials $w_A, w_B, w_C$ not satisfying the zero-check hypercube sum for random $\vec{\rho} \leftarrow\$ F$. Then it samples $h_0, h_1, \ldots, h_n$ and $\lambda \neq 0$ and computes the hypercube sum

$$c_0 = \sum G + \lambda h. \tag{41}$$

It then runs the sumcheck protocol on the valid claim (41). The (wrong) claim $c'$ is chosen adaptively so that $\alpha \cdot 0 + c' = c_0$. For details, see Section **??**.

[5] Alternatively, one may directly use the representation by their values (38).

### 5.3 Masked commitments.

To avoid witness information leakage from commitment openings, we use a slightly different approach than [CFW26]. Instead of committing to a zero-knowledge encoding of witness, we overlay a witness polynomial with a sparse random mask without degree increase: Given witness $p \in \mathscr{P}_m$ and random $\mathsf{msk} \leftarrow\$ F[X]^{<q_i}$, for some small $q_i < 2^m$, we commit to

$$\hat{p} = p + \mathsf{msk}, \tag{42}$$

which is again from $\mathscr{P}_m$. (This secures the witness for up to $q_i$ openings.) Obviously this does not bind the witness, unless we also commit to $\mathsf{msk}$, which is required for ZO0K anyways (see Section 5.4).

For practical reasons, we *jointly* commit all[6]masks required by the protocol in one shot. We concatenate their coefficient vectors, append $q_{zk}$ additional random coefficients, and encode the resulting vector as a single word from

$$C_{zk} = \mathsf{RS}[F, D_{zk}, d_{zk}]. \tag{43}$$

Here the degree bound $d_{zk}$ is at least $q_{zk}$ larger than the total of all mask lengths, and can be chosen again a power of two, if this helps encoding performance. See Section 5.4 for details. (The choice of $q_{zk}$ depends on the number of openings in the final proof of proximity, Protocol 3.)

The rationale behind masked commitments is the following. Adding the small mask $\mathsf{msk}$ in coefficient representation is a local modification of the witness, and the subsequent encoding of $\hat{p}$ is as costly as that of $p$. In contrast, even a tiny degree increase of the witness polynomial would lead to an $O(m)$ overhead in the encoding step, where $m$ is the degree of $p$.

### 5.4 WHIR in zero-knowledge

We describe an adaptation of ZO0K [CFW26] to our technique for randomizing commitments as described in the previous section. The changes to the original protocol are minimal, and based on the fact that we can now also view masked polynomials as multilinears in the sumcheck claim. See Remark 6 for a one-to-one variant of [CFW26].

The sumcheck rounds are randomized using the sparse Libra mask explained in Section 5.2, and information leakage via commitment openings is mitigated through the commitment masks as described above. In this multi-polynomial regime, ZO0K considers WHIR as a sumcheck prover, which gradually reduces composite claims of the form

$$c_i = \left\langle \hat{p}_i, K^{(i)} \right\rangle_{H_{k_i}} + \sum_{j=0}^{i} \Gamma_j^{(i)}(\mathsf{msk}_j) + \sum_{j=1}^{i} \Lambda_j^{(i)}(h_j), \tag{44}$$

[6] These are the masks for the two witness polynomials $w_1$ and $w_2$, and the further masks $\mathsf{msk}_1, \ldots, \mathsf{msk}_r$, $h_1, \ldots, h_r$ required for ZO0K.

involving a costly part, the inner product $\langle \hat{p}_i, K^{(i)} \rangle$ of a large (masked) witness polynomial $\hat{p}_i$ with a public kernel $K^{(i)}$, and a cheap part, certain public linear functionals $\Lambda_j^{(i)}$, $\Gamma_j^{(i)}$ on the (small) masks committed by the prover so far:[7] The sparse sumcheck masks $h_1, \ldots, h_i$ and the witness masks $\mathsf{msk}_0, \ldots, \mathsf{msk}_i$. The cheap part is not further reduced by the sumcheck logic; it is treated as constant function on the hypercube $H_{k_i}$, which is passed through the rounds in a weighted form. The costly part is reduced in the usual sense, by folding. However, not before a fresh sumcheck mask $h_{i+1}$ enters the sum. In this manner, the number of masks in the claim (44) grow from round to round, but the costly part shrinks.

After the final round, the remaining claim, a linear functional in $\hat{p}_r, h_1, \ldots, h_r$, $\mathsf{msk}_0, \ldots, \mathsf{msk}_r$, is proven in zero-knowledge by an elementary proof of proximity, Protocol 3 below.

**The required masks.** As pointed out in Section 5.3, we assume that the prover has committed beforehand to all the masks, as a single word from $C_{\mathrm{zk}}$. Concretely, each sumcheck mask $h_i$ is committed as coefficient vector

$$h_{i,0} \| h_{i,1} \| \ldots \| h_{i,r_i} \in F^{2r_i+1} \tag{45}$$

from $h_{i,0} \leftarrow\$ F$, and $h_{i,j} \leftarrow\$ F[X]^{<2}$, appended by the coefficients of the commitment masks

$$msk_i \in F^{q_i}, \quad i = 0, \ldots, r, \tag{46}$$

from $\mathsf{msk}_i \leftarrow\$ F[X]^{<q_i}$, where $msk_0$ is taken from the subfield $\mathbb{F}_q$, assuming that the first witness polynomial $p_0$ is over $\mathbb{F}_q$. (In our application, our $p_0$ is the composite witness polynomial, both of which have their own basefield mask, and $msk_0 = msk_{0,1} \| msk_{0,2}$.) The number of coefficients is chosen according to how often the commitment of $p_i + \mathsf{msk}_i$ is queried in the course of WHIR:

