Sonata: Certifying Canonical Openings in High-Arity Lattice Folding

Xiang Wang

Guizhou University, Guiyang, China
ee.wangx24@gzu.edu.cn

Abstract. Folding lets a prover certify a computation incrementally by combining committed instances into an accumulator. In lattice-based folding, a linear combination of field values can change their canonical digit representations. Re-encoding the result gives the prover a short opening. The verifier must establish that this opening represents the prescribed fold of the original committed values.
We present Sonata, a high-arity lattice folding scheme for committed quadratic relations. A quadratic sumcheck certifies canonical encoding and the folded equations together. A recursive short-opening argument binds its final evaluation to the original commitments. We prove knowledge soundness for classical rewindable provers by comparing two labelled extraction contexts and bounding the probability that both yield the same invalid candidate. We evaluate complete five-layer transitions for batches of 16 to 1,024 fresh inputs. Across this range, communication increases from 143,165 to 170,361 bytes when the input commitments and setup are shared. At 1,024 inputs, the median prover and verifier times are 258.3 and 226.2 seconds.

Keywords: Lattice-based folding, High-arity folding, Canonical re-encoding, Sumcheck protocols, Recursive short-opening arguments, Adaptive knowledge extraction, Module-SIS

Contents

1 Introduction . . . 3
1.1 Problem: Authenticating Re-Encoded Openings . . . 3
1.2 Our Solution: Sonata . . . 4
1.2.1 Implementation and Experimental Evaluation . . . 4
2 A Technical Overview of Sonata . . . 4
2.1 Certifying Canonical Openings after Folding . . . 4
2.2 A Recursive Argument for Short Openings . . . 5
2.3 Knowledge Extraction from Adaptive Responses . . . 6
2.4 Authenticated Histories and Batch Settlement . . . 7
3 Preliminaries and Relations . . . 8
3.1 Notation and Algebraic Setting . . . 8
3.2 Committed States and Folding Relations . . . 8
3.3 Knowledge-Extraction Model . . . 9
4 The Sonata Construction . . . 10
4.1 Certifying Canonical Encodings . . . 10
4.2 From Quadratic Checks to a Short-Opening Instance . . . 11
4.3 The Online Transition . . . 13
4.4 A Recursive Certification Layer . . . 14
4.5 Terminal Pivot Verification and Completeness . . . 16
5 Knowledge Soundness . . . 18
5.1 Main Extraction Guarantee . . . 18
5.2 Compatible Records and Local Comparison . . . 19
5.3 Labelled Bad-Slope Counting . . . 20
5.4 Recursive Composition and Expected Work . . . 23
6 Authenticated Histories and Settlement . . . 27
6.1 From Accepted Transitions to an Authenticated History . . . 27
6.2 Strict Settlement of a Frozen Window . . . 28
7 Implementation and Evaluation . . . 29
7.1 Concrete Profiles . . . 29
7.2 End-to-End Ordinary Transitions . . . 29
7.3 Verification Tests and Component Costs . . . 30
7.4 Settlement on a Matched Log . . . 30
8 Discussion . . . 31
References . . . 32
A Algebraic Correctness . . . 35
A.1 Canonicality over the Integers . . . 35
A.2 Constructing the Square Representation . . . 36
A.3 Semantic Entries and the Terminal Evaluation . . . 37
A.4 Coefficient Adjoints and Child Linearization . . . 38
A.5 Honest Norm Propagation and Retry Exhaustion . . . 39
B Knowledge-Soundness Proofs . . . 41
B.1 Binding a Submitted Response . . . 41
B.2 Seeded Projection for a Fixed Candidate . . . 43
B.3 From a False Aggregate to a Defective Diagonal . . . 44
B.4 The Two Front-End Checkpoints . . . 45
B.5 Expected Work at an Adaptive Checkpoint . . . 47
B.6 From Correlated Commitment Roles to Module-SIS . . . 49
C Parameters and Reproducibility . . . 51
C.1 Parameter Selection and Audit . . . 51
C.2 Commitment Roles and Encodings . . . 52
C.3 Frozen-Window Dimensions . . . 53
C.4 Experimental Reproduction . . . 53
C.5 Lattice-Attack Estimates . . . 55
C.6 Extraction Cutoff and Concrete Security Budget . . . 57

1 Introduction

A proof of a long computation can be built a step at a time. This is useful when the inputs arrive in a stream or when a prover cannot keep the entire execution in memory. Halo and subsequent work on accumulation showed how a verifier can carry an unresolved verification claim from one step to the next [20,23]. Nova applies this idea to witnesses: a prover folds a fresh rank-one constraint instance into a committed running instance of the relaxed relation [36]. SuperNova extends this approach to computations with different step circuits [34]. The cost of checking the final proof also matters in applications such as on-chain verification [53].

Folding several inputs in one transition reduces the number of times an accumulator must be updated. HyperNova supports multiple instances of a customizable constraint system [35,50]. Protostar derives folding schemes from special-sound protocols, and ProtoGalaxy develops this approach for batches of instances [22,28]. With lattice commitments, a further condition enters at each update. A commitment has the form $C = Ad \bmod q$, and its binding depends on the difficulty of finding a short nonzero kernel vector [1,38]. The prover must therefore preserve both the algebraic relation on the values and a suitable norm bound on the vector d that opens their commitment.

Existing lattice schemes manage this norm in several ways. LatticeFold checks ranges using digit decompositions, while LatticeFold+ proves ranges algebraically and uses sum-check to transform double commitments [14,13]. Neo and SuperNeo support commitments whose cost reflects the bit width of the witness and perform their norm checks over a small field [44]. Cyclo amortizes norm refreshing and uses partial range checks [29]. For high arity, Symphony folds many committed statements at once, and ProtogaLattice combines bounded-depth algebraic folding with a separate norm-reduction step [24,9]. We study a transition in which the next accumulator must open to the canonical digits of a particular field-valued fold.

1.1 Problem: Authenticating Re-Encoded Openings

Folding combines values over a field, while a lattice commitment requires a short representation of those values. A canonical encoding need not preserve linear combinations. Even adding two binary encodings of the value one produces a coefficient two where the encoded sum requires a carry. Directly combining the original openings can therefore produce a noncanonical representation or exceed the required norm bound.

A natural solution is to compute the folded values first and then re-encode them. The prover can commit to the new representation and prove that it is canonical and short. This establishes only part of the required claim. A malicious prover could supply a valid encoding of unrelated values.

How can the verifier authenticate the new opening while preserving its connection to the original commitments? The digits must be canonical, the decoded values must equal the prescribed fold, and both conditions must hold for the same opening. A knowledge argument must establish this link even if the prover changes its responses after seeing a challenge.

1.2 Our Solution: Sonata

We present Sonata, a high-arity lattice folding scheme for committed quadratic relations. An ordinary transition takes an accumulated state and a batch of fresh committed inputs. It produces a new state with a canonical short opening and updates a public pending value with a weighted sum of the fresh residuals. We certify the encoded representation and the equations on its values with one quadratic sumcheck. Its final evaluation is an affine function of the original openings. A recursive argument certifies that evaluation against the input, output and auxiliary commitments.

Our second result is a two-point knowledge extractor for adaptive, classically rewindable provers. A pair of accepting responses determines a candidate for one witness block. Our proof bounds the probability of repeatedly obtaining a particular invalid candidate and converts incompatible short openings into a Module-SIS solution. The extractor retains the first real execution. If it accepts, the required witness must open the commitments in that execution. In a sequence of authenticated transitions, the same guarantee lets a verifier certify a history that returns to zero pending value.

1.2.1 Implementation and Experimental Evaluation We implement Sonata[1] in C++17 with a Python short-challenge sampler and parameter-auditing tools. On a shared AMD EPYC 7742 server, we execute three ordinary transitions at each of four batch sizes: 16, 64, 256 and 1,024 fresh inputs. Every configuration uses four recursive reductions and a terminal verification step. All twelve executions accept with fresh verifier challenges, and public verification rejects an altered terminal commitment in each run.

The total bidirectional transcript grows from 143,165 to 170,361 bytes across these batch sizes when the parties already hold the input commitments and setup material. At 1,024 inputs the median prover and verifier times are 258.3 and 226.2 seconds. We also compare certification schedules for a frozen log of 994 sources in eight steps. The window requires 22 sumcheck rounds and one backend invocation. Certifying each step separately requires 152 rounds and eight invocations. A direct batch for the same strict relation can use the same table and backend as the window. We checked the window construction separately in a five-layer example with a fixed verifier tape.

2 A Technical Overview of Sonata

2.1 Certifying Canonical Openings after Folding

Let z_{acc} denote the 69 accumulator coordinates that fold linearly and let $z_1, \ldots, z_k$ be their counterparts in the fresh inputs. Once the inputs are committed, the verifier samples a field challenge α. The prover computes

$$z^\star = z_{\mathrm{acc}} + \sum_{j=1}^{k} \alpha^j z_j, \qquad b^\star = \mathrm{Enc}_K(z^\star). \tag{1}$$

[1] Implementation and reproduction scripts: https://github.com/cxcx52/sonata.

The verifier checks the fold on the five public coordinates directly. The remaining 64 coordinates enter the proof through their encodings. The prover computes the four error coordinates of the new state from its quadratic residuals. It uses the fresh sources' residuals for a different purpose, updating the public pending value. Figure 1 shows where each of these obligations enters the certificate.

A quadratic certificate for canonical encodings. For each private base-field value, we use 48 bit positions and seven auxiliary positions. We construct an integer quadratic penalty whose zeros are exactly the canonical bit strings together with feasible auxiliary values. In particular, it excludes the 59 invalid strings above the largest representative of our field. For any centered opening W within the extraction norm budget, its total penalty satisfies $0 \le E_{\mathrm{enc}}(W) < q$. Nonnegativity holds on integer lifts. The strict upper bound ensures that a nonzero penalty survives reduction modulo q. A shifted sum-of-squares representation over the extension field then allows us to combine the penalty with the folded relation.

One sumcheck with an authenticated endpoint. Let $\mathcal{A}(W)$ be the affine table derived from the committed openings and the verifier's batching coins ρ, η. Let S denote the public initial claim and let $S_\rho(W)$ collect the discrepancies in the semantic equations and the pending update. The table satisfies

$$\sum_{x \in \{0,1\}^\ell} \widetilde{\mathcal{A}}(W)(x)^2 - S = E_{\mathrm{enc}}(W) + \eta S_\rho(W). \tag{2}$$

The verifier chooses these batching coins after the input, output and auxiliary commitments are fixed. The sumcheck protocol [39] reduces the square sum to the square of a claimed evaluation ν at a verifier point $\mathbf{r}$. Since the table is affine in W, that evaluation must satisfy

$$\nu = o_{\mathbf{r}} + \langle h_{\mathbf{r}}, W \rangle_K. \tag{3}$$

Both the offset and the coefficient row are public. Expanding this equality in a basis of K gives four base-field equations. For each commitment C_g, its public insertion map I_g and fixed public digits f_g give another equation, $A_g(I_g W + f_g) = C_g$. The root statement requires one short opening W satisfying these commitment equations and all four evaluation equations.

2.2 A Recursive Argument for Short Openings

The endpoint asks for an exact linear equation and an upper bound on the norm of its opening. Exact short openings are a distinct target in lattice arguments [18]. LaBRADOR recursively amortizes lattice proofs of quadratic constraints [12], and algebraic reductions of knowledge formalize how extracting a witness for one relation can yield a witness for another [33]. RPS develops such reductions for bounded-norm lattice relations [31]. RoK and Roll uses structured random projections to lower the cost of verification, and SALSAA incorporates sumcheck into reductions for these relations [32,37]. PikkuFold uses layered random projections to certify the norms that occur in a fold [46].

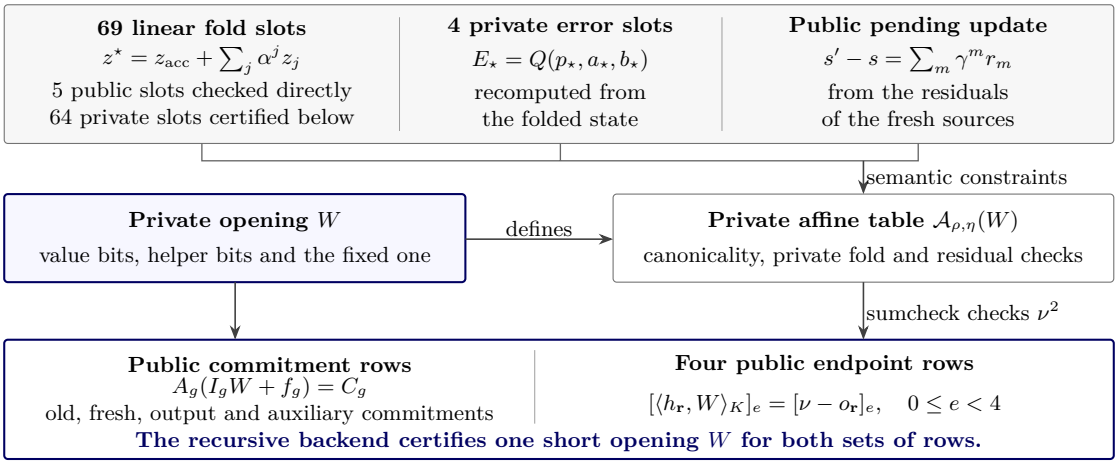

Fig. 1. How the root statement binds the evaluation of the prover's private table to the original commitments.

In Sonata, each nonterminal layer begins with commitments to information about the witness blocks. Only then does the verifier issue a projection seed. The prover submits an image and the verifier checks its norm. The parties aggregate the public equations and the equations for the submitted projection. Symmetric bilinear products between blocks account for the cross terms in the aggregate. The prover commits to their encodings before receiving a short challenge vector. Once that challenge is fixed, the equations on the decomposed response and auxiliary openings are linear. This produces the next short-opening instance. At the terminal layer, the verifier checks the short response and the corresponding ring equations directly.

2.3 Knowledge Extraction from Adaptive Responses

The extractor saves a first execution and rewinds copies of the prover at specified message boundaries. Each returned witness must correspond to the public instance in that first execution. A malicious prover can change its responses across forks, and a second context may contain different auxiliary commitments. We handle these choices in four steps.

Local comparison before denominator clearing. Hold all but coordinate j of a challenge vector fixed. Two distinct accepting points give a response difference X_j and a challenge difference Δ_j. The choice of challenge support makes Δ_j a unit in R_q, a property studied for short challenges and subtractive sets [40,3]. We can therefore recover a candidate X_j/Δ_j modulo q. Its inverse denominator may have a large integer norm. Following *Compare Before Clearing* [51], we retain the differences from both contexts. If the two openings refer to the same block image, they give

$$v_j = X_j \Delta'_j - X'_j \Delta_j, \qquad A_i v_j = 0 \bmod q.$$

If the candidates differ, v_j is nonzero modulo q. Bounds on the response differences and the short challenges give a bound on its integer norm.

Labelled two-point extraction. The pre-projection commitment fixes a label common to the two contexts. Within one context, the commitment sent before the short challenge provides a second label for the auxiliary products. Distinct short openings of a shared label give a kernel immediately. When the pre-projection openings agree across contexts, both candidates refer to the same block image, even though each context has its own auxiliary products. This is the condition needed for the local comparison.

Bounding bad slopes. A further difficulty arises if two contexts produce the same invalid candidate w. Fix w before sampling the second context. At a defective coordinate, restricting the consistency equation to a line of slope w_j gives a quadratic with a nonzero leading coefficient. The center is already a root. Because distinct challenge differences are units, at most one further point on that line can be a root. We apply this count to forks whose labels agree with the center and whose response difference actually yields w_j.

Let D_j be the event that an ordinary center has no effective fork. Let $B_j(w)$ be the event that the center is defective for w and its first effective fork has compatible labels and slope w_j. With M_j possible challenges at this coordinate, counting over the uniform center and the scan of its other points gives

$$\Pr[D_j] + \Pr[B_j(w)] \leq 2/M_j.$$

The center may have been reached on an accepting path. Counting its mass jointly with that of the fork accounts for the bias that acceptance introduces. Adaptive special soundness also distinguishes extracting usable accepting transcripts from recovering a witness from them [7].

Acceptance-weighted composition. If the first candidate is invalid, the extractor independently redraws a context until it obtains a nonempty outcome. Different candidates or conflicting openings yield a short kernel. A repeated invalid candidate is charged according to its probability mass in the two draws. This bounds the total repeated-candidate mass without summing a separate error allowance for every possible w. The statistical error adds across layers, while a separate recurrence bounds the expected work of rewinding.

2.4 Authenticated Histories and Batch Settlement

Each ordinary transition updates a public pending value with a weighted sum of the fresh residuals. Starting from an authenticated zero-pending state, a verifier checks that each transition opens the preceding public state. A linked history that returns to zero pending value certifies its fresh sources up to the extraction error. Its statistical budget counts every attempt that exposed a challenge, including attempts the prover later abandoned.

A verifier may instead retain the original commitments, public fields and fold challenges for several steps, then freeze the ordered log. The prover supplies a shared opening for each state that appears on both sides of a step boundary. A single certificate checks the strict source equations and all of the recorded transitions at the checkpoint. This trades earlier certification of the first sources for fewer sumcheck rounds and backend invocations.

3 Preliminaries and Relations

3.1 Notation and Algebraic Setting

We use the prime $q = 2^{48} - 59$, the extension field $K = \mathbb{F}_q[U]/(U^4 - 4U^2 + 2)$, and the ring $R_q = \mathbb{F}_q[X]/(X^{64} + 1)$. Write $R = \mathbb{Z}[X]/(X^{64} + 1)$.

Elements of K and R_q are represented in their respective power bases. For a field or ring vector w, $\mathrm{cen}(w)$ denotes the coefficient-wise centered integer lift. Its Euclidean norm is written $\|\mathrm{cen}(w)\|_2$.

Sonata uses linear commitments over R_q. The public parameters consist of uniformly sampled ring elements from which the commitment matrices are derived. Each matrix takes a fixed prefix of this shared array in column-major order. Different commitment roles may therefore use overlapping entries. Binding uses the short integer solution problem for these module matrices. Short openings of structured lattice commitments and proofs of exact opening knowledge are studied in [10,18]. Appendix B accounts for the shared prefixes in the reduction.

3.2 Committed States and Folding Relations

Definition 3.1 (Committed state). *A public record is $\xi = (p, C)$, where $p = (u, x_1, \ldots, x_4) \in K^5$ and $C \in R_q^4$. Its witness is $\omega = (a, b, E, d)$, with $a, b \in K^{4\times 8}$, $E \in K^4$, and $d \in R_q^{219}$.*

The 73 slots of v contain p, then the pairs $(a_{\ell,t}, b_{\ell,t})$ in lane order, then the four private errors E_ℓ. The encoding $\mathrm{Enc}(v)$ records each slot in the power basis of K and each base-field component as 48 little-endian bits of its least nonnegative representative. Packing consecutive groups of 64 bits in increasing powers of X gives 219 ring elements. This choice also fixes where to authenticate the errors. Mova, for example, folds R1CS without a commitment to the error or cross terms [27]. Here E is part of the state opening.

For each lane $\ell \in [4]$, put $Q_\ell(p, a, b) = \sum_{t=1}^{8} a_{\ell,t} b_{\ell,t} - u x_\ell$.

Writing $Q = (Q_1, \ldots, Q_4)$, the canonical-opening and accumulated-state relations are

$$\mathcal{R}_{\mathrm{can}}(\xi; \omega) \iff d = \mathrm{Enc}(v) \wedge A_{\mathrm{st}} d = C,$$
$$\mathcal{R}_{\mathrm{acc}}((\xi, s); \omega) \iff \mathcal{R}_{\mathrm{can}}(\xi; \omega) \wedge Q = E.$$

The public scalar $s \in K$ records the pending value of the accumulator. We write $\mathcal{R}^0_{\mathrm{acc}}$ when $s = 0$ is additionally required.

Definition 3.2 (Fresh and strict sources). *A fresh source satisfies the canonical-opening relation with $u = 1$, $x \in \mathbb{F}_q^4$, $a, b \in \mathbb{F}_q^{4\times 8}$, and $E = 0$. We denote this relation by* Fresh.

A strict source also satisfies $Q(p, a, b) = 0$. We denote the conjunction of this equality and $\mathrm{Fresh}(\xi; \omega)$ by $\mathcal{R}_{\mathrm{src}}(\xi; \omega)$.

