ASCENT: Algebraic Structured Compilation of Exact Norms for Succinct Lattice Proofs (Full Version)

Lizhen ZHANG[1], Shang Gao[1], and Bin Xiao[1]

The Hong Kong Polytechnic University, Hong Kong, China SAR
lizh6.zhang@connect.polyu.hk, shanggao@polyu.edu.hk
b.xiao@polyu.edu.hk

Abstract. Norm bounds are semantic constraints in lattice cryptography: satisfying the modular equations is not enough unless the witness also satisfies the prescribed shortness bound. Existing norm reductions either introduce slack, or rely on general sumcheck style machinery whose transcript cost grows with the witness length. We introduce a reduction-of-knowledge compiler for exact ℓ_2-norm constraints. It compiles the original norm language for a committed lattice witness into a narrow structured relation consisting only of commitment-opening constraints, tensor-structured witness-public inner-product claims, and relaxed admissible shortness bounds. Exact norm soundness is enforced inside the reduction, so the remaining proof obligation is structured and algebraic. The compiler adds only $O(1)$ prover-to-verifier communication, measured in ring and field elements, independent of the witness length N, and requires only $O(\log N)$ verifier work. The same techniques can also extend to a folding mechanism for broader quadratic relations, including R1CS-type circuits. At 128-bit security, our concrete estimates give transcripts of about 27.69KB for exact norm reduction and 27.75KB for the R1CS folding application. This is about 3–4× smaller than LatticeFold+, about 2–3× smaller than SALSAA, and remains close to the compact estimates reported for Cyclo.

Keywords: Lattice-based proof systems, Reduction of knowledge

1 Introduction

Recent advances have made lattice-based proof systems a promising foundation for post-quantum succinct arguments involving short witnesses, such as secret keys, commitment openings, preimages, and signatures, as well as for post-quantum verifiable computation [13,11,4]. A substantial line of work has developed efficient lattice-based proof machinery for algebraic relations, including commitment-opening constraints, structured witness-public inner-product claims, and related linear relations over committed witnesses [2,9,25,10,8]. These results suggest a modular route to post-quantum proof design: instead of building

separate proof protocols for application-specific languages, one can try to reduce the application relation to the algebraic interface already handled by these backends, and then delegate the remaining proof task to the backend without losing its succinctness advantage.

A central obstacle to this backend-oriented compilation approach is the treatment of norm bounds. In lattice cryptography, shortness is part of the security semantics that defines the intended cryptographic languages: witnesses such as secret keys, commitment openings, preimages, and signatures are valid only when they satisfy the prescribed norm bound, not merely the associated algebraic equations. The challenge is therefore to preserve this language-level condition while compiling the statement into the algebraic interface supported by efficient lattice proof backends. Several recent lattice-based proof systems [4,22,6,5,12,19] compile or reduce norm constraints using projection or sumcheck techniques.

Projection based approaches, such as Johnson-Lindenstrauss style norm reductions [4,22], relate the norm of the original witness to the norm of a shorter projected vector. The verifier checks the projected norm, together with algebraic consistency constraints showing that the projection was computed correctly. This reduces the norm claim to lower-dimensional algebraic constraints, but it preserves the original ℓ_2 bound only up to slack, typically a constant factor of 2. LatticeFold-style systems handle exact ℓ_∞ bounds by compiling coefficient-wise range conditions into algebraic constraints. These constraints are encoded either as high-degree component-wise vector relations [5], or as polynomial-encoding constraints [6,12], and are then reduced by logarithmic-round sumcheck-style protocols to relations that can be handled efficiently by existing proof systems. SALSAA-style techniques [19,14,15] instead target ℓ_2-type shortness conditions. They express norm claims as polynomial identities over multilinear or low-degree extensions, and verify them through dedicated sumcheck-style reductions.

These approaches reveal a tension between exactness and backend compatibility. Projection-based reductions are efficient, but introduce slack and therefore do not preserve the exact norm language. Sumcheck-based reductions can certify stronger norm statements, but typically add an $O(\log N)$-level transcript overhead for a witness of coefficient length N. This overhead is acceptable when the downstream backend has comparable cost, but becomes a bottleneck when the backend is already highly succinct. For example, lattice polynomial-commitment Greyhound achieves $O(\log \log N)$ proof size for structured backend relations. If norm reduction still contributes an $O(\log N)$ transcript, then norm handling becomes the asymptotically dominant component. In this regime, generic norm reduction is no longer merely a preprocessing step; it prevents the final proof from fully inheriting the succinctness of the backend.

This calls for a norm-reduction method that achieves both exactness and backend-level efficiency. It should preserve the original norm language without slack, while keeping the reduction overhead small enough not to dominate highly succinct downstream backends.

Our Contribution. This work develops a reduction layer for exact norm constraints in lattice proof systems. Our main contribution is a compiler that transforms exact norm statements into a narrow class of structured relations already supported by efficient lattice proof backends [2,9,25].

The reduction introduces only $O(1)$ prover-to-verifier communication, measured in the number of ring and field elements, so its transcript size is independent of the witness length N. The verifier performs only $O(\log N)$ field operations. Thus, the reduction layer preserves the succinctness advantage of highly efficient backends, rather than adding a generic norm-reduction transcript whose cost dominates the final proof. We summarize our main contributions below.

A succinct compiler for exact norm reduction. We construct a three-stage reduction of knowledge framework for compiling exact norm constraints into backend-native structured relations. The first reduction, Π_{alg}, algebraizes the exact ℓ_2-norm statement into quadratic and structured linear constraints, while preserving the exact norm semantics required for knowledge extraction. The second reduction, Π_{comp}, compiles the quadratic constraints into witness-public inner-product constraints against tensor-structured public vectors. The third reduction, Π_{bat}, batches the resulting structured linear constraints into a smaller instance of the same backend relation family. Each reduction contributes only $O(1)$ communication in ring and field elements from the prover, and requires only $O(\log N)$ verifier work in field operations. Consequently, the composed reduction is succinct, verifier-friendly, and suitable for instantiation with highly efficient lattice PCS backends.

Importantly, exact norm soundness is enforced by the reduction layer itself. The downstream backend is only required to prove the final structured algebraic instance, and need not provide exact norm extraction on its own.

An application to folding generalized relations. The quadratic-compilation and batching techniques developed for norm reduction also extend to broader algebraic relations. The key point is that quadratic inner-product relations over committed witnesses can first be compiled into structured linear relations using Π_{comp}. Once in this form, they can be folded and batched by the same mechanism underlying Π_{bat}.

This perspective also yields a folding procedure for general R1CS-type circuit relations, since their constraints can be expressed as quadratic relations in the witness. We give the corresponding R1CS reduction and folding implementation in Appendix E. This application illustrates that the proposed reduction layer is not limited to norm statements; it can also serve as a backend-oriented compiler for generalized quadratic relations.

Concrete efficiency and comparison. We provide concrete parameter estimates for the reduction layer. At the 128-bit security level with negligible soundness error, the transcript of the core exact-norm reduction is approximately 27.69KB. Under the same parameter setting, applying the same machinery to fold R1CS-type relations yields a folding-layer proof of approximately 27.75KB. This result

compares favorably with prior lattice-based folding and norm-reduction techniques.

At a comparable security level, LatticeFold and LatticeFold+ [5,6] have proof sizes around 100KB, SALSAA [19] has proof size around 78KB, and Cyclo [12] achieves a compact proof size around 30KB for folding R1CS-type relations. Thus, our proof size is about 3-4× smaller than LatticeFold-style approaches and about 2-3× smaller than SALSAA. It is also comparable to Cyclo, the most compact point of comparison, while offering a different advantage: the core norm-compiler transcript is length-independent. More precisely, our reduction contains a fixed number of ring and field elements, rather than growing with the witness length. This allows the final proof to better preserve the succinctness of the downstream backend as the input length grows.

1.1 Technique Overview

We now give a high-level overview of the main technical ideas.

Our goal is to compile an exact norm-sensitive statement for a committed lattice witness $\vec{s} \in \mathcal{R}_q^n$, requiring $\|\vec{s}\|_2 \leq B$, into a backend-native algebraic relation. The target relation contains only the constraints already optimized by modern lattice proof backends: commitment-opening constraints, relaxed admissible shortness bounds, and witness-public inner-product constraints against tensor-structured public vectors. Throughout this paper, the term *backend* refers to the downstream proof system used to prove the reduced relation produced by our reduction-of-knowledge framework. Our focus is the reduction layer itself; we abstract away the concrete backend construction and assume any existing proof system satisfying the required relation interface and security properties.

The exact norm bound is not left to the backend. Instead, the reduction layer enforces the exact norm semantics and compiles them into these structured algebraic claims. The reduction consists of three sub-protocols.

- Π_{alg} transforms the original non-algebraic norm statement into an algebraic norm relation containing quadratic constraints.
- Π_{comp} compiles the resulting quadratic constraints into backend-native structured linear constraints.
- Π_{bat} batches these structured linear constraints into a smaller instance of the same backend relation family.

Thus, the overall reduction follows the pipeline

$$\mathcal{R}_{\mathsf{norm,na}} \xrightarrow{\Pi_{\mathsf{alg}}} \mathcal{R}_{\mathsf{norm,al}} \xrightarrow{\Pi_{\mathsf{comp}}} \mathcal{R}_{\mathsf{back}}^{(p)} \xrightarrow{\Pi_{\mathsf{bat}}} \mathcal{R}_{\mathsf{back}}^{(o)}.$$

Here, $\mathcal{R}_{\mathsf{norm,al}}$ denotes the algebraized norm relation, $\mathcal{R}_{\mathsf{back}}^{(p)}$ denotes the preliminary backend instance produced after compilation, and $\mathcal{R}_{\mathsf{back}}^{(o)}$ denotes the final compressed backend instance.

Algebraizing the exact ℓ_2-norm constraint. The first step compiles the exact integer norm condition into algebraic constraints over $\mathcal{R}_q$. For a committed witness $\vec{s} \in \mathcal{R}_q^n$, let $\vec{s} \in \mathbb{Z}_q^{nd}$ denote its coefficient embedding. The squared ℓ_2-norm can then be represented by the quadratic term $\langle \vec{s}, \vec{s} \rangle$, along with a coefficient-wise bound $\|\vec{s}\|_\infty \le B_\infty$ with $nd \cdot B_\infty^2 < q$, which rules out modular wraparound. [1] Under this condition, checking $\langle \vec{s}, \vec{s} \rangle \le B^2 \bmod q$ is equivalent to proving the intended integer statement $\|\vec{s}\|_2 \le B$.

It remains to algebraize the coefficient-wise range bound. For $B_\infty = 1$, the coefficients are binary, so the range condition is $\vec{s} \circ (\vec{s} - \vec{1}) = \vec{0}$. A random challenge $\vec{r} = (1, r, \ldots, r^{nd-1})$ compresses this component-wise condition into the quadratic check $\langle \vec{s} \circ \vec{r}, \vec{s} \rangle = \langle \vec{s}, \vec{r} \rangle$, together with structured linear checks certifying the auxiliary vector $\vec{s} \circ \vec{r}$. For $B_\infty = 2^m - 1$, we decompose $\vec{s} = \sum_{i=0}^{m-1} 2^i \vec{s}_i$ into binary slices, prove each slice binary, and enforce recomposition by a randomized linear check. Thus, Π_{alg} outputs an algebraic norm relation consisting only of quadratic constraints and structured linear constraints.

From quadratic constraints to structured linear constraints. The second step compiles each committed quadratic inner-product constraint into structured linear claims. Consider a quadratic claim over a coefficient vector of length $N = 2^{2t}$. We view the witness as the evaluation table of a multilinear polynomial over the Boolean hypercube, so that the quadratic relation becomes a sum of evaluations of a polynomial with individual degree at most 2. We then apply a two-stage sumcheck-style reduction by splitting the variables into two blocks.

In the first stage, the prover sums over the second block to obtain a partial-sum polynomial h_0, commits to its evaluation table, and sends the commitment. After receiving a verifier challenge $\vec{z}_1$, the prover fixes the first block to $\vec{z}_1$, obtains a second polynomial h_1, and commits to its evaluation table as well. Since h_0 and h_1 have individual degree at most 2, their tables are represented over a degree-2 domain such as $\{0, 1, -1\}^t$. Evaluations of these committed tables at verifier-chosen points can be expressed as inner products with tensor-structured Lagrange selector vectors.

Thus, the consistency checks among h_0, h_1, and the original witness reduce to witness-public inner-product constraints against succinctly described public vectors. Consequently, each quadratic constraint is transformed into backend-native constraints: commitments to the original and intermediate witnesses, together with structured linear inner-product claims.

Batching structured linear constraints. The preliminary backend instance may still contain many structured linear claims. Although each claim is already backend-native, proving them separately would introduce unnecessary overhead.

[1] We use the canonical non-centered representation, interpreting $\mathbb{Z}_q$ in $\{0, 1, \ldots, q-1\}$. We focus on the standard bounded-coefficient regime, where witnesses are sampled, encoded, or decomposed so that the relevant coefficients lie in a small unsigned interval and no single coefficient carries a large fraction of the ℓ_2 mass.

The role of Π_{bat} is to compress these claims while keeping the output in the same backend relation family.

We first normalize all structured linear claims into a common polynomial form. The constraints produced by the previous reductions involve tensor structured public vectors of different arities, mainly arity 2 and arity 3. We show that both types can be expressed as evaluation claims between a multilinear extension and a $\widetilde{\mathsf{eq}}$-type selector polynomial. After normalization, all claims have the same algebraic shape and can be handled by a single bi-variate sumcheck template with common verifier challenges. Claims evaluated at the same points are then combined by random linear combinations. As a result, the preliminary batch of structured linear constraints is compressed into a constant-size instance of the same backend relation family.

2 Preliminaries

We provide the necessary preliminaries in this section, and defer additional background to Appendix A.

2.1 Notation

Let λ be the security parameter, implicitly given to all algorithms. All parameters are bounded by $\mathsf{poly}(\lambda)$ unless explicitly defined. A function $\mathsf{negl}(\lambda)$ is negligible. For $n \in \mathbb{N}$, define $[n] \coloneqq \{0, 1, \ldots, n-1\}$. For integers $a < b$, define $[a\,{:}\,b] \coloneqq \{a, a+1, \ldots, b-1\}$. Let q be a prime modulus. Let $\mathbb{Z}$ denote the set of integers, and define $\mathbb{Z}_q \coloneqq \mathbb{Z}/q\mathbb{Z}$. When working with non-centered representatives, we identify each $x \in \mathbb{Z}_q$ with its canonical representative in $\{0, 1, \ldots, q-1\}$. Vectors in $\mathbb{Z}_q^n$ are written as $\vec{f} = (f_0, \ldots, f_{n-1})$. Matrices over $\mathbb{Z}_q$ are denoted by uppercase letters. Let I_m be the identity matrix of size $m \times m$. Let $\otimes$ be the Kronecker tensor product.

Let d be a power of two. Define $\mathcal{R} \coloneqq \mathbb{Z}[X]/(X^d+1)$ and $\mathcal{R}_q \coloneqq \mathbb{Z}_q[X]/(X^d+1)$. Ring elements are denoted in bold, *e.g.*, $\boldsymbol{f} \in \mathcal{R}_q$, except for challenges $c \in \mathcal{C}$. For $\boldsymbol{f} \coloneqq \sum_{i=0}^{d-1} f_i X^i \in \mathcal{R}_q$, define $\mathsf{ct}(\boldsymbol{f}) \coloneqq f_0$ and $\mathsf{coef}(\boldsymbol{f}) \coloneqq \vec{f} = (f_0, f_1, \ldots, f_{d-1})$. Bold uppercase letters denote matrices over $\mathcal{R}_q$. For vectors $\vec{\boldsymbol{f}}, \vec{\boldsymbol{g}} \in \mathcal{R}_q^n$, their concatenation is $(\vec{\boldsymbol{f}}, \vec{\boldsymbol{g}}) \in \mathcal{R}_q^{2n}$. For matrices $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{R}_q^{m\times n}$, write $[\boldsymbol{A}\ \boldsymbol{B}] \in \mathcal{R}_q^{m\times 2n}$. For $\vec{\boldsymbol{a}} \in \mathcal{R}_q^n$, let $\boldsymbol{a}_i$ or $\boldsymbol{a}[i]$ denote its i-th entry.

Norms. For $\boldsymbol{f} \coloneqq \sum_{i=0}^{d-1} f_i X^i \in \mathcal{R}_q$, we define $\|\boldsymbol{f}\|_1 \coloneqq \sum_{i=0}^{d-1} |f_i|$, $\|\boldsymbol{f}\|_2 \coloneqq \sqrt{\sum_{i=0}^{d-1} f_i^2}$, and $\|\boldsymbol{f}\|_\infty \coloneqq \max_{i\in[d]} |f_i|$. For $\vec{\boldsymbol{f}} \coloneqq (\boldsymbol{f}_0, \ldots, \boldsymbol{f}_{m-1}) \in \mathcal{R}_q^m$, define $\|\vec{\boldsymbol{f}}\|_1 \coloneqq \sum_{i=0}^{m-1} \|\boldsymbol{f}_i\|_1$, $\|\vec{\boldsymbol{f}}\|_2 \coloneqq \sqrt{\sum_{i=0}^{m-1} \|\boldsymbol{f}_i\|_2^2}$, and $\|\vec{\boldsymbol{f}}\|_\infty \coloneqq \max_{i\in[m]} \|\boldsymbol{f}_i\|_\infty$. The ℓ_∞-norm and ℓ_2-norm of $\vec{\boldsymbol{f}}$ satisfy $\|\vec{\boldsymbol{f}}\|_\infty \le \|\vec{\boldsymbol{f}}\|_2 \le \sqrt{md}\,\|\vec{\boldsymbol{f}}\|_\infty$. Unless stated otherwise, $\|\cdot\|$ refers to the ℓ_2-norm.

Conjugation automorphism. Let $\sigma_{-1} : \mathcal{R}_q \to \mathcal{R}_q$ be the automorphism induced by $X \mapsto X^{-1}$ in $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^d+1)$. For $\boldsymbol{f} = \sum_{i=0}^{d-1} f_i X^i$, we write $\sigma_{-1}(\boldsymbol{f}) = \sum_{i=0}^{d-1} f_i X^{-i}$, with exponents interpreted modulo X^d+1. For $\vec{\boldsymbol{a}}, \vec{\boldsymbol{b}} \in \mathcal{R}_q^n$ with coefficient vectors $\vec{a}, \vec{b} \in \mathbb{Z}_q^{nd}$, we have $\langle \vec{a}, \vec{b} \rangle = \mathsf{ct}\Big(\Big\langle \sigma_{-1}(\vec{\boldsymbol{a}}), \vec{\boldsymbol{b}} \Big\rangle\Big)$.

2.2 Lattice Problems

We recall the standard module-SIS (M-SIS) assumption [20].

Definition 1 ($\mathsf{M\text{-}SIS}_{\kappa,n,q,\beta}$). *Given a random matrix $\boldsymbol{A} \in \mathcal{R}_q^{\kappa \times n}$ for $n > \kappa$, M-SIS is to find a non-zero $\vec{\boldsymbol{z}} \in \mathcal{R}_q^n$ such that $\boldsymbol{A}\vec{\boldsymbol{z}} = \vec{\boldsymbol{0}}$ over $\mathcal{R}_q$ and $\|\vec{\boldsymbol{z}}\| \le \beta$.*

We define below the variant of the M-SIS problem in which the ℓ_2-norm bound is replaced by an ℓ_∞-norm bound. It is clear that $\mathsf{M\text{-}SIS}_{\kappa,n,q,\beta_\infty}$ is at least as hard as $\mathsf{M\text{-}SIS}_{\kappa,n,q,\sqrt{dn}\,\beta_\infty}$.