$$q_0 = s_1, \tag{47}$$

$$q_i = s_{i+1} + 1 \text{ for } 1 \leq i \leq r-1, \tag{48}$$

assuming a single DEEP query in each of the rounds, and

$$q_r = s_{\mathrm{final}}, \tag{49}$$

the number of queries on $\hat{p}_r$ in the final opening protocol, Protocol 3. The concatenated coefficient vectors (45) and (46) are encoded in zero-knowlege manner, by appending to the message another $q_{zk} = s_{\mathrm{zk}} + 1$ random coefficients

$$msk_{zk} \in F^{q_{zk}}, \tag{50}$$

[7] The $\Lambda_j^{(i)}(h_j)$ and $\Gamma_j^{(i)}(\mathsf{msk}_i)$ can be again represented as inner products. However, because of their fundamentally different treatment in the protocol, we write them as linear functionals.

before the Reed–Solomon encoding. For reasons to be explained later, another $e = [F : \mathbb{F}_q]$ random elements $r_0 \leftarrow\$ \mathbb{F}_q^e$ are appended to it. Overall, $C_{zk}$ encodes the joint mask vector

$$\mathsf{m} = h_1 \| \ldots \| h_r \| msk_0 \| \ldots \| msk_r \| msk_{zk} \| r_0 \tag{51}$$

parsed as coefficient vector of $F^{m_{\mathrm{zk}}}$,

$$m_{\mathrm{zk}} = r + k_0 - k_r + q_0 + \ldots + q_r + q_{zk} + e, \tag{52}$$

as Reed–Solomon codeword over $F$ with evaluation domain $D_{zk}$.

**Initialization.** The initial claim is on the unmasked polynomial $p_0 = \hat{p}_0 - \mathsf{msk}_0 \in \mathscr{P}_{k_1}^{e_1}(\mathbb{F}_q)$, and of the form

$$c_0 = \langle \hat{p}_0 - \mathsf{msk}_0, K \rangle_{H_{k_0}},$$

where the inner product kernel $K$ is public. This is of the form (44) with

$$K^{(0)} = K, \quad \Lambda_0^{(0)}(\mathsf{msk}_0) = \langle K, \mathsf{msk}_0 \rangle_{H_{k_0}}.$$

The latter can be easily computed as a linear functional on the coefficient vector $msk_0 \in \mathbb{F}_q^{q_0}$: The multilinear representation of $\mathsf{msk}_0 \in F[X]^{<q_0}$ is non-zero only on the first $q_0$ points of the hypercube and thus

$$\langle K, \mathsf{msk}_0 \rangle_{H_{k_0}} = \sum_{j=0}^{q_0-1} K(j) \cdot msk_{0,j},$$

in the index notation of the hypercube.

**Modifications of a WHIR round.** Let us describe the changes of round $i$ in Protocol 2, $i = 1, \ldots, r$. At the beginning of the round, the prover has already committed to $\hat{p}_{i-1}$, $\mathsf{msk}_0, \ldots, \mathsf{msk}_{i-1}$, $h_1, \ldots h_{i-1}$, satisfying the linear constraint

$$c_{i-1} = \left\langle \hat{p}_{i-1}, K^{(i-1)} \right\rangle_{H_{k_{i-1}}} + \sum_{j=0}^{i-1} \Gamma_j^{(i-1)}(\mathsf{msk}_j) + \sum_{j=1}^{i-1} \Lambda_j^{(i-1)}(h_j) \tag{53}$$

We think of the claim as $c_{i-1} = \sum_{\vec{x} \in H_{k_{i-1}}} G_{i-1}(\vec{x})$ for the quadratic expression

$$G_{i-1}(\vec{x}) = \hat{p}_{i-1}(\vec{x}) \cdot K^{(i-1)}(\vec{x}) - 2^{-k_{i-1}} \cdot B_{i-1},$$

where $B_{i-1}$ is the sum of the mask functionals in (53).

*Step 1: Masking phase.* Identical to the sumcheck randomization in Section 5.2. The prover takes the masks $h_{i,0} \in F$, $h_{i,1}, \ldots, h_{i,r_i} \in F[X]^{<2}$ it committed before running the protocol. They define the sumcheck mask

$$h_i(X_1, \ldots, X_{r_i}) = h_{i,0} + X_1 \cdot h_{i,1}(X_1) + \ldots + X_{r_i} \cdot h_{i,r_i}(X_{r_i}), \tag{54}$$

in as many variables as the sumcheck phase of the WHIR round consumes, $r_i = k_i - k_{i-1}$. It claims

$$d_i = \sum_{\vec{x} \in H_{k_{i-1}}} h_i(x_1, \ldots, x_r).$$

The verifier sends a random $\lambda_i \leftarrow\$ F$, which reduces (53) to the combined claim

$$\lambda_i \cdot c_{i-1} + d_i = \sum_{\vec{x} \in H_{k_{i-1}}} \left(\lambda_i \cdot G_{i-1}(\vec{x}) + h_i(x_1, \ldots, x_{r_i})\right). \tag{55}$$