A fresh source may therefore have $E = 0$ while its actual quadratic residuals are nonzero. Sonata records these residuals through the pending update.

Definition 3.3 (Ordinary folding transition). *An ordinary transition takes an accumulated state (ξ_0, s) and k fresh sources $\xi_1, \ldots, \xi_k$. After the inputs are fixed, the verifier samples $\alpha \in K$ and $\gamma \in K^\times$. The prover supplies a new accumulated state $(\xi_\star, s')$ and an auxiliary commitment C_{aux}. Its public instance is the tuple of old and fresh records, α, γ, and the output triple $(\xi_\star, s', C_{\mathrm{aux}})$. Its witness consists of all the corresponding openings and the helper vector H.*

Write $y(v)$ for the first 69 state coordinates, excluding the four error coordinates. The fresh quadratic residuals form the vector $r \in K^{4k}$, obtained by concatenating $Q(p_j, a_j, b_j)$ in source order. The transition satisfies

$$y(v_\star) = y(v_0) + \sum_{j=1}^{k} \alpha^j y(v_j),$$
$$s' - s = \sum_{m=1}^{4k} \gamma^m r_m.$$

The first equation fixes the new folded values. The second specifies the update of the public pending value.

The transition relation $\mathcal{R}_{\mathrm{tr}}$ requires the old and new states to satisfy $\mathcal{R}_{\mathrm{acc}}$, every input source to satisfy Fresh*, and the two equations above to hold. The witness also includes auxiliary bits H for the canonical-encoding certificate. These bits must satisfy the certificate constraints and open C_{aux} through $A_{\mathrm{aux}} \operatorname{Pack}(H) = C_{\mathrm{aux}}$. Packing pads the final ring block with zeros.*

Ordinary transitions permit nonzero fresh residuals. A zero-pending decision additionally checks $s' = 0$. Its stronger guarantee for strict sources follows from Theorem 5.1 at the pre-γ checkpoint.

Definition 3.4 (Short-opening relation). *The recursive backend operates on a public linear system $L(W) = Y$, where $L : R_q^{N_i/64} \to \mathbb{F}_q^{m_i}$ is an $\mathbb{F}_q$-linear map at layer i and W is the private opening. Its relation is*

$$\mathcal{R}_{\mathrm{short}}((L, Y, B); W) \iff L(W) = Y \ \wedge \ \|\operatorname{cen}(W)\|_2 \le B.$$

The root system combines the original commitment equations with the terminal evaluation obtained from sumcheck. Algorithm 1 uses the same relation for each subsequent public child instance.

3.3 Knowledge-Extraction Model

We consider classical rewindable provers that may adapt their messages to the verifier's challenges. The extractor operates from a fixed reachable interaction prefix, including the prover's random tape and internal state. We use the reductions-of-knowledge formulation of extracting a witness for a prescribed relation [33].

First-execution requirement. The extractor first continues the prover with fresh verifier randomness, producing a transcript τ. This is the original execution. Subsequent rewinding queries restore saved prover configurations and use independent continuation randomness.

If τ is accepted, the extractor must recover a witness for its original public instance or produce a nonzero short kernel for a designated public commitment matrix. In particular, a witness recovered from another execution must still open the commitments recorded in τ.

A short kernel v for a matrix A satisfies

$$Av = 0 \pmod{q}, \qquad v \not\equiv 0 \pmod{q}, \qquad \|v\|_2 \le \beta,$$

where β is the bound associated with that matrix. Extraction failure is measured jointly with acceptance of the original execution. Rejected executions carry no witness obligation.

Extraction checkpoints. The ordinary-transition extractor starts after the output record, pending value, and all relevant commitments have been fixed, but before the batching challenges ρ and η.

The zero-pending extractor also uses an earlier checkpoint, immediately before γ is sampled. At this point, α and the input commitments are fixed, while the output record and auxiliary commitment may change between rewound executions.

All rewinding occurs on private copies of the prover's state. The real verifier retains its issued challenges. For a sequence of transitions, it maintains a nonrollback attempt counter, including attempts abandoned after challenge issuance.

4 The Sonata Construction

4.1 Certifying Canonical Encodings

The output of a folding transition is represented by a new canonical opening. To authenticate this representation, the verifier must check both its binary digits and their consistency with the prescribed folded values. Bitness alone admits invalid encodings. Since $q = 2^{48} - 59$, the 48-bit range contains 59 integers that are not canonical representatives of $\mathbb{F}_q$.

We construct a quadratic certificate for this fixed range. Each base-field component is represented by 48 value bits $b_0, \dots, b_{47}$ and seven auxiliary bits $h_0, \dots, h_4, e, m$. The auxiliary bits allow the upper range restriction to be expressed by a quadratic penalty over the integers.

The certificate acts on $x = (b_0, \dots, b_{47}, h_0, \dots, h_4, e, m) \in \mathbb{Z}^{55}$. The helper bits determine $t = h_0 + 2h_1 + 4h_2 + 6h_3 + 7h_4$ and $r = 41 - \sum_{i=6}^{47} b_i + e - 2t$. Its penalty is

$$\begin{aligned} E(x) &= 8B(x) + r(r-1) + F(x), \\ B(x) &= \sum_{i=0}^{54} x_i(x_i - 1), \\ F(x) &= eb_2 - 2em - 2b_2m + 3m + e(b_3 + b_4 + b_5) + m(b_0 + b_1). \end{aligned}$$

The bitness term $B(x)$ vanishes on binary coordinates. The remaining terms enforce the canonical range. In particular, if all upper 42 value bits are one, the auxiliary conditions restrict the lower six bits to the integers from zero through four. These are exactly the admissible representations at the upper end of the field.

The certificate also applies to extracted openings whose coordinates have not yet been shown to be binary. We therefore prove the integer bound before reducing modulo q.

Lemma 4.1 (Canonical encoding certificate). *The quadratic penalty E is nonnegative over $\mathbb{Z}^{55}$. Its zero set projects exactly onto the canonical 48-bit representations of elements in $\mathbb{F}_q$. Every zero-penalty assignment is binary, and every canonical representation admits auxiliary bits making the penalty zero. For a centered opening W containing V active base-field components and satisfying $\|\operatorname{cen}(W)\|_2^2 \le B_0^2$, the total encoding penalty obeys*

$$0 \le E_{\mathrm{enc}}(W) \le 956B_0^2 + 3281V < q.$$

The final inequality holds under the root extraction budgets of all four concrete profiles.

Proof. Rounding an integer coordinate to its nearest binary value bounds the change in the mixed terms of F by the bitness penalty. This establishes nonnegativity. Equality forces all coordinates to be binary and the auxiliary range conditions to hold. These conditions exclude the 59 noncanonical representations, while every canonical representation admits a valid auxiliary assignment. Bounding the quadratic and affine terms by the squared norm of the centered opening gives the stated upper bound. Appendix A.1 gives the coordinatewise inequalities. □

The bound ensures that a nonzero integer penalty cannot vanish after reduction modulo q. Thus a field-level zero check also establishes canonicality for any extracted opening within B_0.

To use the certificate in sumcheck, we express it as a shifted sum of affine squares over K. The factorization in Appendix A.2 provides affine forms $\psi_0, \dots, \psi_{54}$ and a public constant κ_{enc} satisfying $\sum_{j=0}^{54} \psi_j(x)^2 = E(x) + \kappa_{\mathrm{enc}}$. This identity holds for arbitrary inputs over K, including nonbinary coordinates.

4.2 From Quadratic Checks to a Short-Opening Instance

The encoding certificate must be checked together with the prescribed folding relation. We combine these conditions into one quadratic table, then authenticate its terminal evaluation against the original commitments.

Combining the constraints. The semantic checks consist of eight state residual equations, 64 private-coordinate folding equations, and the pending update. The first 72 discrepancies are denoted by $\delta_1(W), \dots, \delta_{72}(W)$. The fresh sources contribute the $4k$ quadratic residuals $r_1(W), \dots, r_{4k}(W)$ from Section 3. With the previously issued pending challenge γ and the new batching challenge ρ, their combined discrepancy is

$$S_\rho(W) = \sum_{j=1}^{72} \rho^j \delta_j(W) + \sum_{m=1}^{4k} \gamma^m r_m(W) - (s' - s).$$

The verifier samples $\rho, \eta \in K^\times$ only after the output and auxiliary commitments have been fixed. These coins batch the discrepancies using the polynomial identity bound [48,54]. The field check combines the encoding penalty and the semantic discrepancy as $E_{\mathrm{enc}}(W) + \eta S_\rho(W)$.

Constructing the square table. The encoding penalty already has an affine-square representation. Quadratic products in the state and source relations can be handled similarly. For $\iota \in K$ satisfying $\iota^2 = -1$, the identity

$$ab = \left(\frac{a+b}{2}\right)^2 + \left(\frac{\iota(a-b)}{2}\right)^2$$

expresses a product as two squares of affine forms. A public coefficient can be incorporated into one factor, and affine terms are obtained by taking the other factor to be one.

Applying these identities produces a table $\mathcal{A}(W)$ whose entries are affine in the opening W. For k fresh sources, the number of active base-field components is $V(k) = 544 + 64k$. The resulting table contains $T(k) = 55V(k) + 64(k+2) + 2$ entries and is padded with zeros to length 2^ℓ, where $\ell = \lceil \log_2 T(k) \rceil$.

Each encoding group contributes the same public shift κ_{enc}. Together with the pending update, this gives the initial claim $S = V(k)\kappa_{\mathrm{enc}} + \eta(s' - s)$. The resulting table satisfies the quadratic identity (2) for every opening W. In particular, the identity does not assume that the extracted opening is already canonical.

Reducing the table. The parties execute a degree-two sumcheck protocol [39] on the square of the multilinear extension of $\mathcal{A}(W)$. In each round, the prover sends only the constant and quadratic coefficients g_0, g_2. The verifier recovers the linear coefficient from the running claim t as $g_1 = t - 2g_0 - g_2$, then samples the next evaluation point.

After ℓ rounds, the prover supplies a terminal value ν. The verifier checks that ν^2 equals the final sumcheck claim.

The remaining task is to show that ν is the evaluation of the table derived from the committed opening. Since every table entry is affine in W, its multilinear extension at the verifier's point r is affine and must satisfy

$$\nu = o_r + \langle h_r, W \rangle_K.$$

Both o_r and h_r are publicly computable from the table description and the verifier's challenges. Expanding this equation in the fixed basis of K gives four linear equations over $\mathbb{F}_q$.

Building the root system. The verifier combines these four equations with the original commitment equations. For a commitment role with a fixed public opening part f and insertion map I, the corresponding equation is $A_{\mathrm{role}} I W = C_{\mathrm{role}} - A_{\mathrm{role}} f$. A fixed-one equation and the required zero-padding equations complete the system.

All rows act on the same opening W. The resulting public instance is (L_0, Y_0, B_0), and its witness must satisfy $\mathcal{R}_{\mathrm{short}}((L_0, Y_0, B_0); W)$.

By Definition 3.4, a valid root opening links the sumcheck endpoint to the state, source, output and auxiliary commitments. Its short-opening witness is certified by the recursive argument below.

4.3 The Online Transition

Protocol 1 certifies the ordinary relation of Definition 3.3. Each input source satisfies the fresh type of Definition 3.2, which permits a nonzero residual. The verifier first fixes the input records and samples the folding challenges. After receiving the output and auxiliary commitments, it samples the batching challenges for sumcheck. The short-opening backend then authenticates the terminal evaluation.

Protocol 1. Online Sonata Transition

Public input. The profile and CRS, the old record (ξ_0, s) and the fresh records $(\xi_j)_{j=1}^{k}$.
Prover witness. Canonical openings of the old state and fresh records.
Output. An updated state $(\xi_\star, s')$ upon acceptance, or rejection.

Prover $\mathcal{P}$	**Message**	**Verifier** $\mathcal{V}$
	$\xleftarrow{\alpha, \gamma}$	Fix the input commitments. Check public types and continuity. Draw $\alpha \leftarrow K$, then $\gamma \leftarrow K^\times$ independently.
Fold the 69 semantic slots, compute $E_\star$ and s', and commit to the canonical output and helper opening.	$\xrightarrow{\xi_\star, s', C_{\mathrm{aux}}}$	Check the five-coordinate public fold and fix the output triple.
Build the affine table on the joint opening.	$\xleftarrow{\rho, \eta}$	Draw $\rho, \eta \leftarrow K^\times$ independently. Initialize claim $t = S$.
At round j compute the constant and quadratic coefficients of g_j.	$\xrightarrow{g_{j,0}, g_{j,2}}$	Recover the linear coefficient from $g_j(0) + g_j(1) = t$.
Restrict the table to r_j.	$\xleftarrow{r_j}$	Draw $r_j \leftarrow K$ and set $t = g_j(r_j)$. Repeat for $j = 1, \ldots, \ell$.
Evaluate the terminal affine table entry.	$\xrightarrow{\nu}$	Check $\nu^2 = t$. Construct $I_0 = (L_0, Y_0, B_0)$ from the public transcript and CRS.
Use the private root opening for the backend call.		Run Protocol 2 at the root, or Protocol 3 if terminal. Accept after all checks succeed. The zero-pending decision also checks $s' = 0$.

The record $\xi_\star = (p_\star, C_\star)$ carries both the public output fields and their state commitment. Each party computes the public root instance locally. The prover constructs the backend witness from its private openings, while the verifier uses the public rows of the root system. These rows and the sumcheck endpoint authenticate the 64 private folded coordinates.

4.4 A Recursive Certification Layer

The backend certifies a sequence of short-opening instances. Each nonterminal layer takes a public instance $I_i = (L_i, Y_i, B_i)$ and produces the next instance of the same relation. The prover's opening remains private throughout this reduction. Earlier lattice arguments authenticate computations using succinct commitment openings or delegation [17,16].

At layer i, the opening is partitioned into s_i blocks $w_0, \ldots, w_{s_i-1} \in R_q^{n_i}$. The public block map A_i gives block images $t_j = A_i w_j$. The profile specifies S_i for the squared norm of an honest opening, G_i for the response energy threshold, and B_i for the extraction norm.

Fixing the block images The prover first commits to radix-d_i decompositions of the block images through $u_1 = M_{t,i}\,\mathrm{Dig}_{d_i}(t)$. This tag is fixed before the projection seed is issued. The seed σ defines the public 384-row ternary matrix P_i. On receipt of the image p, the verifier checks its centered squared norm against $384S_i$ and samples $\zeta \leftarrow K$. The original equations and projection equations, with targets (Y_i, p), are grouped in fours as elements of K and combined using powers of ζ. This gives four ring functionals $(\phi_{e,j})_j$ with base-field targets b_e. Tensor-train projections provide another approach to shortness checks [30]. Appendix B analyzes the seeded projection used here.

Certifying the combined response For $0 \le a \le b < s_i$ and $0 \le e < 4$, let

$$h_{ab}^{(e)} = \tfrac{1}{2}\big(\langle \phi_{e,a}, w_b \rangle_R + \langle \phi_{e,b}, w_a \rangle_R\big), \qquad h_{ba}^{(e)} = h_{ab}^{(e)}.$$

We store the upper triangles. For a challenge vector c put $\Phi_e(c) = \sum_j c_j \phi_{e,j}$ and $H_e(c) = \sum_j c_j^2 h_{jj}^{(e)} + 2\sum_{a<b} c_a c_b h_{ab}^{(e)}$. Then the honest response $z = \sum_j c_j w_j$ satisfies $\langle \Phi_e(c), z \rangle_R = H_e(c)$ and $\sum_j \mathrm{ct}(h_{jj}^{(e)}) = b_e$.

Protocol 2. Nonterminal Short-Opening Certification

Public input. I_i, the nonterminal layer profile and its public keys $A_i, M_{t,i}, M_{h,i}$.
Prover witness. An opening $W_i = (w_j)_j$ for $L_i(W_i) = Y_i$. For honest completeness, $\|\operatorname{cen}(W_i)\|_2^2 \le S_i$.
Output. A public child instance I_{i+1} and its certification verdict. The prover retains the corresponding child witness.

Prover $\mathcal{P}$	**Message**	**Verifier** $\mathcal{V}$
Decompose $t_j = A_i w_j$ in radix d_i and compute $u_1 = M_{t,i} d_t$.	$\xrightarrow{u_1}$	Fix the first tag.
Expand P_i and compute $p = P_i W_i$.	$\xleftarrow{\sigma}$	Draw a fresh projection seed.
Retry, or submit the bounded image p.	$\xrightarrow{0 \text{ or } (1,p)}$	On retry draw another seed. On submission check the length and $\|\operatorname{cen}(p)\|_2^2 \le 384 S_i$.
Derive ϕ, b and compute the upper triangles h.	$\xleftarrow{\zeta}$	Draw $\zeta \leftarrow K$. Derive the same ϕ, b from public data.
Set $u_2 = M_{h,i} d_h$ for the radix digits of h.	$\xrightarrow{u_2}$	Fix the second tag.
Compute $z = \sum_j c_j w_j$.	$\xleftarrow{c}$	Draw independent coordinates from $\mathcal{C}$.
Retry, or split z and retain its private child opening.	$\xrightarrow{0 \text{ or } 1}$	On retry draw another c. On submission run Algorithm 1 and certify its child exactly once. Accept iff the child accepts.

The challenge support $\mathcal{C}$ consists of ring elements with squared coefficient norm at most 128 and multiplication operator norm at most 26. Both backend protocols use one control byte per projection or response attempt. Zero requests a fresh draw at that stage. One closes the stage. There are at most R_P projection seeds with the same first tag and R_C response challenges with all preceding messages fixed. Exhausting either budget rejects. A malformed submission or a failed child rejects the layer immediately. The first tag binds the block images across projection contexts. The second binds the auxiliary products within one context before the short challenge is chosen. Section 5 uses these labels to compare responses obtained by rewinding an adaptive prover.

Constructing the child instance The prover checks $\|\operatorname{cen}(z)\|_2^2 \le G_i$ and constructs the child witness privately as $W_{i+1} = (z_{\mathrm{lo}}, z_{\mathrm{hi}}, d_t, d_h)$ with the required zero padding. Its squared norm is at most S_{i+1} by Appendix A.5. Once c is fixed, the constraints on these coordinates are linear. Algorithm 1 constructs the public system using formal reconstruction expressions for z, t, h.

Algorithm 1. BuildChild: Constructing the Next Short-Opening Instance

Public input. The parent I_i, its profile and keys, the submitted $(u_1, \sigma, p, \zeta, u_2, c)$ and the next layer's dimensions and extraction bound.
Output. $I_{i+1} = (L_{i+1}, Y_{i+1}, B_{i+1})$.

1. Derive ϕ, b from (I_i, σ, p, ζ). Allocate formal ring coordinates $(z_{\mathrm{lo}}, z_{\mathrm{hi}}, d_t, d_h)$ of lengths $n_i, n_i, 16 s_i \ell_i, 4\binom{s_i+1}{2}\ell_i$, respectively, where $\ell_i = \lceil 48/\log_2 d_i \rceil$.
2. As linear expressions in those coordinates, form $z = z_{\mathrm{lo}} + d_i z_{\mathrm{hi}}$ and radix-recompose each group of ℓ_i digits in d_t, d_h into $t_j, h_{ab}^{(e)}$. Set $h_{ba}^{(e)} = h_{ab}^{(e)}$.
3. Add the tag, block-image, principal and diagonal-sum rows, in order:

$$
\begin{aligned}
M_{t,i} d_t &= u_1, \qquad & M_{h,i} d_h &= u_2, \\
A_i z - \sum_j c_j t_j &= 0, & & \\
\langle \Phi_e(c), z \rangle_R - H_e(c) &= 0 & & (0 \le e < 4), \\
\sum_j \mathrm{ct}(h_{jj}^{(e)}) &= b_e & & (0 \le e < 4).
\end{aligned}
$$

   Expand each ring equality into 64 base-field rows. The four diagonal-sum equations are already base-field rows.
4. Let n_{raw} be 64 times the total number of ring coordinates in step 1. Extend the formal opening to N_{i+1} coefficients and append a separate identity row with target zero for every coefficient from n_{raw} to $N_{i+1} - 1$ (zero-based). Return the resulting map, its targets and B_{i+1}.