2.3 Low-Degree Extensions, Multilinear Extensions, and Bi-variate Sum-Checks

We recall the polynomial-extension notation used throughout the paper. Let $\mathbb{F} = \mathbb{Z}_q$ for a prime modulus q. For a finite domain $\mathcal{D} \subseteq \mathbb{F}$ and $a \in \mathcal{D}$, let $\lambda_a(X) := \prod_{a' \in \mathcal{D},\, a' \neq a} \frac{X - a'}{a - a'}$ be the univariate Lagrange basis polynomial over $\mathcal{D}$. For $\vec{b} = (b_0, \ldots, b_{\mu-1}) \in \mathcal{D}^\mu$, define $\lambda_{\vec{b}}(\vec{X}) := \prod_{i=0}^{\mu-1} \lambda_{b_i}(X_i)$.

Low-degree extensions. Let $N = |\mathcal{D}|^\mu$, and fix a bijection to-int : $\mathcal{D}^\mu \to [N]$. For $\vec{f} \in \mathbb{F}^N$, its low-degree extension over $\mathcal{D}^\mu$ is defined as $\mathsf{LDE}_{\mathcal{D}}[\vec{f}](\vec{X}) := \sum_{\vec{b} \in \mathcal{D}^\mu} f(\text{to-int}(\vec{b}))\lambda_{\vec{b}}(\vec{X})$. It is the unique polynomial with individual degree at most $|\mathcal{D}| - 1$ that agrees with $\vec{f}$ on $\mathcal{D}^\mu$.

Multilinear extensions. The multilinear extension is the special case $\mathcal{D} = \{0, 1\}$. For $N = 2^\mu$ and $\vec{f} \in \mathbb{F}^N$, define $\mathsf{MLE}[\vec{f}](\vec{X}) := \sum_{\vec{b} \in \{0,1\}^\mu} f(\text{to-int}(\vec{b}))\mathsf{eq}(\vec{b}, \vec{X})$, where $\mathsf{eq}(\vec{b}, \vec{X}) := \prod_{i=0}^{\mu-1} \big((1 - b_i)(1 - X_i) + b_i X_i\big)$. Thus, $\mathsf{eq}(\vec{b}, \vec{X})$ is the tensor-product Lagrange basis polynomial over the Boolean hypercube.

Schwartz-Zippel lemma. We recall the standard Schwartz-Zippel lemma [26,30], which bounds the probability that a nonzero low-degree polynomial vanishes at a uniformly random point.

Lemma 1 (Schwartz-Zippel). *Let $\mathbb{F}$ be a field, let $f \in \mathbb{F}[X_1, \ldots, X_n]$ be a nonzero polynomial of total degree at most D, and let $S \subseteq \mathbb{F}$ be finite. If $\vec{r} \leftarrow S^n$ is sampled uniformly at random, then $\Pr[f(\vec{r}) = 0] \le \frac{D}{|S|}$. In particular, when $S = \mathbb{F}$, we have $\Pr[f(\vec{r}) = 0] \le D/|\mathbb{F}|$.*

Bi-variate sum-check. We recall the two-stage variant sum-check reduction.

Lemma 2 (Bi-variate sum-check). *Let μ be even, and $f \in \mathbb{F}[X_0, \ldots, X_{\mu-1}]$ have individual degree at most d. For the claim $s = \sum_{\vec{b}\in\{0,1\}^\mu} f(\vec{b})$, the following protocol has soundness error at most $\mu d/|\mathbb{F}|$.*

1. *The prover sends $f_0(\vec{X}) := \sum_{\vec{b}\in\{0,1\}^{\mu/2}} f(\vec{X}, \vec{b})$.*
2. *The verifier checks $\sum_{\vec{a}\in\{0,1\}^{\mu/2}} f_0(\vec{a}) \stackrel{?}{=} s$ and samples $\vec{z}_0 \leftarrow\$ \mathbb{F}^{\mu/2}$.*
3. *The prover sends $f_1(\vec{X}) := f(\vec{z}_0, \vec{X})$.*
4. *The verifier checks $\sum_{\vec{b}\in\{0,1\}^{\mu/2}} f_1(\vec{b}) \stackrel{?}{=} f_0(\vec{z}_0)$ and samples $\vec{z}_1 \leftarrow\$ \mathbb{F}^{\mu/2}$.*
5. *The prover proves the terminal consistency claim $f_1(\vec{z}_1) = f(\vec{z}_0, \vec{z}_1)$.*

2.4 Challenge Sets and Extraction Lemma

A set $\mathcal{C} \subseteq \mathcal{R}_q$ is a challenge space if $c - c'$ is invertible in $\mathcal{R}_q$ for every distinct $c, c' \in \mathcal{C}$. We also require public bounds $\tau, T \in \mathbb{Z}$ such that $\|c\| \le \tau$ and $\|c\|_{\mathsf{op},\infty} \le T$ for all $c \in \mathcal{C}$, where $\|c\|_{\mathsf{op},\infty} := \sup_{\boldsymbol{v}\in\mathcal{R}_q\setminus\{0\}} \|c\boldsymbol{v}\|_\infty / \|\boldsymbol{v}\|_\infty$. In our instantiations, $\mathcal{C}$ is chosen so that every nonzero difference $c - c'$ lies in the invertible regime of Lemma 3.

Lemma 3 (Invertibility in polynomial rings [23]). *Let $d > 1$ be a power of two, and let q be a prime with $q \equiv 5 \bmod 8$. Then every $\boldsymbol{f} \in \mathcal{R}_q$ satisfying $0 < \|\boldsymbol{f}\|_2 < \sqrt{q/2}$ is invertible in $\mathcal{R}_q$.*

We next recall the notation used by the rewinding extractor. For $\vec{x}, \vec{y} \in S^\ell$ and $i \in [\ell]$, write $\vec{x} \equiv_i \vec{y}$ if $\vec{x}$ and $\vec{y}$ differ only in the i-th coordinate, namely $x_i \ne y_i$ and $x_j = y_j$ for all $j \in [\ell] \setminus \{i\}$. For integers $\ell, k \ge 1$, let $K := \ell(k-1)+1$. Define $\mathsf{SS}(S, \ell, k)$ as the set of all tuples $(\vec{x}_1, \ldots, \vec{x}_K) \in (S^\ell)^K$ for which there exists a central index $e \in [K]$ such that for every $i \in [\ell]$, there is a set $J_i \subseteq [K] \setminus \{e\}$ with $|J_i| = k-1$ and $\vec{x}_e \equiv_i \vec{x}_j$ for every $j \in J_i$. Equivalently, for each coordinate i, the tuple contains k vectors that vary only in the i-th coordinate.

Lemma 4 (Rewinding extractor [10, Lemma 7.1]). *Fix $\mu \in \mathbb{N}$, let $\widetilde{S}$ be a finite set, $U := \widetilde{S}^\mu$. Let $\Psi : U \times T \to \{0,1\}$ be any predicate, let $\mathcal{A} : U \to T$ be a probabilistic algorithm, and define $\epsilon_\Psi(\mathcal{A}) := \Pr_{\mathbf{u}\stackrel{\$}{\leftarrow}U}[\Psi(\mathbf{u}, \mathcal{A}(\mathbf{u})) = 1]$. Then there exists an oracle algorithm $\mathcal{E}$ such that, on input $\mathbf{u}_0 \stackrel{\$}{\leftarrow} U$ and $y_0 \leftarrow \mathcal{A}(\mathbf{u}_0)$, the algorithm $\mathcal{E}^{\mathcal{A}}(\mathbf{u}_0, y_0)$ outputs, with probability at least $\epsilon_\Psi(\mathcal{A}) - \mu/|\widetilde{S}|$, accepting pairs $(\mathbf{u}_0, y_0), (\mathbf{u}_1, y_1), \ldots, (\mathbf{u}_\mu, y_\mu)$ such that $y_i \leftarrow \mathcal{A}(\mathbf{u}_i)$, $\Psi(\mathbf{u}_i, y_i) = 1$ for all $i \in \{0, \ldots, \mu\}$, and $\mathbf{u}_{i+1} \equiv_i \mathbf{u}_0$ for all $i \in [\mu]$. Moreover, $\mathcal{E}$ makes $1 + \mu$ oracle calls to $\mathcal{A}$ in expectation.*

3 Problem Formulation and Target Interface

In this section, we formalize the norm-sensitive statements considered in this work and the backend-native algebraic interface produced by our reduction.

Problem formulation. We consider lattice statements in which a committed witness must satisfy an exact norm bound, as captured by the following relation:

$$\mathcal{R}_{\mathsf{norm,na}} = \{\mathbf{x} = (\mathsf{cm}_s, B),\ \mathbf{w} = \vec{\boldsymbol{s}} \mid \mathsf{Com}(\vec{\boldsymbol{s}}) = \mathsf{cm}_s,\ \|\vec{\boldsymbol{s}}\|_2 \le B\}, \tag{1}$$

where Com is a lattice commitment scheme and $\vec{\boldsymbol{s}} \in \mathcal{R}_q^n$ defines the witness.

The difficulty is that the bound $\|\vec{\boldsymbol{s}}\|_2 \le B$ is a semantic condition on the integer coefficients of the witness, not a backend-native algebraic constraint. Existing lattice proof backends are typically optimized for commitment-opening constraints, linear identities, and structured inner-product claims. Thus, before $\mathcal{R}_{\mathsf{norm,na}}$ can be proved by such a backend, the exact norm condition must be compiled into explicit algebraic constraints. Our goal is to perform this compilation while preserving the exact norm language, and to output a relation class that can be efficiently handled by existing lattice-based backends.

Backend relation class. We now define the backend-native relation produced by our reduction as

$$\mathcal{R}_{\mathsf{back}} = \left\{ \begin{array}{l} \mathbf{x} = \big((\mathsf{cm}_i)_{i\in[\mu]}, \\ (\vec{t}_j, v_j, \sigma(j))_{j\in[\nu]}, B_\infty\big), \\ \mathbf{w} = \big((\vec{\boldsymbol{w}}_i)_{i\in[\mu]}\big) \end{array} \middle| \begin{array}{l} \mathsf{Com}(\vec{\boldsymbol{w}}_i) = \mathsf{cm}_i\ \forall i \in [\mu], \\ \langle \mathsf{coef}(\vec{\boldsymbol{w}}_{\sigma(j)}), \vec{t}_j \rangle = v_j\ \forall j \in [\nu], \\ \|\vec{\boldsymbol{w}}_i\|_\infty \le B_\infty,\ \forall i \in [\mu]. \end{array} \right\}. \tag{2}$$

Here, $\sigma : [\nu] \to [\mu]$ specifies which committed witness participates in each inner-product constraint, and each public vector $\vec{t}_j$ is assumed to admit a succinct tensor description.

We stress that the shortness condition in $\mathcal{R}_{\mathsf{back}}$ is not intended to encode the exact norm semantics of the original statement. The exact norm condition has already been compiled by the preceding reductions into explicit algebraic constraints. The shortness bound in $\mathcal{R}_{\mathsf{back}}$ is only an admissibility condition: it ensures that all committed openings remain within the parameter regime required for the underlying lattice binding assumption.

Accordingly, it suffices to instantiate the backend with a proof system whose extractor may incur slack in the shortness bound, as long as the extracted witnesses satisfy the following relation:

$$\mathcal{R}^*_{\mathsf{back}} = \left\{ \begin{array}{l} \mathbf{x} = \big((\mathsf{cm}_i)_{i\in[\mu]}, \\ (\vec{\boldsymbol{t}}_j, \boldsymbol{v}_j, \sigma(j))_{j\in[\nu]}, B_\infty, \eta, \tau\big), \\ \mathbf{w} = \big((\vec{\boldsymbol{w}}_i, c_i)_{i\in[\mu]}\big) \end{array} \middle| \begin{array}{l} \mathsf{Com}(\vec{\boldsymbol{w}}_i) = \mathsf{cm}_i\ \forall i \in [\mu], \\ \langle \vec{\boldsymbol{w}}_{\sigma(j)}, \vec{\boldsymbol{t}}_j \rangle = \boldsymbol{v}_j\ \forall j \in [\nu], \\ \|c_i \vec{\boldsymbol{w}}_i\|_\infty \le \eta \cdot B_\infty, \\ \|c_i\|_1 \le \tau,\ \forall i \in [\mu]. \end{array} \right\}. \tag{3}$$

Backend instantiation requirements. We now state the requirements on a downstream backend used to instantiate the reduction. At a high level, any backend that efficiently proves instances of $\mathcal{R}_{\mathsf{back}}$ can be used.

Completeness and relaxed knowledge soundness. The backend must be complete for $\mathcal{R}_{\mathsf{back}}$. That is, for every instance and witness satisfying (2), the backend prover produces an accepting proof. For soundness, we allow relaxed shortness extraction. Given any accepting proof for an instance of $\mathcal{R}_{\mathsf{back}}$, the backend extractor must output witnesses $(\widetilde{\vec{\boldsymbol{w}}}_i, \widetilde{c}_i)_{i\in[\mu]}$ satisfying the relaxed relation $\mathcal{R}^*_{\mathsf{back}}$. Equivalently, the extracted witnesses must open the public commitments, satisfy all witness-public inner-product constraints, and obey the relaxed admissible shortness bounds $\|\widetilde{c}_i\widetilde{\vec{\boldsymbol{w}}}_i\|_\infty \le \eta B_\infty$ and $\|\widetilde{c}_i\|_1 \le \tau$ for all $i \in [\mu]$. Thus, slack is allowed only in the shortness condition, not in the algebraic constraints.

Batching support. Batching is not required for the abstract reduction, but is desirable for efficient instantiations. The reduction outputs a uniform family of same-type constraints: commitment-opening checks and structured witness-public inner-product relations. A backend with efficient batching can prove the whole reduced instance in one backend workflow, rather than invoking the backend separately for each constraint. This preserves the intended backend-native interface and avoids repeated per-instance overhead.

Several existing lattice proof systems already provide suitable backend candidates for this interface, including Greyhound [25], SLAP [2], the construction of [9], and related systems [10,1,7,27,8] supporting commitment and structured inner-product relations.

4 The Exact Norm Reduction Framework

This section presents the main reduction framework. Starting from the exact norm relation $\mathcal{R}_{\mathsf{norm,na}}$, we reduce it to the backend relation $\mathcal{R}_{\mathsf{back}}$ defined above.

The reduction proceeds in three stages: algebraize the norm constraint into explicit algebraic conditions, compile the resulting constraints into a preliminary batch of backend-native structured claims, and then batch this instance into a smaller relation of the same backend type.

4.1 Π_{alg}: From $\mathcal{R}_{\mathsf{norm,na}}$ to $\mathcal{R}_{\mathsf{norm,al}}$

We begin by algebraizing the non-algebraic norm condition in $\mathcal{R}_{\mathsf{norm,na}}$.

Let $\vec{\boldsymbol{s}} \in \mathcal{R}_q^n$, and let $\vec{s} \in \mathbb{Z}_q^{nd}$ denote its coefficient embedding under canonical integer representatives. Using the involution σ_{-1} and the constant-term map ct, the squared coefficient norm is represented by $\|\vec{s}\|_2^2 = \langle \vec{s}, \vec{s}\rangle = \mathsf{ct}(\langle \vec{\boldsymbol{s}}, \sigma_{-1}(\vec{\boldsymbol{s}})\rangle)$ over the integers. Hence the exact norm condition $\|\vec{s}\|_2 \le B$ can be checked through the quadratic relation $\mathsf{ct}(\langle \vec{\boldsymbol{s}}, \sigma_{-1}(\vec{\boldsymbol{s}})\rangle) \le B^2$, provided that the modular computation over $\mathcal{R}_q$ coincides with its integer interpretation.

To ensure this, we additionally enforce a coefficient-wise bound $\|\vec{s}\|_\infty \le B_\infty$ with $nd \cdot B_\infty^2 < q$. Under this condition, we have $\|\vec{s}\|_2^2 = \sum_{i=1}^{nd} s_i^2 \le nd \cdot B_\infty^2 < q$, and therefore no modular wraparound can occur.

Consequently, proving $\mathsf{ct}(\langle \vec{s}, \sigma_{-1}(\vec{s}) \rangle) \bmod q \leq B^2$ together with $\|\vec{s}\|_\infty \leq B_\infty$ is equivalent to proving the intended exact integer statement $\|\vec{s}\|_2 \leq B$. It remains to express the coefficient-wise bound algebraically, we do so by first treating the binary case and then handling general ranges via binary decomposition.

Case 1: $B_\infty = 1$. Under the non-centered representation, the constraint $\|\vec{s}\|_\infty \leq 1$ is equivalent to the Boolean constraint $\vec{s} \circ (\vec{s} - \vec{1}) = \vec{0}$. To check this compactly, the verifier samples $r \in \mathbb{Z}_q$ and defines $\vec{r} = (1, r, \ldots, r^{nd-1})$. Then the Boolean constraint is reduced to the randomized quadratic check $\langle \vec{s} \circ \vec{r}, \vec{s} \rangle = \langle \vec{s}, \vec{r} \rangle$.

Let $\vec{s}_r \coloneqq \vec{s} \circ \vec{r}$, and let $\vec{\boldsymbol{s}}_r \in \mathcal{R}_q^n$ be its ring lifting. Since $\vec{\boldsymbol{s}}_r$ may not be short enough to commit directly, the prover decomposes it in base b as $\vec{\boldsymbol{s}}_r = \sum_{j=0}^{\ell-1} b^j \vec{\boldsymbol{s}}_{r,j}$, where each piece $\vec{\boldsymbol{s}}_{r,j}$ is short and committed separately.

To certify that this auxiliary witness is correctly formed, the verifier samples $\alpha \in \mathbb{Z}_q$, with coefficient vector $\vec{\alpha}$, and checks the linear consistency relation $\langle \vec{s}_r, \vec{\alpha} \rangle = \langle \vec{s}, \vec{\alpha} \circ \vec{r} \rangle$. Thus, in the binary case, the exact ℓ_∞ constraint is reduced to: (i). a quadratic relation linking $\vec{\boldsymbol{s}}_r$ and $\vec{\boldsymbol{s}}$, and (ii). a collection of linear consistency checks certifying that $\vec{\boldsymbol{s}}_r$ is correctly formed.

Case 2: $B_\infty = 2^m - 1$. For the general unsigned range, each coefficient of $\vec{\boldsymbol{s}}$ must lie in $\{0, \ldots, 2^m - 1\}$. The prover writes $\vec{\boldsymbol{s}} = \sum_{i=0}^{m-1} 2^i \vec{\boldsymbol{s}}_i$ where each slice $\vec{\boldsymbol{s}}_i$ is intended to be binary, and commits to all slices. Correctness of the decomposition is checked by a randomized linear test: for a verifier challenge γ with coefficient vector $\vec{\gamma}$, the protocol requires $\left\langle \sum_{i=0}^{m-1} 2^i \vec{s}_i, \vec{\gamma} \right\rangle = \langle \vec{s}, \vec{\gamma} \rangle$. It remains only to prove that every slice $\vec{\boldsymbol{s}}_i$ is binary, which is done by applying the binary-check procedure from Case 1 to each slice. Thus, the range bound $\|\vec{\boldsymbol{s}}\|_\infty \leq 2^m - 1$ is reduced to binary subrelations, each consisting of one quadratic constraint and structured linear consistency checks.

Relaxed shortness constraints. The algebraization step introduces auxiliary witnesses only to encode the exact norm semantics of the original witness $\vec{\boldsymbol{s}}$. These witnesses are not part of the original statement. Once the exact norm condition has been reduced to quadratic and linear algebraic constraints, the reduced relation no longer needs to impose exact norm bounds on all committed witnesses. It suffices to require relaxed shortness bounds ensuring that the original and auxiliary openings remain in the admissible parameter regime for the underlying lattice hardness assumptions.