*Step 2: Sumcheck phase.* Both prover and verifier proceed with the $r_i$ sumcheck rounds on the combined claim (69). In the course of it, the prover provides the quadratic refinement polynomials $\hat{q}_{i,1}(X), \ldots, \hat{q}_{i,r_i}(X)$, the verifier has sent $\vec{\beta}_i = (\beta_{i,1}, \ldots, \beta_{i,r_i})$. After the last sumcheck round, the claim (55) is reduced to

$$\begin{aligned}\hat{q}_{i,r_i}(\beta_{i,r_i}) &= \sum_{\vec{x}' \in H_{k_i}} \Big(\lambda_i \cdot G_{i-1}(\vec{\beta}_i, \vec{x}') + h_i(\vec{\beta}_i)\Big) \\ &= \lambda_i \cdot \Big\langle \hat{p}_{i-1}(\vec{\beta}_i,\ .\,), K^{(i-1)}(\vec{\beta}_i,\ .\,) \Big\rangle_{H_{k_i}} + 2^{k_i - k_{i-1}} \cdot \lambda_i \cdot B_i + 2^{k_i} \cdot h_i(\vec{\beta}_i)\end{aligned}$$

which linear in $\hat{p}_{i-1}(\vec{\beta}_i,\ .\,)$ and the masks, now including $h_i$.

*Step 3: Commit.* The prover takes $\mathsf{msk}_i \in F[X]^{<q_i}$, and commits to

$$\hat{p}_i = \hat{p}_{i-1}(\vec{\beta}_i,\ .\,) + \mathsf{msk}_i(\ .\,)$$

as a word from $\mathscr{P}_{k_i}^{e_i}$. This masked commitment is required also in the last round $i = r$. The new ensemble of polynomials, $\hat{p}_i, \mathsf{msk}_i$, and $\mathsf{msk}_0, \ldots, \mathsf{msk}_{i-1}$, $h_1, \ldots, h_i$, is subject to the linear constraint

$$\begin{aligned}\hat{q}_{i,r_i}(\beta_{i,r_i}) = \Big\langle \hat{p}_i - \mathsf{msk}_i, \lambda_i \cdot K^{(i-1)}(\vec{\beta}_i,\ .\,) \Big\rangle_{H_{k_i}} \\ + 2^{-r_i} \cdot \lambda_i \cdot B_{i-1} + 2^{k_i} \cdot h_i(\vec{\beta}_i),\end{aligned} \tag{56}$$

which are of the form (44).

*Step 4: Query phase.* In the query phase, the verifier asks the masked polynomial $\hat{p}_i$ for the value at the out-of-domain sample $x_{i,0} \leftarrow\$ F$ and uses the values of the previous $\hat{p}_{i-1}$ to constrain $\hat{p}_i - \mathsf{msk}_i$ at the points $x_{i,1}, \ldots, x_{i,s_i} \leftarrow\$ D_i$. In terms of the new ensemble, these claims are the linear constraints

$$\sum_{\vec{x}' \in H_{k_i}} \hat{p}_i(\vec{x}') \cdot L_{x_{i,0}}(\vec{x}') = v_0, \tag{57}$$

$$\sum_{\vec{x}' \in H_{k_i}} (\hat{p}_i(\vec{x}') - \mathsf{msk}_i(\vec{x}')) \cdot L_{x_{i,k}}(\vec{x}') = v_k, \tag{58}$$

for $k = 1, \ldots, s_i$, where $L_{x_{i_k}}$ are the univariate Lagrangians of degree $2^{k_{i-1}}$. These constraints are again of the form (44).

*Step 4: Aggregation of claims.* The verifier samples a random $\gamma_i \leftarrow\$ F$ for combining the above claims (56), (57) and (58) on the new ensemble into a single claim of the form (44), with the following linear functionals:

$$K^{(i)} = \lambda_i \cdot K^{(i-1)}(\vec{\beta}_i, \,.\,) + \sum_{k=0}^{s_i} \gamma_i^{k+1} \cdot L_{x_{i,k}}(\,.\,), \tag{59}$$

$$\Gamma_j^{(i)} = 2^{r_i} \cdot \lambda_i \cdot \Lambda_j^{(i-1)}, \quad j = 1, \ldots, i-1, \tag{60}$$

$$\Gamma_i^{(i)} = -\langle \,.\,, K^{(i)} - \gamma_i \cdot L_{x_{i,0}} \rangle_{H_{k_i}} \tag{61}$$

$$\Lambda_j^{(i)} = 2^{r_i} \cdot \lambda_i \cdot \Lambda_j^{(i-1)}, \quad j = 1, \ldots, i-1, \tag{62}$$

$$\Lambda_i^{(i)}(h_i) = 2^{-k_i} \cdot h_i(\vec{\beta}_i). \tag{63}$$

As before, the inner product (61) on $\mathsf{msk}_i$ and the evaluation (63) on $h_i$ are inexpensive to compute.

**Remark 6.** In closer alignment with [CFW26], one may think of the claims (44) in terms of the unmasked polynomial $p_i = \hat{p}_i - \mathsf{msk}_i$, rather than of the masked one, i.e.

$$c_i = \left\langle p_i, K^{(i)} \right\rangle_{H_{k_i}} + \sum_{j=0}^{i} \Gamma_j^{(i)}(\mathsf{msk}_j) + \sum_{j=1}^{i} \Lambda_j^{(i)}(h_j).$$

In Step 3, the prover commits to $\hat{p}_i = p_{i-1}(\vec{\beta}_i, \,.\,) + \mathsf{msk}_i(\,.\,)$ instead of $\hat{p}_i = \hat{p}_{i-1}(\vec{\beta}_i, \,.\,) + \mathsf{msk}_i(\,.\,)$, and the consistency relation for the $s_i$ in-domain queries would then be

$$\sum_{\vec{x}' \in H_{k_i}} \left( \hat{p}_i(\vec{x}') - \mathsf{msk}_i(\vec{x}') + \mathsf{msk}_{i-1}(\vec{\beta}_i, \,.\,) \right) \cdot L_{x_{i,j}}(\vec{x}') = v_j,$$

instead of (58), leading to a different update rule of $\Gamma_{i-1}^{(i)}$.