4.5 Terminal Pivot Verification and Completeness

The final layer checks the short response directly. Before receiving the projection seed, the prover supplies the non-pivot block images $t_1, \ldots, t_{s_i - 1}$ and a commitment $T_0 = M_{\mathrm{pivot}} \, \mathrm{Dig}_8(t_0)$ to the 16 canonical balanced base-eight digits of each coefficient of t_0. The first challenge coordinate is drawn from $\mathcal{C} \setminus \{0\}$, whose elements are units modulo q under the chosen parameters.

The projection and aggregation stages proceed as in a nonterminal layer. After the aggregation challenge, the prover supplies the upper-triangular auxiliary values for the four principal equations, omitting the four diagonal coefficients $\mathrm{ct}(h_{00}^{(e)})$ that can be recovered from the public targets. Write $\widehat{h}$ for this message. Its coefficients are fixed before the verifier samples c.

Protocol 3. Terminal Pivot Verification

Public input. The terminal instance I_i, its profile and keys A_i, M_{pivot}.
Prover witness. An opening $W_i = (w_j)_j$ for $L_i(W_i) = Y_i$, of honest squared norm at most S_i.
Output. The verifier's accept or reject decision.

Prover $\mathcal{P}$	Message	Verifier $\mathcal{V}$
Compute the block images and tag the pivot digits.	$\xrightarrow{T_0, (t_j)_{j>0}}$	Fix the tag and non-pivot images.
Expand P_i and compute the image.	$\xleftarrow{\sigma}$	Draw a fresh projection seed.
Retry, or submit $p = P_i W_i$.	$\xrightarrow{0 \text{ or } (1,p)}$	On retry redraw the seed. Otherwise check image length and squared norm $\le 384 S_i$.
Compute all four symmetric upper triangles.	$\xleftarrow{\zeta}$	Draw $\zeta \leftarrow K$ and derive the public ϕ, b.
Omit one diagonal constant per lane.	$\xrightarrow{\widehat{h}}$	Fix the received coefficients.
Compute $z = \sum_j c_j w_j$.	$\xleftarrow{c}$	Draw c_0 from $\mathcal{C} \setminus \{0\}$ and the other coordinates independently from $\mathcal{C}$.
Retry, or submit the bounded response.	$\xrightarrow{0 \text{ or } (1,z)}$	On retry redraw c. Otherwise check shape and $\|\operatorname{cen}(z)\|_2^2 \le G_i$. Accept iff the pivot checks below pass.

The verifier enforces the prescribed shapes and canonical encodings on all submitted fields. Since c_0 is a unit, the block-image equation and the four diagonal sums determine the missing values:

$$t_0 = c_0^{-1}\left(A_i z - \sum_{j>0} c_j t_j\right), \qquad \operatorname{ct}(h_{00}^{(e)}) = b_e - \sum_{j>0} \operatorname{ct}(h_{jj}^{(e)}).$$

The verifier recomputes the base-eight digit tag of this t_0 and compares it with T_0. With the restored constants it checks all four principal identities $\langle \Phi_e(c), z\rangle_R = H_e(c)$. A failed check rejects the submitted response.

Proposition 4.1 (Honest completeness). *For a valid old accumulated state and canonical fresh inputs, the honest prover constructs an ordinary transition accepted with probability at least* $1 - \varepsilon_{\text{comp}}$*, where*

$$\varepsilon_{\text{comp}} \le \sum_{i=0}^{4} \left[\left(\frac{1}{2} + \delta_i\right)^{R_P} + 2^{-R_C}\right].$$

Here δ_i *bounds the fixed-input acceptance-probability difference between the seeded projection and an independent ternary projection at layer* i*. With* $R_P = R_C = 160$*, all four five-layer profiles satisfy* $\varepsilon_{\text{comp}} < 11 \cdot 2^{-160}$*.*

Proof. Canonical encoding gives the honest prover a root opening within its prescribed norm bound. The square-table and affine-endpoint identities hold exactly, so the front-end field checks accept for every verifier challenge.

At backend layer i, the expected squared norm of an independently projected honest opening is at most $192S_i$. Markov's inequality bounds rejection at the threshold $384S_i$ by $1/2$, with an additional δ_i for the seeded projection. The short-challenge distribution similarly gives expected squared response norm at most $128S_i$, so one draw fails the response test with probability at most $1/2$.

Successful submissions satisfy the next-layer linear equations and honest norm budget. At the terminal pivot, the recovered block image and auxiliary products agree with the honest values. Summing the retry-exhaustion probabilities over the five layers proves the claim. Appendix A.5 supplies the integer norm analysis.

5 Knowledge Soundness

We establish knowledge soundness of Sonata against adaptive classical provers. Successful extraction must explain the commitments and responses of the original accepted execution. As in adaptive special soundness, the extractor selects further challenges using the responses it has seen [7]. Rewinding may produce different responses and auxiliary openings. We recover candidates from local challenge differences and compare them across independent contexts. A labelled counting argument bounds the probability that both contexts return the same invalid candidate.

5.1 Main Extraction Guarantee

The public records and admissible source openings are specified in Definitions 3.1 and 3.2. The extractor for Protocol 1 must recover a witness for the transition in Definition 3.3. It retains the first execution τ under the prescribed verifier randomness, then rewinds private copies of the prover at selected checkpoints.

Theorem 5.1 (Knowledge soundness). *Fix any admissible parameter profile and trusted public CRS. Protocol 1 admits a uniform classical black-box extractor against any resettable classical prover with bounded running time on every queried branch. From any reachable checkpoint after the output and auxiliary commitments and before the draw of ρ, η, it retains the original verifier continuation τ. On acceptance its output is a witness for $\mathcal{R}_{\mathrm{tr}}$ at the public instance of τ, a nonzero short kernel under a designated public matrix, or $\perp$, with joint failure mass*

$$\Pr[\mathrm{Accept}(\tau) \wedge \mathrm{Ext} = \perp] \leq \kappa_{\mathrm{tr}}. \tag{4}$$

The guarantee also applies from the earlier checkpoint immediately before γ is sampled, with α and the input records fixed. In this experiment, extraction has joint error at most κ_{zero}. If the original execution accepts with $s' = 0$, a successfully extracted witness additionally has $s = 0$ and satisfies all fresh quadratic relations strictly. Every kernel output is nonzero modulo q, lies in the kernel of its designated public matrix, and satisfies that matrix's integer norm bound.

The error budget separates the recursive backend from the front-end field checks. At backend layer i, the witness is divided into s_i blocks, and challenge coordinate j has support size M_{ij}. The aggregation polynomial has degree at most $D_i = \lceil (m_i+384)/4 \rceil - 1$, where m_i includes the alignment equations. For any fixed opening exceeding the extraction norm budget B_i, a fresh seeded projection passes its norm test with probability at most e_i.

With d backend layers and ℓ sumcheck rounds, the statistical bounds are

$$\begin{aligned} \varepsilon_i &= R_P e_i + \frac{D_i}{|K|} + 2R_C \sum_{j=0}^{s_i-1} \frac{1}{M_{ij}}, \\ \kappa_{\mathrm{tr}} &= \sum_{i=0}^{d-1} \varepsilon_i + \frac{72}{|K|-1} + \frac{2\ell}{|K|}, \\ \kappa_{\mathrm{zero}} &= \kappa_{\mathrm{tr}} + \frac{4k}{|K|-1}. \end{aligned} \tag{5}$$

The terms in ε_i charge the projected norm test on a fixed long opening, a false aggregate, and unsuccessful coordinate extraction with at most R_P projection seeds and R_C short challenge vectors issued per layer. The front-end terms account for semantic batching, sumcheck and the zero-pending check.

Appendix B proves the short-kernel bounds (Lemma B.2) and the fixed-candidate projection estimate (Lemma B.3). The computational reduction in Theorem 6.1 samples the shared CRS uniformly and bounds the probability of a kernel using six time-bounded Module-SIS advantages.

5.2 Compatible Records and Local Comparison

The backend extractor starts from Protocols 2 and 3. At a nonterminal layer a child opening for the short relation of Definition 3.4 certifies the block-image and auxiliary equations produced by Algorithm 1. The terminal pivot checks the corresponding equations directly. We call either ordinary outcome a *certified record*. Lemma B.2 bounds its centered response norm by $Z_i = \sqrt{1+d_i^2}\, B_{i+1}$ at a nonterminal layer and by $Z_i = \sqrt{G_i}$ at the pivot.

Recovering a candidate from two challenges Consider an ordinary certified center (c, z). For a fork at coordinate j, restore the snapshot saved before drawing c and leave c_{-j} unchanged. Suppose a distinct coordinate value x gives another certified ordinary record, with response z'. Write $\Delta_j = x - c_j$ for the difference of the integer challenges.

The two records must first agree on the openings of the auxiliary tags fixed before c. If they disagree, subtracting their short openings gives a nonzero kernel of the associated commitment matrix. Otherwise, they use the same block images t_j. Subtracting their certified block-image equations gives

$$A_i X_j = \Delta_j t_j \pmod q, \qquad X_j = \mathrm{cen}(z') - \mathrm{cen}(z).$$

By Lemma B.1, Δ_j is a unit in R_q. Hence $W_j = \Delta_j^{-1} X_j$ satisfies $A_i W_j = t_j \bmod q$.

This equation determines the candidate over R_q, but says little about its integer norm. The challenge difference is short, while its modular inverse may have a large integer representation. Following *Compare Before Clearing* [51], we retain X_j and Δ_j instead of using the inverse to establish a norm bound.

Comparing independent contexts If the first candidate fails the original linear system or its norm bound, the extractor returns to the earlier checkpoint immediately after the pre-projection tag u_1. Each new continuation receives independent projection and aggregation coins and may produce a different second tag u_2'. At the pivot the shared first message consists of the non-pivot block images and the commitment to the first image.

At a nonterminal layer the second tag remains fixed only during the coordinate forks of its own context. The extractor compares the first-tag openings across contexts. At the pivot it checks the opening of the common first-image commitment.

If these openings differ, subtraction gives a short commitment kernel. Otherwise the block images agree across the two contexts, even though the projections and second tags may differ.

Suppose two contexts produce candidates W_j and W_j', with local difference pairs (X_j, Δ_j) and (X_j', Δ_j'). Their comparison yields

$$v_j = X_j \Delta_j' - X_j' \Delta_j, \qquad A_i v_j = 0 \pmod q, \qquad \|v_j\|_2 \le 8\Gamma Z_i. \tag{6}$$

Indeed, the response differences have norms at most $2Z_i$, and multiplication by either challenge difference has operator norm at most 2Γ. The stated bound follows by the triangle inequality. If the candidates differ, $v_j \not\equiv 0 \pmod q$, because $\Delta_j \Delta_j'$ is a unit.

Binding the candidate to the original response A candidate passing the original linear equations and norm bound must also explain the response in the retained first execution. The extractor forms the following integer lift of their difference, with each c_j taken in its short integer representation.

$$a = \operatorname{cen}(z) - \sum_j c_j \operatorname{cen}(W_j).$$

The block-image equations give $A_i a = 0 \bmod q$. By the triangle inequality and Cauchy–Schwarz, $\|a\|_2 \le Z_i + \Gamma\sqrt{s_i} B_i$. Each profile checks this against its block-kernel radius. If a is nonzero modulo q, return it. Otherwise W reconstructs z on the retained first challenge and can supply the witness required by the parent.

5.3 Labelled Bad-Slope Counting

If independent contexts return the same invalid candidate, their cross-product difference vanishes. We bound this event while allowing the prover to adapt both the auxiliary openings and the responses.

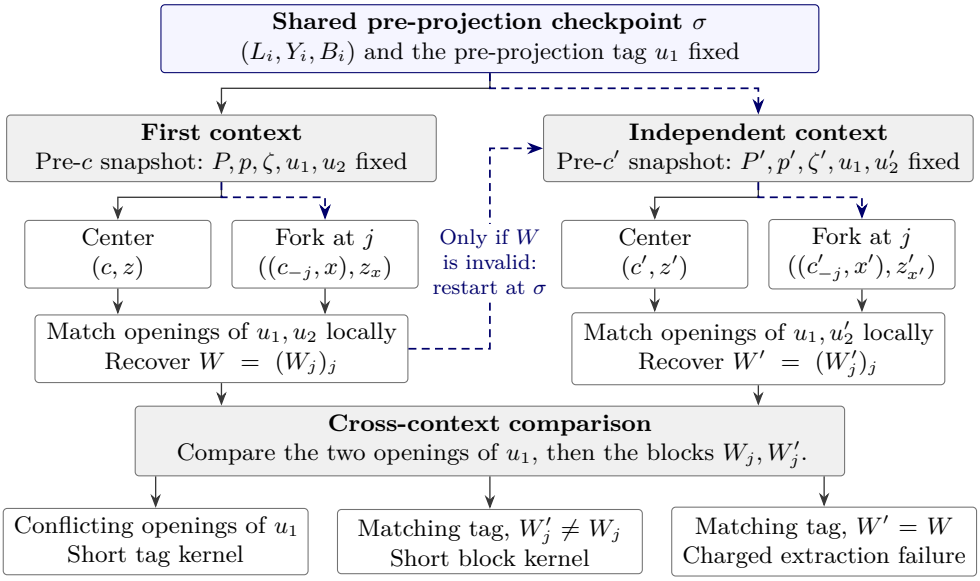

Fig. 2. Two rewind levels in labelled extraction. Each fork fixes its local second tag. Only an invalid first candidate triggers independent trials from the shared checkpoint.

Accounting for adaptive responses Fix the checkpoint immediately before the short challenge c, including the prover state and its random-tape position. Associate an independent verifier and child-extractor suffix with every complete challenge. This defines a conceptual table. Each entry is obtained by applying the actual prover strategy to the chosen challenge and restoring the saved state for that query.

The table can be sampled as entries are queried. In particular, the original challenge remains uniform and independent of the table, and its execution uses the continuation randomness assigned to that entry.

A point is effective if immediate submission returns a certified ordinary record or a short kernel. An immediate retry, rejection or failed child extraction gives an empty entry. The scan stops at the first effective point and compares an ordinary record only after testing its shared openings.

Bounding a fixed bad slope Fix a candidate w independently of the projection and consider a coordinate j. Its challenge fiber consists of the M_{ij} points obtained by varying c_j while keeping every other coordinate fixed.

For this coordinate, an ordinary center is defective for w if some lane e satisfies $\langle \phi_{e,j}, w_j \rangle_R \neq h_{jj}^{(e)}$ at that center.

Two events capture the possible extraction failures. The event D_j occurs when an ordinary certified center has no effective neighbor. The event $B_j(w)$ occurs when a defective ordinary center has a first effective neighbor which is ordinary, has compatible tag openings, and gives the prescribed slope w_j.

Lemma 5.1 (Labelled fiber bound). *For any candidate w fixed before projection, a uniform center and a uniform scan without replacement of the remaining points on a*

coordinate-j fiber satisfy

$$\Pr[D_j] + \Pr[B_j(w)] \le \frac{2}{M_{ij}}. \tag{7}$$

Proof. Fix the response table and the challenge coordinates other than j. Consider a defective ordinary center with challenge c_j and response z. If a neighboring point with coordinate T has compatible auxiliary openings and the same slope w_j, its response must be $z + (T - c_j)w_j$.

Write $c(T)$ for the challenge vector with coordinate j replaced by T. Substituting this response into the principal equation gives

$$\begin{aligned} g_e(T) = &\left\langle \sum_a c_a(T)\phi_{e,a}, z + (T - c_j)w_j \right\rangle_R \\ &- \sum_{a,b} c_a(T)c_b(T)h_{ab}^{(e)}. \end{aligned}$$

The center c_j is a root of this polynomial. Its quadratic coefficient is $\langle \phi_{e,j}, w_j \rangle_R - h_{jj}^{(e)}$, which is nonzero by the defective-diagonal condition.

Any three distinct supported challenge values have pairwise unit differences in R_q. Their Vandermonde matrix is therefore invertible, so a quadratic polynomial with a nonzero leading coefficient cannot vanish at all three values. There is at most one other root besides the center.

We now account for how the extractor chooses that neighbor. Suppose the fiber contains a effective points. If $a = 0$, neither event occurs. If $a = 1$, only a missing fork is possible, with probability at most $1/M_{ij}$.

For $a \ge 2$, every effective center has another effective point. Each defective ordinary center has at most one compatible neighbor with slope w_j, giving at most a directed center-neighbor pairs. Since the scan visits the other effective points in uniformly random order, each pair is selected with probability $1/[M_{ij}(a-1)]$. Their total probability is at most $a/[M_{ij}(a-1)] \le 2/M_{ij}$. □

This counts the joint event that the uniform center is defective and its *first effective neighbor* is ordinary, has compatible openings and has slope w_j. Conditioning on a particular defective center could give bad-slope probability one.

Algorithm 2 applies this scan once per coordinate, retaining the first forward execution. It enters the independent repeat only if the first ordinary candidate is invalid.

Algorithm 2. Labelled Two-Point Extraction

Input. Layer i, its public instance (L_i, Y_i, B_i), and a restorable snapshot σ after the first tag and before any projection seed.
Output. A witness reconstructing the response of the retained first execution, a nonzero short kernel under a designated public matrix, or $\perp$.

Procedure RawNode(σ)

1. Follow a verifier continuation from σ through its permitted retries. Save the pre-c snapshot *before* each short challenge is drawn. On rejection return $\perp$. On submission call the child extractor, or check the terminal pivot. Propagate a kernel or $\perp$. For an ordinary center retain its challenge c, response z, tag openings, and the snapshot saved for that draw.
2. For each j, perform this step and the next. Scan the other supported values of c_j in uniform order without replacement. For every query restore the same pre-c snapshot and keep c_{-j} fixed. Skip immediate retries and failed side extractions. Stop at the first certified ordinary record or kernel. Return $\perp$ on exhaustion and propagate a kernel immediately.
3. Compare the center and fork openings of the tags fixed in this context. A discrepancy yields a short tag kernel. Otherwise store $\Delta_j = x - c_j$, $X_j = \mathrm{cen}(z_x) - \mathrm{cen}(z)$ and $W_j = X_j/\Delta_j \bmod q$. Retain X_j, Δ_j as integer lifts. After all coordinates, return the center, openings and block candidates with their differences.

Procedure ExtractNode(σ)

1. Run RawNode(σ) on the first execution. Propagate $\perp$ or a kernel. If the returned W satisfies $L_i W = Y_i$ and $\|\mathrm{cen}(W)\|_2 \le B_i$, compare $\sum_j c_j W_j$ with the first record's z. Return W on equality modulo q and the short anchor kernel on inequality.
2. For an invalid W, restore σ for each independent RawNode trial. Use fresh suffix coins and repeat until nonempty. If the nonempty result is a kernel, propagate it immediately. For an ordinary result W', compare the two openings of the common pre-projection tag and return the tag kernel if they differ.
3. With matching openings, return the block kernel (6) for any $W'_j \neq W_j$. If all blocks agree, return $\perp$.

The coordinate scans use the checkpoint immediately before c, which keeps both local tags fixed. Independent repeats restore the earlier checkpoint before the projection and may obtain different second tags, as shown in Figure 2.

5.4 Recursive Composition and Expected Work

Lemma 5.1 bounds the probability of obtaining a particular bad slope. The extractor, however, does not choose its output candidate in advance. Its distribution may place different probability masses on many invalid openings. The extractor samples from this adaptive output distribution.

From fixed candidates to adaptive outputs At a pre-projection checkpoint, write m for the unconditional probability that an ordinary center is followed by a missing fork. For each invalid candidate w, write r_w for the unconditional probability that RawNode returns w. A candidate is invalid if it fails the original linear equations or exceeds the extraction norm bound B_i.

Fix w before projection. For a submitted image p, compress $(L_i w - Y_i, P_i w - p)$ into a polynomial over K. If the polynomial is identically zero and w is invalid, w is long and the issued seed passed its norm test, at cost at most $R_P e_i$ across seeds. Otherwise the polynomial is nonzero and fixed before the single aggregation draw ζ, so its vanishing costs at most $D_i/|K|$.

Outside these exceptions, the candidate has a defective diagonal at some coordinate. Applying Lemma 5.1 across the coordinate scans and at most R_C reached short-challenge checkpoints gives, for every fixed invalid w,

$$m + r_w \le R_P e_i + \frac{D_i}{|K|} + 2R_C \sum_{j=0}^{s_i-1} \frac{1}{M_{ij}} = \varepsilon_i.$$

The missing-fork mass m is common to all candidates. Count a virtual scan for every coordinate at each reached checkpoint before conditioning on success at earlier coordinates. Lemma B.4 integrates these unconditional masses over the reached checkpoints.