We now define the algebraized relation output by Π_{alg}. For each $i \in [m]$, let $\vec{r}_i = (1, r_i, \ldots, r_i^{nd-1}) \in \mathbb{Z}_q^{nd}$, and $\vec{\alpha}_i = (1, \alpha_i, \ldots, \alpha_i^{nd-1})$. We write $\vec{\boldsymbol{s}}_{r_i} \coloneqq \sum_{j=0}^{\ell-1} b^j \vec{\boldsymbol{s}}_{r_i,j}$ for the lifted auxiliary witness associated with the i-th binary slice.

We define the reduced relation $\mathcal{R}_{\mathsf{norm,al}}$ as follows:

$$\mathcal{R}_{\mathsf{norm,al}} = \left\{ (\mathbf{x}, \mathbf{w}) \;\middle|\; \begin{array}{l} \textbf{Commitment consistency:} \\ \quad \mathsf{Com}(\vec{\boldsymbol{s}}) = \mathsf{cm}_s, \mathsf{Com}(\vec{\boldsymbol{s}}_i) = \mathsf{cm}_{s,i}, \forall i \in [m], \\ \quad \mathsf{Com}(\vec{\boldsymbol{s}}_{r_i,j}) = \mathsf{cm}_{r,i,j} \quad \forall (i,j) \in [m] \times [\ell], \\ \textbf{Quadratic } \ell_2 \textbf{ constraint:} \\ \quad \mathsf{ct}\big(\langle \vec{\boldsymbol{s}}, \sigma_{-1}(\vec{\boldsymbol{s}}) \rangle\big) \bmod q = v_s \leq B^2, \\ \textbf{Decomposition consistency} : \\ \quad \langle \mathsf{coef}(\vec{\boldsymbol{s}}_i), \vec{\gamma} \rangle = v_{\mathsf{dec},i},\ \langle \mathsf{coef}(\vec{\boldsymbol{s}}), \vec{\gamma} \rangle = v_{\mathsf{dec}}, \\ \quad \sum_{i=0}^{m-1} 2^i \cdot v_{\mathsf{dec},i} = v_{\mathsf{dec}}. \\ \textbf{Correctness of binary checks: } (\forall i \in [m] :) \\ \quad \mathsf{ct}\Big(\Big\langle \sum_{j=0}^{\ell-1} b^j \vec{\boldsymbol{s}}_{r_i,j}, \sigma_{-1}(\vec{\boldsymbol{s}}_i) \Big\rangle\Big) = \langle \mathsf{coef}(\vec{\boldsymbol{s}}_i), \vec{r}_i \rangle = v_{r,i}, \\ \quad \langle \mathsf{coef}(\vec{\boldsymbol{s}}_{r_i,j}), \vec{\alpha}_i \rangle = v_{\alpha,i,j},\ \langle \mathsf{coef}(\vec{\boldsymbol{s}}_i), \vec{\alpha}_i \circ \vec{r}_i \rangle = v_{\alpha,i}, \\ \quad \sum_{j=0}^{\ell-1} b^j \cdot v_{\alpha,i,j} = v_{\alpha,i}. \\ \textbf{Shortness of witnesses:} \\ \quad \|\vec{\boldsymbol{s}}\|_\infty, \|\vec{\boldsymbol{s}}_i\|_\infty, \|\vec{\boldsymbol{s}}_{r_i,j}\|_\infty \leq b_{w,\infty}, \forall i \in [m], j \in [\ell]. \end{array} \right\}, \tag{4}$$

where

$$\begin{aligned} \mathbf{x} = \Big(& \mathbf{x}_{\mathsf{pp}},\ \mathsf{cm}_s, \vec{\gamma},\ v_s,\ v_{\mathsf{dec}}, v_{\mathsf{dec},i}, (\mathsf{cm}_{s,i}, (\mathsf{cm}_{r,i,j})_{j\in[\ell]})_{i\in[m]} \\ & (\vec{r}_i, \vec{\alpha}_i, v_{r,i}, v_{\alpha,i}, (v_{\alpha,i,j})_{j\in[\ell]})_{i\in[m]} \Big), \\ \mathbf{w} = \Big(& \vec{\boldsymbol{s}},\ (\vec{\boldsymbol{s}}_i,\ (\vec{\boldsymbol{s}}_{r_i,j})_{j\in[\ell]})_{i\in[m]} \Big), \end{aligned}$$

for $\mathbf{x}_{\mathsf{pp}} = \big(B, B_\infty, b, m, \ell, b_{w,\infty} = \max(\frac{b}{2}, B_\infty)\big)$. The protocol Π_{alg} implementing this reduction is given in Figure 1.

Theorem 1. *The protocol Π_{alg} in Figure 1 is a reduction of knowledge from $\mathcal{R}_{\mathsf{norm,na}}$ to $\mathcal{R}_{\mathsf{norm,al}}$. More precisely, it is publicly reducible, perfectly complete, and knowledge sound with soundness error at most $\varepsilon_{\mathsf{alg}} \leq \frac{(1+2m)nd}{q}$.*

Proof. We prove the three properties separately.

Public reducibility. The output statement is determined by the input instance and the public transcript. Indeed, given $\mathbf{x}_{\mathsf{in}}$ and a transcript tr containing the public challenges, commitments, and values generated in Π_{alg}, the verifier can compute the corresponding output instance $\mathbf{x}_{\mathsf{out}}$ for $\mathcal{R}_{\mathsf{norm,al}}$ without using the witness. Hence the reduction is publicly reducible.

Perfect completeness. Let $(\mathbf{x}_{\mathsf{in}}, \mathbf{w}_{\mathsf{in}}) \in \mathcal{R}_{\mathsf{norm,na}}$. An honest prover constructs $\mathbf{w}_{\mathsf{out}}$ by keeping the committed witness $\vec{\boldsymbol{s}}$ and adding the prescribed decomposition and auxiliary witnesses. The identity $\|\vec{s}\|_2^2 = \mathsf{ct}(\langle \vec{\boldsymbol{s}}, \sigma_{-1}(\vec{\boldsymbol{s}}) \rangle)$ holds over the integers, and the imposed coefficient-wise bound rules out wraparound modulo q. The decomposition and binary constraints are satisfied by construction. Thus $(\mathbf{x}_{\mathsf{out}}, \mathbf{w}_{\mathsf{out}}) \in \mathcal{R}_{\mathsf{norm,al}}$, so completeness is perfect.

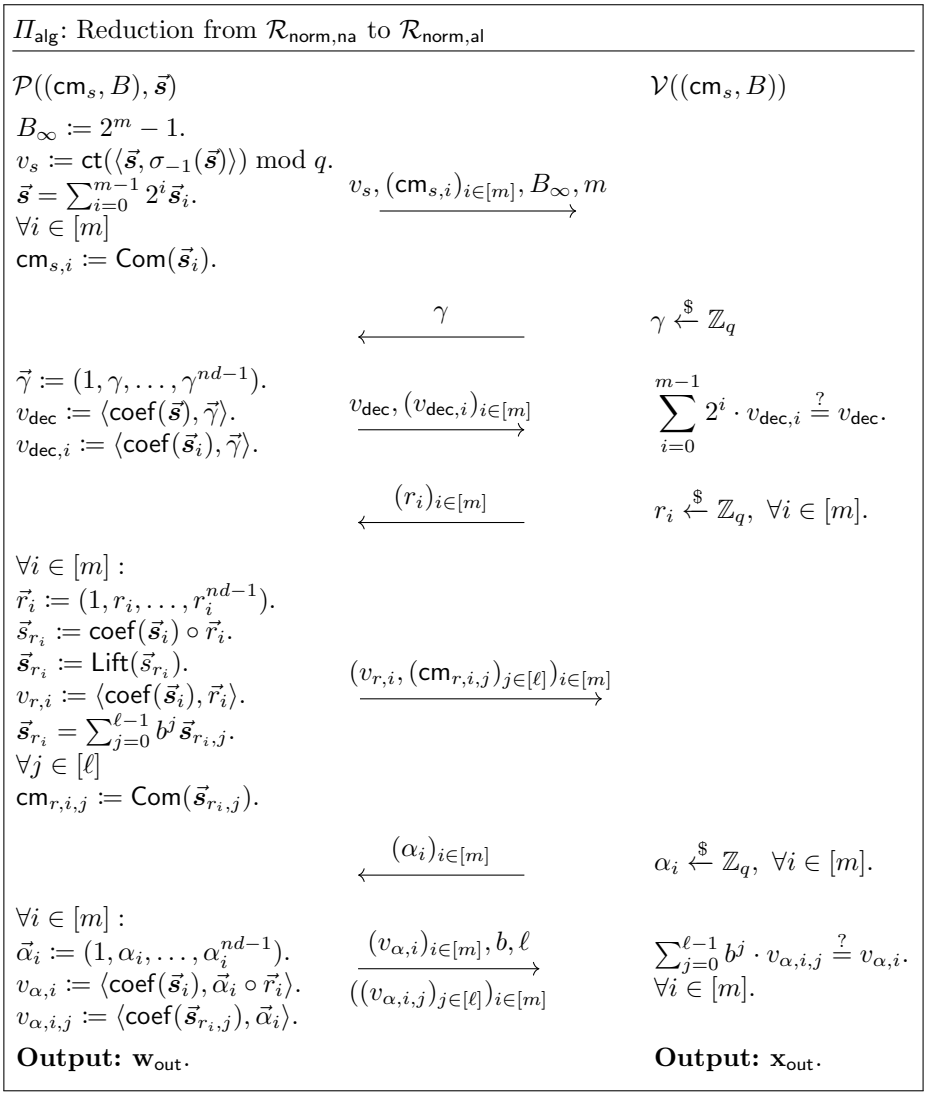


Fig. 1. Π_{alg}: reduction from $\mathcal{R}_{\mathsf{norm,na}}$ to $\mathcal{R}_{\mathsf{norm,al}}$.

Knowledge soundness. Let a prover output an accepting pair $(\mathbf{x}_{\mathsf{out}}, \mathbf{w}_{\mathsf{out}}) \in \mathcal{R}_{\mathsf{norm,al}}$ for an output statement of Π_{alg}. The extractor $\mathcal{E}_{\mathsf{alg}}$ parses $\mathbf{w}_{\mathsf{out}}$ and outputs the committed vector $\vec{s}$ as the witness for $\mathcal{R}_{\mathsf{norm,na}}$. By the commitment constraint, $\mathsf{Com}(\vec{s}) = \mathsf{cm}_s$. It remains to show $\|\vec{s}\|_2 \le B$.

The algebraic norm constraint gives $\langle \vec{s}, \vec{s} \rangle \equiv v_s \pmod q$ with $v_s \le B^2$. The relation also contains slices $\vec{s}_0, \ldots, \vec{s}_{m-1}$ and checks the decomposition $\vec{s} = \sum_{i=0}^{m-1} 2^i \vec{s}_i$ by a randomized linear test. An invalid decomposition is accepted with probability at most nd/q. For each slice $\vec{s}_i$, the relation checks binaryness

through the randomized condition corresponding to $\vec{s}_i \circ (\vec{s}_i - \vec{1}) = \vec{0}$, together with the auxiliary consistency check. For each i, these checks fail to detect an invalid binary slice with probability at most $2nd/q$. By a union bound, the total probability that the extracted decomposition is invalid is at most $(1+2m)nd/q$. Conditioned on the complement of this event, the extracted $\vec{\boldsymbol{s}}$ has a valid binary decomposition and therefore satisfies the required coefficient-wise bound.

By the no-wraparound condition, $\langle \vec{s}, \vec{s} \rangle \equiv v_s \pmod q$ with $v_s \leq B^2$ then implies the integer inequality $\|\vec{s}\|_2^2 = \langle \vec{s}, \vec{s} \rangle \leq B^2$. Hence $(\mathbf{x}_{\mathsf{in}}, \vec{\boldsymbol{s}}) \in \mathcal{R}_{\mathsf{norm,na}}$ except with probability at most $\frac{(1+2m)nd}{q}$. This proves knowledge soundness. □

Efficiency. Assume that B_∞, m, and ℓ are fixed small constants by the security parameter λ, and that each commitment consists of $\kappa = O(1)$ ring elements. Then the prover-side communication of Π_{alg} is

$$\begin{aligned}\mathsf{Comm}_{\mathcal{P}}(\Pi_{\mathsf{alg}}) = & \underbrace{\left(3m + m\ell + 2 + \frac{1}{\ell}\right)\log q}_{|v_s|+|v_{\mathsf{dec}}|+\sum_{i\in[m]}(|v_{\mathsf{dec},i}|+|v_{r,i}|+|v_{\alpha,i}|)+\sum_{i,j\in[m\times\ell]}|v_{\alpha,i,j}|+|b|} \\ & + \underbrace{(m\kappa + m\ell\kappa)\cdot|\mathcal{R}_q|}_{\sum_{i\in[m]}|\mathsf{cm}_{s,i}|+\sum_{i,j\in[m\times\ell]}|\mathsf{cm}_{r,i,j}|} + \underbrace{O(1)}_{B_\infty,m,\ell} \\ & = O(|\mathcal{R}_q| + \log q)\text{bits},\end{aligned}$$

Thus, the communication is independent of the witness length nd in the number of ring and field elements. The prover performs $O(nd)$ operations over $\mathbb{Z}_q$, dominated by the construction of auxiliary witnesses and the computation of the corresponding inner-product claims. The verifier samples a constant number of challenges, derives tensor-structured public vectors of description length $O(\log(nd))$, and checks a constant number of scalar consistency equations. Thus, the verifier work is $O(\log(nd))$ field operations.

Remark 1. Unless stated otherwise, coefficient bounds in this work are interpreted under the canonical non-centered embedding $\mathbb{Z}_q \to \{0, 1, \ldots, q-1\}$. For centered witnesses, we use a public offset representation. Namely, for $s_k \in [-B_\infty, B_\infty]$, set $a_k = s_k + B_\infty \in [0, 2B_\infty]$ and apply the unsigned range proof to a_k. The squared norm of the original centered witness is then enforced through

$$\sum_{k=0}^{nd-1} s_k^2 = \sum_{k=0}^{nd-1} a_k^2 - 2B_\infty \sum_{k=0}^{nd-1} a_k + ndB_\infty^2.$$

Thus, the same unsigned range-checking machinery supports centered bounded witnesses after a deterministic offset transformation.

4.2 Π_{comp}: From $\mathcal{R}_{\mathsf{norm,al}}$ to $\mathcal{R}_{\mathsf{back}}^{(p)}$

We next compile the quadratic constraints in $\mathcal{R}_{\mathsf{norm,al}}$ into backend-native structured linear constraints. The reduction follows a bi-variate sumcheck template: a

quadratic relation is written as a sum of evaluations of a low-degree polynomial, and the resulting claim is reduced to a bounded number of structured linear consistency checks.

We illustrate the idea for the representative constraint $\langle \vec{s}, \vec{s} \rangle \equiv v_s \pmod{q}$, where $\vec{s} \in \mathbb{Z}_q^N$ and $N = nd$. Assume for simplicity that $N = 2^{2t}$, and let $\widetilde{s}(\vec{X}_1, \vec{X}_2)$ be the multilinear extension of $\vec{s}$ over $\{0,1\}^{2t}$. Then the quadratic constraint is equivalent to $\sum_{(\vec{b}_1, \vec{b}_2) \in \{0,1\}^{2t}} \widetilde{s}(\vec{b}_1, \vec{b}_2)^2 = v_s \pmod{q}$. Let $g_s(\vec{X}_1, \vec{X}_2) \coloneqq \widetilde{s}(\vec{X}_1, \vec{X}_2)^2$. Since $\widetilde{s}$ is multilinear, g_s has individual degree at most 2. We now reduce this sum claim using a two-stage sumcheck-style procedure.

First stage. The first stage sums out the second block of Boolean variables. Define $h_0(\vec{X}_1) \coloneqq \sum_{\vec{b} \in \{0,1\}^t} g_s(\vec{X}_1, \vec{b})$. Since g_s has individual degree at most 2, so does h_0, therefore can be uniquely determined by its evaluations on any product grid of size 3^t. Let $\lambda_{-1}, \lambda_0, \lambda_1$ be the univariate degree-2 Lagrange basis polynomials over $\{0, \pm 1\}$:

$$\lambda_0(X) = (1+X)(1-X), \quad \lambda_1(X) = \frac{1}{2}X(X+1), \quad \lambda_{-1}(X) = \frac{1}{2}X(X-1).$$

They satisfy $\lambda_u(v) = 1[u = v]$ for all $u, v \in \{0, \pm 1\}$. And define $\widetilde{\ell}(\vec{X}, \vec{u}) \coloneqq \prod_{k=0}^{t-1} \lambda_{u_k}(X_k)$ for $\vec{u} \in \{0, \pm 1\}^t$. Then h_0 has the unique interpolation of the form $h_0(\vec{X}_1) = \sum_{\vec{u} \in \{0,\pm 1\}^t} h_0(\vec{u}) \widetilde{\ell}(\vec{X}_1, \vec{u})$.

We next encode the evaluation table of h_0 as a vector. To keep all intermediate vectors at the common length $N = 4^t$, we pad this table to the domain $\mathcal{D}^t$, where $\mathcal{D} \coloneqq \{0, 1, -1, \star\}$. Entries indexed by $\{0, \pm 1\}^t$ store the true evaluations of h_0, while entries containing $\star$ are set to 0. This gives a padded vector $\vec{h}_0 \in \mathbb{Z}_q^{4^t}$. The prover commits to $\vec{h}_0$ and sends the commitment to the verifier.

Let $\chi_{\mathsf{bool}} \coloneqq \bigotimes_t (1, 1, 0, 0)$ be the selector for the Boolean slice $\{0,1\}^t \subseteq \mathcal{D}^t$ under the ordering $(0, 1, -1, \star)$. The first-stage sum claim is then captured by the linear constraint

$$\left\langle \vec{h}_0, \chi_{\mathrm{bool}} \right\rangle = \sum_{\vec{b} \in \{0,1\}^t} h_0(\vec{b}) = v_s.$$

Second stage. The second stage handles the remaining block of variables. After receiving the verifier challenge $\vec{z}_1 \in \mathbb{Z}_q^t$, we fix the first block and define $h_1(\vec{X}_2) \coloneqq g_s(\vec{z}_1, \vec{X}_2)$.

As before, h_1 has individual degree at most 2, so we encode it by a padded evaluation vector $\vec{h}_1 \in \mathbb{Z}_q^{4^t}$ over $\mathcal{D}^t$, where $\mathcal{D} = \{0, 1, -1, \star\}$. The entries indexed by $\{0, \pm 1\}^t$ contain the true evaluations of h_1, and entries containing $\star$ are set to 0. Similarly, the commitment of $\vec{h}_1$ is sent to the verifier. By construction, $\sum_{\vec{b} \in \{0,1\}^t} h_1(\vec{b}) = h_0(\vec{z}_1)$. Thus, this stage is captured by the consistency check $\langle \vec{h}_1, \chi_{\mathsf{bool}} \rangle = \langle \vec{h}_0, \chi_{\vec{z}_1} \rangle$, where $\chi_{\vec{z}_1} \coloneqq \bigotimes_{k=0}^{t-1} (\lambda_0(z_{1,k}), \lambda_1(z_{1,k}), \lambda_{-1}(z_{1,k}), 0)$ is the tensor selector evaluating the padded table of h_0 at $\vec{z}_1$.

Consistency Check. It remains to check that the committed table $\vec{h}_1$ is consistent with the original witness $\vec{s}$. The verifier samples $\vec{z}_2 \in \mathbb{Z}_q^t$. By definition, we have $h_1(\vec{z}_2) = g_s(\vec{z}_1, \vec{z}_2) = \widetilde{s}(\vec{z}_1, \vec{z}_2)^2$.