**The final opening protocol.** The final claim after round $r$ is of the form

$$c_r = \langle \hat{p}_r, K^{(r)} \rangle_{H_{k_r}} + \sum_{j=0}^{r} \Gamma_j^{(r)}(\mathsf{msk}_j) + \sum_{j=1}^{r} \Lambda_j^{(r)}(h_j). \tag{64}$$

It is proven via an independent proof of proximity to $\hat{p}_r(X) \in \mathscr{P}_{k_r}^{e_r}$, and $\mathsf{m}(X) \in F[X]^{<m_{\mathsf{zk}}}$, satisfying the joint claim (64). For improving soundness, we constrain the prover to a single proximate mask by asking it for the value

$$v = \mathsf{m}(z)$$

at a random out-of-domain point $z \leftarrow\$ F$, at commit time of $\mathsf{m}$. (See the full protocol description in Section A.) We use the same random point to prove invariance of the mask component $msk_0$ under the Frobenius automorphism $\phi : F \longrightarrow F$, which leaves $\mathbb{F}_q$ invariant. The prover also provides the value of the component

$$\hat{\mathsf{msk}}_0(X) = \mathsf{msk}_0(X) + X^{|\mathsf{msk}_0|} \cdot r_0(X) \tag{65}$$

at $z$, i.e.

$$v' = V_z(\mathsf{m}) = \hat{\mathsf{msk}}_0(z). \tag{66}$$

If $\phi(v') = V_{\phi(z)}(\mathsf{m})$, then with high probability $\hat{\mathsf{msk}}_0(X) = \phi(\hat{\mathsf{msk}}_0)(X)$, proving that $msk_0$ (and $r_0$) is over $\mathbb{F}_q$.

**Protocol 3.** *The prover commits to full entropy masks $H \leftarrow\$ \mathscr{P}_r^{e_r}$ and $M \leftarrow\$ F^{m_{zk}}$, as words from $C_r$ and $C_{zk}$, respectively. It parses $M$ as $H_1 \| \ldots \| H_r \| M_0 \| \ldots \| M_r \| M_{zk} \| R_0$ according to* (51)*, and claims*

$$d = \langle H, K^{(r)} \rangle_{H_{k_r}} + \sum_{j=0}^{r} \Gamma_j^{(r)}(M_j) + \sum_{j=1}^{r} \Lambda_j^{(r)}(H_j), \tag{67}$$

*and*

$$v_M = M(z), \quad v'_M = V_z(M), \quad v''_M = V_{\phi(z)}(M). \tag{68}$$

*The verifier replies with $\lambda \leftarrow\$ F$. The prover reveals the combined messages $H' = \lambda \cdot \hat{p}_r + H$, $M' = \lambda \cdot \mathsf{m} + M$, the latter parsed as $H'_1 \| \ldots \| H'_r \| M'_0 \| \ldots \| M'_r \| M'_{zk} \| R'_0$.*

*The verifier queries the commitments of $\hat{p}_r$ and $\mathsf{m}$ independently at random points $x_1, \ldots x_s \leftarrow\$ D_r$ and $y_1, \ldots, y_{s_{zk}} \leftarrow\$ D_{zk}$, respectively, and checks collinearity of the revealed values with those of the encodings of $H'$ and $M'$. If the collinearity checks hold, and if*

$$\lambda \cdot c_r + d = \langle H', K \rangle_{H_{k_r}} + \sum_{j=0}^{r} \Gamma_j(M'_j) + \sum_{j=1}^{r} \Lambda_j(H'_j), \tag{69}$$

*and*

$$\lambda \cdot v + v_M = M'(z),\ \lambda \cdot v' + v'_M = V_z(M'),\ \lambda \cdot \phi(v') + v''_M = V_{\phi(z)}(M'), \tag{70}$$

*then the verifier accepts.*

### 5.5 Impact on soundness (tbc)

While all other measures from Section 5.1, 5.2, 5.3 do not affect soundness, except for introducing another round for combining

This is different for the ZO0K modification of WHIR, Section 5.4. The final opening proof, Protocol 3, is an independent proof of proximity for the final masked WHIR oracle $\hat{p}_r$ and the joint commitment $\mathsf{m}$ of all masks and blinders. As a consequence we are only able to bound the number of proximates pairs by

$$\ell(\theta_r) \cdot \ell(\theta_{\mathrm{zk}}),$$

and likewise the additional multiplier $\ell(\theta_{\mathrm{zk}})$ is present in all other steps of the protocol.

$$\mathcal{R}_\theta = \left\{ \begin{array}{c} f \in \left(\mathbb{F}_q^D\right)^e, \\ g \in \mathbb{F}^{D_{\mathrm{zk}}} \end{array} : \begin{array}{c} \exists \hat{p}(X) \in \mathscr{P}_{m_1}, \hat{q}(X) \in F[X]^{<q_{\mathrm{zk}}}, \\ d(\hat{p}, f) \leq \theta \ \wedge \ d(\hat{q}, g) \leq \theta_{\mathrm{zk}}, \\ \text{and} \\ \text{the coefficients } (c_i) \text{ of} \\ p(X) = \hat{p}(X) - \mathsf{msk}_{0,0}(X) \\ \text{satisfy} \\ (c_i) \in \mathscr{R}_{main} \cap \mathscr{R}_{aux}, \\ (c_1, \ldots, c_{t_1}) = \vec{in} \end{array} \right\}, \tag{71}$$

where $\mathsf{msk}_{0,0}(X)$ are the corresponding

Ulrich: It is sufficient to prove a proximate $\hat{p}(X) \in \mathscr{P}_{m_1}(F)$ that solves the $A_1$ $B_1$ and $C_1$ part. Because the entries of the matrices are in $\mathbb{F}_q$, the $\mathbb{F}_q$-component of $p(X) = \hat{p}(X) - \mathsf{msk}_{0,0}(X)$ are a solution of $(A_1, B_1, C_1)$ with the corresponding inputs. WRONG! We need to prove that $\mathsf{msk}_0(X)$ is over $\mathbb{F}_q$.