The extractor can now compare an invalid first candidate against an independently generated second context. Conflicting labels or different candidates yield a short kernel. The only remaining failure occurs when the second context returns the same invalid candidate.

Lemma 5.2 (Acceptance-weighted comparison). *At a pre-projection checkpoint let u be the unconditional mass of the event that the first path accepts and extraction at its submitted child fails. The joint probability that the first path accepts and* ExtractNode *fails is at most $u + \varepsilon_i$.*

Proof. Write t for the probability that RawNode returns any nonempty result, including a kernel. When $t > 0$, the first nonempty result obtained by independent repetition returns a particular invalid candidate w with probability r_w/t.

The first context returns w with unconditional probability r_w. Hence the probability that both contexts return this invalid candidate is r_w^2/t. Conflicting tag openings may still yield a kernel. Summing the coincidence probabilities over invalid candidates and including missing forks and child failures bounds the failure mass by

$$u + m + \sum_{w \text{ bad}} \frac{r_w^2}{t} \le u + m + \max_{w \text{ bad}} r_w \le u + \varepsilon_i. \tag{8}$$

The first inequality uses $\sum_{w \text{ bad}} r_w \le t$. If no bad opening has positive mass, including when $t = 0$, repetition never starts. In that case the missing-fork bound alone gives $m \le R_C \sum_j 1/M_{ij} \le \varepsilon_i$. □

The stopping rule weights each candidate by its mass in two independent trials. The fixed-candidate inequality controls the largest such mass without summing a separate error bound over all openings.

Composing the backend layers At the terminal pivot, a certified response is checked directly, so there is no child extraction failure and $u = 0$. At a nonterminal layer, the original execution enters at most one child submission. That submission is fixed before the child's continuation randomness is sampled.

The child guarantee can therefore be averaged over the reached parent checkpoints, giving $u \leq \kappa_{i+1}$. Failed side queries remain empty entries in the fiber analysis and do not replace the original submitted child. The recursive error consequently satisfies $\kappa_i \leq \kappa_{i+1} + \varepsilon_i$, with $\kappa_d = 0$. This is a reduction of knowledge from a parent relation to a submitted child relation [33,31].

Front-end extraction The ordinary extractor restarts after α, γ, the input, output and auxiliary commitments and the new pending value have been fixed, immediately before either ρ or η is sampled. Each independent continuation draws both field challenges, the sumcheck coins and the backend coins afresh. Its extracted opening authenticates the endpoint of its own field transcript.

The extracted openings are compared through their original commitment labels. If two short openings of a shared commitment disagree, their difference gives a short commitment kernel. Otherwise, they describe the same committed values and auxiliary coordinates.

For a fixed short opening w compatible with those commitments and the publicly fixed root coordinates, the square-table identity of Appendix A.3 gives the following discrepancy. The true sum minus the prescribed initial claim equals $E_{\mathrm{enc}}(w) + \eta S_\rho(w)$, where S_ρ batches the semantic discrepancies and includes the pending discrepancy as its constant term.

If $E_{\mathrm{enc}}(w) > 0$ over the integers, Lemma 4.1 ensures that this penalty remains nonzero modulo q. For each fixed ρ, at most one nonzero η can cancel it. If the encoding penalty is zero but the transition relation fails, the semantic discrepancy is a nonzero polynomial in ρ of degree at most 72. The two cases are disjoint for fixed w, giving a field-test error of at most $72/(|K| - 1)$.

An incorrect sumcheck claim may also become consistent at a later round. At the first such round, the prover's polynomial of degree at most two differs from the correct polynomial for the fixed opening. The verifier's fresh challenge is a root of their nonzero difference with probability at most $2/|K|$. Summing over ℓ rounds contributes at most $2\ell/|K|$.

The backend binds the terminal affine evaluation to the extracted opening. Repeating from this pre-ρ, η checkpoint until an opening or kernel is obtained and comparing the two raw results weights an identical bad opening by its squared mass divided by the total nonempty mass, as in Lemma 5.2. Adding the original backend failure proves the stated bound κ_{tr}. Lemma B.5 gives the full two-checkpoint argument.

The zero-pending guarantee The stronger extraction begins before γ is sampled. At this checkpoint, the old state, α, and the fresh input commitments are already fixed, while subsequent output and auxiliary commitments may change.

For fixed canonical fresh values, a transition satisfying the ordinary relation with $s' = 0$ must satisfy

$$s + \sum_{m=1}^{4k} \gamma^m r_m = 0.$$

If $s \neq 0$ or any fresh residual $r_m \neq 0$, this is a nonzero polynomial in γ of degree at most $4k$. Since γ is sampled uniformly from $K^\times$, its root probability is at most $4k/(|K| - 1)$.

Only a first ordinary witness with $s' = 0$ that violates the strengthened conclusion triggers a repeat. Restore the pre-γ checkpoint and run independent continuations until one returns a kernel or another certified violating zero-pending record. Propagate a kernel immediately. Different canonical source openings of a shared fresh commitment give a short kernel. For identical source values the fixed polynomial root bound controls the repeated-label mass. The original ordinary failure is charged once, giving κ_{zero}.

Expected extraction work The statistical analysis allows the extractor to scan challenge fibers and repeat independent contexts. To obtain an expected-time bound, we account for the cost of these operations over the same unconditional distribution.

On a fixed coordinate-j fiber, any given point is visited as a side query with probability at most $1/M_{ij}$ over the center and scan order. After averaging over other coordinates, the expected visit multiplicity of any complete point, including the center and all s_i scans, is at most $1 + s_i$ times its mass under a uniform product challenge. Since the original path can submit only once, the conditional child budgets sum to at most $(1 + s_i)T_{i+1}$ for a single raw call. The R_C factor applies to pre-submission query work.

Independent repetition contributes at most one further copy of the expected RawNode cost. If a call has expected cost μ, returns an invalid candidate with probability b, and returns any nonempty result with probability t, then $b \leq t$. The expected cost of repeating until a nonempty result is obtained, weighted by the probability of entering that repetition, is $b\mu/t \leq \mu$ for $t > 0$. For $t = 0$ the first call cannot trigger repetition. This calculation allows a trial's own cost and outcome to be correlated. Entry into the trial depends only on previous trials.

Consequently, the expected work at backend layer i satisfies

$$T_i \leq 2\left[C_i^{\mathrm{loc}} + R_C(1 + s_i)H_i + (1 + s_i)T_{i+1}\right]. \tag{9}$$

Here C_i^{loc} covers local sampling, restoration, arithmetic, and public checks, while H_i bounds the work of one immediate short-challenge query before its retry or submission decision. The terminal cost is bounded by T_d.

The ordinary front-end extractor requires at most two backend extraction budgets, and the zero-pending extractor requires at most four, in addition to their local work. For the $k = 1{,}024$ profile, expanding the five-layer recurrence gives a coefficient of 786,240 on T_d.

The statistical losses add across layers. For a growing parameter family, polynomial expected extraction time also requires polynomial bounds on local work, challenge

encodings, the ambient-to-filtered sampling ratios, and the expanded factor $\prod_i 2(1+s_i)$. These expected costs concern a resettable prover under extraction. The honest acceptance probability follows from Proposition 4.1.

Appendix B.6 embeds the short kernels under the shared CRS into six representative Module-SIS problems. Section 6 applies the extraction guarantee to authenticated histories.

6 Authenticated Histories and Settlement

We apply the extraction guarantee to linked transitions and describe how a frozen window can be certified at a single checkpoint.

6.1 From Accepted Transitions to an Authenticated History

A verifier starts from an authenticated accumulator with pending value zero and advances its state only after accepting a transition. Each subsequent transition must use the preceding output's public fields, state commitment, and pending value. This continuity links the accepted transitions to the original anchor. Ordinary transitions may carry nonzero fresh residuals; the stronger guarantee follows when the accepted history returns to zero pending value.

The verifier processes attempts serially and maintains a global counter that cannot be rolled back. Each attempt is charged before α is issued, even if the prover later aborts, times out, or fails verification. This prevents selective aborts from creating uncounted opportunities to obtain favorable challenges.

For example, a fresh source with residuals $(-1, 1, 0, 0)$ at pending value zero contributes $\gamma(\gamma - 1)$ to the pending update. A prover could abandon every attempt with $\gamma \neq 1$ and complete only those for which the update vanishes. Counting completed transitions alone would omit these additional attempts.

Theorem 6.1 (Soundness of authenticated histories).
Consider a serialized verifier that starts from an authenticated zero-pending anchor and uses one of the four five-layer profiles throughout the history. The shared CRS is sampled uniformly, and the input history and prover auxiliary state are generated efficiently. At most H online attempts are issued. The history extractor has expected cost at most T_H and is stopped after $U > 0$ units of work. Its joint failure event F occurs when the selected continuous chain is accepted with zero final pending value, but the truncated extractor does not recover compatible transition witnesses with every fresh source in $\mathcal{R}_{\mathrm{src}}$. Then

$$\Pr[F] \le H\kappa_{\mathrm{zero}} + \frac{T_H}{U} + \sum_{g=1}^{6} \mathrm{Adv}_{\mathrm{MSIS},g}(U + t_g).$$

Here $\mathrm{Adv}_{\mathrm{MSIS},g}(t)$ is the time-t success probability for the g-th representative Module-SIS problem. Its dimensions and radius are determined by the chosen profile. The quantity

t_g is a deterministic bound on the reduction's additional setup, input-generation, and output-conversion work, measured in the same time units.

Proof. The combined ordinary and zero-pending guarantee of Theorem 5.1 holds at every reachable checkpoint. Averaging over the public history preceding each issued attempt and summing over at most H attempts gives a statistical loss of at most $H\kappa_{\mathrm{zero}}$.

On the accepted chain, adjacent extracted openings must agree on their shared state commitment. Otherwise, their difference gives a short state-commitment kernel. Starting from the final zero-pending transition, the strengthened guarantee yields zero pending at the preceding state and strict witnesses for that transition's fresh sources. Applying the same argument backward recovers compatible witnesses for the entire chain.

The extractor uses one clock for all continuations, rewinds, and boundary-opening comparisons. Markov's inequality bounds cutoff failure by T_H/U. The remaining short-kernel events yield the six computational terms through the shared-CRS reductions in Appendix B.6.

For $H = 16{,}384$, the statistical term $H\kappa_{\mathrm{zero}}$ is below 2^{-130} in each of the four profiles.

6.2 Strict Settlement of a Frozen Window

Certification can also be deferred until a checkpoint. Starting from an authenticated anchor, the verifier retains an ordered log of the original source and state commitments, public fields, folding challenges, and pending values. Adjacent transitions share one boundary-state record, while intermediate outputs await settlement.

Before issuing the batch challenges, the verifier freezes the log, the parameter profile, and the auxiliary commitment. The window relation requires every fresh source to satisfy $\mathcal{R}_{\mathrm{src}}$, together with the accumulated-state, folding, and pending-update equations of every recorded transition. The prover compiles these conditions into one square table. The parties run a single sumcheck and certify its affine endpoint against the original commitments through one invocation of the recursive backend. The verifier advances its authenticated anchor only after this certification succeeds. A settlement attempt is charged once its first batch challenge has been issued, even if the prover subsequently aborts.

A direct batch proof for the same strict window relation can also share intermediate state openings and invoke the backend once. Since all fresh residuals must vanish, it can replace each pending-update equation with the public check $s_t = s_{t-1}$. These checks must hold at every boundary: checking only the two endpoint values would also admit a pending trace such as $0, 1, 0$.

Our comparison therefore holds the recorded log, strict source relation, and settlement point fixed. Both one-shot constructions share the reduction in backend invocations relative to certifying the transitions separately, and both defer certification of earlier inputs until the checkpoint.

7 Implementation and Evaluation

7.1 Concrete Profiles

We instantiate the four-lane quadratic relation at $k \in \{16, 64, 256, 1024\}$ and run the complete protocol at each arity. We select each profile before timing from the same family of 18,000 five-layer schedules, minimizing the number of projection coefficients subject to the parameter screens in Appendix C. Each profile has four recursive reductions followed by one pivot, uses 384 projection rows, and allows at most 160 draws at each projection or short-challenge stage.

Table 1. Five-layer profiles and statistical history exponents for $H = 16{,}384$ issued attempts.

k	Blocks	Radix	Packed CRS (B)	History
16	$(4, 4, 4, 4, 5)$	128	2,217,984	131.3725
64	$(6, 4, 4, 4, 5)$	128	4,085,760	131.2412
256	$(8, 6, 4, 4, 5)$	64	11,175,936	131.0099
1,024	$(12, 8, 6, 4, 5)$	64	29,079,552	130.6355

The target is a classical lattice-attack cost of 2^{128}. For all six representatives we require estimated Core-SVP work exponents of at least 144 under classical pricing and 136 under quantum pricing. The audit also checks single-component CRT strategies. Table 1 reports the statistical component of Theorem 6.1. For $k = 1024$ the actual 16×4733 root block matrix determines the packed CRS size of 29,079,552 bytes. The full set of 24 representative instances and cost estimates appears in Table 7.

7.2 End-to-End Ordinary Transitions

Each run starts with one old state and k fresh committed inputs. One fresh source has a nonzero residual, so the verifier must accept an ordinary transition that updates the pending value. Reproducible fixtures supply the inputs and CRS. The verifier draws fresh challenges in message order, including all sumcheck, projection and response challenges. The prover and verifier each write one encoded stream. After the private witness leaves scope, the verifier decodes both streams, checks the challenges against its own retained randomness, and rebuilds the root and all four child statements from public data. Every one of the twelve trials accepted the original and rejected a changed coefficient in the pivot commitment. An independent Python decoder also checked the field recurrence, challenge membership, bounded vectors, padding, retry counts and message lengths.

The shared AMD EPYC 7742 server ran Ubuntu 24.04.4 LTS with a 200 GiB container memory limit and a CPU quota equivalent to 40 logical CPUs. At each arity the three trials ran concurrently, each pinned to a distinct physical core with a separate last-level cache. Table 2 charges the prover for output and auxiliary commitments, encoding, sumcheck, all backend layers, witness consistency checks and serialization. The verifier total includes issuing and encoding fresh challenges and replaying the submitted messages with the public inputs. The separate wall clock includes setup, file I/O and a final negative check. Table 5 gives both public replay and wall time.

Table 2. Complete ordinary transitions. Times and bytes are medians of three runs. RSS is the largest process peak.

Fresh inputs	Prover (s)	Verifier (s)	$P \to V$ (B)	$V \to P$ (B)	Total (B)	RSS (MiB)
16	19.737	18.071	106,775	36,390	143,165	86.65
64	34.227	31.068	110,127	39,119	149,246	167.50
256	83.797	74.946	118,100	41,407	159,507	523.89
1,024	258.324	226.150	126,145	44,216	170,361	2,014.43

All twelve runs submitted on the first draw at each stage. Their actual stream lengths therefore agree with the no-retry budgets. At $k = 1024$, the parties exchange 126,145 and 44,216 bytes, respectively. Median proving and verification take 258.324 and 226.150 seconds. The full run takes 487.709 seconds and the largest observed process peak is 2,014.43 MiB.

7.3 Verification Tests and Component Costs

A separate live fixture exercises two consecutive steps with one and two fresh inputs. The verifier obtains operating-system randomness for the field and backend challenges and requests short ring challenges from the exact filtered sampler. Sixteen sumcheck rounds lead to one recursive reduction and a terminal pivot. The honest window passes. Changing the auxiliary commitment or an intermediate pending value causes the terminal principal check to fail. Committing a source with an incorrect residual fails the field recurrence. Substituting the previous state commitment fails continuity before the next online challenge, and changing the terminal response fails the pivot binding equation.

Table 6 gives four independent microbenchmarks covering canonical encoding, square sumcheck, projection and short-challenge sampling. These measurements use a separate Intel Core i9-13980HX under WSL2. Each time covers the workload listed in its row. The projection benchmark separately times expansion of the seeded matrix and application to the input vector, giving medians of 785.778 ms and 13.205 ms, respectively.

7.4 Settlement on a Matched Log

Two of the eight steps contain 125 fresh sources and the remaining six contain 124 each. All policies start from the same authenticated anchor and retain the same ordered log. The stepwise policy certifies the strict sources after each step. The frozen window and the matched direct batch described in Section 6 perform one certification at the end.

Table 3 gives the resulting front-end counts. Under either one-shot policy the first source waits seven more steps, and the shared table takes 22 sumcheck rounds and one backend invocation. With the optimized $k = 1024$ envelope the one-shot no-retry budget is 168,633 bytes (Appendix C.3). A separate deterministic full-profile window fixture using an earlier backend schedule accepted the 994-source log after rebuilding all five public layers and rejected a modified terminal commitment. The optimized four-arity timings in Table 2 concern ordinary transitions.

Table 3. Settlement policies for the same retained log and strict source relation. The delay is the number of additional steps before the earliest input can be certified. Bytes count field messages and auxiliary tags.

	Stepwise	Window	Direct
Table entries	3,802,896	3,697,714	3,697,714
sumcheck rounds	152	22	22
Backend calls	8	1	1
Maximum delay	0	7	7
Front-end bytes	23,808	3,192	3,192

8 Discussion

Alternative approaches. A direct approach combines the original commitment openings using the folding challenge. The resulting vector has the required linear image, but field addition can introduce carries and change its canonical digits. Re-encoding restores short canonical digits. A proof of canonicality alone then leaves open whether the new vector represents the specified fold of the original commitments. One could prove fold consistency and canonicality separately, provided both arguments are bound to the same opening. Sonata combines their constraints in a single quadratic check and authenticates their common opening through the sumcheck endpoint.

A separate range argument could certify a committed interval, as in Bulletproofs [21]. Lattice arguments also support products of committed values [8]. In either approach a fold verifier would still need an equation tying the certified digits to the input openings. Our endpoint provides this equation. Libra and Spartan use sumcheck for layered circuits and R1CS, respectively [52,49]. Sumcheck arguments can establish opening knowledge for compatible commitments [15], while Greyhound and Hachi authenticate evaluations of lattice-committed polynomials [43,42]. In Sonata the verifier already has the linear description of the table entries in the state and helper openings. It can therefore append the evaluation directly to their commitment equations.

Scope of the security guarantee. The extraction theorem covers a classical rewindable prover that can choose its later responses after seeing verifier challenges. It preserves the first real execution and accounts for both statistical failure and the cost of obtaining an incompatible pair of openings. Theorem 6.1 retains the global timeout and six time-bounded Module-SIS advantages. An estimated Core-SVP work factor specifies the cost of a particular attack model. A concrete bound on adversarial success would also require an advantage curve at the reduction time limits. Likewise, our construction and experiments instantiate the specific four-lane quadratic relation. Extending the claim to arbitrary R1CS would require a new embedding and an audit of its dimensions, degree and norm growth.

The analysis uses an interactive verifier. Applying Fiat–Shamir would require a transcript specification that binds setup, public inputs, each commitment and earlier messages before deriving each challenge, together with a security analysis for that transformation. Omitting public transcript inputs can break knowledge soundness in deployed proof systems [26].

Implementation trade-offs. Increasing arity from 16 to 1,024 raises the encoded exchange from 143,165 to 170,361 bytes, about 19%, while median proving increases from 19.737 to 258.324 seconds and peak RSS increases from 86.65 to 2,014.43 MiB. The four choices all use five backend layers and five pivot blocks, but the root projections grow with the witness dimensions. At $k = 1024$ each full materialization of the root projection occupies $384 \cdot 3{,}634{,}944$ one-byte entries. The component measurement suggests evaluating a streamed projection or a different seed-expansion implementation. Its cost would have to include any repeated expansion when both prover and verifier reconstruct their matrices.

For a sequence of transitions, settlement frequency determines when fresh sources become authenticated. Deferring certification in our eight-step example removes seven backend calls at the expense of seven steps of delay for the earliest input. A direct proof of the same frozen strict relation obtains the same call count. Applications that must act on each fresh input immediately therefore face a different choice from applications that can retain the log until a checkpoint.

Comparison with High-Arity Folding Schemes. Symphony [24] and Sonata both use lattice-based high-arity folding. Symphony builds a SNARK for batch proving generalized R1CS statements. Sonata certifies that a short, canonically re-encoded opening represents the prescribed fold of committed quadratic instances. Our benchmarks therefore measure the cost of proving and verifying this transition on Sonata's native relation.