For $\vec{z} \in \mathbb{Z}_q^t$, let $\widetilde{\mathsf{eq}}_{\vec{z}} \coloneqq \bigotimes_{i=0}^{t-1}(\widetilde{\mathsf{eq}}(0, z_i), \widetilde{\mathsf{eq}}(1, z_i))$. Then $\widetilde{s}(\vec{z}_1, \vec{z}_2) = \langle \vec{s}, \widetilde{\mathsf{eq}}_{\vec{z}_1} \otimes \widetilde{\mathsf{eq}}_{\vec{z}_2} \rangle$, while the padded h_1 is restricted by $h_1(\vec{z}_2) = \langle \vec{h}_1, \chi_{\vec{z}_2} \rangle$. Thus the terminal consistency condition is $\langle \vec{h}_1, \chi_{\vec{z}_2} \rangle = \left(\langle \vec{s}, \widetilde{\mathsf{eq}}_{\vec{z}_1} \otimes \widetilde{\mathsf{eq}}_{\vec{z}_2} \rangle \right)^2$.

This links $\vec{h}_1$ back to the original witness and completes the two-stage reduction. Figure 2 gives the complete reduction protocol for a committed quadratic inner-product constraint.

Theorem 2. *Let Π_{en} be a proof system for the constraints appearing in the "Engage to prove" part of Figure 2. Assume that Π_{en} is perfectly complete and knowledge-sound, with knowledge soundness error at most ϵ_{en}. Then the protocol in Figure 2 for the committed quadratic inner-product statement is perfectly complete and knowledge-sound with total soundness error at most $\epsilon_{\mathsf{quad}} \leq \epsilon_{\mathsf{en}} + \frac{4t}{q}$.*

Proof. We prove completeness and knowledge soundness separately.

Perfect completeness. Assume that the prover holds witnesses $\vec{\boldsymbol{s}}_1, \vec{\boldsymbol{s}}_2 \in \mathcal{R}_q^n$ such that $\mathsf{Com}(\vec{\boldsymbol{s}}_i) = \mathsf{cm}_{s,i}$ for $i \in \{1, 2\}$ and $\langle \vec{s}_1, \vec{s}_2 \rangle = v$, where $\vec{s}_i = \mathsf{coef}(\vec{\boldsymbol{s}}_i)$. The honest prover defines

$$g(\vec{X}_1, \vec{X}_2) = \widetilde{s}_1(\vec{X}_1, \vec{X}_2)\widetilde{s}_2(\vec{X}_1, \vec{X}_2), \qquad h_0(\vec{X}_1) = \sum_{\vec{b} \in \{0,1\}^t} g(\vec{X}_1, \vec{b}),$$

and, after receiving $\vec{z}_1$, $h_1(\vec{X}_2) = g(\vec{z}_1, \vec{X}_2)$.

By construction, $\sum_{\vec{a} \in \{0,1\}^t} h_0(\vec{a}) = \langle \vec{s}_1, \vec{s}_2 \rangle = v$, $\sum_{\vec{b} \in \{0,1\}^t} h_1(\vec{b}) = h_0(\vec{z}_1)$, and $h_1(\vec{z}_2) = \widetilde{s}_1(\vec{z}_1, \vec{z}_2)\widetilde{s}_2(\vec{z}_1, \vec{z}_2)$. Thus all constraints in the final engagement relation are satisfied by the honestly generated witnesses and claims. Perfect completeness follows from the perfect completeness of Π_{en}.

Knowledge soundness. Let $\mathcal{P}^*$ be any prover that makes the verifier accept. By the knowledge soundness of Π_{en}, except with probability at most ϵ_{en}, we extract openings $\vec{\boldsymbol{s}}_1, \vec{\boldsymbol{s}}_2$ and committed tables inducing degree-2 polynomials $\widehat{h}_0$ and $\widehat{h}_1$ such that all final linear constraints hold.

Define the honest target polynomials $H_0(\vec{X}_1) \coloneqq \sum_{\vec{b} \in \{0,1\}^t} \widetilde{s}_1(\vec{X}_1, \vec{b})\widetilde{s}_2(\vec{X}_1, \vec{b})$ and $H_1(\vec{X}_2) \coloneqq \widetilde{s}_1(\vec{z}_1, \vec{X}_2)\widetilde{s}_2(\vec{z}_1, \vec{X}_2)$. Since the commitment to $\widehat{h}_1$ is fixed before $\vec{z}_2$ is sampled, if $\widehat{h}_1 \neq H_1$, then $\widehat{h}_1(\vec{z}_2) = H_1(\vec{z}_2)$ occurs with probability at most $2t/q$ by Schwartz–Zippel. Thus, except with probability $2t/q$, the terminal consistency check implies $\widehat{h}_1 = H_1$.

Conditioned on this event, the extracted consistency relation gives

$$\widehat{h}_0(\vec{z}_1) = \sum_{\vec{b} \in \{0,1\}^t} \widehat{h}_1(\vec{b}) = \sum_{\vec{b} \in \{0,1\}^t} H_1(\vec{b}) = H_0(\vec{z}_1).$$

Private short input: $\vec{s}_1, \vec{s}_2 \in \mathcal{R}_q^n$.

Public input: $\mathsf{cm}_{s,1}, \mathsf{cm}_{s,2}, v,\ N = nd = 4^t$.

Claim: $\mathsf{cm}_{s,i} = \mathsf{Com}(\vec{s}_i), \quad i \in \{1,2\}, \qquad \langle \vec{s}_1, \vec{s}_2 \rangle = v, \qquad \vec{s}_i = \mathsf{coef}(\vec{s}_i)$.

$\mathcal{P}$		$\mathcal{V}$
$g(\vec{X}_1, \vec{X}_2) \coloneqq$ $\widetilde{s}_1(\vec{X}_1, \vec{X}_2)\widetilde{s}_2(\vec{X}_1, \vec{X}_2)$. $h_0(\vec{X}_1) \coloneqq \sum_{\vec{b} \in \{0,1\}^t} g(\vec{X}_1, \vec{b})$. $\vec{h}_0 \coloneqq (h_0(\vec{u}))_{\vec{u} \in \mathcal{D}^t}$, $\vec{\boldsymbol{h}}_0 \coloneqq \mathsf{Lift}(\vec{h}_0)$. $(\vec{\boldsymbol{h}}_{0,j})_{j \in [\ell]} \coloneqq \mathsf{Decom}_b(\vec{\boldsymbol{h}}_0)$. $\mathsf{cm}_{h_0,j} \coloneqq \mathsf{Com}(\vec{\boldsymbol{h}}_{0,j})$, $v_{h_0,j} \coloneqq \langle \mathsf{coef}(\vec{\boldsymbol{h}}_{0,j}), \chi_{\mathsf{bool}} \rangle$.	$\xrightarrow{(\mathsf{cm}_{h_0,j})_{j \in [\ell]}}$ $\xleftarrow{(v_{h_0,j})_{j \in [\ell]}}$	$\sum_{j=0}^{\ell-1} b^j v_{h_0,j} \stackrel{?}{=} v$.
	$\xleftarrow{\vec{z}_1}$	$\vec{z}_1 \xleftarrow{\$} \mathbb{Z}_q^t$
$v_{h_0,z,j} \coloneqq \langle \mathsf{coef}(\vec{\boldsymbol{h}}_{0,j}), \chi_{\vec{z}_1} \rangle$. $h_1(\vec{X}_2) \coloneqq g(\vec{z}_1, \vec{X}_2)$. $\vec{h}_1 \coloneqq (h_1(\vec{u}))_{\vec{u} \in \mathcal{D}^t}$, $\vec{\boldsymbol{h}}_1 \coloneqq \mathsf{Lift}(\vec{h}_1)$. $(\vec{\boldsymbol{h}}_{1,j})_{j \in [\ell]} \coloneqq \mathsf{Decom}_b(\vec{\boldsymbol{h}}_1)$. $\mathsf{cm}_{h_1,j} \coloneqq \mathsf{Com}(\vec{\boldsymbol{h}}_{1,j})$, $v_{h_1,j} \coloneqq \langle \mathsf{coef}(\vec{\boldsymbol{h}}_{1,j}), \chi_{\mathsf{bool}} \rangle$.	$\xrightarrow{(v_{h_0,z,j}, v_{h_1,j})_{j \in [\ell]},}$ $\xleftarrow{(\mathsf{cm}_{h_1,j})_{j \in [\ell]}}$	$\sum_{j=0}^{\ell-1} b^j v_{h_0,z,j} \stackrel{?}{=} \sum_{j=0}^{\ell-1} b^j v_{h_1,j}$
	$\xleftarrow{\vec{z}_2}$	$\vec{z}_2 \xleftarrow{\$} \mathbb{Z}_q^t$
$v_{h_1,z,j} \coloneqq \langle \mathsf{coef}(\vec{\boldsymbol{h}}_{1,j}), \chi_{\vec{z}_2} \rangle$. $u_i \coloneqq \widetilde{s}_i(\vec{z}_1, \vec{z}_2), \quad i \in \{1,2\}$.	$\xrightarrow{u_1, u_2}$ $\xleftarrow{(v_{h_1,z,j})_{j \in [\ell]}}$	$\sum_{j=0}^{\ell-1} b^j v_{h_1,z,j} \stackrel{?}{=} u_1 u_2$

Engage to prove:

$\vec{s}_1, \vec{s}_2, (\vec{\boldsymbol{h}}_{0,j}, \vec{\boldsymbol{h}}_{1,j})_{j \in [\ell]}$ satisfy the shortness bounds.

$\mathsf{Com}(\vec{s}_i) = \mathsf{cm}_{s,i}, \quad i \in \{1,2\}$.

$\mathsf{Com}(\vec{\boldsymbol{h}}_{0,j}) = \mathsf{cm}_{h_0,j}, \quad \mathsf{Com}(\vec{\boldsymbol{h}}_{1,j}) = \mathsf{cm}_{h_1,j}, \quad j \in [\ell]$.

$\langle \mathsf{coef}(\vec{s}_i), \widetilde{\mathsf{eq}}_{(\vec{z}_1, \vec{z}_2)} \rangle = u_i, \quad i \in \{1,2\}$.

$\langle \mathsf{coef}(\vec{\boldsymbol{h}}_{0,j}), \chi_{\mathsf{bool}} \rangle = v_{h_0,j}, \quad \langle \mathsf{coef}(\vec{\boldsymbol{h}}_{0,j}), \chi_{\vec{z}_1} \rangle = v_{h_0,z,j}$.

$\langle \mathsf{coef}(\vec{\boldsymbol{h}}_{1,j}), \chi_{\mathsf{bool}} \rangle = v_{h_1,j}, \quad \langle \mathsf{coef}(\vec{\boldsymbol{h}}_{1,j}), \chi_{\vec{z}_2} \rangle = v_{h_1,z,j}$.

Fig. 2. Reduction of a quadratic inner-product constraint over committed witnesses.

Since the commitment to $\widehat{h}_0$ is fixed before $\vec{z}_1$ is sampled, if $\widehat{h}_0 \neq H_0$, then $\widehat{h}_0(\vec{z}_1) = H_0(\vec{z}_1)$ occurs with probability at most $2t/q$. Therefore, except with probability at most $4t/q$, we have $\widehat{h}_0 = H_0$ and $\widehat{h}_1 = H_1$.

Using the extracted first-stage relation $\sum_{\vec{a}\in\{0,1\}^t} \widehat{h}_0(\vec{a}) = v$, we obtain

$$v = \sum_{\vec{a}\in\{0,1\}^t} H_0(\vec{a}) = \sum_{(\vec{a},\vec{b})\in\{0,1\}^{2t}} \widetilde{s}_1(\vec{a},\vec{b})\widetilde{s}_2(\vec{a},\vec{b}) = \langle \vec{s}_1, \vec{s}_2 \rangle.$$

The extracted witnesses therefore open the public commitments and satisfy the original quadratic inner-product claim. Combining this with the extraction error of Π_{en} gives $\epsilon_{\mathsf{quad}} \leq \epsilon_{\mathsf{en}} + 4t/q$. □

Efficiency of protocol in Figure 2. We analyze the asymptotic cost of reducing one quadratic inner-product claim to structured linear constraints.

Recall that $N = nd = 4^t$. The prover sends 2ℓ commitments to the intermediate table witnesses and $4\ell + 2$ field elements. If each commitment contains $\kappa = O(1)$ ring elements, the prover-side communication is $2\ell\kappa|\mathcal{R}_q| + (4\ell+2)\log q$ bits. Thus, for fixed small constants ℓ and κ, the communication is independent of N in the number of ring and field elements.

The prover time is dominated by constructing the table of h_0. Indeed, we have $h_0(\vec{X}_1) = \sum_{\vec{b}\in\{0,1\}^t} \widetilde{s}_1(\vec{X}_1, \vec{b})\widetilde{s}_2(\vec{X}_1, \vec{b})$ is evaluated over $\{0, 1, -1\}^t$. This requires 3^t evaluation points, each summing over 2^t Boolean points, and therefore costs $O(3^t \cdot 2^t) = O(6^t)$ field operations.

The table of h_1 can be computed more efficiently after $\vec{z}_1$ is fixed. For each $a \in \{1, 2\}$, the restricted table $S_{a,\vec{z}_1}(\vec{b}_2) = \sum_{\vec{b}_1\in\{0,1\}^t} s_a(\vec{b}_1, \vec{b}_2)\widetilde{\mathsf{eq}}(\vec{b}_1, \vec{z}_1)$ has length 2^t and can be computed in $O(4^t)$ time. Its multilinear extension is then expanded from $\{0, 1\}^t$ to $\{0, 1, -1\}^t$ using the identity $L(-1) = 2L(0) - L(1)$ for univariate linear polynomials. This costs $O(t3^t)$ additional operations, and the pointwise product forming $h_1 = S_{1,\vec{z}_1} \cdot S_{2,\vec{z}_1}$ costs $O(3^t)$. Hence h_1 costs $O(4^t + t3^t) = O(4^t)$.

Therefore, the prover time for one quadratic constraint is dominated by $O(6^t) = O(N^{\log_4 6}) \approx O(N^{1.292})$.

On the verifier side, the verifier samples $\vec{z}_1, \vec{z}_2 \in \mathbb{Z}_q^t$ and derives tensor descriptions of $\widetilde{\mathsf{eq}}_{(\vec{z}_1,\vec{z}_2)}$, $\chi_{\vec{z}_1}$, and $\chi_{\vec{z}_2}$. These descriptions have size $O(t) = O(\log N)$. The remaining checks are scalar recombination checks involving ℓ terms, and hence are constant-time for fixed small constant $\ell = O(1)$. Thus, the verifier work is $O(\log N)$ field operations.

Applying this compilation to every quadratic constraint in $\mathcal{R}_{\mathsf{norm,al}}$ removes all quadratic checks. The resulting preliminary backend relation, denoted $\mathcal{R}^{(p)}_{\mathsf{back}}$, consists only of committed short witnesses and witness-public inner-product claims against tensor-structured public vectors.

The full reduction protocol from $\mathcal{R}_{\mathsf{norm,al}}$ to $\mathcal{R}^{(p)}_{\mathsf{back}}$ is given in Figure 3. Its output is a pair $(\mathbf{x}_{\mathsf{out}}, \mathbf{w}_{\mathsf{out}})$, where the public instance is

$$\begin{aligned}\mathbf{x}_{\mathsf{out}} := \Big(&\mathbf{x}_{\mathsf{pp}}, \mathsf{cm}_s, \vec{\gamma}, v_{\mathsf{dec}}, (v_{\mathsf{dec},i})_{i\in[m]}, \widetilde{\mathsf{eq}}_{\vec{z}_1}, \widetilde{\mathsf{eq}}_{\vec{z}_2}, u_s, \chi_{\mathsf{bool}}, \chi_{\vec{z}_1}, \chi_{\vec{z}_2},\\ &\big(\mathsf{cm}^{(s)}_{h_0,j}, \mathsf{cm}^{(s)}_{h_1,j}, v^{(s)}_{h_0,j}, v^{(s)}_{h_0,z,j}, v^{(s)}_{h_1,j}, v^{(s)}_{h_1,z,j}\big)_{j\in[\ell]},\\ &\big(\mathsf{cm}_{s_i}, \vec{r}_i, v_{r,i}, \vec{\alpha}_i, v_{\alpha,i}, (v_{\alpha,i,j})_{j\in[\ell]}, (\mathsf{cm}_{r,i,j})_{j\in[\ell]}\big)_{i\in[m]},\\ &\big(\widetilde{\mathsf{eq}}^{(i)}_{\vec{z}_1}, \widetilde{\mathsf{eq}}^{(i)}_{\vec{z}_2}, u_{s_i}, u_{s_r,i}, (u_{s_r,i,j})_{j\in[\ell]}, \chi_{\vec{z}^{(i)}_1}, \chi_{\vec{z}^{(i)}_2},\\ &\big(\mathsf{cm}^{(i)}_{h_0,j}, \mathsf{cm}^{(i)}_{h_1,j}, v^{(i)}_{h_0,j}, v^{(i)}_{h_0,z,j}, v^{(i)}_{h_1,j}, v^{(i)}_{h_1,z,j}\big)_{j\in[\ell]}\big)_{i\in[m]}\Big).\end{aligned}$$

and the witness is

$$\mathbf{w}_{\mathsf{out}} := \Big(\vec{\boldsymbol{s}}, (\vec{\boldsymbol{h}}^{(s)}_{0,j}, \vec{\boldsymbol{h}}^{(s)}_{1,j})_{j\in[\ell]}, \big(\vec{\boldsymbol{s}}_i, (\vec{\boldsymbol{s}}_{r_i,j}, \vec{\boldsymbol{h}}^{(i)}_{0,j}, \vec{\boldsymbol{h}}^{(i)}_{1,j})_{j\in[\ell]}\big)_{i\in[m]}.\Big).$$

The output pair $(\mathbf{x}_{\mathsf{out}}, \mathbf{w}_{\mathsf{out}})$ is required to satisfy the following constraints. The resulting preliminary backend instance consists of the following constraints.

1. **Commitment consistency.** Every committed witness is consistent with its public commitment.
2. **Structured constraints for the main quadratic relation.** The reduction of $\mathsf{ct}(\langle \vec{\boldsymbol{s}}, \sigma_{-1}(\vec{\boldsymbol{s}})\rangle) = v_s$ produces the standard structured constraints from Figure 2. Namely, the protocol enforces the final evaluation claim for $\vec{\boldsymbol{s}}$, the Boolean-slice sum of the decomposed h_0 table, the consistency between $h_0(\vec{z}_1)$ and the Boolean sum of h_1, and the terminal check $h_1(\vec{z}_2) = u_s^2$. All these checks are expressed as witness-public inner-product constraints against tensor-structured vectors such as $\widetilde{\mathsf{eq}}_{\vec{z}_1} \otimes \widetilde{\mathsf{eq}}_{\vec{z}_2}$, χ_{bool}, $\chi_{\vec{z}_1}$, and $\chi_{\vec{z}_2}$.
3. **Structured constraints for the slice-wise quadratic relations.** For each binary slice $i \in [m]$, the reduction of $\mathsf{ct}\big(\langle \sum_{j=0}^{\ell-1} b^j \vec{\boldsymbol{s}}_{r_i,j}, \sigma_{-1}(\vec{\boldsymbol{s}}_i)\rangle\big) = v_{r,i}$ produces the same type of structured constraints. These include the final evaluation claims for $\vec{\boldsymbol{s}}_i$ and the decomposed pieces $\vec{\boldsymbol{s}}_{r_i,j}$, the Boolean-slice sum of the corresponding $h_0^{(i)}$ table, the consistency between $h_0^{(i)}(\vec{z}_1^{(i)})$ and the Boolean sum of $h_1^{(i)}$, and the terminal check enforcing the product $u_{s_i} \cdot u_{s_r,i}$. The recombinations over the base-b pieces are enforced by the corresponding scalar equalities.
4. **Inherited structured linear constraints.** The linear constraints already present in $\mathcal{R}_{\mathsf{norm,al}}$ are inherited unchanged. These include the binary decomposition consistency of $\vec{\boldsymbol{s}}$, the randomized binary-check claims $\langle \mathsf{coef}(\vec{\boldsymbol{s}}_i), \vec{r}_i\rangle = v_{r,i}$, and the well-formedness checks for $\vec{\boldsymbol{s}}_{r_i} = \sum_{j=0}^{\ell-1} b^j \vec{\boldsymbol{s}}_{r_i,j}$. Each of them is already a witness-public inner-product constraint against a public structured vector, together with a scalar recombination equality.
5. **Relaxed shortness.** All committed witnesses satisfy the relaxed admissible bound $b_{w,\infty} := \max\{b/2, B_\infty\}$.