The prover declares the value

$$v = \langle \pi_0(\hat{q}), L_z \rangle$$

at random $z \in F$, where $\pi_0$ grabs the coefficients for $\mathsf{msk}_0$, and $\hat{q}$ is subject to the invariance constraints

$$\begin{aligned} \langle \pi_0(\hat{q}), L_z \rangle &= v \\ \langle \pi_0(\hat{q}), L_{\phi(z)} \rangle &= \phi(v), \end{aligned}$$

which prove that with high probability $\phi(\pi_0(\hat{q})) = \pi_0(\hat{q})$. However, to make this zk we need to append $e$ many $\mathbb{F}_q$ to $\mathsf{msk}_0$. We can use this value for reducing to a single proximate anyways.

# 6 The non-interactive STARK (tbc)

## 6.1 Merkle commitments (tbc)

## 6.2 The

## 6.3 Fiat–Shamir / BCS transform (tbc)

## 6.4 Example parameters (tbc)

## 6.5 Benchmarks (tbc)

# 7 Outlook v3 (tbc)

# References


ACFY24. Gal Arnon, Alessandro Chiesa, Giacomo Fenzi, and Eylon Yogev. WHIR: Reed–solomon proximity testing with super-fast verification. In *IACR preprint archive 2024/1586*, 2024. `https://eprint.iacr.org/2024/1586`.

BCH+25. Eli Ben-Sasson, Dan Carmon, Ulrich Haböck, Swastik Kopparty, and Shubhangi Saraf. On proximity gaps for Reed–Solomon codes (full version). Cryptology ePrint Archive, Paper 2025/2055, 2025. `https://eprint.iacr.org/2025/2055`.

BCHO22. Jonathan Bootle, Alessandro Chiesa, Yuncong Hu, and Michele Orr'u. Gemini: Elastic SNARKs for diverse environments. In *EUROCRYPT 2022*, 2022. Full paper: `https://eprint.iacr.org/2022/420`.

BCI+20. Eli Ben-Sasson, Dan Carmon, Yuval Ishai, Swastik Kopparty, and Shubhangi Saraf. Proximity gaps for Reed-Solomon codes. In *FOCS 2020*, 2020. Full paper: `https://eprint.iacr.org/2020/654`.

CBBZ22. Binyi Chen, Benedikt Bünz, Dan Boneh, and Zhenfei Zhang. Hyperplonk: PLONK with a linear-time prover and high-degree costum gates. In *IACR ePrint Archive 2020/1355*, 2022. `https://eprint.iacr.org/2022/1355`.

CFW26. Alessandro Chiesa, Giacomo Fenzi, and Guy Weissenberg. Zero-knowledge IOPPs for constrained interleaved codes. Cryptology ePrint Archive, Paper 2026/391, 2026.

GG25. Rohan Goyal and Venkatesan Guruswami. Optimal proximity gaps for subspace-design codes and (random) reed-solomon codes. Cryptology ePrint Archive, Paper 2025/2054, 2025.

Gos96. David Goss. *Basic Structures of Function Field Arithmetic*, volume 35 of *Ergebnisse der Mathematik und ihrer Grenzgebiete*. Springer, Berlin, 1996.

Hab24. Ulrich Haböck. Basefold in the list decoding regime. 2024. `https://eprint.iacr.org/2024/1571`.

Hab25. Ulrich Haböck. A note on mutual correlated agreement for reed-solomon codes. Cryptology ePrint Archive, Paper 2025/2110, 2025.

Ham22. Adrian Hamelink. Gemini. hackmd.io, 2022. `https://hackmd.io/@adrian-aztec/BJxoyeCqj`.

HHM25. Ulrich Haböck, Adrian Hamelink, and Andrew Milson. Lifted FRI: A uniform multi-domain polynomial commitment scheme. 2025. personal communciation.

HK24. Ulrich Haböck and Al Kindi. A note on adding zero-knowledge to STARKs. Cryptology ePrint Archive, Paper 2024/1037, 2024.

KT23. Tohru Kohrita and Patrick Towa. Zeromorph: Zero-knowledge multilinear-evaluation proofs from homomorphic univariate commitments. In *IACR preprint archive 2023/917*, 2023. `https://eprint.iacr.org/2023/917`.

Set20. Srinath Setty. Spartan: Efficient and general-purpose zkSNARKs without trusted setup. In *CRYPTO 2020*, 2020. Full paper: `https://eprint.iacr.org/2019/550`.

XZZ+19. Tiancheng Xie, Jiaheng Zhang, Yupeng Zhang, Charalampos Papamanthou, and Dawn Song. Libra: Succinct zero-knowledge proofs with optimal prover computation. Cryptology ePrint Archive, Paper 2019/317, 2019.