Declaration of AI Use. The author used GPT-6 Astra to polish selected passages, chiefly in the knowledge soundness section and the appendices. The model suggested a heuristic approach to the bad-slope argument in the knowledge extraction proof and helped review the arguments for rigor. The author designed the code structure and used the model to help complete the implementation. The author reviewed all AI-assisted text and code and takes responsibility for the paper and its implementation.

References

1. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing (STOC 1996). pp. 99–108. ACM (1996). https://doi.org/10.1145/237814.237838, https://eccc.weizmann.ac.il/report/1996/007/
2. Albrecht, M.R., Gheorghiu, V., Postlethwaite, E.W., Schanck, J.M.: Estimating quantum speedups for lattice sieves. In: Advances in Cryptology—ASIACRYPT 2020, Part II. pp. 583–613 (2020), https://eprint.iacr.org/2019/1161
3. Albrecht, M.R., Lai, R.W.F.: Subtractive Sets over Cyclotomic Rings: Limits of Schnorr-like Arguments over Lattices. Cryptology ePrint Archive, Paper 2021/202 (2021), https://eprint.iacr.org/2021/202
4. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. Journal of Mathematical Cryptology **9**(3), 169–203 (2015). https://doi.org/10.1515/jmc-2015-0016, https://doi.org/10.1515/jmc-2015-0016
5. Albrecht, M.R., et al.: Lattice Estimator. Software, revision 53da59825977 (2026), https://github.com/malb/lattice-estimator
6. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange—a new hope. In: 25th USENIX Security Symposium. pp. 327–343 (2016), https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim

7. Attema, T., Klooß, M., Lai, R.W.F., Yatsyna, P.: Adaptive special soundness: A new framework for knowledge extraction with applications to Kilian's protocol. Cryptology ePrint Archive, Paper 2024/2038 (2024), https://eprint.iacr.org/2024/2038
8. Attema, T., Lyubashevsky, V., Seiler, G.: Practical product proofs for lattice commitments. Cryptology ePrint Archive, Paper 2020/517 (2020), https://eprint.iacr.org/2020/517
9. Balbás, D., Nitulescu, A., Plançon, M.: ProtogaLattice: Lattice-based algebraic folding. Cryptology ePrint Archive, Paper 2026/1317 (2026), https://eprint.iacr.org/2026/1317
10. Baum, C., Damgård, I., Lyubashevsky, V., Oechsner, S., Peikert, C.: More efficient commitments from structured lattice assumptions. Cryptology ePrint Archive, Paper 2016/997 (2016), https://eprint.iacr.org/2016/997
11. Becker, A., Ducas, L., Gama, N., Laarhoven, T.: New directions in nearest neighbor searching with applications to lattice sieving. In: Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 10–24 (2016)
12. Beullens, W., Seiler, G.: LaBRADOR: Compact proofs for R1CS from module-SIS. Cryptology ePrint Archive, Paper 2022/1341 (2022), https://eprint.iacr.org/2022/1341
13. Boneh, D., Chen, B.: LatticeFold+: Faster, simpler, shorter lattice-based folding for succinct proof systems. In: Advances in Cryptology—CRYPTO 2025, Part VII. Lecture Notes in Computer Science, vol. 16006, pp. 327–361. Springer (2025). https://doi.org/10.1007/978-3-032-01907-3_11, https://eprint.iacr.org/2025/247
14. Boneh, D., Chen, B.: LatticeFold: A lattice-based folding scheme and its applications to succinct proof systems. In: Advances in Cryptology—ASIACRYPT 2025, Part III. Lecture Notes in Computer Science, vol. 16247, pp. 330–362. Springer (2026). https://doi.org/10.1007/978-981-95-5099-9_11, https://eprint.iacr.org/2024/257
15. Bootle, J., Chiesa, A., Sotiraki, K.: Sumcheck arguments and their applications. Cryptology ePrint Archive, Paper 2021/333 (2021), https://eprint.iacr.org/2021/333
16. Bootle, J., Chiesa, A., Sotiraki, K.: Lattice-based succinct arguments for NP with polylogarithmic-time verification. Cryptology ePrint Archive, Paper 2023/930 (2023), https://eprint.iacr.org/2023/930
17. Bootle, J., Lyubashevsky, V., Nguyen, N.K., Seiler, G.: A non-PCP approach to succinct quantum-safe zero-knowledge. Cryptology ePrint Archive, Paper 2020/737 (2020), https://eprint.iacr.org/2020/737
18. Bootle, J., Lyubashevsky, V., Seiler, G.: Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. Cryptology ePrint Archive, Paper 2019/642 (2019), https://eprint.iacr.org/2019/642
19. Bostan, A., Lecerf, G., Schost, É.: Tellegen's principle into practice. In: Proceedings of ISSAC 2003. pp. 37–44. ACM (2003), https://specfun.inria.fr/bostan/publications/BoLeSc03.pdf
20. Bowe, S., Grigg, J., Hopwood, D.: Recursive proof composition without a trusted setup. Cryptology ePrint Archive, Paper 2019/1021 (2019), https://eprint.iacr.org/2019/1021
21. Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., Maxwell, G.: Bulletproofs: Short proofs for confidential transactions and more. Cryptology ePrint Archive, Paper 2017/1066 (2017), https://eprint.iacr.org/2017/1066
22. Bünz, B., Chen, B.: Protostar: Generic efficient accumulation/folding for special-sound protocols. In: Advances in Cryptology—ASIACRYPT 2023, Part II. Lecture Notes in Computer Science, vol. 14439, pp. 77–110. Springer (2023). https://doi.org/10.1007/978-981-99-8724-5_3, https://eprint.iacr.org/2023/620
23. Bünz, B., Chiesa, A., Mishra, P., Spooner, N.: Recursive proof composition from accumulation schemes. In: Theory of Cryptography—TCC 2020, Part II. Lecture Notes in Computer Science, vol. 12551, pp. 1–18. Springer (2020). https://doi.org/10.1007/978-3-030-64378-2_1, https://doi.org/10.1007/978-3-030-64378-2_1
24. Chen, B.: Symphony: Scalable SNARKs in the random oracle model from lattice-based high-arity folding. Cryptology ePrint Archive, Paper 2025/1905 (2025), https://eprint.iacr.org/2025/1905
25. Chen, Y., Nguyen, P.Q.: BKZ 2.0: Better lattice security estimates. In: Advances in Cryptology—ASIACRYPT 2011. Lecture Notes in Computer Science, vol. 7073, pp. 1–20. Springer (2011), https://www.iacr.org/archive/asiacrypt2011/70730001/70730001.pdf

26. Dao, Q., Miller, J., Wright, O., Grubbs, P.: Weak Fiat–Shamir attacks on modern proof systems. In: IEEE Symposium on Security and Privacy (2023), https://eprint.iacr.org/2023/691
27. Dimitriou, N., Garreta, A., Manzur, I., Vlasov, I.: Mova: Nova folding without committing to error terms. Cryptology ePrint Archive, Paper 2024/1220 (2024), https://eprint.iacr.org/2024/1220
28. Eagen, L., Gabizon, A.: ProtoGalaxy: Efficient ProtoStar-style folding of multiple instances. Cryptology ePrint Archive, Paper 2023/1106 (2023), https://eprint.iacr.org/2023/1106
29. Garreta, A., Lipmaa, H., Luhaäär, U., Osadnik, M.: Cyclo: Lightweight lattice-based folding via partial range checks. In: Advances in Cryptology—EUROCRYPT 2026, Part VII. Lecture Notes in Computer Science, vol. 16547, pp. 33–62. Springer (2026). https://doi.org/10.1007/978-3-032-25336-1_2, https://eprint.iacr.org/2026/359
30. Geng, Z., Plançon, M.: Succinct shortness check under a few kilobytes via tensor train random projections. Cryptology ePrint Archive, Paper 2026/2146 (2026), https://eprint.iacr.org/2026/2146
31. Klooß, M., Lai, R.W.F., Nguyen, N.K., Osadnik, M.: RoK, paper, SISsors – toolkit for lattice-based succinct arguments. Cryptology ePrint Archive, Paper 2024/1972 (2024), https://eprint.iacr.org/2024/1972
32. Klooß, M., Lai, R.W.F., Nguyen, N.K., Osadnik, M.: RoK and roll – verifier-efficient random projection for $\tilde{O}(\lambda)$-size lattice arguments. Cryptology ePrint Archive, Paper 2025/1220 (2025). https://doi.org/10.1007/978-981-95-5099-9_10, https://eprint.iacr.org/2025/1220
33. Kothapalli, A., Parno, B.: Algebraic reductions of knowledge. In: Advances in Cryptology—CRYPTO 2023, Part IV. Lecture Notes in Computer Science, vol. 14084, pp. 669–701. Springer (2023). https://doi.org/10.1007/978-3-031-38551-3_21, https://eprint.iacr.org/2022/009
34. Kothapalli, A., Setty, S.: SuperNova: Proving universal machine executions without universal circuits. Cryptology ePrint Archive, Paper 2022/1758 (2022), https://eprint.iacr.org/2022/1758
35. Kothapalli, A., Setty, S.: HyperNova: Recursive arguments for customizable constraint systems. In: Advances in Cryptology—CRYPTO 2024, Part X. Lecture Notes in Computer Science, vol. 14929, pp. 345–379. Springer (2024). https://doi.org/10.1007/978-3-031-68403-6_11, https://eprint.iacr.org/2023/573
36. Kothapalli, A., Setty, S., Tzialla, I.: Nova: Recursive zero-knowledge arguments from folding schemes. In: Advances in Cryptology—CRYPTO 2022, Part IV. Lecture Notes in Computer Science, vol. 13510, pp. 359–388 (2022). https://doi.org/10.1007/978-3-031-15985-5_13, https://eprint.iacr.org/2021/370
37. Kuriyama, S., Lai, R.W.F., Osadnik, M., Tucci, L.: SALSAA – sumcheck-aided lattice-based succinct arguments and applications. Cryptology ePrint Archive, Paper 2025/2124 (2025), https://eprint.iacr.org/2025/2124
38. Langlois, A., Stehlé, D.: Worst-case to average-case reductions for module lattices. Designs, Codes and Cryptography **75**(3), 565–599 (2015). https://doi.org/10.1007/s10623-014-9938-4, https://eprint.iacr.org/2012/090
39. Lund, C., Fortnow, L., Karloff, H., Nisan, N.: Algebraic methods for interactive proof systems. Journal of the ACM **39**(4), 859–868 (1992). https://doi.org/10.1145/146585.146605
40. Lyubashevsky, V., Seiler, G.: Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In: Advances in Cryptology—EUROCRYPT 2018, Part I. Lecture Notes in Computer Science, vol. 10820, pp. 204–224. Springer (2018). https://doi.org/10.1007/978-3-319-78381-9_8, https://eprint.iacr.org/2017/523
41. MATZOV: Report on the security of LWE: Improved dual lattice attack. Tech. rep., MATZOV (2022). https://doi.org/10.5281/zenodo.6412487, https://zenodo.org/records/6412487
42. Nguyen, N.K., O'Rourke, G., Zhang, J.: Hachi: Efficient lattice-based multilinear polynomial commitments over extension fields. Cryptology ePrint Archive, Paper 2026/156 (2026), https://eprint.iacr.org/2026/156
43. Nguyen, N.K., Seiler, G.: Greyhound: Fast polynomial commitments from lattices. Cryptology ePrint Archive, Paper 2024/1293 (2024). https://doi.org/10.1007/978-3-031-68403-6_8, https://eprint.iacr.org/2024/1293
44. Nguyen, W., Setty, S.: Neo and SuperNeo: Post-quantum folding with pay-per-bit costs over small fields. In: Advances in Cryptology—CRYPTO 2026, Part IX. Lecture Notes in Computer Science,

vol. 16808, pp. 492–523. Springer (2026). https://doi.org/10.1007/978-3-032-35424-2_17, https://eprint.iacr.org/2026/242
45. Nisan, N.: Pseudorandom generators for space-bounded computation. Combinatorica **12**(4), 449–461 (1992). https://doi.org/10.1007/BF01305237
46. Osadnik, M.: PikkuFold: Efficient folding in a few kilobytes. Cryptology ePrint Archive, Paper 2026/1809 (2026), https://eprint.iacr.org/2026/1809
47. Schnorr, C.P.: Lattice reduction by random sampling and birthday methods. In: STACS 2003. Lecture Notes in Computer Science, vol. 2607, pp. 145–156. Springer (2003). https://doi.org/10.1007/3-540-36494-3_14
48. Schwartz, J.T.: Fast probabilistic algorithms for verification of polynomial identities. Journal of the ACM **27**(4), 701–717 (1980). https://doi.org/10.1145/322217.322225
49. Setty, S.: Spartan: Efficient and general-purpose zkSNARKs without trusted setup. Cryptology ePrint Archive, Paper 2019/550 (2019), https://eprint.iacr.org/2019/550
50. Setty, S., Thaler, J., Wahby, R.: Customizable constraint systems for succinct arguments. Cryptology ePrint Archive, Paper 2023/552 (2023), https://eprint.iacr.org/2023/552
51. Wang, X., Fu, S.: Compare Before Clearing: Exact Integral-Comparison Frontiers for Lattice Extraction. Cryptology ePrint Archive, Paper 2026/1962 (2026), https://eprint.iacr.org/2026/1962
52. Xie, T., Zhang, J., Zhang, Y., Papamanthou, C., Song, D.: Libra: Succinct zero-knowledge proofs with optimal prover computation. Cryptology ePrint Archive, Paper 2019/317 (2019), https://eprint.iacr.org/2019/317
53. Zhao, J., Setty, S., Cui, W., Zaverucha, G.: MicroNova: Folding-based arguments with efficient (on-chain) verification. In: IEEE Symposium on Security and Privacy. pp. 1964–1982 (2025). https://doi.org/10.1109/SP61157.2025.00168, https://eprint.iacr.org/2024/2099
54. Zippel, R.: Probabilistic algorithms for sparse polynomials. In: Ng, E.W. (ed.) Symbolic and Algebraic Computation—EUROSAM 1979. Lecture Notes in Computer Science, vol. 72, pp. 216–226. Springer (1979). https://doi.org/10.1007/3-540-09519-5_73

Appendix A Algebraic Correctness

A.1 Canonicality over the Integers

Proof of Lemma 4.1. Consider the encoding penalty $E(x) = 8B(x)+r(x)(r(x)-1)+F(x)$. Rounding each integer coordinate x_i to $u_i = 0$ when $x_i \le 0$, and to $u_i = 1$ otherwise, gives a binary vector u. The coordinate penalty $p_i = x_i(x_i-1)$ is nonnegative and bounds both $2|x_i - u_i|$ and $(x_i - u_i)^2$. Expanding a mixed term therefore gives $|x_i x_j - u_i u_j| \le p_i + p_j$.

These inequalities control the change from $F(x)$ to $F(u)$. For each coordinate, the sum of its incident absolute quadratic coefficients, together with half its absolute linear coefficient, is at most $15/2$. Thus $|F(x) - F(u)| \le (15/2)B(x)$, and

$$E(x) \ge \frac{1}{2}B(x) + r(x)(r(x)-1) + F(u) \ge 0.$$

The last inequality uses two facts. First, $r(x)$ is an integer, so $r(x)(r(x)-1) \ge 0$. Second, $F(u)$ is nonnegative on binary inputs. Its first four terms enforce $m = eb_2$, and its remaining terms are products of nonnegative bits. Consequently, $E(x) = 0$ forces every coordinate to be binary, $r(x) \in \{0, 1\}$, and $F(x) = 0$.

It remains to identify the value bits admitted by these conditions. The quantity $s = 42 - \sum_{i=6}^{47} b_i$ counts the zeros among the upper 42 bits. When $s = 0$, the equation $r = s - 1 + e - 2t \in \{0, 1\}$ forces $e = 1$ and $t = 0$. The condition $F(x) = 0$ then implies $m = b_2$, $b_3 = b_4 = b_5 = 0$, and $b_2(b_0 + b_1) = 0$. Hence the lower six bits encode an integer

between zero and four. Together with the upper bits, these are precisely the admissible values at the upper end of the canonical range.

When $s \geq 1$, at least one upper bit is zero, so the encoded value is already below q. Every such encoding admits $e = m = 0$ and $t = \lfloor (s-1)/2 \rfloor$, which makes r either zero or one. The helper weights $1, 2, 4, 6, 7$ represent every integer from zero through 20, covering all possible values of t. Conversely, when $s = 0$, each admissible lower-bit pattern has a valid assignment with $e = 1$, all h-bits zero, and $m = b_2$. This proves that the zero set of E projects exactly onto the canonical encodings.

Protocol 1 builds its honest root opening with these assignments. For $s \geq 1$, choose $e = m = 0$. Each selected count helper contributes at least one to t, so the group contains at most $42 - s + 6 + t \leq 47$ nonzero bits. For $s = 0$, the lower bits and m contribute at most two, giving at most 45 bits in total. The state opening of Definition 3.1 fixes the 48 value bits of each component. Several helper assignments may certify those bits. Definition 3.3 binds the particular helpers used in the transition through the auxiliary commitment.

To bound the penalty on an arbitrary integer opening, we use $B(x) \leq 2\|x\|_2^2$ and $|F(x)| \leq 6\|x\|_2^2$. The affine expression $r(x) = 41 + a^\top x$ has $\|a\|_2^2 = 467$. Expanding $r(r-1)$ and bounding its linear term gives $r(x)(r(x)-1) \leq 934\|x\|_2^2 + 3281$. Summing over the V encoding groups therefore yields

$$0 \leq E_{\mathrm{enc}}(W) \leq 956\|\operatorname{cen}(W)\|_2^2 + 3281V.$$

The honest squared-norm budget is $S_0 = 47V + 1$, including the fixed one. For an opening recovered by the extractor, Definition 3.4 supplies the larger budget $B_0^2 = (38400/4121)S_0$. Substituting this latter bound into the penalty gives an expression increasing in $V = 544 + 64k$. Its largest value among the four profiles occurs at $k = 1{,}024$ and rounds up to $27{,}883{,}328{,}406 < q$. Hence a zero penalty modulo q implies zero integer penalty throughout the admitted extraction range.

A.2 Constructing the Square Representation

The integer argument above characterizes the valid encodings. To check the same penalty through sumcheck, we now express it as a shifted sum of affine squares over K.

In the coordinate order $x = (b_0, \ldots, b_{47}, h_0, \ldots, h_4, e, m)$, the range expression is $r(x) = 41 + a^\top x$, where $a = (0^{[6]}, (-1)^{[42]}, -2, -4, -8, -12, -14, 1, 0)$. Here the superscript $[n]$ denotes n repetitions. The symmetric matrix J records the mixed terms of F: a term $c_{ij}x_ix_j$, with $i < j$, contributes $c_{ij}/2$ to both J_{ij} and J_{ji}, and all other entries are zero. Expanding the penalty gives $E(x) = x^\top Mx + \lambda^\top x + c$, with $M = 8I + aa^\top + J$, $\lambda = 81a - 81\mathbf{1} + 3e_{54}$, and $c = 1640$.

We factor $M = LDL^\top$ over the rationals, using a unit lower-triangular L and eliminating coordinates in their given order. The entries follow the usual recurrences $D_j = M_{jj} - \sum_{t<j} L_{jt}^2 D_t$ and $L_{ij} = (M_{ij} - \sum_{t<j} L_{it}D_tL_{jt})/D_j$. The first six pivots are 8, while the high-bit pivots are $D_j = 8(j+3)/(j+2)$ for $6 \leq j \leq 47$. The final seven pivots, in helper-coordinate order, are 216/25, 280/27, 536/35, 1112/67, 1896/139, 14963/1896,

and 3687425/478816. Their numerators and denominators are nonzero modulo q, so the same factorization is valid after reduction to $\mathbb{F}_q$.

The diagonal entries have square roots in K. We fix the square-root convention by choosing the smaller nonnegative root whenever a nonzero square lies in $\mathbb{F}_q$. If D_j is a base-field square, this gives $\sigma_j = \mathrm{root}(D_j)$. Otherwise, $D_j/2$ is a square because $q \equiv 5 \pmod 8$, and we use $\sigma_j = \mathrm{root}(D_j/2)(U^2 - 2)$. The defining polynomial of K gives $(U^2 - 2)^2 = 2$, so $\sigma_j^2 = D_j$ in either case.