In addition to the constraints above, the padded tables $\vec{h}_0$ and $\vec{h}_1$ implicitly satisfy a zero-padding condition: all entries indexed by tuples containing $\star$ are

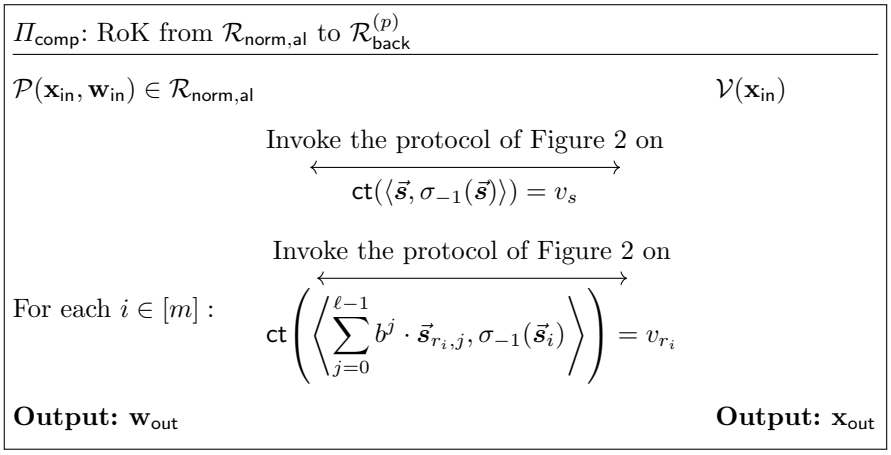


Fig. 3. Reduction of knowledge Π_{comp} from $\mathcal{R}_{\mathsf{norm,al}}$ to $\mathcal{R}^{(p)}_{\mathsf{back}}$. The protocol invokes the quadratic-to-linear reduction of Figure 2 on the self inner-product constraint for $\vec{s}$ and on each auxiliary consistency constraint indexed by $i \in [m]$.

zero. At this stage, this condition need not be enforced explicitly, since all corresponding tensor selectors vanish on the $\star$-indexed coordinates and hence these entries do not affect any of the above inner-product claims. We will enforce the zero-padding condition explicitly in the next section, where it becomes necessary for the subsequent tensor normalization. The full set of constraints is deferred to Appendix F.

Theorem 3. *The protocol Π_{comp} in Figure 3 is a reduction of knowledge from $\mathcal{R}_{\mathsf{norm,al}}$ to $\mathcal{R}^{(p)}_{\mathsf{back}}$. In particular, it satisfies public reducibility, perfect completeness, and knowledge soundness with soundness error $\epsilon_{\mathsf{comp}} \leq (m+1)\epsilon_{\mathsf{quad}}$.*

Proof. The protocol Π_{comp} applies the quadratic-to-linear reduction of Figure 2 to the $(m+1)$ quadratic constraints in $\mathcal{R}_{\mathsf{norm,al}}$, namely the main norm constraint and the m slice-wise binary-check constraints. All other constraints in $\mathcal{R}_{\mathsf{norm,al}}$ are already backend-native and are copied unchanged into the output instance.

Public reducibility. Given the input instance $\mathbf{x}_{\mathsf{in}}$ and the public transcript tr, the output instance $\mathbf{x}_{\mathsf{out}}$ is determined by the inherited public constraints, the commitments, verifier challenges, and scalar claims generated in the $(m+1)$ invocations of Figure 2. Thus there is a deterministic polynomial-time algorithm $\mathsf{Red}_{\mathsf{comp}}$ such that $\mathsf{Red}_{\mathsf{comp}}(\mathbf{x}_{\mathsf{in}}, \mathsf{tr}) = \mathbf{x}_{\mathsf{out}}$. The transformation is witness-independent, so the reduction is publicly reducible.

Perfect completeness. Let $(\mathbf{x}_{\mathsf{in}}, \mathbf{w}_{\mathsf{in}}) \in \mathcal{R}_{\mathsf{norm,al}}$. The non-quadratic constraints of $\mathcal{R}_{\mathsf{norm,al}}$ are copied directly and remain valid. For each quadratic constraint, the honest prover runs Figure 2 with the corresponding committed witnesses. By the

perfect completeness of the quadratic-to-linear reduction, every generated structured constraint is satisfied by the produced intermediate witnesses. Combining these witnesses with the inherited ones gives $\mathbf{w}_{\mathsf{out}}$ such that $(\mathbf{x}_{\mathsf{out}}, \mathbf{w}_{\mathsf{out}}) \in \mathcal{R}^{(p)}_{\mathsf{back}}$. Hence the protocol is perfectly complete.

Knowledge soundness. Suppose that the output instance $\mathbf{x}_{\mathsf{out}}$ has a witness $\mathbf{w}_{\mathsf{out}}$ satisfying $\mathcal{R}^{(p)}_{\mathsf{back}}$. The extractor parses from $\mathbf{w}_{\mathsf{out}}$ the original witnesses appearing in $\mathcal{R}_{\mathsf{norm,al}}$, together with the intermediate witnesses produced by each quadratic-to-linear reduction. For the inherited backend-native constraints, validity in $\mathcal{R}^{(p)}_{\mathsf{back}}$ immediately gives the corresponding constraints of $\mathcal{R}_{\mathsf{norm,al}}$. For each of the $(m+1)$ quadratic constraints, the structured constraints in $\mathcal{R}^{(p)}_{\mathsf{back}}$ are exactly the output of Figure 2. By the soundness of Figure 2, except with probability at most ϵ_{quad} per invocation, these structured constraints imply the original quadratic inner-product relation. Applying a union bound over the $(m+1)$ invocations, all quadratic constraints of $\mathcal{R}_{\mathsf{norm,al}}$ hold except with probability at most $(m+1)\epsilon_{\mathsf{quad}}$. Therefore the parsed witness satisfies $(\mathbf{x}_{\mathsf{in}}, \mathbf{w}_{\mathsf{in}}) \in \mathcal{R}_{\mathsf{norm,al}}$ except with probability at most $\epsilon_{\mathsf{comp}} \leq (m+1)\epsilon_{\mathsf{quad}}$. This proves knowledge soundness. □

Efficiency. The protocol Π_{comp} invokes the quadratic-to-linear reduction of Figure 2 once for the main norm constraint and once for each of the m slice-wise quadratic constraints, hence $m+1$ times in total. All other constraints inherited from $\mathcal{R}_{\mathsf{norm,al}}$ are already backend-native and are copied into the output instance. Thus, the communication, prover time, and verifier time are those of Figure 2 multiplied by $m+1$. For fixed $m = O(1)$, Π_{comp} has the same asymptotic efficiency as a single quadratic-to-linear reduction.

Remark 2. For clarity of presentation, Π_{comp} reduces the $m+1$ quadratic constraints separately. In an implementation, these constraints can first be batched by a random linear combination, and Figure 2 can then be applied once to the batched claim. This is sound because all claims use the same evaluation domain, and a random linear combination remains degree 2 in each variable. This optimization reduces the concrete cost by roughly a factor of $m+1$.

4.3 Π_{bat}: From $\mathcal{R}^{(p)}_{\mathsf{back}}$ to $\mathcal{R}^{(o)}_{\mathsf{back}}$

At this point, the original exact norm statement has been reduced to a preliminary backend instance containing a bounded batch of same-type structured constraints. The witness part contains $(1+2\ell)+m(1+3\ell)$ committed witnesses: one block for $\vec{s}$ and its auxiliary tables, and, for each $i \in [m]$, one block for $\vec{s}_i$, $\vec{s}_{r_i}$, and the corresponding auxiliary tables. The instance contains $(2+4\ell+4m+6m\ell)$ structured constraints. Our next goal is to batch these constraints internally, producing a smaller instance of the same backend relation family. This reduces the cost of proving the reduction output in the downstream backend.

We use the same bi-variate sumcheck principle to batch the structured constraints in $\mathcal{R}^{(p)}_{\mathsf{back}}$. We first classify the public vectors appearing in these constraints into two forms.

Arity-2 tensor-structured constraints. The first class consists of arity-2 tensor-structured vectors of the form $\bigotimes_{j=0}^{L-1}(a_{j,0}, a_{j,1})$. This includes vectors such as $\widetilde{\mathsf{eq}}_{\vec{z}_1} \otimes \widetilde{\mathsf{eq}}_{\vec{z}_2}$ and power vectors such as $\vec{\gamma} = (1, \gamma, \ldots, \gamma^{N-1})$. There are $n_1 \coloneqq 2 + 4m + 2m\ell$ constraints of this type.

For this class, each constraint can be normalized into a Boolean-sum claim. Concretely, if $a_{j,0} + a_{j,1} \neq 0$ for all j, then we have $\bigotimes_{j=0}^{L-1}(a_{j,0}, a_{j,1}) = C \cdot \widetilde{\mathsf{eq}}_{\vec{\rho}}$, where $C \coloneqq \prod_{j=0}^{L-1}(a_{j,0} + a_{j,1})$ and $\rho_j \coloneqq a_{j,1}/(a_{j,0} + a_{j,1})$.

Thus, the constraint $\langle \vec{w}, \bigotimes_{j=0}^{L-1}(a_{j,0}, a_{j,1}) \rangle = v$ is equivalent to

$$\sum_{\vec{b} \in \{0,1\}^L} \widetilde{w}(\vec{b}) \cdot \widetilde{\mathsf{eq}}(\vec{b}, \vec{\rho}) = \frac{v}{C},$$

where $\widetilde{w}$ is the multilinear extension of the vector $\vec{w}$. The summand has individual degree at most 2, and is therefore suitable for the bi-variate reduction.

Following this idea, the constraint $\left\langle \mathsf{coef}(\vec{s}), \widetilde{\mathsf{eq}}_{\vec{z}_1} \otimes \widetilde{\mathsf{eq}}_{\vec{z}_2} \right\rangle = u_s$ is normalized to $\sum_{\vec{b} \in \{0,1\}^{2t}} \widetilde{s}(\vec{b}) \cdot \widetilde{\mathsf{eq}}(\vec{b}, (\vec{z}_1, \vec{z}_2)) = u_s$. And the power-vector constraint $\langle \mathsf{coef}(\vec{s}), \vec{\gamma} \rangle = v_{\mathsf{dec}}$ can be normalized as $\sum_{\vec{b} \in \{0,1\}^{2t}} \widetilde{s}(\vec{b}) \cdot \widetilde{\mathsf{eq}}(\vec{b}, \vec{\gamma}') = v'_{\mathsf{dec}}$, where $\gamma'_j = \gamma^{2^j}/(1 + \gamma^{2^j})$ and $v'_{\mathsf{dec}} = v_{\mathsf{dec}} / \prod_{j=0}^{2t-1}(1 + \gamma^{2^j})$. This is well defined except on the negligible bad event that $1 + \gamma^{2^j} = 0$ for some j.

Arity-4 tensor-structured constraints. The second class consists of arity-4 tensor-structured vectors. It contains the Boolean selector $\chi_{\mathsf{bool}} = \bigotimes_t (1, 1, 0, 0)$ and evaluation selectors of the form $\chi_{\vec{z}} = \bigotimes_{j=0}^{t-1}(\lambda_{-1}(z_j), \lambda_0(z_j), \lambda_1(z_j), 0)$. There are $n_2 \coloneqq 2\ell + 2m\ell$ constraints using χ_{bool}, and $n_3 \coloneqq 2\ell + 2m\ell$ constraints using evaluation selectors $\chi_{\vec{z}}$.

Boolean selector case. The Boolean selector is an arity-4 tensor vector, but it can be normalized to the same Boolean-sum form by reindexing each arity-4 coordinate with two Boolean bits. Specifically, we have [2]

$$(1, 1, 0, 0) = (1, 0) \otimes (1, 1) = 2 \cdot \widetilde{\mathsf{eq}}_{(0,1/2)}.$$

Hence

$$\chi_{\mathsf{bool}} = \bigotimes_t (1, 1, 0, 0) = 2^t \cdot \widetilde{\mathsf{eq}}_{\rho_{\mathsf{bool}}}, \quad \rho_{\mathsf{bool}} = (0, 1/2, 0, 1/2, \ldots, 0, 1/2) \in \mathbb{Z}_q^{2t}.$$

[2] Throughout this work, tensor products are ordered lexicographically; namely, $(a_1, a_2) \otimes (b_1, b_2) = (a_1 b_1, a_1 b_2, a_2 b_1, a_2 b_2)$.

Therefore, a constraint of the form $\langle \mathsf{coef}(\vec{\boldsymbol{h}}), \chi_{\mathsf{bool}} \rangle = v$ is equivalently written as $\sum_{\vec{b} \in \{0,1\}^{2t}} \widetilde{h}(\vec{b})\, \widetilde{\mathsf{eq}}(\vec{b}, \rho_{\mathsf{bool}}) = v/2^t$, where $\widetilde{h}$ is the multilinear extension of the vector $\vec{h}$. Thus every χ_{bool}-constraint is reduced to the same Boolean-sum form as the arity-2 constraints.

Degree-2 selector case. The remaining arity-4 selectors are evaluation selectors for padded degree-2 tables. For $\vec{z} \in \mathbb{Z}_q^t$, the selector is

$$\begin{aligned}\chi_{\vec{z}} &= \bigotimes_{j=0}^{t-1} \big(\lambda_0(z_j), \lambda_1(z_j), \lambda_{-1}(z_j), 0\big) \\ &= \bigotimes_{j=0}^{t-1} \big((1+z_j)(1-z_j), \frac{1}{2} z_j(z_j+1), \frac{1}{2} z_j(z_j-1), 0\big)\end{aligned}.$$

Recall that the padded domain is $\mathcal{D} = \{0, 1, -1, \star\}$, and all witness entries indexed by tuples containing $\star$ are set to 0. Under this padding convention, each local selector can be replaced by the decomposable vector

$$(1 + z_j, -z_j/2) \otimes (1 - z_j, z_j/2) = (\lambda_0(z_j), \lambda_1(z_j), \lambda_{-1}(z_j), -z_j^2/4).$$

This differs from the original selector only in the $\star$ coordinate, and therefore gives the same inner product once the zero-padding condition is enforced. We then normalize this arity-2 tensor as before.

Assuming $1 + z_j/2 \neq 0$ and $1 - z_j/2 \neq 0$ for all j, define

$$\rho_{\vec{z}} = \left(\frac{-z_0/2}{1+z_0/2}, \frac{z_0/2}{1-z_0/2}, \ldots, \frac{-z_{t-1}/2}{1+z_{t-1}/2}, \frac{z_{t-1}/2}{1-z_{t-1}/2} \right) \in \mathbb{Z}_q^{2t},$$

and

$$C_{\vec{z}} = \prod_{j=0}^{t-1} \left(1 + \frac{z_j}{2}\right) \left(1 - \frac{z_j}{2}\right) = \prod_{j=0}^{t-1} \left(1 - \frac{z_j^2}{4}\right).$$

Thus a constraint $\langle \mathsf{coef}(\vec{\boldsymbol{h}}), \chi_{\vec{z}} \rangle = v$ is rewritten as $\sum_{\vec{b} \in \{0,1\}^{2t}} \widetilde{h}(\vec{b})\, \widetilde{\mathsf{eq}}(\vec{b}, \rho_{\vec{z}}) = v/C_{\vec{z}}$.

This normalization is sound only after enforcing the zero-padding condition. Without such a check, the rewritten arity-2 tensor may read nonzero $\star$-entries, whereas the original selector $\chi_{\vec{z}}$ ignores them. We therefore add a zero-check for the $\star$-related entries before applying this normalization.

Zero-check for $\star$-related entries. We must ensure that all padded entries indexed by tuples containing $\star$ are zero. In order to achieve this goal, we first encode $\mathcal{D} = \{0, 1, -1, \star\}$ by Boolean bits, e.g., $0 \mapsto 00$, $1 \mapsto 01$, $-1 \mapsto 10$, and $\star \mapsto 11$. Thus a padded table over $\mathcal{D}^t$ can be viewed as a Boolean table over $\{0,1\}^{2t}$.

For each $i \in [t]$, let $\mathcal{P}_i$ be the set of Boolean indices whose i-th encoded symbol is $(1, 1)$, namely whose i-th $\mathcal{D}$-coordinate is $\star$:

$$\mathcal{P}_i = \left\{ \vec{b} \in \{0,1\}^{2t} \;\middle|\; (b_{2i}, b_{2i+1}) = (1,1) \right\} = \{0,1\}^{2i} \times \{(1,1)\} \times \{0,1\}^{2(t-i-1)}.$$

It suffices to prove that all entries indexed by $\mathcal{P}_\star := \bigcup_{i\in[t]} \mathcal{P}_i$ are zero. For each $i \in [t]$, the verifier samples random coordinates $\vec{\rho} \in \mathbb{Z}_q^{2t-2}$ and forms the point

$$\vec{\zeta}^{(i)} = (\rho_0, \ldots, \rho_{2i-1}, 1, 1, \rho_{2i}, \ldots, \rho_{2t-3}) \in \mathbb{Z}_q^{2t}.$$

The zero-padding check requires $\langle \vec{h}, \widetilde{\mathsf{eq}}_{\vec{\zeta}^{(i)}} \rangle = 0$ for every $i \in [t]$. Equivalently, we require $\sum_{\vec{b}\in\{0,1\}^{2t}} \tilde{h}(\vec{b}) \cdot \tilde{\mathsf{eq}}(\vec{b}, \vec{\zeta}^{(i)}) = 0$ for every $i \in [t]$.

Each check is a random linear combination of all entries whose i-th $\mathcal{D}$-coordinate is $\star$. If some padded entry is nonzero, then for some i, $\widetilde{h}(\vec{\zeta}^{(i)})$ is a nonzero polynomial in the random coordinates of $\vec{\rho}$, and the check fails except with probability at most $(2t-2)/q$.

Inner batching process. At this point, all structured constraints have been normalized into Boolean-sum claims for polynomials of individual degree at most 2. Let $\mathcal{N} := n_1 + n_2 + (t+1)n_3$ be the total number of such claims. Each claim has the form

$$\sum_{\vec{b}\in\{0,1\}^{2t}} F_i(\vec{b}) = \sum_{\vec{b}\in\{0,1\}^{2t}} \widetilde{f}_i(\vec{b})\widetilde{\mathsf{eq}}(\vec{b}, \vec{r}_i^{(f)}) = v_i,$$

where $\widetilde{f}_i$ is the multilinear extension of the corresponding witness table, and each F_i has individual degree at most 2.

To batch them, the verifier samples random coefficients β_i, and the prover forms $g_b(\vec{X}) := \sum_{i\in[\mathcal{N}]} \beta_i F_i(\vec{X})$.