ZCF23. Hadas Zeilberger, Binyi Chen, and Ben Fisch. BaseFold: efficient field-agnostic polynomial commitment schemes from foldable codes. In *IACR preprint archive 2023/1705*, 2023. `https://eprint.iacr.org/2023/1705`.

## A The complete protocol (tbc)

The complete protocol takes all the measures from Section 5 into account.

1. The first stage witness $\vec{w}_1$ is
   (a) Appended by the zero-check masks $h_0 \in F, h_1, \ldots, h_m \in F^3$ laid out as basefield vectors.
   (b) Appended by $e^2 = [F : \mathbb{F}_q]^2$ basefield randomizers for $w_A, w_B, w_C$ at $\vec{\alpha}$, as described in Section 5.1 (at any position)
   The propagation constraints for the basefield randomizers are placed according to Lemma 5. No randomness appended to $\vec{w}_2$.
2. Commitment masks as in Section 5.4: The masks $msk_0, \ldots, msk_r$ and $h_1, \ldots, h_r$ are committed as codeword in $C_{zk}$. Here $msk_0$ for the combined witness poly splits as $msk_{0,1}, msk_{0,2} \in \mathbb{F}_q^{s_1}$, and is over the basefield. (The other masks are over $F$.)

## B Correlated agreement (tbc)

Let $\mathbb{F}$ be an arbitrary finite field,

$$\Gamma : \mathscr{P}_k(\mathbb{F}) \times \mathbb{F} \longrightarrow \mathbb{F},$$

be a polynomial (in the coordinates of an arbitrary basis) of total degree $d = \deg \Gamma = 2$.

**Theorem 3 ([BCI+20,BCH+25]).** *For $C = \mathsf{RS}[F, D, k]$ any Reed–Solomon code over an arbitrary finite field $F$, with minimum distance $\delta$, and $\Gamma : \mathscr{P}_k \times F \longrightarrow F$ as above. Then, for any $\theta \in \left[\frac{\delta}{2}, 1 - \sqrt{1-\delta}\right)$, the following holds. If the set*

$$S = \left\{ z \in F \ : \ \begin{array}{c} \exists p_z(X) \in \mathscr{P}_k, d((1-z) \cdot f_0 + z \cdot f_1, p_z) \leq \theta \\ \wedge \Gamma(p_z, z) = 0 \end{array} \right\}$$

*is of size*

$$|S| \geq \ell_{GS}(\theta) \cdot \left( \frac{2\ell_{GS}(\theta)^4}{3} \cdot (1-\delta) + 1 \right) \cdot |D|,$$

*then there exist polynomials $q_0(X)$, $q_1(X) \in \mathscr{P}_k$ with $d((q_0, q_1), (f_0, f_1)) \leq \theta$, and $\Gamma(q_0, 0) = \Gamma(q_1, 1) = 0$.*

The theorem is direct consequence of line decodability [], also called collinearity of proximates property in [Hab24]. Given a of proximates of the random linear combinations, one for each, then a notable fraction of it lies on a line.

# C Proofs: IOPP soundness (tbc)

## C.1 Protocol 1

Let us outline the soundness properties of the protocol. In the following we write

$$\ell_1 = \ell_1(\theta_1) \tag{72}$$

for the Johnson list size bound on the number of proximates from $C_1$ at distance $\theta_1$.

*Phase 1:* The first phase is a reduction $\mathcal{R}_\theta \longrightarrow \mathcal{R}_{\text{phase 1}}$ depending on the verifier challenges $\vec{\xi} \leftarrow\$ \mathbb{F}_q^t$, where

$$\mathcal{R}_{\text{phase 1}} = \left\{ \begin{array}{c} f_1 \in \left(\mathbb{F}_q^D\right)^e, \\ f_2^* \in \left(\mathbb{F}_q^D\right)^e \end{array} : \begin{array}{c} \exists (p_1(X), p_2(X)) \in \mathscr{P}_{m_1}^2 \\ d((p_1, p_2), (f_1, f_2^*)) \leq \theta_1, \\ \text{and} \\ (p_1(X), \vec{\xi}, p_2(X)) \\ \text{satisfy the lifted R1CS} \\ \text{on input } \vec{in} \end{array} \right\},$$

where as before, the distance refers to the interleaved representations of the polynomials. The phase is a single-step reduction, and its soundness error is as follows.

**Lemma 7.** *Phase 1 in Protocol 1 is a round-by-round sound oracle reduction $\mathscr{R}_{\theta_1} \longrightarrow R_{phase\ 1}$ with soundness error*

$$\varepsilon_{phase\ 1} \leq \ell_1 \cdot \varepsilon_{R1CS},$$

*where $\varepsilon_{R1CS}$ is the soundness error of the randomized R1CS.*

*Proof.* Assume that for more than $\kappa = \ell_1(\theta)\varepsilon_{\text{R1CS}} \cdot |\mathbb{F}_q|^t$ distinct verifier challenges $\vec{\xi}$, there exists a potential prover answer $f_2 \in \mathbb{F}_q^{D'}$ which extends $f_1$ to a pair $(f_1, f_2^*)$ in $\mathscr{R}_{\text{phase 1}}$. For each such challenge $\vec{\xi_i}$, $i = 1, \ldots, \kappa$, we select one of the $\theta_1$-proximates $(p_1(X), p_2(X))$ stated by $\mathscr{R}_{\text{phase 1}}$, and obtain a list

$$\mathscr{L} = \left\{ (p_{1,i}(X), \vec{\xi_i}, p_{2,i}(X))) \ : \ \begin{array}{c} d(p_{1,i}(X), f_1) \leq \theta_1, \\ \text{and} \\ (p_{1,i}(X), \vec{\xi_i}, p_{2,i}(X)) \\ \text{satisfy the lifted R1CS} \\ \text{on input } \vec{in} \end{array} \right\}$$

with more than $\kappa$ entries.