To match the linear term, we solve $Lt = \lambda/2$ and set $\beta = D^{-1}t$. The affine forms are then $\psi_j(x) = \sigma_j(\beta_j + \sum_{i \geq j} L_{ij} x_i)$. Expanding their squared sum recovers the quadratic and linear terms of E, since $LDL^\top = M$ and $2LD\beta = \lambda$. The constant difference is $\kappa_{\mathrm{enc}} = \sum_j t_j \beta_j - c = 204388862566065 \pmod q$, giving $\sum_j \psi_j(x)^2 = E(x) + \kappa_{\mathrm{enc}}$. This is a polynomial identity over K, so it also holds for nonbinary coordinates.

The script `verify_ldl.py` supplied with the manuscript checks the rational pivots, the coefficient identities and the reduced constants against the implementation tables.

A.3 Semantic Entries and the Terminal Evaluation

The same square representation accommodates the semantic constraints. A weighted product ωab is represented by the two affine entries $(\omega a + b)/2$ and $\iota(\omega a - b)/2$, where $\iota^2 = -1$. Their squared sum is ωab. Taking one factor to be the constant one handles the remaining affine term.

The old state, new state, and k fresh sources contribute $32(k + 2)$ quadratic products. Their representations use $64(k + 2)$ table entries, and the final affine term uses two more. Together with the $55V$ encoding entries, this gives $T(k) = 55V + 64(k + 2) + 2$. Each encoding group contributes the same public shift κ_{enc}, so the initial claim is $S = V\kappa_{\mathrm{enc}} + \eta(s' - s)$. Summing their squares gives

$$\sum_{j=0}^{T(k)-1} \mathcal{A}_j(W)^2 - S = E_{\mathrm{enc}}(W) + \eta S_\rho(W).$$

The identity holds for arbitrary W, since each entry was obtained by an algebraic expansion. Protocol 1 uses powers $\rho^1, \ldots, \rho^8$ on the old and output state equations and $\rho^9, \ldots, \rho^{72}$ on the 64 private fold coordinates. The fresh residuals use $\gamma^1, \ldots, \gamma^{4k}$. Thus subtracting S retains any discrepancy between their weighted sum and the advertised pending increment. This is the ordinary transition predicate in Definition 3.3, including when a fresh residual is nonzero.

To authenticate the terminal evaluation, write the padded table as $\mathcal{A}(W) = a_{\mathrm{tab}} + H_{\mathrm{tab}} W$. At the final sumcheck point r, the Boolean equality weights $\epsilon_j = \mathrm{eq}(r, \mathrm{bin}(j))$ give $\widetilde{\mathcal{A}}(r) = o_r + \langle h_r, W \rangle_K$, with $o_r = \epsilon^\top a_{\mathrm{tab}}$ and $h_r = \epsilon^\top H_{\mathrm{tab}}$. Zero-padding contributes nothing to either expression.

The factorization in Appendix A.2 also gives an efficient way to compute this row. Within one encoding group, the public offset is $\sum_j \epsilon_j \sigma_j \beta_j$, and coordinate x_i receives coefficient $\sum_{j \leq i} \epsilon_j \sigma_j L_{ij}$. For the high-bit columns, $L_{ij} = -a_i/(j + 3)$ when $i > j$ and $6 \leq j \leq 47$. Their contributions can therefore be accumulated through prefix sums. A

high value bit receives its diagonal contribution and the preceding prefix. The five count helpers receive the complete prefix scaled by $2(1, 2, 4, 6, 7)$, and the switch coordinate receives its negative. The remaining entries follow directly from the factorization.

For a semantic product pair with equality weights ϵ_+ and ϵ_-, the coefficients on ωa and b are $(\epsilon_+ + \iota\epsilon_-)/2$ and $(\epsilon_+ - \iota\epsilon_-)/2$, respectively. Thus the optimized calculation produces the same affine row as evaluating the full table.

At the handoff in Protocol 1, each state or source commitment contributes $A_{\mathrm{st}} I_g W = C_g - A_{\mathrm{st}} f_g$. For slot a, power-basis component t, and bit j, the physical coefficient index is $192a + 48t + j$. The five public slots contribute fixed bits through f_g. Fresh objects use only base-field components in the 64 private value slots. The other three components and the four reserved error slots are fixed to zero, as required by Definition 3.2. For active group u, the auxiliary insertion places $W_{55u+48+j}$ at physical coefficient $7u + j$ for $0 \le j < 7$.

The verifier stacks the commitment rows for the old state, all fresh sources, the output state, and the auxiliary tag, in this order. Next come the four power-basis coordinates of $\langle h_r, W \rangle_K = \nu - o_r$, then $W_{55V} = 1$. This gives $256(k + 3) + 5$ scalar rows on $55V + 1$ active coefficients. Each alignment slot contributes an identity row with target zero. The resulting (L_0, Y_0, B_0) is the public input to the backend. Its witness in Definition 3.4 is a single vector satisfying every row. All formulas for L_0 and its public target apply to arbitrary coefficient vectors.

A.4 Coefficient Adjoints and Child Linearization

The root and child systems are evaluated through their linear maps and adjoints. The transposed map pulls a scalar test back to the opening coordinates [19]. Negacyclic multiplication determines the coefficient convention. For $a \in R_q$, the involution $a^\dagger = a(X^{-1})$ satisfies $\mathrm{ct}(a^\dagger x) = \sum_{j=0}^{63} a_j x_j$. A scalar coefficient-weight vector is therefore packed into a ring element and transformed by $\dagger$ before it is used in a constant-coefficient pairing. Under this convention, a functional is pulled through a ring matrix by its ordinary transpose.

At the root, the commitment adjoint is applied first, and the resulting coefficient weights are mapped back through $I^\top$ to the private opening coordinates. The endpoint contributes a weighted combination of its four base-field rows. The fixed-one row places its weight on the designated constant coordinate, while each alignment row selects one trailing coefficient. These contributions produce the functional on the opening. The corresponding public target is aggregated separately.

After the unique nonterminal submission in Protocol 2, Algorithm 1 defines the child map on $(z_{\mathrm{lo}}, z_{\mathrm{hi}}, d_t, d_h)$. Radix recomposition gives $z = z_{\mathrm{lo}} + d_i z_{\mathrm{hi}}$ and the auxiliary values t, h. These formal expressions are linear even for arbitrary, noncanonical digit arrays. Before padding, the two six-row tag equations, the 16-row block-image equation and the four principal equations contribute $64(6 + 6 + 16 + 4)$ scalar rows. The four scalar diagonal sums give a total of 2,052. The public targets in this order are the coefficients of u_1, u_2, zeros in the block-image and principal rows, and $b_0, \ldots, b_3$.

To compute the adjoint, assign ring weights ξ_t, ξ_h to the two tag equations, ring weights μ_r to the block-image equation, ring weights ν_e to the four principal equations, and scalar weights τ_e to the diagonal sums. These ring weights are already expressed in the constant-coefficient convention. With $\Phi_e(c) = \sum_j c_j \phi_{e,j}$ as in Protocol 2, the row acting on the reconstructed response is $g = A_i^\top \mu + \sum_e \nu_e \Phi_e(c)$. Its contributions to the low and high response parts are consequently g and $d_i g$.

The digit rows begin with the commitment contributions $M_{t,i}^\top \xi_t$ and $M_{h,i}^\top \xi_h$. Digit level ℓ of block-image entry (j, r) receives the additional coefficient $-d_i^\ell c_j \mu_r$. For an upper-triangular entry $h_{ab}^{(e)}$, with $a \le b$, the additional coefficient is

$$d_i^\ell \left[-\nu_e (2 - \mathbf{1}_{a=b}) c_a c_b + \mathbf{1}_{a=b} \tau_e \right].$$

The factor two accounts for both occurrences of an off-diagonal entry in the symmetric product sum. The diagonal coefficient embeds τ_e as a constant ring element. Algorithm 1 appends a zero-target identity row for each padding coordinate. Its transpose puts the corresponding test weight on that coordinate. Thus for any scalar test vector λ and any opening x of the prescribed shape, the computed row $L^*\lambda$ satisfies

$$\sum_t \lambda_t L(x)_t = \sum_j \mathrm{ct}\left((L^*\lambda)_j x_j \right).$$

This identity holds independently of bitness, norm, and satisfaction of the child equations. Protocols 2 and 3 use it when aggregating a reached system together with its projection rows.

A.5 Honest Norm Propagation and Retry Exhaustion

Proof of Proposition 4.1. Fix the CRS, an old state satisfying Definition 3.1, and fresh inputs of the prescribed types and openings in Definition 3.2. For an ordinary transition, their residuals may be nonzero. Protocol 1 computes the new state errors from the folded values and sets $s' = s + \sum_m \gamma^m r_m$ using the issued γ. This satisfies the pending update of Definition 3.3 for every residual vector. The auxiliary assignments in Appendix A.1 give an honest root opening of squared norm at most $S_0 = 47V + 1$, including the fixed one. The identities in Appendices A.2 and A.3 hold at every sumcheck point, so all field checks pass.

At each reached layer, the completeness invariant is $L_i(W_i) = Y_i$ and $\| \mathrm{cen}(W_i) \|_2^2 \le S_i$. The larger extraction budget $B_i^2 = (38400/4121) S_i$ is used when interpreting a recovered witness for Definition 3.4. Under the honest invariant, each independent ternary projection coefficient has mean zero and variance $1/2$, as it is the difference of two independent unbiased bits. Thus a row has expected squared inner product equal to half the squared opening norm. Centering after reduction modulo q can only decrease that norm. Across 384 rows, the expected squared image norm is at most $192 S_i$, and Markov's inequality bounds rejection at threshold $384 S_i$ by $1/2$.

The space-bounded-generator analysis [45] in Appendix B shows that the seeded projection changes the acceptance probability of this fixed-input test by at most δ_i.

The honest opening remains fixed across retries, and each retry receives an independent complete seed. The probability of exhausting all R_P attempts is consequently at most $(1/2 + \delta_i)^{R_P}$.

For the short-challenge stage, the argument takes place over the integer ring before modular reduction. Both challenge distributions—including the nonzero restriction at the pivot—are invariant under negation and the automorphisms $X \mapsto X^a$ for odd a modulo 128. Negation gives $\mathbb{E}[c_j] = 0$. Galois invariance makes $\mathbb{E}[c_j^\dagger c_j]$ rational, with constant coefficient $\mathbb{E}\|c_j\|_2^2 \le 128$. Independence then removes the cross-block terms and gives

$$\mathbb{E}\Big\| \sum_j c_j \operatorname{cen}(w_j) \Big\|_2^2 = \sum_j \mathbb{E}\|c_j\|_2^2 \, \| \operatorname{cen}(w_j)\|_2^2 \le 128 S_i.$$

Reduction and centering preserve this upper bound. At nonterminal layers, the response threshold is $G_i = 256 S_i - 1$. At the pivot, it is $G_4 = 256 S_4$. Since squared coefficient norms are integers, rejection in either case requires energy at least $256 S_i$. Each draw therefore fails with probability at most $1/2$. The auxiliary tag or terminal triangle remains fixed during these retries, giving an exhaustion probability of at most 2^{-R_C}.

We next bound the child opening after a successful nonterminal submission. For the positive even radix d_i, a centered integer coefficient x is split using $u^+ = \lfloor (|x| + d_i/2 - 1)/d_i \rfloor$ and $\ell = |x| - d_i u^+$. Restoring the sign of x to both parts gives an integer decomposition with low coefficient bounded in magnitude by $d_i/2$. The response retains this low part and the entire high quotient, while the auxiliary arrays are decomposed recursively into digits.

Writing $\Lambda_i = 64 n_i (d_i/2)^2$, the low response block has squared norm at most Λ_i. The identity $d_i z_{\mathrm{hi}} = \operatorname{cen}(z) - z_{\mathrm{lo}}$ then bounds the high part by $(\sqrt{G_i} + \sqrt{\Lambda_i})^2 / d_i^2$. The child also contains digit decompositions of $16 s_i$ ring block-image entries and $2 s_i (s_i + 1)$ ring auxiliary-product entries. If $F_{\mathrm{dig}}(d_i)$ denotes the largest squared digit norm of a centered base-field coefficient, the resulting budget is

$$\begin{aligned} S_{i+1} = \Lambda_i + \left\lceil \frac{(\sqrt{G_i} + \sqrt{\Lambda_i})^2}{d_i^2} \right\rceil \\ + 64\big(16 s_i + 2 s_i (s_i + 1)\big) F_{\mathrm{dig}}(d_i). \end{aligned}$$

After a nonterminal submission in Protocol 2, Algorithm 1 evaluates its rows on these digits. The tag equations hold by construction. Linearity gives $A_i z = \sum_j c_j t_j$. Expanding $\langle \Phi_e(c), z \rangle_R$ and grouping (a, b) with (b, a) gives the symmetric product $H_e(c)$. The four diagonal sums are the four coordinates of the honest aggregate. Zero padding satisfies the remaining identity rows and adds no energy. Thus the submitted child satisfies both parts of the honest invariant at layer $i + 1$.

At the pivot, Protocol 3 receives the other $s_i - 1$ block images and the tag T_0 before issuing its projection seed. For an accepted short draw the response satisfies the same block-image equation. The unit c_0 therefore gives $c_0^{-1}(A_i z - \sum_{j>0} c_j t_j) = t_0$, so recomputing the deterministic base-eight digits (16 per coefficient) reproduces T_0. For each lane e, the missing constant $\operatorname{ct}(h_{00}^{(e)})$ is $b_e - \sum_{j>0} \operatorname{ct}(h_{jj}^{(e)})$. Restoring the four

constants recovers the honest triangles and all four principal identities hold. Once the pivot response is submitted, these algebraic checks introduce no further honest failure.

All these bounds hold conditional on the honest state at each reached layer, including earlier retries and challenges. Each submitted child is certified once. Summing the two retry-exhaustion probabilities over the five layers gives Proposition 4.1. The certified projection-transfer exponents are at least 430, 436, 445, and 453 for the four profiles, so every relevant $\delta_i \le 2^{-430}$. With $R_P = R_C = 160$, the bound $(1/2 + \delta_i)^{160} \le 2^{-160}/(1 - 320\delta_i)$ gives $\varepsilon_{\mathrm{comp}} < 11 \cdot 2^{-160}$. For a zero-pending decision starting from $s = 0$ with strict fresh sources, every $r_m = 0$, hence $s' = 0$ and the additional public check in Protocol 1 passes with certainty.

Appendix B Knowledge-Soundness Proofs

We prove the extraction guarantee for Protocol 1 in Theorem 5.1. We first analyze Algorithm 2 at one backend layer, then account for the two front-end comparisons and the history reduction of Theorem 6.1. Norms of modular ring vectors use their centered coefficient lifts. Products of stored integer lifts used to exhibit a kernel are taken in R before reduction modulo q.

B.1 Binding a Submitted Response

Section 5.2 uses invertible differences of short challenges in R_q. Short units in partially splitting cyclotomics and subtractive sets provide related challenge constructions [40,3]. The following calculation establishes the exact property needed at Sonata's modulus.

Lemma B.1 (Unit differences). *Any two distinct elements of the short-challenge support differ by a unit in R_q. Every nonzero element of the support is also a unit.*

Proof. Consider a nonzero difference Δ between two supported challenge values. In the integral cyclotomic ring $R = \mathbb{Z}[X]/(X^{64} + 1)$, multiplication by Δ has a nonzero integer determinant because R embeds into the cyclotomic field of order 128. The multiplicative order of q modulo 128 is 32. Thus $X^{64} + 1$ factors into two irreducible polynomials of degree 32 over $\mathbb{F}_q$. If Δ were not invertible in R_q, its multiplication matrix would have nullity at least 32 modulo q. Its integer determinant would consequently be divisible by q^{32}. Every supported challenge has multiplication operator norm at most $\Gamma = 26$, so the difference has operator norm at most 2Γ. Its multiplication determinant therefore satisfies $|\det(\Delta)| \le (2\Gamma)^{64} < q^{32}$, a contradiction. Since the full challenge support contains zero, comparison with zero also proves that every nonzero supported element is a unit. □

Lemma B.1 justifies both the divided differences in Algorithm 2 and the recovery of t_0 in Protocol 3. The norm estimate in (6) uses the stored integer numerators.

Lemma B.2 (Certified records and their conflicts). *At a fixed pre-short-challenge checkpoint, every certified ordinary record determines a response z, its block-image equation, the four principal equations, and the four diagonal-sum equations. Two certified*

records either open their shared auxiliary tags identically or yield a nonzero short kernel under a tag-commitment matrix. If their tag openings agree and only challenge coordinate j changes, their response difference satisfies $A_i X_j = \Delta_j t_j \pmod q$.

Proof. At a nonterminal layer, the response submission in Protocol 2 fixes the challenge c and the public child constructed by Algorithm 1. An ordinary result from that child's extractor satisfies Definition 3.4 for this submitted instance. The child identity rows force its alignment tail to zero. The remaining coordinates are $(z_{\mathrm{lo}}, z_{\mathrm{hi}}, d_t, d_h)$. Radix recomposition gives $z = z_{\mathrm{lo}} + d_i z_{\mathrm{hi}}$, the block images t and the symmetric entries h. The two tag rows, block-image rows, four ring principal equations and four scalar diagonal sums in Algorithm 1 hold on this opening. In particular, $A_i z = \sum_j c_j t_j$. These are statements about the values obtained by recomposition and hold for adversarial digits as well. If the child opening satisfies $\|\operatorname{cen}(W_{i+1})\|_2 \le B_{i+1}$, the reconstructed response obeys

$$\|\operatorname{cen}(z)\|_2 \le \sqrt{1 + d_i^2}\, B_{i+1} =: Z_i.$$

If two records open a shared tag with different digit vectors modulo q, subtract their respective centered lifts. The two openings have the same image under $M_{t,i}$ or $M_{h,i}$, so the difference is a kernel nonzero modulo q and has integer norm at most $2B_{i+1}$. This comparison uses the digit vectors themselves even when they recompose to the same value.

At the pivot, Protocol 3 transmits z and checks its norm, so $Z_i = \sqrt{G_i}$. The common pre-projection message is $(T_0, (t_j)_{j>0})$. The nonzero c_0 is a unit by Lemma B.1, so the block-image equation uniquely recovers t_0 from z. Distinct canonical base-eight digit vectors opening T_0 give a nonzero kernel of norm at most $8\sqrt{256 \cdot 64} = 1024$. Before c is sampled, the four targets b_e and the transmitted $\widehat{h}$ already determine every omitted constant $\operatorname{ct}(h_{00}^{(e)}) = b_e - \sum_{j>0} \operatorname{ct}(h_{jj}^{(e)})$. Thus every record at this short-challenge checkpoint uses the same completed upper triangles. For two compatible records, the reconstructed block images agree. Subtracting their block-image equations when only c_j changes gives $A_i X_j = \Delta_j t_j \pmod q$, as required. □

In the coordinate scan of Algorithm 2, a fork restores the pre-c state reached after all earlier retries on that path. Both tags (or the pivot data and $\widehat{h}$) are shared there. For the independent repeat the extractor instead restores the pre-projection state. Only the first-stage data are shared. Once its openings agree, the block images t_j agree across contexts, even if the two u_2 tags or terminal upper triangles differ. Distinct block candidates then yield the kernel in (6).

For a candidate $W = (W_0, \ldots, W_{s_i - 1})$ that satisfies the original linear equations and norm bound, the extractor makes one additional check against its original response. The centered difference $a = \operatorname{cen}(z) - \sum_j c_j \operatorname{cen}(W_j)$ satisfies $A_i a = 0 \pmod q$ and

$$\|a\|_2 \le Z_i + \Gamma \sqrt{s_i}\, B_i.$$

The profiles bound this quantity by $8\Gamma Z_i$. If $a \not\equiv 0 \bmod q$, the extractor returns this short kernel. Otherwise, W explains the center from the first forward execution modulo q. This is the anchor check in Algorithm 2. Its center uses the original submitted child's certified opening at a nonterminal layer and the original transmitted z at the pivot.

B.2 Seeded Projection for a Fixed Candidate

Modular norm projections are also used in lattice reductions of knowledge [32]. Here the verifier reveals a compact seed. We compare the resulting test for a fixed candidate with the test under independent ternary rows. Fix an arbitrary candidate label w at the pre-projection checkpoint in Algorithm 2. This fixes the test to analyze. The adversary may still adapt its image and the candidate that it eventually returns to every challenge it sees.