The batched claim is $\sum_{\vec{b}\in\{0,1\}^{2t}} g_b(\vec{b}) = \sum_{i\in[\mathcal{N}]} \beta_i v_i$. Since g_b still has individual degree at most 2, we apply the same bi-variate reduction to this single batched claim. By the standard random-linear-combination argument, if any original claim is false, then the batched claim is accepted with probability at most $1/q$ over the choice of the β_i. The bi-variate reduction introduces two intermediate table witnesses for the partial-sum polynomials and reduces the batched claim to backend-native structured inner-product constraints. Finally, constraints paired with the same public tensor vector are folded by random linear combinations of their witnesses.

The folding challenges are sampled from a prescribed small challenge space, ensuring that the folded witnesses remain within the relaxed shortness regime while keeping the additional folding error controlled. Thus, the preliminary backend instance is compressed into a smaller instance of the same relation family. In particular, the final witness part contains $1 + 2\ell$ committed witnesses: one folded witness and the two decomposed intermediate table witnesses introduced by the final bi-variate reduction.

The full batching protocol reducing $\mathcal{R}^{(p)}_{\mathsf{back}}$ to $\mathcal{R}^{(o)}_{\mathsf{back}}$ is given in Figure 4. The output pair $(\mathbf{x}_{\mathsf{out}}, \mathbf{w}_{\mathsf{out}})$ is defined as follows:

$$\begin{aligned}\mathbf{x}_{\mathsf{out}} &:= \Big(\mathbf{x}_{\mathsf{pp}},\, \mathsf{cm}_f,\, u_\theta,\, B_{f,\infty},\, \widetilde{\mathsf{eq}}_{(\vec{z}_1,\vec{z}_2)},\, \chi_{\mathsf{bool}},\, \chi_{\vec{z}_1},\, \chi_{\vec{z}_2},\\ &\qquad (\mathsf{cm}_{h_0,j}, \mathsf{cm}_{h_1,j}, v_{0,j}, v_{0,z,j}, v_{1,j}, v_{1,z,j})_{j\in[\ell]}\Big),\\ \mathbf{w}_{\mathsf{out}} &:= \Big(\vec{\boldsymbol{f}}_\theta,\, (\vec{\boldsymbol{h}}_{0,j}, \vec{\boldsymbol{h}}_{1,j})_{j\in[\ell]}\Big).\end{aligned}$$

The output pair is required to satisfy the following constraints.

1. **Commitment consistency.** The folded witness and intermediate tables open their public commitments: $\mathsf{Com}(\vec{\boldsymbol{f}}_\theta) = \mathsf{cm}_f$, and $\mathsf{Com}(\vec{\boldsymbol{h}}_{a,j}) = \mathsf{cm}_{h_a,j}$ for every $a \in \{0,1\}$ and $j \in [\ell]$.
2. **Structured inner-product constraints.** The folded witness satisfies the terminal evaluation claim $\langle \mathsf{coef}(\vec{\boldsymbol{f}}_\theta), \widetilde{\mathsf{eq}}_{(\vec{z}_1,\vec{z}_2)}\rangle = u_\theta$. For every $j \in [\ell]$, the intermediate tables satisfy

$$\begin{aligned}\langle \mathsf{coef}(\vec{\boldsymbol{h}}_{0,j}), \chi_{\mathsf{bool}}\rangle &= v_{0,j}, \quad \langle \mathsf{coef}(\vec{\boldsymbol{h}}_{0,j}), \chi_{\vec{z}_1}\rangle = v_{0,z,j},\\ \langle \mathsf{coef}(\vec{\boldsymbol{h}}_{1,j}), \chi_{\mathsf{bool}}\rangle &= v_{1,j}, \quad \langle \mathsf{coef}(\vec{\boldsymbol{h}}_{1,j}), \chi_{\vec{z}_2}\rangle = v_{1,z,j}.\end{aligned}$$

3. **Relaxed shortness.** The committed witnesses satisfy $\|\vec{\boldsymbol{f}}_\theta\|_\infty \le B_{f,\infty}$ and $\|\vec{\boldsymbol{h}}_{a,j}\|_\infty \le b/2$ for all $a \in \{0,1\}$ and $j \in [\ell]$.

We prove that Π_{bat} in Figure 4 satisfying public reducibility, perfect completeness, and relaxed knowledge soundness.

Theorem 4. *Let $\mathcal{C} \subseteq \mathcal{R}_q$ be the folding challenge space, and assume that every $c \in \mathcal{C}$ satisfies $\|c\| \le \tau$ and $\|c\|_{\mathsf{op},\infty} \le T$. Let $b_{w,\infty} := \max\{B_\infty, b/2\}$, and let $\mathcal{N} = n_1 + n_2 + (t+1)n_3$ be the number of normalized Boolean-sum claims, where $n_1 = 2+4m+2m\ell$, $n_2 = 2\ell+2m\ell$, and $n_3 = 2\ell+2m\ell$. Set $B_{f,\infty} := \mathcal{N}\cdot T\cdot b_{w,\infty}$. Then Π_{bat} in Figure 4 satisfies public reducibility and perfect completeness with folded-witness bound $B_{f,\infty}$.*

Proof. We prove the two properties seperately.

Public reducibility. Given the input instance $\mathbf{x}_{\mathsf{in}}$ and the public transcript tr, the output instance $\mathbf{x}_{\mathsf{out}}$ is determined without using the witness. Indeed, normalization is deterministic from $\mathbf{x}_{\mathsf{in}}$ and the public challenges. The transcript contains the challenges, the commitments, the scalar claims, and the terminal folded claim u_θ. The folded commitment cm_f is computed homomorphically from the input commitments using the public folding coefficients. Thus there is a deterministic polynomial-time map $\mathsf{Red}_{\mathsf{bat}}(\mathbf{x}_{\mathsf{in}}, \mathsf{tr}) = \mathbf{x}_{\mathsf{out}}$, proving public reducibility.

Π_{bat}: Reduction from $\mathcal{R}^{(p)}_{\mathsf{back}}$ to $\mathcal{R}^{(o)}_{\mathsf{back}}$

1. $\mathcal{P}$ and $\mathcal{V}$: Normalize all structured constraints to $\mathcal{N}$ Boolean-sum claims.
2. $\mathcal{P}$ engage to prove the claim:

$$C_i : \sum_{\vec{b}\in\{0,1\}^{2t}} F_i(\vec{b}) = v_i, F_i(\vec{X}) \coloneqq \widetilde{f}_i(\vec{X}) \cdot \widetilde{\mathsf{eq}}(\vec{X}, \vec{r}_i^{(f)}), \forall i \in [\mathcal{N}].$$

$\mathcal{P}(\mathbf{x}_{\mathsf{in}}, \mathbf{w}_{\mathsf{in}}) \in \mathcal{R}^{(p)}_{\mathsf{back}}$		$\mathcal{V}(\mathbf{x}_{\mathsf{in}})$
$g_b(\vec{X}_1, \vec{X}_2)$ $\coloneqq \sum_{i=0}^{\mathcal{N}-1} \beta_i F_i(\vec{X}_1, \vec{X}_2)$. $v_b \coloneqq \sum_{i=0}^{\mathcal{N}-1} \beta_i v_i$.	$\xleftarrow{\quad \vec{\beta} \quad}$	$\vec{\beta} \xleftarrow{\$} \mathbb{Z}_q^{\mathcal{N}}$
$h_0(\vec{X}_1) \coloneqq \sum_{\vec{b}\in\{0,1\}^t} g_b(\vec{X}_1, \vec{b})$. $\vec{h}_0 \coloneqq (h_0(\vec{u}))_{\vec{u}\in\mathcal{D}^t}$, $\vec{\boldsymbol{h}}_0 \coloneqq \mathsf{Lift}(\vec{h}_0)$. $(\vec{\boldsymbol{h}}_{0,j})_{j\in[\ell]} \coloneqq \mathsf{Decom}_b(\vec{\boldsymbol{h}}_0)$. $\mathsf{cm}_{h_{0,j}} \coloneqq \mathsf{Com}(\vec{\boldsymbol{h}}_{0,j})$, $v_{0,j} \coloneqq \langle \mathsf{coef}(\vec{\boldsymbol{h}}_{0,j}), \chi_{\mathsf{bool}} \rangle$.	$\xrightarrow{\quad (\mathsf{cm}_{h_{0,j}}, v_{0,j})_{j\in[\ell]} \quad}$	$v_b \coloneqq \sum_{i=0}^{\mathcal{N}-1} \beta_i v_i$, $\sum_{j=0}^{\ell-1} b^j v_{0,j} \stackrel{?}{=} v_b$
	$\xleftarrow{\quad \vec{z}_1 \quad}$	$\vec{z}_1 \xleftarrow{\$} \mathbb{Z}_q^t$
$v_{0,z,j} \coloneqq \langle \mathsf{coef}(\vec{\boldsymbol{h}}_{0,j}), \chi_{\vec{z}_1} \rangle$. $h_1(\vec{X}_2) \coloneqq g_b(\vec{z}_1, \vec{X}_2)$. $\vec{h}_1 \coloneqq (h_1(\vec{u}))_{\vec{u}\in\mathcal{D}^t}$, $\vec{\boldsymbol{h}}_1 \coloneqq \mathsf{Lift}(\vec{h}_1)$. $(\vec{\boldsymbol{h}}_{1,j})_{j\in[\ell]} \coloneqq \mathsf{Decom}_b(\vec{\boldsymbol{h}}_1)$. $\mathsf{cm}_{h_{1,j}} \coloneqq \mathsf{Com}(\vec{\boldsymbol{h}}_{1,j})$, $v_{1,j} \coloneqq \langle \mathsf{coef}(\vec{\boldsymbol{h}}_{1,j}), \chi_{\mathsf{bool}} \rangle$.	$\xrightarrow{\quad (v_{0,z,j}, \mathsf{cm}_{h_{1,j}}, v_{1,j})_{j\in[\ell]} \quad}$	$\sum_{j=0}^{\ell-1} b^j v_{0,z,j} \stackrel{?}{=} \sum_{j=0}^{\ell-1} b^j v_{1,j}$
	$\xleftarrow{\quad \vec{z}_2 \quad}$	$\vec{z}_2 \xleftarrow{\$} \mathbb{Z}_q^t$
$v_{1,z,j} \coloneqq \langle \mathsf{coef}(\vec{\boldsymbol{h}}_{1,j}), \chi_{\vec{z}_2} \rangle$. $u_i \coloneqq \widetilde{f}_i(\vec{z}_1, \vec{z}_2), \quad i \in [\mathcal{N}]$.	$\xrightarrow{\quad (u_i)_{i\in[\mathcal{N}]}, (v_{1,z,j})_{j\in[\ell]} \quad}$	$e_i \coloneqq \widetilde{\mathsf{eq}}((\vec{z}_1, \vec{z}_2), \vec{r}_i^{(f)})$. $\sum_{j=0}^{\ell-1} b^j v_{1,z,j} \stackrel{?}{=} \sum_{i=0}^{\mathcal{N}-1} \beta_i u_i e_i$.
	$\xleftarrow{\quad \vec{\theta} \quad}$	$\vec{\theta} \xleftarrow{\$} \mathcal{C}^{\mathcal{N}}$
$\vec{\boldsymbol{f}}_\theta \coloneqq \sum_{i=0}^{\mathcal{N}-1} \theta_i \vec{\boldsymbol{f}}_i$. $u_\theta \coloneqq \sum_{i=0}^{\mathcal{N}-1} \theta_i u_i$. Output $\mathbf{w}_{\mathsf{out}}$.		$\mathsf{cm}_f = \sum_{i=0}^{\mathcal{N}-1} \theta_i \mathsf{cm}_{f_i}$. Output $\mathbf{x}_{\mathsf{out}}$.

Fig. 4. Π_{bat}: internal batching protocol for the preliminary backend relation $\mathcal{R}^{(p)}_{\mathsf{back}}$.

Perfect completeness. Assume $(\mathbf{x}_{\mathsf{in}}, \mathbf{w}_{\mathsf{in}}) \in \mathcal{R}^{(p)}_{\mathsf{back}}$. After normalization, every structured constraint gives a valid Boolean-sum claim $C_i : \sum_{\vec{b}\in\{0,1\}^{2t}} F_i(\vec{b}) = v_i$ for $i \in [\mathcal{N}]$. For batching coefficients β_i, the aggregated polynomial $g_b(\vec{X}) := \sum_{i\in[\mathcal{N}]} \beta_i F_i(\vec{X})$ satisfies $\sum_{\vec{b}\in\{0,1\}^{2t}} g_b(\vec{b}) = \sum_{i\in[\mathcal{N}]} \beta_i v_i$. The honest prover computes the partial-sum tables h_0, h_1 exactly as in the bi-variate reduction, so all scalar consistency checks hold. At the terminal point $(\vec{z}_1, \vec{z}_2)$, write $u_i = \widetilde{f_i}(\vec{z}_1, \vec{z}_2)$. Then $g_b(\vec{z}_1, \vec{z}_2) = \sum_{i\in[\mathcal{N}]} \beta_i u_i \widetilde{\mathsf{eq}}((\vec{z}_1, \vec{z}_2), \vec{r}_i^{(f)})$. For folding challenge $\vec{\theta}$, the folded witness $\vec{\boldsymbol{f}}_\theta = \sum_{i\in[\mathcal{N}]} \theta_i \vec{\boldsymbol{f}}_i$ satisfies $\langle \mathsf{coef}(\vec{\boldsymbol{f}}_\theta), \widetilde{\mathsf{eq}}_{(\vec{z}_1,\vec{z}_2)}\rangle = u_\theta$.

It remains to check shortness. Since each input witness has norm at most $b_{w,\infty}$ and each folding coefficient has operator norm at most T,

$$\|\vec{\boldsymbol{f}}_\theta\|_\infty \le \sum_{i\in[\mathcal{N}]} \|\theta_i \vec{\boldsymbol{f}}_i\|_\infty \le \mathcal{N} T b_{w,\infty} = B_{f,\infty}.$$

The intermediate table witnesses satisfy their prescribed shortness bounds by honest construction. Therefore proving perfect completeness.

Theorem 5. *Assume that the commitment scheme is additively homomorphic and binding for openings of norm at most $\mathcal{B} := \max\{2B_{f,\infty}, 4TB_{f,\infty}\}$, under the corresponding MSIS assumption.*

Then the protocol Π_{bat} in Figure 4 satisfies relaxed knowledge soundness from $\mathcal{R}^{(p)}_{\mathsf{back}}$ to $\mathcal{R}^{(o)}_{\mathsf{back}}$. More precisely, for any prover $\mathcal{P}^*$ that makes the verifier accept with probability ϵ^*, there exists an extractor $\mathcal{E}_{\mathsf{bat}}$ which, whenever $\epsilon^* - \mathcal{N}/|\mathcal{C}|$ is non-negligible, runs in expected polynomial time and outputs openings for the input commitments of $\mathcal{R}^{(p)}_{\mathsf{back}}$. The extracted openings satisfy all algebraic constraints of $\mathcal{R}^{(p)}_{\mathsf{back}}$ and a relaxed shortness bound. The soundness error is at most $\epsilon_{\mathsf{bat}} \le \epsilon_{\mathsf{N}} + \frac{4t+1}{q} + \frac{\mathcal{N}}{|\mathcal{C}|}$, where ϵ_{N} is the soundness error of the normalization step. For coordinate-wise zero-padding checks, we have $\epsilon_{\mathsf{N}} \le \frac{n_3 \cdot t(2t-2)}{q}$.

The complete proof is given in Appendix B.

Efficiency. We analyze the asymptotic efficiency of Π_{bat}. Recall that $N = nd = 4^t$, and let (n_1, n_2, n_3) denote the numbers of arity-2 tensor, χ_{bool}, and $\chi_{\vec{z}}$ constraints, respectively. After adding the t coordinate-wise zero-padding checks for each $\chi_{\vec{z}}$-constraint, the number of Boolean-sum claims is $\mathcal{N} = n_1 + n_2 + (t+1)n_3$. The number of distinct witness tables is only $\mathcal{M} := n_1 + n_2 + n_3$, since each $\chi_{\vec{z}}$-claim and its padding checks share one witness table and differ only in the public selector.

The normalization phase is communication-free. In the batching phase, the prover sends 2ℓ commitments to intermediate tables, 4ℓ scalar claims for the two partial-sum tables, and $\mathcal{M}$ terminal evaluations. Therefore, the prover-to-verifier communication is $\big(2\ell\kappa|\mathcal{R}_q| + (4\ell + \mathcal{M})\log q\big)$ bits. For fixed constants κ, m, ℓ, this communication is independent of N in the number of ring and field elements.

We next consider the computational cost. In the normalization phase, arity-2 and χ_{bool} constraints are normalized from succinct tensor descriptions in $O(\log N)$

field operations per constraint. For each $\chi_{\tilde{z}}$-constraint, the t zero-padding selectors share the same random coordinates and differ only in the two positions fixed to encode $\star$. Hence their descriptions can be generated in $O(t) = O(\log N)$ total time per group. Thus the total normalization cost is $O((n_1 + n_2 + n_3) \log N)$, which is $O(\log N)$ for fixed m, ℓ. In the batching phase, for the prover, forming the batched polynomial $g_b = \sum_i \beta_i F_i$ costs $O(\mathcal{N}N)$ operations. The subsequent bi-variate reduction on g_b costs $O(\cdot 6^t)$ field operations in the direct implementation. The remaining scalar evaluations are lower order.

On the verifier side, computing the batched target value costs $O(\mathcal{N})$ field operations. Since n_1, n_2, n_3 are fixed and $\mathcal{N} = O(t)$, this is $O(\log N)$. The equality factors and tensor selectors can also be derived in $O(t) = O(\log N)$ time using their shared tensor structure. Scalar recombination checks add only $O(1)$ work for fixed ℓ. Therefore, for fixed m, ℓ, κ of $O(1)$, Π_{bat} has prover-side communication independent of N , prover time $O(N^{\log_4 6})$, and verifier time $O(\log N)$.

4.4 Protocol Composition

We now compose the three reductions. Let $\Pi_{\mathsf{r}} \coloneqq \Pi_{\mathsf{alg}} \diamond \Pi_{\mathsf{comp}} \diamond \Pi_{\mathsf{bat}}$, where $\diamond$ denotes sequential composition. The composed protocol reduces the original exact ℓ_2-norm relation to the final backend relation $\mathcal{R}^{(o)}_{\mathsf{back}}$. The backend is only required to handle structured algebraic constraints with admissible shortness bounds; exact norm semantics are enforced by the reduction. We obtain the following theorem.

Theorem 6. *The composed protocol Π_{r} is a reduction of knowledge from $\mathcal{R}_{\mathsf{norm,na}}$ to $\mathcal{R}^{(o)}_{\mathsf{back}}$. It satisfies public reducibility, perfect completeness, and exact knowledge soundness with soundness error $\epsilon_{\mathsf{r}} \leq \epsilon_{\mathsf{alg}} + \epsilon_{\mathsf{comp}} + \epsilon_{\mathsf{bat}}$.*

Proof. Public reducibility and perfect completeness follow directly from the corresponding properties of Π_{alg}, Π_{comp}, and Π_{bat} under sequential composition. It remains to prove knowledge soundness.

Given an accepting final backend instance, Theorem 5 implies that, except with probability ϵ_{bat}, the extractor recovers openings satisfying the preliminary backend relation $\mathcal{R}^{(p)}_{\mathsf{back}}$, up to the relaxed admissible shortness bound. Given such openings, Theorem 3 implies that, except with probability ϵ_{comp}, the extractor recovers a witness satisfying the algebraized norm relation $\mathcal{R}_{\mathsf{norm,al}}$. Finally, Theorem 1 implies that, except with probability ϵ_{alg}, this witness satisfies the original relation $\mathcal{R}_{\mathsf{norm,na}}$, including the exact ℓ_2-norm bound.