By the list size bound, there at most $\ell_1$ distinct $p_{1,i}(X)$ mentioned in $\mathscr{L}$. One of them, $p_1(X) = p_{1,i_0}(X)$, appears more than

$$\kappa' = \frac{\kappa}{\ell_1} = \varepsilon_{\text{R1CS}} \cdot |\mathbb{F}_q|^t$$

many times in the list. This $p_1(X)$ has more than $\kappa'$ many $\vec{\xi}_i, p_{2,i}(X)$ satisfying answers for the lifted randomized R1CS on input $\vec{in}$. Since the soundness error of the lifted R1CS equals that of the original R1CS, $p_1(X)$ is a solution of the first stage R1CS. This proves that $f_1 \in \mathscr{R}_{\theta_1}$. $\square$

*Phase 2:* The second phase, the zero-check, reduces $\mathcal{R}_{\text{phase 1}}$ to $\mathcal{R}_{\text{phase 2}}$, where

$$\mathcal{R}_{\text{phase 2}} = \left\{ \begin{array}{c} f_1 \in \left(\mathbb{F}_q^{D_1}\right)^{e_1}, \\ f_2 \in \left(\mathbb{F}_q^{D_k}\right)^{e_k}, \\ q_1(X), \ldots, q_m(X) \in F[X]^{\leq 3} \\ v_A, v_B, v_C \in F \end{array} : \begin{array}{c} \exists (p_1(X), p_2(X)) \in \mathscr{L}_{m_1} \times \mathscr{L}_{m_2} \\ d(p_1, f_1) \leq \theta_1, d(p_2, f_2) \leq \theta_2 \\ \text{and} \\ 0 = q_1(0) + q_1(1), \\ q_i(\alpha_i) = q_{i+1}(0) + q_{i+1}(1), \\ \text{for } i = 1, \ldots, m-1, \\ q_m(\alpha_m) = (v_A \cdot v_B - v_C) \cdot eq(\vec{\alpha}, \vec{\rho}) \end{array} \right\},$$

depends on the verifier challenges $\vec{\rho}, \vec{\alpha}$ (and the preceding $\vec{\xi}$). The soundness error of this reduction is

$$\varepsilon_{\text{phase 2}} \leq \ell_1 \cdot \ell_2 \cdot \left( \frac{m}{|F|} + m \cdot \frac{3}{|F|} \right),$$

again scaled by the number of joint $\theta_1$- and $\theta_2$-proximates of $f_1$ and $f_2$. The components correspond to the roundwise soundness errors of that phase. The first term is the probability of that $\vec{\rho}$ is a zero of the multilinear extension of $a(\vec{x}) \cdot b(\vec{x}) - c(\vec{x})$, $\vec{x} \in H_m$, despite being non-trivial. The second term is for the $m$ rounds of the sumcheck. Each $\frac{3}{|F|}$ stands for the probability that the specialization of the sumcheck claim to the random $\alpha_i$ is satisfied by a joint proximate $(p_1(X), p_2(X))$, although not both claims for $X_i = 0$ and $X_i = 1$ hold.

*Phase 3:* The third phase reduces $\mathscr{R}_{phase2}$ to the final verifier relation $\mathscr{R}_V$ of the protocol. This is

$$\mathscr{R}_V = \left\{ \begin{array}{c} f_1 \in \left(\mathbb{F}_q^{D_1}\right)^{e_1}, \\ f_2 \in \left(\mathbb{F}_q^{D_k}\right)^{e_k}, \\ q_1(X), \ldots, q_m(X) \in F[X]^{\leq 3} \\ v_A, v_B, v_C \in F \\ g_2 \in \left(\mathbb{F}_q^{D_2}\right)^{e_2}, u_2 \in F \\ \vdots \\ g_{r-1} \in \left(\mathbb{F}_q^{D_r}\right)^{e_r}, u_{r-1} \in F \\ g_r \in \mathscr{L}_{k_r} \end{array} : \begin{array}{c} 0 = q_1(0) + q_1(1), \\ q_i(\alpha_i) = q_{i+1}(0) + q_{i+1}(1), \\ \text{for } i = 1, \ldots, m-1, \\ q_m(\alpha_m) = (v_A \cdot v_B - v_C) \cdot eq(\vec{\alpha}, \vec{\rho}) \\ \text{and} \\ g_r \text{ satisfies the aggregated WHIR claim} \\ \bar{v} = \langle K_r, g_r \rangle_{H_{k_r}}, \\ \text{which involves the specializations} \\ M_i(\vec{\alpha}, \vec{\beta} \| \, . \, ), \\ M = A, B, C, \; i = 1, 2 \end{array} \right\},$$

where $g_2$, ..., $g_{r-1}$, are the folding proposals committed by in the first $r-1$ rounds of WHIR, together with their values $u_2$, ..., $u_{r-1}$ at the verifier DEEP queries $z_2, \ldots, z_{r-1}$, and $g_r$ is the final folding proposal revealed in plain.

The soundness error of this phase is that of joint WHIR,

$$\varepsilon_{\text{phase 3}} = \varepsilon_{WHIR}(\theta_1, \ldots, \theta_r)$$

and will be described in the following section.