Lemma B.3 (Fixed-test seed transfer). *Consider a deterministic machine with at most n_{st} states that reads a sequence of b-bit blocks. Under the binary-tree generator of height h used by Sonata, the difference between its acceptance probabilities on a uniformly seeded stream and on 2^h independent uniform blocks is at most $(2^h-1)n_{\mathrm{st}}2^{-b/2}$. The machine and its tested input are fixed before sampling the seed.*

Proof. The Nisan-style generator [45] is defined recursively by $G_0(x) = x$ and $G_j(x) = G_{j-1}(x) \,\|\, G_{j-1}(H_j(x))$, where the H_j are independently sampled from a pairwise-independent hash family over b-bit strings. The lower-level hashes are shared by the two subtrees. For each level take $H(x)_i = v_i + \sum_{r=0}^{b-1} a_{i+r}x_r$ over $\mathbb{F}_2$, with independent uniform strings a of length $2b-1$ and v of length b. For a fixed nonzero input difference, use its last nonzero coordinate as a pivot. Each successive output equation introduces a fresh bit of a, giving a uniform output difference. The independent shift v then makes the two outputs pairwise independent.

Fix all lower-level hashes. Each input x induces a deterministic state transition F_x over the height-$(j-1)$ subtree. Let T be its expected matrix for a uniform x. For the fresh hash H at level j, the two-subtree transition matrix has entry

$$T_H[u,v] = 2^{-b}\sum_x \mathbf{1}\{F_{H(x)}(F_x(u)) = v\}.$$

Uniformity gives $\mathbb{E}_H T_H = T^2$. For distinct x the indicators at a fixed entry are pairwise independent. For each x exactly one destination occurs, so summing the variances of the averaged entries over v gives $\sum_v \mathrm{Var}_H(T_H[u,v]) \le 2^{-b}$ for each u. Cauchy–Schwarz and Jensen now yield $\mathbb{E}_H\|T_H - T^2\|_\infty \le \sqrt{n_{\mathrm{st}}\sum_{u,v}\mathrm{Var}_H(T_H[u,v])} \le n_{\mathrm{st}}2^{-b/2}$.

For row-stochastic matrices, the maximum row-sum norm is submultiplicative. If δ_j denotes the expected difference at height j, replacing the two lower subtrees by their ideal transitions contributes at most $2\delta_{j-1}$. Consequently, $\delta_j \le n_{\mathrm{st}}2^{-b/2} + 2\delta_{j-1}$, with $\delta_0 = 0$. Solving this recurrence gives the stated bound. □

For Sonata, the machine computes $P_i w \bmod q$ for a fixed opening w and checks whether its centered squared norm is at most $384S_i$. The projection uses 384 rows and $768N_i$ input bits, with each pair of independent bits producing one ternary coefficient. The machine records the stream position, an unfinished bit pair, the current row sum modulo q, and the accumulated squared norm saturated at $384S_i + 1$. It has at most

$$n_{\mathrm{st},i} \le (768N_i + 1)\, 4q\, (384S_i + 2)$$

states.

Now consider a candidate with $\|\operatorname{cen}(w)\|_2 > B_i$. The modular ternary tail bound of Osadnik [46, Thm. 2 and Table 1] applies when $B_i \le q/176$. It bounds the probability that independent rows give squared image norm at most $41.21B_i^2 = 384S_i$ by 2^{-192}. The parameter audit checks this condition at every layer. Combining this tail bound with Lemma B.3 gives the fixed-candidate projection bound

$$e_i = 2^{-192} + \delta_i^{\mathrm{seed}}, \qquad \delta_i^{\mathrm{seed}} \le (2^{h_i} - 1)n_{\mathrm{st},i}2^{-b_i^{\mathrm{seed}}/2}.$$

The prover sees the seed before it submits p in Protocols 2 and 3. If a fixed long w satisfies $P_i w = p$ and the submitted image passes its norm check, that issued seed must pass the fixed-w test. Conditioning on the state before each fresh seed and summing over at most R_P draws charges at most $R_P e_i$, including seeds after earlier retries.

B.3 From a False Aggregate to a Defective Diagonal

The projection bound handles candidates that exceed the permitted norm. A candidate satisfying the norm bound may still violate the original linear system. We show that such a candidate either passes an incorrect aggregation check or creates a defective diagonal that is detected by the labelled fiber analysis.

Lemma B.4 (Fixed bad-candidate mass). *Fix an invalid candidate w at a pre-projection checkpoint. If m is the unconditional probability that RawNode encounters a missing fork, and r_w is the probability that it returns this candidate, then*

$$m + r_w \le R_P e_i + \frac{D_i}{|K|} + 2R_C \sum_{j=0}^{s_i-1} \frac{1}{M_{ij}} = \varepsilon_i. \tag{10}$$

Proof. Consider a reached projection submission p. The discrepancy between the original linear system and the candidate opening is $L_i w - Y_i$. The projection contributes the additional discrepancy $P_i w - p$. Padding these base-field coordinates to a multiple of four, grouping them as extension-field coefficients, and using the groups as coefficients of successive powers of T gives a polynomial $E_w(T)$ of degree at most D_i. If E_w is identically zero, all original and projection equations hold for w. Since the candidate is invalid, its centered norm must exceed B_i. It can then be certified only if a fresh projection seed passes the norm test for this fixed long opening. Summing over at most R_P issued seeds gives probability at most $R_P e_i$.

Otherwise E_w is a nonzero polynomial fixed before ζ. In either backend protocol, submitting the single p closes the projection stage. Only one ζ is drawn, so the probability that $E_w(\zeta) = 0$ is at most $D_i/|K|$ [48,54].

Outside these two events, the aggregated discrepancy is nonzero. By the coefficient-adjoint identity of Appendix A.4, at least one of its four base-field coordinates is a nonzero difference between the candidate's aggregated linear evaluation and the public

target. Lemma B.2 gives the diagonal sums for the ordinary center's certified auxiliary products. Thus, for some lane e and block j,

$$\mathrm{ct}\Big(\langle \phi_{e,j}, w_j \rangle_R - h_{jj}^{(e)}\Big) \neq 0.$$

For each such center choose one defective pair (e, j) by a fixed order. Lemma B.2 gives a fixed auxiliary opening at that center. A neighbor that changes its opening of a shared tag yields a kernel. For a compatible neighbor returning the fixed slope w_j, the principal equation becomes the quadratic in Lemma 5.1 with nonzero leading coefficient. If RawNode returns w, this coordinate therefore realizes $B_j(w)$. An exhausted scan at any coordinate realizes its D_j.

We apply Lemma 5.1 to each reached pre-c checkpoint before conditioning on the success of earlier coordinate scans. For counting purposes, every coordinate has a virtual scan, even if the actual procedure would already have stopped. This only enlarges the failure event and preserves the uniform scan distribution. The lemma bounds the combined missing-fork and fixed bad-slope contribution by $2\sum_j M_{ij}^{-1}$ per checkpoint. There are at most R_C reached short-challenge checkpoints. Integrating the bound over their actual reach probabilities gives the last term in ε_i. Only one defective lane is needed at each center, so no additional factor of four arises. Adding the projection and aggregation exceptions proves the claim. □

At the pivot an actual submission is checked directly. At a nonterminal layer, Protocol 2 permits one irrevocable child submission on the first forward path. Conditional on its reached pre-projection checkpoint, the child guarantee bounds the joint event of acceptance and empty extraction. Averaging over these reached checkpoints charges that failure once. In a side scan, an immediate retry or an empty child result marks that point ineffective. The scan continues from the same pre-c state. These entries affect the first-effective-neighbor law in Lemma 5.1 and incur no separate child failure term. Combining this observation with Lemmas 5.2 and B.4 gives $\kappa_i \leq \kappa_{i+1} + \varepsilon_i$, with $\kappa_d = 0$. In particular, m in (10) is the same unconditional missing-fork mass for every fixed w in the acceptance-weighted sum (8). Each independent RawNode call uses fresh suffix coins and a new counterfactual table at each reached pre-c state. Within a scan, the chosen center retains its original continuation.

B.4 The Two Front-End Checkpoints

The backend extracts an opening of Definition 3.4 for the root in Protocol 1. A witness for Definition 3.3 must also establish canonical encodings, the private fold and the pending update. We use two checkpoints for the ordinary and zero-pending guarantees in Theorem 5.1.

The ordinary checkpoint, denoted by σ_F, occurs after α, γ, the new pending value, and all original input, output, and auxiliary commitments have been fixed, but before either ρ or η is sampled. An independent continuation from this checkpoint rebuilds its own sumcheck transcript and root system. The backend opening extracted from that continuation authenticates its terminal evaluation against the commitments fixed at σ_F.

Each label identifies the opening coordinates under each old, fresh, output and helper commitment and the public offsets fixed at σ_F. Each restart preserves these records while drawing fresh ρ, η, all round points and all backend coins. Roots in different continuations can have different terminal rows.

Lemma B.5 (Field and pending comparisons). *Ordinary front-end extraction adds at most* $72/(|K|-1) + 2\ell/|K|$ *to the backend failure bound. The zero-pending extraction from the earlier pre-*γ *checkpoint adds at most* $4k/(|K|-1)$ *beyond the ordinary guarantee.*

Proof. Fix a label w satisfying the shared commitment rows, the fixed-coordinate equations and the root extraction budget. A label failing any of these checks has no mass as a certified opening. For this fixed w, the square-table identity of Section 4.2 gives the actual initial square sum minus the public claim as

$$E_{\mathrm{enc}}(w) + \eta \left(\sum_{a=1}^{72} \rho^a A_a(w) + \delta(w) \right),$$

where $A_a(w)$ are the state and private-folding discrepancies, and $\delta(w)$ is $\sum_{m=1}^{4k} \gamma^m r_m(w) - (s' - s)$.

Suppose first that $E_{\mathrm{enc}}(w) > 0$ over the integers. The no-wrap bound in Lemma 4.1 ensures that this energy remains nonzero modulo q. For every fixed value of ρ, at most one nonzero value of η can cancel it. This event has probability at most $1/(|K|-1)$. If the encoding energy is zero but w violates the transition relation, at least one of the semantic discrepancies is nonzero. Their weighted sum is then a nonzero polynomial in ρ, of degree at most 72, with the pending discrepancy as its constant term. Since ρ is uniform over $K^\times$, this event has probability at most $72/(|K|-1)$. The two cases are disjoint, and together they are bounded by the latter quantity.

An incorrect initial sumcheck claim might still reach a correct terminal value. Consider the first round in which an incorrect running claim becomes consistent with the true partial sum for the fixed opening. The prover's round polynomial and the correct round polynomial differ by a nonzero polynomial of degree at most two. Their difference is fixed before the verifier samples that round's evaluation point, even when the prover adapted its earlier messages. The probability of equality is therefore at most $2/|K|$ per round. The terminal opening equations of Appendix A.3 bind the value ν sent in Protocol 1 to this w. The verifier checks ν^2 against its last claim. Therefore an incorrect initial claim must have become correct at some round, at total cost at most $2\ell/|K|$ by the sumcheck argument [39].

The extractor uses the independent-repeat calculation of Lemma 5.2. If the two openings of a particular shared commitment differ, subtract their centered lifts. Their public offsets cancel, and the difference of the inserted free coordinates has norm at most $2B_0$ and nonzero residue. Every free value coordinate belongs to a state or source commitment in Definitions 3.1 and 3.2. Every free helper coordinate belongs to the auxiliary commitment of Definition 3.3. The fixed-one and alignment equations cover the remaining coordinates. Thus two distinct ordinary root openings yield a kernel even

when their terminal rows differ. For a bad first root opening, repeat independently from σ_F until the trial returns an opening or a kernel. If the total nonempty mass is $t > 0$ and the raw mass for bad w is a_w, the failure contribution from identical labels is at most $\sum_w a_w^2/t \le \max_w a_w$. When $t = 0$ a bad first output cannot trigger the repeat. Together with the original backend error this proves the ordinary bound.

For the strengthened conclusion, restore σ_γ after α but before γ in Protocol 1. The old pending value and every old and fresh input record are fixed. A fresh γ can lead to different output and auxiliary commitments. The label for this comparison is the tuple of canonical fresh-source openings under the original input commitments. Only ordinary witnesses for violating zero-pending continuations enter the comparison; a verified kernel terminates it immediately. Suppose the first execution has an ordinary witness but fails the stronger conclusion. Its old pending value or some fresh quadratic residual is nonzero. For fixed canonical source values, a zero-pending continuation must satisfy

$$s + \sum_{m=1}^{4k} r_m(w)\gamma^m = 0.$$

This is a nonzero polynomial of degree at most $4k$. Its probability of vanishing for a uniform $\gamma \in K^\times$ is at most $4k/(|K| - 1)$.

The extractor independently retries from σ_γ until it obtains another certified violating zero-pending record or a kernel. Two different labels differ on at least one shared fresh commitment. In Definition 3.2, the public and reserved slots agree. The differing physical bits for this source lie among $64 \cdot 48$ positions, so their difference gives a state-key kernel of norm at most $\sqrt{64 \cdot 48}$. For matching labels the pending polynomial is fixed. If a_λ is the unconditional mass of a certified violating record with source label λ, then $a_\lambda \le 4k/(|K| - 1)$. With t_γ the total mass of such records and kernels, the joint matching mass is at most $\sum_\lambda a_\lambda^2/t_\gamma \le \max_\lambda a_\lambda$ when $t_\gamma > 0$. At $t_\gamma = 0$ the repeat cannot be triggered. Only the ordinary extraction failure on the first execution is charged separately. This proves the stated additional loss. □

For the frozen-window embedding of Appendix C.3, use the logical projection onto the committed value and helper coordinates when comparing front-end openings. Sum over preimages differing only on ignored physical coordinates. The ordinary alignment rows still enforce their prescribed zeros.

B.5 Expected Work at an Adaptive Checkpoint

The extraction analysis above bounds failure probability, but it does not immediately bound the number of prover continuations. In particular, a coordinate scan may examine many challenge values before finding an effective neighbor. We now bound its expected cost under the same probability distribution used in the soundness proof.

Fix the counterfactual response table of Lemma 5.1 before drawing its center. Consider the coordinate scan in Algorithm 2 on a fiber of size M containing a effective points. Conditional on the other challenge coordinates the center is uniform. If $a \ge 2$, a particular effective point can be visited as a fork only from another effective center, giving visit

probability at most $(a-1)/M \cdot 1/(a-1) = 1/M$. An ineffective point may also be visited during a scan. For $a \geq 1$, the probability of choosing an effective center is a/M. A kernel center ends the call and only an ordinary center starts a scan. Given a scanning center, an ineffective point precedes all $a-1$ other effective points with probability $1/a$. Its visit probability is again at most $1/M$. The same bound covers fibers with zero or one effective point. Stopping early because of a kernel or a previous failed coordinate only decreases the number of visits.

After averaging over the other coordinates, a given product challenge has expected visit multiplicity at most $1 + s_i$ times its mass under the uniform product law, counting both center and scans.

Attach to each presampled entry the realized cost of its child extraction if it submits immediately, and zero if it retries. The visit bound holds for every fixed table and hence for entries marked with these costs. In Protocol 2, immediate submission precedes any child verifier coin. Condition on the public child from Algorithm 1 and, if reached, its first pre-projection tag. The child extraction budget is uniform over these checkpoints. Averaging the marks over their randomness bounds the contribution under one uniform challenge by its immediate-submission probability times T_{i+1}. Local work up to the child's first tag or early rejection is included in the local cost. Summing over the reached short-challenge checkpoints does not introduce another factor of R_C on child extraction. Along the original execution, only the first actual submission enters a child. Earlier checkpoints ended in retries. The expected child cost of a RawNode call is consequently at most $(1 + s_i)T_{i+1}$. Local query work, which can occur at every reached checkpoint, is bounded separately by $R_C(1 + s_i)H_i$.

Sampling the challenge fibers. The challenge support filters an ambient integer coefficient ball by operator norm. The pivot also excludes zero from its first coordinate. The extractor samples a uniformly random permutation of ambient coefficient-ball ranks other than the center lazily. Each visited rank is decoded, tested for support membership, and discarded if it fails the filter. Filtering a uniform permutation preserves the relative order of the eligible challenges, so the coordinate scans retain the distribution required by Lemma 5.1. For an ambient ball $\mathcal{B}$ and eligible support $\mathcal{C}$, the expected number of additional membership tests, weighted by entry into the scan, is at most $|\mathcal{B}|/|\mathcal{C}|$. This bound also covers a fiber containing only one effective point, where an unsuccessful scan may otherwise visit every eligible challenge. The lazy permutation stores only the ranks actually visited. Its sampling and membership-test costs are included in C_i^{loc}. Thus the expected-work claim does not require materializing the full challenge space.

Independent repetition. A bad first candidate triggers a second context. Repeat RawNode independently until its output is nonempty. Suppose one RawNode call has unconditional expected cost μ, invalid-return probability b, and nonempty-return probability t, with $b \leq t$. The second context is entered with probability b. When $t > 0$, the expected total cost of its independent repetitions is

$$b \sum_{\ell \geq 1} (1-t)^{\ell-1} \mu = \frac{b\mu}{t} \leq \mu.$$

The cost of a trial may be correlated with its own result. This does not affect the calculation, since the decision to enter that trial depends only on the results of previous trials. When $t = 0$, the second context is never entered.

Including both contexts and the public relation and kernel checks gives the expected-work recurrence (9). Here H_i bounds one immediate challenge query up to the retry or submission decision. The remaining local work, including restoration and sampling, is covered by C_i^{loc}. The terminal verification cost is bounded by T_d.

The same calculation applies twice in Lemma B.5, first at σ_F using the mass of certified openings and kernels and then at σ_γ using only the mass of violating zero-pending records and kernels. In each comparison the trigger is a subset of the repeat's stopping event. Ordinary extraction therefore incurs at most two backend call budgets, and the strengthened extraction incurs at most four, plus local work at the respective checkpoints. For the four concrete profiles, expanding the recurrence gives coefficients of T_d equal to 120,000, 168,000, 302,400, and 786,240. In particular, the largest profile has block counts $(s_0, \ldots, s_4) = (12, 8, 6, 4, 5)$, giving the terminal coefficient $2^5(13)(9)(7)(5)(6) = 786{,}240$. The recurrence is uniform over all reachable checkpoints and averages over accepting, rejecting, and abandoned continuations. For a growing parameter family, expected-polynomial extraction additionally requires polynomial bounds on the local operations, challenge encodings, ambient-to-filtered sampling ratios, and the expanded recursion factor.

B.6 From Correlated Commitment Roles to Module-SIS

The extractor may return a kernel under several public matrices. These matrices are derived from a shared CRS array, so their entries are not necessarily independent across commitment roles. We show that the joint distribution still permits reductions to standard Module-SIS instances.

For a representative group g we use the homogeneous Module-SIS search problem [38]. Write $\mathrm{Adv}_{\mathrm{MSIS},g}(t)$ for the maximum success probability of a time-t solver given a uniform matrix $A \in R_q^{a_g \times b_g}$. It must output $v \in R^{b_g}$ with $Av = 0 \bmod q$, $v \neq 0 \bmod q$, and integer coefficient norm $\|v\|_2 \leq \beta_g$. Centering modulo q preserves its residue and can only reduce this norm.

The first representative covers state and helper commitments. Lemma B.5 gives radius $2B_0$ for disagreeing root openings at σ_F and $\sqrt{64 \cdot 48}$ for disagreeing canonical source openings at σ_γ. Disagreement at a shared history boundary gives radius $\sqrt{14016}$, since Definition 3.1 packs $73 \cdot 4 \cdot 48$ binary coefficients into the state opening. At a nonterminal layer, disagreement on the digits opening either tag of Protocol 2 gives radius $2B_{i+1}$ by Lemma B.2. The two six-row representatives cover the root tags and the later tags, respectively. The response-anchor check and the block comparison (6) in Algorithm 2 give kernels under A_i of radius at most $8\Gamma Z_i$, using the audited inequality $Z_i + \Gamma\sqrt{s_i}B_i \leq 8\Gamma Z_i$. The root block matrix and later block matrices form two more groups. For the two-row pivot matrix in Protocol 3, the digit-opening radius is 1024 by Lemma B.2.