The key point is that the exact norm guarantee is not derived from the relaxed backend shortness condition. Rather, it has already been compiled into the algebraic constraints of $\mathcal{R}_{\mathsf{norm,al}}$. Thus, once the extracted witness satisfies these algebraic constraints, the original exact norm semantics follow. By a union bound, the total soundness error is at most $\epsilon_{\mathsf{alg}} + \epsilon_{\mathsf{comp}} + \epsilon_{\mathsf{bat}}$. □

Table 1. Asymptotic efficiency of Π_{alg}, Π_{comp}, Π_{bat} and the composed reduction Π_{r}.

	Communication (prover-to-verifier, bits)	**Prover time** ($\mathbb{Z}_q$ operations)	**Verifier time** ($\mathbb{Z}_q$ operations)
Π_{alg}	$O(\lvert\mathcal{R}_q\rvert + \log q)$	$O(N)$	$O(\log N)$
Π_{comp}	$O(\lvert\mathcal{R}_q\rvert + \log q)$	$O\big(N^{\log_4 6}\big)$	$O(\log N)$
Π_{bat}	$O(\lvert\mathcal{R}_q\rvert + \log q)$	$O\big(N^{\log_4 6}\big)$	$O(\log N)$
Π_{r}	$O(\lvert\mathcal{R}_q\rvert + \log q)$	$O\big(N^{\log_4 6}\big)$	$O(\log N)$

4.5 Efficiency Analysis

We now analyze the asymptotic and concrete costs of the reduction layer. This section presents the asymptotic complexity and parameter-selection criteria, while the concrete cost estimates are deferred to Appendix C.

Asymptotic reduction overhead. We first analyze the asymptotic overhead of the composed reduction $\Pi_{\mathsf{r}} = \Pi_{\mathsf{alg}} \diamond \Pi_{\mathsf{comp}} \diamond \Pi_{\mathsf{bat}}$. Let $\vec{\boldsymbol{s}} \in \mathcal{R}_q^n$ be the committed witness and set $N = nd = 4^t$, where d is the ring dimension. We measure communication in bits, and prover and verifier costs in operations over $\mathbb{Z}_q$.

The cost of Π_{r} is obtained by summing the costs of Π_{alg}, Π_{comp}, and Π_{bat}. Throughout this asymptotic accounting, we treat the decomposition parameters m and ℓ, the commitment dimension κ, and the admissible bound B_∞ as fixed constant sof $O(1)$ by the security parameter. Under this parameter regime, the reduction layer has prover-side communication independent of the input dimension when measured in ring and field elements, subquadratic prover overhead dominated by the bi-variate reduction, and logarithmic verifier overhead. The resulting asymptotic costs are summarized in Table 1.

Parameter Selection for Concrete Efficiency. Let λ be the target security parameter and let the input witness be a vector over $\mathcal{R}_q$ of length n. We choose the modulus q, the commitment dimension κ, the admissible coefficient bound $B_\infty = 2^m - 1$, decomposition parameters (b, ℓ) with $\ell = \lceil \log_b q \rceil$, and a challenge space $\mathcal{C}$ so that the following conditions hold.

1. The total soundness error of the composed reduction satisfies $\epsilon_{\mathsf{r}} = \epsilon_{\mathsf{alg}} + \epsilon_{\mathsf{comp}} + \epsilon_{\mathsf{bat}} \leq 2^{-\lambda}$. Thus q and $|\mathcal{C}|$ are chosen large enough to make the Schwartz–Zippel, batching, zero-padding, and rewinding errors negligible. If a smaller modulus is used in the concrete estimate, we amplify the reduction soundness by independent repetitions.
2. The modulus q is large enough to prevent wraparound in the integer quantities encoded by the norm-reduction constraints. In particular, we require $q > nd \cdot B_\infty^2$.
3. Under the norm bound $\mathcal{B}$ and commitment dimension κ, the underlying MSIS used for the commitment scheme remains hard.

4. m, ℓ, and κ are treated as fixed constants, *i.e.*, $O(1)$ with respect to the input dimension $N = nd$. Under this regime, the reduction-layer communication is independent of N when measured in ring and field elements.

Remark 3 (Packed-commitment optimization). The concrete communication can be reduced by packing each family of decomposed auxiliary witnesses into a single longer witness and committing to it with one block-structured Ajtai commitment. The corresponding linear constraints are updated by tensoring the original public selector with the decomposition coefficients, so the algebraic statement and backend relation form remain unchanged. The same packing can be propagated through Π_{bat} by batching and folding the corresponding blocks jointly, incurring only an additional negligible soundness loss from the extra random linear combinations. Since the packing width is $w = \max\{m, \ell\} = O(1)$ in our parameter regime, this optimization does not change the asymptotic complexity, while substantially reducing the number of commitment elements in the concrete transcript. We give the full construction, soundness analysis, and resulting concrete savings in Appendix D.

5 Beyond Norm Reduction: Folding Quadratic-Relation Instances via Backend Interface

We next explain how the same reduction machinery applies beyond exact norm constraints. The key observation is that Π_{comp} applies to any committed quadratic inner-product relation. After this reduction, the resulting backend-native structured linear constraints can be folded and batched using the same mechanism as in Π_{bat}. We consider the following committed quadratic relation:

$$\mathcal{R}_{\mathsf{qua}} = \left\{ \begin{array}{l} \mathbf{x} = (\mathsf{cm}_1, \mathsf{cm}_2, B, v), \\ \mathbf{w} = (\vec{\boldsymbol{s}}_1, \vec{\boldsymbol{s}}_2) \end{array} \;\middle|\; \begin{array}{l} \mathsf{Com}(\vec{\boldsymbol{s}}_1) = \mathsf{cm}_1, \mathsf{Com}(\vec{\boldsymbol{s}}_2) = \mathsf{cm}_2, \\ \mathsf{ct}\left(\langle \vec{\boldsymbol{s}}_1, \sigma_{-1}(\vec{\boldsymbol{s}}_2)\rangle\right) = v \pmod q, \\ \|\vec{\boldsymbol{s}}_1\|_2 \le B, \quad \|\vec{\boldsymbol{s}}_2\|_2 \le B. \end{array} \right\}. \tag{5}$$

Reduction from $\mathcal{R}_{\mathsf{qua}}$ to $\mathcal{R}^{(o)}_{\mathsf{back}}$. Given $(\mathbf{x}^{(0)}, \mathbf{w}^{(0)})$ and $(\mathbf{x}^{(1)}, \mathbf{w}^{(1)})$ of $\mathcal{R}_{\mathsf{qua}}$, we first reduce them independently. For each instance, we apply the same pipeline: norm constraints, if present, are algebraized by Π_{alg}; quadratic constraints are compiled by Π_{comp}; and the resulting structured linear constraints are batched by Π_{bat}. This produces two compact instances of the final backend relation $\mathcal{R}^{(o)}_{\mathsf{back}}$.

Reduction from $(\mathcal{R}^{(o)}_{\mathsf{back}})^2$ to $\mathcal{R}^{(o)}_{\mathsf{back}}$. The folding step is performed at the backend layer. Given two instances of $\mathcal{R}^{(o)}_{\mathsf{back}}$, we first normalize their structured linear constraints into Boolean-sum claims. We then batch these claims by a random linear combination and apply the same bi-variate batching reduction as in Figure 4. This produces a single instance of the same relation $\mathcal{R}^{(o)}_{\mathsf{back}}$. For multiple instances, this two-to-one step can be iterated in a standard split-and-fold framework. The witnesses are decomposed into smaller-norm pieces before folding, and the batching step is applied to these pieces to control norm growth.

Efficiency and scope. Both reductions described above preserve the structure of the core protocol in Section 4 and inherit its efficiency. Hence, the reduction-layer communication of the folding step is independent of the input dimension when measured in ring and field elements. The folded proof obligation also remains in the same backend-native relation family, so the final succinctness is inherited from the downstream backend.

More generally, our reduction acts as a compiler from quadratic-linear algebraic relations to a foldable backend interface. This view also applies to broader relations such as R1CS: one first rewrites the constraints as quadratic relations plus linear consistency checks, and then reduces them to the foldable backend relation family. We give this reformulation and the corresponding folding implementation in Appendix E. This can be seen as a potential optimization layer for LatticeFold+-style folding schemes [6].

Acknowledgment

This work was partially supported by the Hong Kong RGC General Research Fund (GRF) under project numbers 15202123 and 15205624; the RGC Early Career Scheme (ECS) under project number 25207926; the RGC Collaborative Research Fund (CRF) under project number C5097-25G; the NSFC Young Scientists Fund (No. 62302418).

References

1. Albrecht, M.R., Cini, V., Lai, R.W.F., Malavolta, G., Thyagarajan, S.A.K.: Lattice-based SNARKs: Publicly verifiable, preprocessing, and recursively composable - (extended abstract). In: CRYPTO Part II. pp. 102–132 (2022). https://doi.org/10.1007/978-3-031-15979-4_4
2. Albrecht, M.R., Fenzi, G., Lapiha, O., Nguyen, N.K.: SLAP: Succinct lattice-based polynomial commitments from standard assumptions. In: EUROCRYPT Part VI. pp. 90–119 (2024). https://doi.org/10.1007/978-3-031-58754-2_4
3. Attema, T., Fehr, S., Klooß, M.: Fiat-shamir transformation of multi-round interactive proofs (extended version). J. Cryptol. **36**(4), 36 (2023). https://doi.org/10.1007/S00145-023-09478-Y
4. Beullens, W., Seiler, G.: LaBRADOR: Compact proofs for R1CS from module-SIS. In: CRYPTO Part V. pp. 518–548 (2023). https://doi.org/10.1007/978-3-031-38554-4_17
5. Boneh, D., Chen, B.: Latticefold: A lattice-based folding scheme and its applications to succinct proof systems. In: Advances in Cryptology - ASIACRYPT 2025 - 31st International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, VIC, Australia, December 8-12, 2025, Proceedings, Part III. pp. 330–362. Lecture Notes in Computer Science, Springer (2025). https://doi.org/10.1007/978-981-95-5099-9_11, https://doi.org/10.1007/978-981-95-5099-9_11
6. Boneh, D., Chen, B.: LatticeFold+: Faster, simpler, shorter lattice-based folding for succinct proof systems. In: CRYPTO Part VII. pp. 327–361 (2025). https://doi.org/10.1007/978-3-032-01907-3_11
7. de Castro, L., Peikert, C.: Functional commitments for all functions, with transparent setup and from SIS. In: EUROCRYPT Part III. pp. 287–320 (2023). https://doi.org/10.1007/978-3-031-30620-4_10
8. Cini, V., Lai, R.W.F., Malavolta, G.: Lattice-based succinct arguments from vanishing polynomials - (extended abstract). In: CRYPTO Part II. pp. 72–105 (2023). https://doi.org/10.1007/978-3-031-38545-2_3
9. Cini, V., Malavolta, G., Nguyen, N.K., Wee, H.: Polynomial commitments from lattices: Post-quantum security, fast verification and transparent setup. In: CRYPTO Part X. pp. 207–242 (2024). https://doi.org/10.1007/978-3-031-68403-6_7
10. Fenzi, G., Moghaddas, H., Nguyen, N.K.: Lattice-based polynomial commitments: Towards asymptotic and concrete efficiency. J. Cryptol. **37**(3), 31 (2024). https://doi.org/10.1007/S00145-024-09511-8
11. Gao, S., Zheng, T., Guo, Y., Peng, Z., Xiao, B.: Lattice-based zero-knowledge proofs for blockchain confidential transactions. In: Public-Key Cryptography - PKC 2025 - 28th IACR International Conference on Practice and Theory of Public-Key Cryptography, Røros, Norway, May 12-15, 2025, Proceedings, Part V. Lecture Notes in Computer Science, vol. 15678, pp. 137–168. Springer (2025), https://doi.org/10.1007/978-3-031-91832-2_5
12. Garreta, A., Lipmaa, H., Luhaäär, U., Osadnik, M.: Cyclo: Lightweight lattice-based folding via partial range checks. IACR Cryptol. ePrint Arch. **2026**, 359 (2026), https://eprint.iacr.org/2026/359
13. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008. pp. 197–206. ACM (2008), https://doi.org/10.1145/1374376.1374407

14. Klooß, M., Lai, R.W.F., Nguyen, N.K., Osadnik, M.: Rok, paper, sissors toolkit for lattice-based succinct arguments - (extended abstract). In: Advances in Cryptology - ASIACRYPT 2024 - 30th International Conference on the Theory and Application of Cryptology and Information Security, Kolkata, India, December 9-13, 2024, Proceedings, Part V. pp. 203–235. Lecture Notes in Computer Science, Springer (2024). https://doi.org/10.1007/978-981-96-0935-2_7
15. Klooß, M., Lai, R.W.F., Nguyen, N.K., Osadnik, M.: Rok and roll - verifier-efficient random projection for $o\tilde{}(\lambda)$-size lattice arguments - (extended abstract). In: Advances in Cryptology - ASIACRYPT 2025 - 31st International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, VIC, Australia, December 8-12, 2025, Proceedings, Part III. pp. 297–329. Lecture Notes in Computer Science, Springer (2025). https://doi.org/10.1007/978-981-95-5099-9_10
16. Kothapalli, A., Parno, B.: Algebraic reductions of knowledge. In: Advances in Cryptology - CRYPTO 2023 - 43rd Annual International Cryptology Conference, CRYPTO 2023, Santa Barbara, CA, USA, August 20-24, 2023, Proceedings, Part IV. pp. 669–701. Lecture Notes in Computer Science, Springer (2023), https://doi.org/10.1007/978-3-031-38551-3_21
17. Kothapalli, A., Setty, S.T.V.: Hypernova: Recursive arguments for customizable constraint systems. In: Advances in Cryptology - CRYPTO 2024 - 44th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2024, Proceedings, 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
18. Kothapalli, A., Setty, S.T.V., Tzialla, I.: Nova: Recursive zero-knowledge arguments from folding schemes. In: Advances in Cryptology - CRYPTO 2022 - 42nd Annual International Cryptology Conference, CRYPTO 2022, Santa Barbara, CA, USA, August 15-18, 2022, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 13510, pp. 359–388. Springer (2022). https://doi.org/10.1007/978-3-031-15985-5_13
19. Kuriyama, S., Lai, R.W.F., Osadnik, M., Tucci, L.: SALSAA - sumcheck-aided lattice-based succinct arguments and applications. IACR Cryptol. ePrint Arch. **2025**, 2124 (2025), https://eprint.iacr.org/2025/2124
20. Langlois, A., Stehlé, D.: Worst-case to average-case reductions for module lattices. Des. Codes Cryptogr. **75**(3), 565–599 (2015). https://doi.org/10.1007/S10623-014-9938-4
21. Liu, X., Gao, S., Zheng, T., Xiao, B.: Snarkfold: Efficient SNARK proof aggregation from split incrementally verifiable computation. IACR Cryptol. ePrint Arch. **2023**, 1946 (2023)
22. Lyubashevsky, V., Nguyen, N.K., Plançon, M.: Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In: CRYPTO Part II. pp. 71–101 (2022). https://doi.org/10.1007/978-3-031-15979-4_3
23. Lyubashevsky, V., Seiler, G.: Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In: EUROCRYPT Part I. pp. 204–224 (2018). https://doi.org/10.1007/978-3-319-78381-9_8
24. Micciancio, D., , Regev, O.: Lattice-based cryptography. In: Post-Quantum Cryptography, pp. 147–191. Springer (2009). https://doi.org/10.1007/978-3-540-88702-7_5
25. Nguyen, N.K., Seiler, G.: Greyhound: Fast polynomial commitments from lattices. In: CRYPTO Part X. pp. 243–275 (2024). https://doi.org/10.1007/978-3-031-68403-6_8

26. Schwartz, J.T.: Fast probabilistic algorithms for verification of polynomial identities. J. ACM **27**(4), 701–717 (1980). https://doi.org/10.1145/322217.322225
27. Wee, H., Wu, D.J.: Succinct vector, polynomial, and functional commitments from lattices. In: EUROCRYPT Part III. pp. 385–416 (2023). https://doi.org/10.1007/978-3-031-30620-4_13
28. Zheng, T., Gao, S., Guo, Y., Xiao, B.: Kilonova: Non-uniform PCD with zero-knowledge property from generic folding schemes. IACR Cryptol. ePrint Arch. **2023**, 1579 (2023), https://eprint.iacr.org/2023/1579
29. Zheng, T., Gao, S., Liu, X.: Zero-knowledge proof-carrying data from accumulation schemes. IACR Cryptol. ePrint Arch. **2026**, 289 (2026), https://eprint.iacr.org/2026/289
30. Zippel, R.: Probabilistic algorithms for sparse polynomials. In: Symbolic and Algebraic Computation, EUROSAM '79, An International Symposiumon Symbolic and Algebraic Computation, Marseille, France, June 1979, Proceedings. Lecture Notes in Computer Science, vol. 72, pp. 216–226. Springer (1979). https://doi.org/10.1007/3-540-09519-5_73

A Additional Preliminaries

This appendix collects additional preliminaries used in the technical sections.

Gadget-matrix decomposition. Let $a \in \mathbb{N}$. Define the gadget vector $\vec{g}_a \coloneqq (1, a, a^2, \ldots, a^{\iota-1})$, where $\iota \coloneqq \lceil \log_a q \rceil$. Define the gadget matrix $G_{a,m} \coloneqq I_m \otimes \vec{g}_a^{\mathsf{T}} \in \mathbb{Z}_q^{m \times m\cdot\iota}$. We use $G_{a,m}^{-1}(\cdot)$ as notation for the (base-a) gadget decomposition operator, not a matrix inverse. For $\boldsymbol{A} \in \mathcal{R}_q^{m\times n}$, define $\tilde{\boldsymbol{A}} \coloneqq G_{a,m}^{-1}(\boldsymbol{A}) \in \mathcal{R}_q^{\iota m\times n}$ (applied coefficient-wise over $\mathcal{R}_q$) such that $\boldsymbol{A} = G_{a,m}\tilde{\boldsymbol{A}}$. We write G_m^{-1} when the basis a is clear.

Integer-to-ring mapping. We formally define the integer-to-ring mapping.

Definition 2. *Given a vector $\vec{f} = (f_0, f_1, \ldots, f_{nd-1}) \in \mathbb{Z}_q^{nd}$, $M_R \colon \mathbb{Z}_q^{nd} \to \mathcal{R}_q^n$ is defined as below. Each component is interpreted as an $\mathcal{R}_q$ element.*

$$M_R(\vec{f}) = \left(\sum_{j=0}^{d-1} f_j X^j,\ \sum_{j=0}^{d-1} f_{d+j} X^j,\ \ldots,\ \sum_{j=0}^{d-1} f_{(n-1)d+j} X^j \right).$$

A.1 Reduction-of-Knowledge Protocols

We recall the notion of reduction-of-knowledge protocols, following [16]. Informally, such a protocol transforms an instance of one relation into an instance of another relation, while preserving the ability to extract a witness for the original relation from any prover that successfully produces a valid reduced instance.

Definition 3 (Reduction of knowledge). *Let λ be the security parameter, and let $\mathcal{R}_1$ and $\mathcal{R}_2$ be two relations. A reduction-of-knowledge (rok) protocol Π from $\mathcal{R}_1$ to $\mathcal{R}_2$ consists of the following algorithms.*

- $\mathsf{Setup}(1^\lambda) \to \mathsf{pp}$ *outputs public parameters* pp.
- $\langle \mathcal{P}(\mathsf{pp}, u_1, w_1), \mathcal{V}(\mathsf{pp}, u_1) \rangle \to (u_2, w_2)$ *is an interactive protocol between a prover* $\mathcal{P}$ *and a verifier* $\mathcal{V}$. *The prover holds a witness* w_1 *such that* $(u_1, w_1) \in \mathcal{R}_1$. *At the end of the protocol, the verifier outputs either a reduced statement* u_2 *or* $\perp$, *and the honest prover additionally outputs a witness* w_2.