### C.2 Soundness ZO0K. (tbc)

For now, only few notes:

For simplicity, we assume the masks as given. (The full story is a bit more complicated as it needs to take into account the proximates of the masks.) Suppose that with notable probability in $\vec{\beta}_i$ the prover is able to provide $\hat{f}_i$ which close to a poly $\hat{p}_i$, so that

- the new ensemble with $\hat{p}_i$ and $\mathsf{msk}_i$ satisfies the $\vec{\beta}_i$-specialized claim (56),
- the folding $\mathsf{fold}_{\vec{\beta}_i} \hat{f}_{i-1}$ agrees on a density $\alpha$ subset of $D_{i-1}$ with $\hat{p}_i - \mathsf{msk}_i$. (Which is caused by (57) and (58) satisfied with notable prob.)

Then, with the help of line decodability, all intermediate foldings are proximate to polys, the mv rep of which satisfy the specialized claim, up to that $\hat{f}_{i-1}$ has a close poly $\hat{p}_{i-1}$, satisfying the combined claim (69).

### C.3 The complete protocol (tbc)

## D Proofs: IOPP zero-knowledge (tbc)

### D.1 ZO0K (tbc)

As elaborated in Section 5.2, the sumcheck polynomials of each round will be uniformly distributed over the relation of consistent polynomials, independent of the witness.

Further, by construction the queried values (in- and out of domain) are identically distributed field elements ($F$ or $\mathbb{F}_q$ if the first round oracles), independent of the witness.

The simulator can be constructed similar to the sumcheck simulator.

### D.2 The complete protocol (tbc)

## E Alternatives to ZO0K

**Proposal 1: VEIL** In its most simple variant, the followings steps need to be taken.

**Securing the witness queries.** We need to randomize the witnesses $w_1$ and $w_2$ for the number of queries from $D$. (No R1CS modification necessary). Recall we commit the witnesses in interleaved manner, using the FFT decomposition, and thus we need to randomize the witnesses properly, so that each component is affected equally: Thinking of $\vec{w}_1$ as the not yet decomposed vector, add a *contiguous* interval of $e \cdot \mathrm{num}_{\mathrm{queries}}$ random values, where $e$ is the interleaving depth, at any place in the not used padding space of it. For $w_2$ likewise, but we may take into account that less than $e$ many components are non-zero.

**Isolator polynomials.** In its most simple variant, the prover samples addionally to two random polynomials over the extension field $F$,

$$H_1 \leftarrow\$ F[X_1, \ldots, X_{m_1}], \quad H_2 \leftarrow\$ F[X_1, \ldots, X_{m_2}],$$

and commits them together with $w_1$ and $w_2$, respectively. These 'isolator polynomials' are combined together with $w_1$ and $w_2$ in an additional (virtual) batching step before entering WHIR:

1. After we combined (21) and (22) into a single inner product claim $\langle K, w\rangle_{H_{m_1+1}}$ on the combined witness poly, the prover reveals

   $$v = \langle K, H\rangle_{H_{m_1+1}}$$

   on the combined isolator polynomial $H = (1 - X_0) \cdot H_1 + X_0 \cdot H_2^*$. with $w_1$ and $w_2^*$ replaced by $H_1$ and $H_2^*$.
2. The verifier draws a further randomness $\gamma_2$ to combine the two claims into a claim on the linear combination $w+\gamma\cdot H$. This combination is not committed, the values of it are computed from those of $w$ and $H$.
3. Run WHIR on the virtual linear combination $w + \gamma \cdot H$.

The isolator polynomial entirely decouples the distribution of the future run of WHIR from the past (and thus the witnesses). WHIR is run entirely in non-zk mode, no changes here.

While the prove size increases only marginally (by one extension field element per $w_1$ query, likewise for $w_2$), prover cost does, since the prover has to hash the

isolator polys. Moreover, we most likely hit memory bounds in this variant of VEIL.

Ulrich: The witness polys $w_1$ is about 1/4-th the size as in the BN254 case, and $w_2$ is about 3/4-th of the size. However, $H_1$ and $H_2$ together are 3/4-th of the entire BN254 witnesses, which is why I believe we will hit memory bounds for the largest circuits we want to prove.

*A more prover friendly variant.* To mitigate the memory bounds, we insert isolator polynomials in the first round of WHIR instead. Ulrich: This is a hacky solution, which requires additional sumcheck blinders, I am not a fan of it.

1. The prover only samples a single

$$H \leftarrow\$ C_1$$

   from the base code of $w_1$ and $w_2^*$, and commits either together with $w_1$. This is much less overhead (both compute and memory).
2. Modified first round of WHIR:
   (a) The sumcheck phase is randomized in the usual manner, using

$$h_1(X), \ldots, h_{k_1}(X) \leftarrow\$ F[X]^{\leq 3}.$$

   These are committed separately, and in zk manner.
   (b) After specializing the sumcheck claim to $X_1 = \beta_1, \ldots, X_k = \beta_{k_1}$, the prover reveals

$$v = \langle K_0(\beta_1, \ldots, \beta_{k_1}, \,.\,), H \rangle_{H_{m_1 - k_1}},$$

   and the claims for $w(\beta_1, \ldots, \beta_{k_1}, \,.\,)$ and $H$ are combined via another verifier randomness $\gamma_2 \leftarrow\$ F$.
   (c) We now proceed with the virtual linear combination $w(\beta_1, \ldots, \beta_{k_1}, \,.\,) + \gamma_2 \cdot H$. (The query phase, the aggregation phase.)

The value $h_1(\beta_1) + \ldots + h_{k_1}(\beta_{k_1})$ is proven separately.