For each group, the representative width is the largest width among roles with that common row count, and the radius is the largest required bound, rounded upward. These maxima may arise from different roles. Table 7 gives all six representatives at each arity.

Embedding the Module-SIS challenge. Fix a representative group g. The reduction flattens its uniform $a_g \times b_g$ challenge column by column into the first $a_g b_g$ entries of the shared CRS array and draws the remaining entries independently and uniformly. Every commitment role reads a fixed prefix of this array, using its prescribed row count and column-major indexing. Since the embedded challenge is itself uniform, the complete array has exactly the distribution of an honestly generated CRS. In particular, all overlaps among different commitment roles are preserved.

When a kernel belongs to a narrower matrix within group g, the reduction pads its vector with zero ring elements to the representative width. The padded vector remains nonzero modulo q, has unchanged centered norm, and lies in the kernel of the representative challenge matrix. The reduction runs the extractor and stops at its first verified kernel, using a fixed order for ties. In the experiment embedding group g, the reduction outputs the padded kernel only if its role is assigned to g. The simulated CRS has the same joint distribution for each choice of g. Under this common distribution the six first-kernel events are disjoint, so their probabilities sum to the kernel term in Theorem 6.1.

Composition over authenticated histories. Fix one profile for the serialized history and an efficiently generated input and auxiliary state. As in Theorem 6.1, the verifier fixes the next input record and charges each attempt before issuing α in Protocol 1, including any attempt later abandoned. Suppose at most H online attempts are issued. We may pad the sequence of attempts to length H for analysis. Conditional on the public history before each issue, the strengthened extraction bound applies to the corresponding pre-γ checkpoint. The probability of a required extraction failure is therefore at most κ_{zero} times the probability that the attempt is issued. Summing these contributions gives $H\kappa_{\mathrm{zero}}$, even if the prover decides adaptively when to abort or stop. Private rewinding changes neither the verifier's issued challenges nor its attempt counter. Hence abandoned attempts are charged without treating the extractor's internal queries as new real executions.

Consider an accepted continuous history whose initial and final pending values are zero. Adjacent transitions must use the same public state record at their common boundary. If their extracted canonical openings disagree, subtracting the two openings gives a short state-commitment kernel. Otherwise, the common opening is consistent across the boundary. Starting from the final transition, the zero-pending guarantee of Lemma B.5 forces its old pending value and all its fresh residuals to be zero, unless a charged failure or short kernel occurs. The same argument can then be applied to the preceding transition. Continuing backward establishes witnesses for the original recorded transitions with every source in the strict relation of Definition 3.2.

The reduction puts one clock on the whole history extractor, including the prover's continuations, state restoration and adjacent-record comparisons. If its expected cost is bounded by T_H, Markov's inequality charges at most T_H/U for exceeding cutoff $U > 0$.

For group g, the Module-SIS reduction additionally requires time t_g for setup embedding, input generation, and output conversion. With a deterministic bound on this extra work, the probability of obtaining an assigned kernel is at most $\mathrm{Adv}_{\mathrm{MSIS},g}(U + t_g)$.

Combining the statistical failures, time cutoff, and six computational reductions gives the history bound

$$\Pr[F] \le H\kappa_{\mathrm{zero}} + \frac{T_H}{U} + \sum_{g=1}^{6} \mathrm{Adv}_{\mathrm{MSIS},g}(U + t_g), \tag{11}$$

Here F is the joint history failure in Theorem 6.1. If the extra work for g has only expected bound I_g, apply a second cutoff V_g to that work and charge $\mathrm{Adv}_{\mathrm{MSIS,g}}(U + V_g) + I_g/V_g$.

For one ordinary transition use κ_{tr} and the corresponding single-transition time budget. With $H = 16{,}384$ issued attempts, Table 1 gives $H\kappa_{\mathrm{zero}} < 2^{-130}$ in each of the four profiles. Appendix C.6 specifies the separate cutoff and six time-bounded advantage conditions required for a numerical bound on F.

Appendix C Parameters and Reproducibility

We describe the parameter selection, concrete security calculations, and reproduction procedure underlying the evaluation in Section 7.

C.1 Parameter Selection and Audit

Each batch size is instantiated by searching the same family of five-layer schedules. The root block count ranges over $\{4, 6, 8, 10, 12, 16, 20, 21, 24\}$, and each of the next three layers independently uses a block count from $\{3, 4, 6, 8, 10\}$. The terminal pivot always uses five blocks. The root radix and the common radix of the next three layers are selected from $\{16, 32, 64, 128\}$, giving 18,000 candidate schedules per batch size.

The search first applies the necessary coordinate-error condition $320H \sum_i s_i/M \le 2^{-130}$, where $H = 16{,}384$ and M is the common lower bound on the challenge-support size. This leaves 17,808 schedules. An analytic geometric-series screen then checks the six coefficient-lattice representatives. Among the surviving schedules, the search minimizes $\sum_i N_i$, with ties resolved by the size of the largest required CRS prefix and then by lexicographic schedule order. Since every layer uses 384 projection rows, this objective also minimizes the number of coefficients expanded by one projection at each layer.

The selected profiles are audited using exact arithmetic. The audit recomputes balanced digit norms by dynamic programming over carries, propagates the child dimensions and norm budgets, and evaluates the projection and extraction errors. In increasing order of batch size, the rounded-up bounds on $956B_0^2 + 3281V$ are 661,646,998, 1,957,917,542, 7,142,999,715 and 27,883,328,406. Each is below q, as required by the integer-lifting argument in Appendix A.

Table 4 gives the layer parameters for the largest profile. For example, its first child contains $64(2 \cdot 4733 + 1536 + 2496) = 863{,}872$ active coefficients. The two additional widths account for the block-image digits and the symmetric-product digits. Padding to

eight equal blocks gives $N_1 = 864{,}256$. The remaining layers follow the same dimension and energy recurrences.

Table 4. Five backend layers at $k = 1024$. Dimensions include alignment. Transfer exponents are lower bounds.

Layer	Blocks × ring width	N_i	m_i	S_i	Seed (b, h)	Transfer exponent
0	$12 \times 4{,}733$	3,634,944	263,460	3,105,761	$(1174, 22)$	453
1	$8 \times 1{,}688$	864,256	2,436	572,903,234	$(1200, 20)$	463
2	6×926	355,584	2,308	289,934,645	$(1186, 18)$	460
3	4×823	210,688	2,052	173,504,448	$(1235, 17)$	486
4	5×496	158,720	2,180	119,655,711	$(1172, 17)$	456

Challenge-support audit. The statistical calculation uses a certified lower bound on the filtered challenge support. The ambient set $\mathcal{B}$ consists of integer coefficient vectors with squared norm at most 128. The filter retains vectors whose multiplication operator norm is at most 26.

Negacyclic multiplication is diagonalized by evaluation at the odd powers of a primitive complex 128th root of unity ω. Its 32 conjugate pairs have the same twelfth moment over $\mathcal{B}$, because cyclotomic automorphisms act by signed coefficient permutations. Consequently,

$$|\mathcal{C}| \geq |\mathcal{B}| - \left\lfloor \frac{32}{26^{12}} \sum_{c \in \mathcal{B}} |c(\omega)|^{12} \right\rfloor .$$

The audit computes both quantities with integer arithmetic. It splits $c(X)$ as $a(X^2) + Xb(X^2)$ and partitions the coefficient energy between the two halves. On each energy shell in degree d, invariance under multiplication by X makes a mixed moment vanish unless $2d$ divides the difference of its exponents. Binomial expansion then reduces the calculation to signed integer convolutions, with direct enumeration at degree one. The implementation is provided in `parameters/challenge_moment.py`.

The resulting support lower bound exceeds $2^{158.0867}$. The statistical budget uses the computed integer bound and subtracts one for the first pivot coordinate, which excludes zero.

C.2 Commitment Roles and Encodings

The sixteen public-matrix roles are assigned to the six representative Module-SIS instances used in the shared-CRS reduction. We retain the representative names from the parameter files. The four-row representative A_{aux} covers the state and auxiliary commitments. The block-image representatives A_0 and A_1 cover the root and later layers, respectively. The six-row representatives D_0, D_1 cover the two digit tags at the root and at subsequent nonterminal layers. The two-row representative B_{pivot} covers the terminal digit commitment.

Within each group, the representative uses the maximum matrix width and maximum kernel radius separately. These maxima need not come from the same role. At $k = 16$,

for example, the auxiliary matrix has 172 columns while the state matrix has 219, so their shared representative has 219 columns. A narrower role kernel is padded with zeros, preserving its norm, nonzero residue, and kernel equation. This is the assignment used by the shared-CRS reduction.

The serialized representation uses six bytes per base-field coefficient, 24 bytes per extension-field element, and 384 bytes per ring element. A short challenge is encoded by a 20-byte rank in the ambient coefficient ball. The decoder checks the operator-norm filter and the pivot's nonzero restriction. Projection images and terminal responses use padded signed Rice encodings.

Each projection or short-challenge draw also carries one control byte. A further projection draw adds its complete seed to the verifier stream, while a further short-challenge draw adds $20s_i$ challenge bytes at layer i. These conventions allow the summary script to check transcript lengths against the actual retry sequence.

For $k = 1{,}024$, the shared CRS contains 75,728 ring elements and occupies 29,079,552 bytes in packed form. The implementation stores coefficients in 64-bit words, giving 38,772,736 bytes of element storage in memory.

C.3 Frozen-Window Dimensions

The settlement comparison in Section 7 uses eight transitions with 994 fresh sources. More generally, a window with h transitions and n fresh sources has $V = 64n + 272(h+1)$ active private base-field components. Its root system contains $55V + 1$ active coefficient slots and $256(n + h + 2) + 5$ public scalar rows.

For the evaluated log, $V = 66{,}064$. The square table contains 3,697,714 entries and requires 22 sumcheck rounds. The window has 4,532 semantic equations. Replacing the pending equations with public adjacent-pending checks in the matched direct batch reduces this count to 4,524. Both constructions retain the same square-table size.

The common-root embedding separates logical witness coordinates from physical capacity. It maps the active window coordinates into the $k = 1{,}024$ root and keeps the fixed-one equation active. The root also enforces its 543 ordinary alignment zeros. Within the logical envelope, 880 additional slots are unused by the window equations, and 5,888 zero rows extend the active row count to 262,917 before ordinary alignment. Commitment comparisons use the logical fields, while the norm projection acts on the full physical vector.

This embedding also gives a communication budget for the one-shot constructions. The ordinary $k = 1{,}024$ transcript contains 4,920 front-end bytes, while the matched window requires 3,192. Replacing that component gives a no-retry budget of $170{,}361 - 4{,}920 + 3{,}192 = 168{,}633$ bytes with the log already shared. Acquiring the log requires at least another 1,540,608 bytes for the 994 source commitments and nine state commitments alone. These values are derived from the window dimensions and message encodings.

C.4 Experimental Reproduction

The measured ordinary-transition binaries correspond to revision `bf78a20`. Revision `776e7e2` packages the results and pins the estimator. From the artifact root, `python3`

`parameters/sweep_profiles.py -check` verifies the generated ledger and C++ constants. A CMake Release build provides the `sonata_sweep` target. The driver in `benchmarks/run_four_arities.py` takes its executable path, a fresh results directory and three CPUs on distinct physical cores with separate last-level caches. Our trials used CPUs 4, 36 and 68.

The full-transition environment uses GCC 13.3.0, CMake 3.28.3, and Python 3.12.3 under Ubuntu 24.04.4 LTS with kernel 5.15.0-161-generic. Each group of three trials completes before the next batch size begins. The input and CRS fixtures are reproducible, while verifier challenges are sampled afresh during each execution.

The summary script, `benchmarks/summarize_four_arities.py`, independently decodes the two message streams and checks their lengths against the recorded challenges and retries. It produces per-trial and aggregate timings, communication counts, packed CRS sizes, and attack-estimate tables.

Table 5 supplements the prover and verifier times in Section 7. Public replay is already part of verifier time. Wall time additionally covers input and CRS preparation, file operations, and the negative test.

Table 5. Supplementary ordinary-transition timings in seconds. Both columns report medians of three executions per arity.

Fresh inputs k	Public replay (s)	Wall (s)
16	18.036	37.912
64	31.033	65.544
256	74.905	159.603
1,024	226.093	487.709

The live verification fixture uses the `sonata_interactive` target and `examples/run_interactive.py`. The four component benchmarks use `sonata_bench` and `benchmarks/run.py`. We ran each workload in five fresh processes on an Intel Core i9-13980HX under Ubuntu 24.04 LTS on WSL2 (kernel `6.18.33.2-microsoft-standard-WSL2`), with GCC 13.3.0 at `-O3 -DNDEBUG` and Python 3.12.3. Processes were pinned sequentially to logical CPU 4 without a warm-up. The RSS high-water mark includes input preparation and runtime overhead (Table 6).

Table 6. Component measurements on one pinned logical CPU. Each timing covers the batch in its row. Peak RSS is the maximum process-lifetime high-water mark over five runs.

Workload	Median (ms)	Range (ms)	RSS (MiB)
Canonically encode 66,064 field scalars	4.073	3.950–4.309	31.281
Square sumcheck on 44,386 entries, 16 rounds	4.502	4.323–4.769	4.531
Expand and apply a $384 \times 44{,}160$ projection	799.013	774.381–836.909	19.531
Sample 21 filtered degree-64 ring challenges	11.704	10.321–12.929	19.309

The `sonata_epoch` executable reproduces the front-end settlement comparison with `-target`, `-target-sequential` and `-target-direct`. The capacity audit is implemented in `parameters/epoch.py`. The separate `sonata_e2e` window fixture runs via `exampl`

`es/run_e2e.py`. It uses deterministic verifier coins and the earlier five-layer backend schedule of revision `677889d`. Its recorded public verification accepted the honest pivot and rejected the modified terminal commitment. The budget in Appendix C.3 applies to the optimized envelope.

C.5 Lattice-Attack Estimates

The six representative instances determine which lattice problems must be priced for each profile. Table 7 records their dimensions, kernel radii, and generic coefficient-lattice estimates.

A matrix with a ring rows and b ring columns induces $n = 64a$ modular coefficient equations in $m = 64b$ unknowns. An attacker may restrict the unknowns to D coefficients. When the selected columns have full row rank, the resulting kernel lattice has determinant q^n. The geometric-series model of basis quality [47,25] predicts a sufficiently short vector at BKZ block size $B \leq D$ when

$$(D-1)\log \delta_B + \frac{n}{D}\log q \leq \log \beta_g,$$

where δ_B is the root-Hermite factor and β_g is the representative kernel radius. The model treats the signed rotations within a ring block as achieving the same reduction quality as a generic coefficient lattice with the corresponding dimension and determinant.

For each instance we maximize the permitted $\log \delta_B$ over integer dimensions, checking the two integers surrounding the continuous optimum and both endpoints. Independent dimension selection in the Euclidean SIS routine of Lattice Estimator gives the same block sizes for all 24 representatives [4,5]. We used SageMath 10.9. The artifact pins revision `53da598` as a Git submodule, verifies its full commit hash before running `parameters/estimate_sis.py`, and records that hash with the estimates.

The estimator follows Core-SVP pricing [6], which assigns base-two work exponents $0.292B$ and $0.265B$ to classical and quantum sieving. These prices model a lattice sieve in the BKZ block dimension [11,2]. The smallest estimates occur for A_0 at $k = 64$, with exponents 150.672 under classical pricing and 136.740 under quantum pricing. The artifact also reports MATZOV [41] classical operation counts and quantum depth-times-width estimates. For this instance, their logarithms are 176.546 and 167.671. The MATZOV values for A_{aux} at $k = 16$ and $k = 64$ are extrapolations because their block sizes, 1,359 and 1,131, exceed the fitted endpoint of 1,024. Quantum pricing describes the lattice-attack model. The knowledge extractor uses classical rewinding.

Single-component CRT checks. The audit also considers short solutions supported on only one CRT component. Such checks exploit the relation between coefficient length and invertibility in partially splitting cyclotomic rings [40]. In R, the modulus factors as $q = \pi\bar{\pi}$, with $\pi = 12993454 + 10613441X^{32}$. Multiplication by X^{32} is orthogonal and skew-symmetric, giving $\|\pi y\|_2^2 = q\|y\|_2^2$. A lift supported on one component belongs to (π) or $(\bar{\pi})$, so a nonzero such lift has norm at least $\sqrt{q}$.

Table 7. Module-SIS radii, BKZ block sizes and estimated $\log_2$ Core-SVP work. Stars mark blocks beyond the MATZOV fit range.

k	Role	Ring shape	β_g	B	Classical	Quantum
16	A_{aux}	4×219	1,658	1359*	396.828	360.135
16	A_0	16×337	1,306,281,810	536	156.512	142.040
16	A_1	16×361	1,333,281,192	535	156.220	141.775
16	D_0	6×448	98,126	729	212.868	193.185
16	D_1	6×448	100,154	726	211.992	192.390
16	B_{pivot}	2×256	1,024	647	188.924	171.455
64	A_{aux}	4×508	2,852	1131*	330.252	299.715
64	A_0	16×665	1,759,232,804	516	150.672	136.740
64	A_1	16×648	1,515,546,845	526	153.592	139.390
64	D_0	6×672	132,150	680	198.560	180.200
64	D_1	6×448	113,845	704	205.568	186.560
64	B_{pivot}	2×256	1,024	647	188.924	171.455
256	A_{aux}	4×1852	5,446	921	268.932	244.065
256	A_0	16×1819	656,411,926	588	171.696	155.820
256	A_1	16×969	536,820,017	605	176.660	160.325
256	D_0	6×1152	98,608	728	212.576	192.920
256	D_1	6×768	80,643	764	223.088	202.460
256	B_{pivot}	2×256	1,024	647	188.924	171.455
1,024	A_{aux}	4×7228	10,760	751	219.292	199.015
1,024	A_0	16×4733	972,750,044	558	162.936	147.870
1,024	A_1	16×1688	692,007,305	584	170.528	154.760
1,024	D_0	6×2496	146,129	665	194.180	176.225
1,024	D_1	6×1152	103,955	719	209.948	190.535
1,024	B_{pivot}	2×256	1,024	647	188.924	171.455

The radii of $A_{\mathrm{aux}}, D_0, D_1$, and B_{pivot} are below this threshold in every profile. For A_0, A_1, substituting $x = \pi y$, or its conjugate, leaves $32a$ modular coefficient equations and the bound $\|y\|_2^2 \le \lfloor \beta_g^2/q \rfloor$. The audit searches block sizes from 40 through $\min(64b, 65{,}536)$, choosing the most favorable eligible dimension for each.

The first satisfying block size is 20,566 for A_1 at $k = 16$, and 12,624 and 15,532 for A_0, A_1 at $k = 64$. All three are beyond the MATZOV calibration range. The remaining five block-matrix cases have no satisfying block size within their search intervals. These checks evaluate the single-component attack under the same geometric-series model.

C.6 Extraction Cutoff and Concrete Security Budget

The statistical errors and lattice work estimates enter the concrete analysis in different ways. For $H = 2^{14}$ issued attempts, the values of $-\log_2(H\kappa_{\mathrm{zero}})$ are 131.3725, 131.2412, 131.0099, and 130.6355 in increasing order of batch size. These determine the statistical term in Theorem 6.1.

The time-cutoff term requires an upper bound $\overline{T}_H \ge T_H$ on the expected cost of the entire history extractor. Choosing $U = \lceil 2^{130}\overline{T}_H \rceil$ gives $T_H/U \le 2^{-130}$. For example, a cutoff of 2^{200} bit operations requires $\overline{T}_H \le 2^{70}$ in the same cost model.

If each of the six time-bounded advantages $\mathrm{Adv}_{\mathrm{MSIS},g}(U + t_g)$ is additionally at most 2^{-132}, the total failure bound becomes

$$2^{-130} + 2^{-130} + 6 \cdot 2^{-132} = \frac{7}{8}\, 2^{-128}.$$

Using the exact statistical terms gives conditional total-failure exponents of 128.471, 128.453, 128.417, and 128.348 for the four profiles.

This allocation separates the three quantities needed to instantiate the theorem, namely the statistical error, the extraction cutoff, and the computational advantages. The estimates in Appendix C.5 provide model-based attack work factors. Establishing the stated probability budget additionally requires the six time-bounded advantage bounds and a cost model for the malicious prover under rewinding.