A reduction-of-knowledge protocol is required to satisfy the following properties.

Completeness. For every honestly generated public parameter $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ and every pair $(u_1, w_1) \in \mathcal{R}_1$, an honest execution of Π outputs (u_2, w_2) such that $(u_2, w_2) \in \mathcal{R}_2$, except with negligible probability.

Knowledge soundness. For every expected polynomial-time adversary $\mathcal{A}$ and malicious prover $\mathcal{P}^*$, there is an expected polynomial-time extractor $\mathcal{E}$ such that given $\mathsf{pp} \leftarrow \mathsf{Setup}(1^\lambda)$ and $(u_1, w_1^*) \leftarrow \mathcal{A}(\mathsf{pp})$,

$$\Pr\left[\left(u_1, \mathcal{E}(\mathsf{pp}, u_1, w_1^*)\right) \in \mathcal{R}_1\right] \approx \Pr\left[\langle \mathcal{P}^*(\mathsf{pp}, u_1, w_1^*), \mathcal{V}(\mathsf{pp}, u_1)\rangle \in \mathcal{R}_2\right].$$

Public reducibility. The protocol is publicly reducible if there exists a deterministic polynomial-time algorithm Reduce such that, for every transcript tr generated in an execution of Π on input (pp, u_1), the reduced statement is determined by public information alone. That is, if the verifier outputs $u_2 \neq \perp$, then $\mathsf{Reduce}(\mathsf{pp}, u_1, \mathsf{tr}) = u_2$. In particular, the statement transformation does not depend on the witness.

We will use the standard composition properties of reduction-of-knowledge protocols.

Theorem 7 (Sequential composition [16, Theorem 5]). *Let* $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3$ *be relations. If* Π_1 *is a reduction of knowledge from* $\mathcal{R}_1$ *to* $\mathcal{R}_2$ *and* Π_2 *is a reduction of knowledge from* $\mathcal{R}_2$ *to* $\mathcal{R}_3$, *then the sequentially composed protocol* $\Pi_1 \diamond \Pi_2$ *is a reduction of knowledge from* $\mathcal{R}_1$ *to* $\mathcal{R}_3$.

Theorem 8 (Parallel composition [16, Theorem 6]). *Let* $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \mathcal{R}_4$ *be relations. If* Π_1 *is a reduction of knowledge from* $\mathcal{R}_1$ *to* $\mathcal{R}_2$ *and* Π_2 *is a reduction of knowledge from* $\mathcal{R}_3$ *to* $\mathcal{R}_4$, *then the parallel composition* $\Pi_1 \times \Pi_2$ *is a reduction of knowledge from* $\mathcal{R}_1 \times \mathcal{R}_3$ *to* $\mathcal{R}_2 \times \mathcal{R}_4$.

B Proof of Theorem 5

Proof. Let $\mathcal{P}^*$ be a prover that makes the verifier accept with probability ϵ^*. We construct an extractor $\mathcal{E}_{\mathsf{bat}}$ that outputs openings for the input commitments in $\mathcal{R}_{\mathsf{back}}^{(p)}$ and then proves that these openings satisfy the preliminary backend relation.

The extractor proceeds as follows.

1. $\mathcal{E}_{\mathsf{bat}}$ runs $\mathcal{P}^*$ until the point immediately before the folding challenge $\vec{\theta}$ is sampled.

2. The extractor rewinds $\mathcal{P}^*$ at the folding-challenge point and queries it on challenges chosen from the special-soundness challenge space $SS(\mathcal{C}, \mathcal{N}, 2)$. The extractor runs the prover continuation on $\vec{\theta}^{(0)}, \vec{\theta}^{(1)}, \ldots, \vec{\theta}^{(\mathcal{N})}$ and keeps the rewinding attempt only if all $\mathcal{N}+1$ continuations are accepting. Hence the extractor reads the folded openings

$$\vec{\boldsymbol{f}}_{\theta^{(0)}}, \vec{\boldsymbol{f}}_{\theta^{(1)}}, \ldots, \vec{\boldsymbol{f}}_{\theta^{(\mathcal{N})}}.$$

3. For every $i \in [\mathcal{N}]$, the extractor computes

$$\widetilde{\vec{\boldsymbol{f}}}_i := \delta_i^{-1} \left(\vec{\boldsymbol{f}}_{\theta^{(i)}} - \vec{\boldsymbol{f}}_{\theta^{(0)}} \right).$$

for $\delta_i := \theta_i^{(i)} - \theta_i^{(0)}$. By the choice of the challenge space, every δ_i is invertible.

Expected running time. We apply Lemma 4 with $\mu = \mathcal{N}$ and $\widetilde{S} = \mathcal{C}$. For the fixed prefix, the algorithm $\mathcal{A}$ is the continuation of $\mathcal{P}^*$ after receiving the folding challenge $\vec{\theta}$, and the predicate Ψ checks whether the continuation is accepting.

By Lemma 4, the extractor obtains an accepting special-soundness fork with probability at least $\epsilon^* - \frac{\mathcal{N}}{|\mathcal{C}|}$. Therefore, by repeating the rewinding procedure until success, the expected running time of $\mathcal{E}_{\mathsf{bat}}$ is $O\left(\frac{1}{\epsilon^* - \mathcal{N}/|\mathcal{C}|}\right)$. Hence $\mathcal{E}_{\mathsf{bat}}$ runs in expected polynomial time whenever $\epsilon^* - \mathcal{N}/|\mathcal{C}|$ is non-negligible.

Commitment consistency and relaxed shortness. For every accepting continuation $r \in \{0, \ldots, \mathcal{N}\}$, the output witness satisfies $\mathsf{Com}(\vec{\boldsymbol{f}}_{\theta^{(r)}}) = \mathsf{cm}_f^{(r)}$. Moreover, the folded commitment is computed homomorphically as $\mathsf{cm}_f^{(r)} = \sum_{j=0}^{\mathcal{N}-1} \theta_j^{(r)} \mathsf{cm}_{f_j}$. Subtracting the center commitment equation from the i-th neighboring equation gives

$$\mathsf{Com}\left(\vec{\boldsymbol{f}}_{\theta^{(i)}} - \vec{\boldsymbol{f}}_{\theta^{(0)}} \right) = \delta_i \mathsf{cm}_{f_i}.$$

By homomorphism and invertibility of δ_i, we obtain $\mathsf{Com}(\widetilde{\vec{\boldsymbol{f}}}_i) = \mathsf{cm}_{f_i}$. Thus the extracted vectors open the corresponding input commitments of $\mathcal{R}_{\mathsf{back}}^{(p)}$.

Each accepting output witness satisfies the output admissible bound $\|\vec{\boldsymbol{f}}_{\theta^{(r)}}\|_\infty \leq B_{f,\infty}$. Thus we have

$$\|(\delta_i)\widetilde{\vec{\boldsymbol{f}}}_i\|_\infty = \left\| \left(\vec{\boldsymbol{f}}_{\theta^{(i)}} - \vec{\boldsymbol{f}}_{\theta^{(0)}} \right) \right\|_\infty \leq 2B_{f,\infty}.$$

Terminal evaluation claims. For every accepting continuation r, the output witness satisfies $\left\langle \mathsf{coef}(\vec{\boldsymbol{f}}_{\theta^{(r)}}), \widetilde{\mathsf{eq}}_{(\vec{z}_1, \vec{z}_2)} \right\rangle = \sum_{j=0}^{\mathcal{N}-1} \theta_j^{(r)} u_j$. Subtracting the center equation from the i-th neighboring equation gives $\left\langle \mathsf{coef}\left(\vec{\boldsymbol{f}}_{\theta^{(i)}} - \vec{\boldsymbol{f}}_{\theta^{(0)}} \right), \widetilde{\mathsf{eq}}_{(\vec{z}_1, \vec{z}_2)} \right\rangle = \delta_i u_i$. Dividing by δ_i yields

$$\left\langle \mathsf{coef}(\widetilde{\vec{\boldsymbol{f}}}_i), \widetilde{\mathsf{eq}}_{(\vec{z}_1, \vec{z}_2)} \right\rangle = u_i \qquad \forall i \in [\mathcal{N}].$$

Therefore the extracted openings satisfy all terminal evaluation claims.

Validity of the normalized sum-check claims. We define the bad events under which the extraction may fail to prove validity of the preliminary constraints.

$\mathsf{Bad}_{\mathsf{N}}$: the normalization step is unsound, namely some structured constraint is not equivalent to its Boolean-sum claim. By definition, $\Pr[\mathsf{Bad}_{\mathsf{N}}] \leq \epsilon_{\mathsf{N}}$. For coordinate-wise zero-padding checks, $\epsilon_{\mathsf{N}} \leq \frac{n_3 \cdot t(2t-2)}{q}$.

$\mathsf{Bad}_{\mathsf{batch}}$: at least one normalized claim is false, but the random linear combination using $\vec{\beta}$ is true. Let $\Delta_i := \sum_{\vec{b} \in \{0,1\}^{2t}} F_i(\vec{b}) - v_i$. If some claim is false, then $(\Delta_i)_{i \in [\mathcal{N}]} \neq 0$. The batched claim can still hold only if $\sum_{i=0}^{\mathcal{N}-1} \beta_i \Delta_i = 0$. Since $\vec{\beta}$ is sampled uniformly over $\mathbb{Z}_q^{\mathcal{N}}$ after the claims are fixed, $\Pr[\mathsf{Bad}_{\mathsf{batch}}] \leq \frac{1}{q}$.

$\mathsf{Bad}_{\mathsf{sc}}$: the bi-variate reduction accepts although the committed partial-sum tables are inconsistent with $g_b(\vec{X}) = \sum_i \beta_i F_i(\vec{X})$. Since g_b has individual degree at most 2 in $2t$ variables, each consistency failure is detected except with probability at most $2t/q$. There are two such checks, and hence $\Pr[\mathsf{Bad}_{\mathsf{sc}}] \leq \frac{4t}{q}$.

Conditioned on none of these bad events occurring, all normalized Boolean-sum claims are valid for the extracted openings $(\widetilde{\vec{f}}_i)_{i \in [\mathcal{N}]}$ and these openings satisfy the original structured inner-product constraints of $\mathcal{R}^{(p)}_{\mathsf{back}}$.

Combining the commitment consistency, terminal evaluation claims, normalized claim validity, and relaxed shortness bound, the extractor outputs a witness satisfying the relaxed version of $\mathcal{R}^{(p)}_{\mathsf{back}}$.

By a union bound over the bad events and the failure probability of the special-soundness rewinding, the total soundness error is

$$\epsilon_{\mathsf{bat}} \leq \epsilon_{\mathsf{N}} + \frac{1}{q} + \frac{4t}{q} + \frac{\mathcal{N}}{|\mathcal{C}|}.$$

This proves relaxed knowledge soundness.

□

C Concrete Cost Accounting and Evaluation

This section provides a concrete cost analysis of the reduction layer under the parameter-selection criteria of Section 4.5.

We instantiate the challenge space with sparse small-norm ring elements. Fix integers $w_0, w_1, w_2 \in [d]$ with $w_0 + w_1 + w_2 = d$, and define

$$\mathcal{P}_{w_0,w_1,w_2} := \left\{ c \in \mathcal{R}_q \;\middle|\; \begin{array}{l} \mathsf{coef}(c) \in \{0, \pm 1, \pm 2\}^d, \\ \#\{|\mathsf{coef}(c)_j| = i\} = w_i \text{ for } i \in \{0,1,2\} \end{array} \right\}.$$

For any distinct $c, c' \in \mathcal{P}_{w_0,w_1,w_2}$, the difference $c - c'$ has at most $2(w_1 + w_2)$ nonzero coefficients, each bounded by 4 in absolute value. Hence $0 < \|c - c'\|_2 \leq 4\sqrt{2(w_1 + w_2)}$. Therefore, if q is prime with $q \equiv 5 \bmod 8$ and $q > 64(w_1 + w_2)$, then every nonzero difference $c - c'$ is invertible in $\mathcal{R}_q$ by Lemma 3.

For polynomial multiplication in $\mathcal{R}_q$, we have $\|c\mathbf{v}\|_\infty \leq \|c\|_1 \cdot \|\mathbf{v}\|_\infty$, and hence $\|c\|_{\mathsf{op},\infty} \leq \|c\|_1 = w_1 + 2w_2$. Thus we may set $T := w_1 + 2w_2$.

Table 2. Concrete parameter setting for Π_{r}.

q	d	(w_1, w_2)	$\lvert\mathcal{C}\rvert$	m	ℓ	κ
32-bits	128	$(24, 3)$	$\approx 2^{128}$	1	8	11

The pool size is $|\mathcal{P}_{w_0,w_1,w_2}| = \binom{d}{w_1}\binom{d-w_1}{w_2}2^{w_1+w_2}$. We choose (w_0, w_1, w_2) so that this pool is large enough for the required challenge space, and take a subset $\mathcal{C} \subseteq \mathcal{P}_{w_0,w_1,w_2}$.

We then choose the commitment parameters so that MSIS remains hard for the extracted bound $\mathcal{B}$. Following the heuristic methodology of [24], an MSIS instance over a ring of degree d and modulus q is estimated to be hard for Euclidean norm bound $\mathcal{B}$ as long as

$$\mathcal{B} \leq \min\Big(2^{2\sqrt{d\kappa \log q \log \delta}},\, q\Big),$$

where δ is the target root-Hermite factor. For a concrete estimate targeting 128-bit security, one may take $\delta \approx 1.0045$ and require, up to lower-order constants, $q/2 > 2^{2\sqrt{d\kappa \log \delta \cdot \log q}}$. Equivalently, writing logarithms in base 2, it suffices that $\log q > 4d\kappa \log \delta$. For the concrete parameters below, the extracted admissible bound remains well below the modulus in the admissible input range, and the commitment scheme can be instantiated so that binding holds for all openings that may be extracted by the reduction.

C.1 Concrete efficiency estimate.

We use the parameter set in Table 2.

With $d = 128$ and $(w_1, w_2) = (24, 3)$, the challenge pool has size $|\mathcal{P}| = \binom{128}{24}\binom{104}{3}2^{27} \approx 2^{130.116}$, while the corresponding operator-norm bound is $T = w_1 + 2w_2 = 30$.

Since q is chosen as a 32-bit prime, the soundness error of each challenge-dependent phase is not negligible by itself. We therefore repeat the challenge-dependent phase $t = \left\lceil \frac{128}{\log_2 q - \log_2 n} \right\rceil$ times, which reduces the corresponding soundness error to at most 2^{-128}.

For the norm bound $\mathcal{B} = 4TB_{f,\infty}$, we require it to lie within the secure extraction bound. Concretely, it should satisfy $\sqrt{dn} \cdot 4TB_{f,\infty} \leq 2^{2\sqrt{d\kappa \log_2 q \log_2 \delta}}$. Equivalently, after taking logarithms, this condition becomes

$$\frac{1}{2}\log_2 d + \frac{1}{2}\log_2 n + \log_2\big(2T^2\mathcal{N}b\big) < 2\sqrt{d\kappa \log_2 q \log_2 \delta}.$$

Under our parameter setting, this yields a secure supported length of approximately 2^{19} ring elements, corresponding to approximately 2^{26} elements over $\mathbb{Z}_q$. Larger input lengths can be supported by adjusting the parameters accordingly.

The admissible input dimension is chosen to satisfy both the no-wraparound condition and the extracted-bound condition from Theorems 4 and 5.

Under the parameter setting of Table 2, a direct communication accounting yields a reduction-layer transcript of approximately 154 KB. By applying the packed-commitment optimization described in Appendix D, the transcript size is reduced to 27.69KB under a baseline soundness-amplification setting with 4 repetitions. Using a more conservative repetition criterion that accounts for the one-shot soundness error and the Fiat-Shamir security loss [3], the required number of repetitions increases to approximately 10-15, resulting in a transcript size of about 28.21KB.

We emphasize that this cost does not grow with the input dimension when measured in ring and field elements. Once the parameters are fixed and the input length lies in the admissible range, increasing N affects prover computation and the succinct tensor descriptions used by the verifier, but not the number of commitments or scalar claims in the reduction transcript.

We then estimate the verifier cost. The verifier's work is dominated by constructing succinct descriptions of the structured public vectors and computing the equality factors appearing in Π_{bat}. The number of $\mathbb{Z}_q$ operations is approximately

$$\underbrace{m + m\ell}_{\Pi_{\mathsf{alg}}} + \underbrace{(m+1)\cdot 4\ell}_{\Pi_{\mathsf{comp}}} + \underbrace{(2+\kappa)\mathcal{N} + 3\ell + \mathcal{M}\log(nd)}_{\Pi_{\mathsf{bat}}}.$$

For $n = 2^{20}$ under the above parameters, this is about 4.8×10^3 operations over $\mathbb{Z}_q$. For a 32-bit modulus this corresponds to sub-millisecond verification in an optimized implementation; even with a conservative high-level implementation, the expected verification time remains around the millisecond scale.

D Optimization for More Succinct Proof Size

The dominant contribution to the prover-side communication of the composed reduction Π_{r} comes from commitments to decomposed auxiliary witnesses. We describe a packing optimization that reduces these multiple commitments to a single commitment per witness family.

Let $w := \max\{m, \ell\}$. For a vector $\widehat{\boldsymbol{f}} = (\vec{\boldsymbol{f}}^{(0)}, \ldots, \vec{\boldsymbol{f}}^{(w-1)}) \in \mathcal{R}_q^{wn}$, where each block $\vec{\boldsymbol{f}}^{(i)} \in \mathcal{R}_q^n$, define the packed commitment

$$\mathsf{Com}^{\mathsf{pack}}_{wn}(\widehat{\boldsymbol{f}}) := \sum_{i=0}^{w-1} \mathsf{Com}_{p_i,n}(\vec{\boldsymbol{f}}^{(i)}),$$

where the public parameters $p_0, \ldots, p_{w-1}$ are independently sampled. Equivalently, this is an Ajtai-style commitment to a length-wn witness using the block matrix induced by the parameters $(p_i)_i$.

Binding of the packed commitment follows from the binding of this joint commitment, or equivalently from the corresponding MSIS assumption for the concatenated public matrix and the extracted norm bound.

This construction is compatible with the original commitments. For example, the original commitment to $\vec{\boldsymbol{s}}$ can be viewed as the packed commitment to

$(\vec{s}, \vec{0}, \ldots, \vec{0})$, namely

$$\mathsf{Com}^{\mathsf{pack}}_{wn}(\vec{s}, \vec{0}, \ldots, \vec{0}) = \mathsf{Com}_{p_0,n}(\vec{s}) = \mathsf{cm}_s.$$

The main use of packing is for decomposed auxiliary witnesses. Instead of sending commitments $(\mathsf{cm}_{h_0,j}, \mathsf{cm}_{h_1,j})_{j\in[\ell]}$ to the pieces $(\vec{\boldsymbol{h}}_{0,j}, \vec{\boldsymbol{h}}_{1,j})_{j\in[\ell]}$, we form

$$\widehat{\boldsymbol{h}}_0 := (\vec{\boldsymbol{h}}_{0,0}, \ldots, \vec{\boldsymbol{h}}_{0,\ell-1}, \vec{\mathbf{0}}, \ldots), \qquad \widehat{\boldsymbol{h}}_1 := (\vec{\boldsymbol{h}}_{1,0}, \ldots, \vec{\boldsymbol{h}}_{1,\ell-1}, \vec{\mathbf{0}}, \ldots)$$

as length-wn witnesses, padded with zero blocks when necessary, and send only

$$\widehat{\mathsf{cm}}_{h_0} := \mathsf{Com}^{\mathsf{pack}}_{wn}(\widehat{\boldsymbol{h}}_0), \qquad \widehat{\mathsf{cm}}_{h_1} := \mathsf{Com}^{\mathsf{pack}}_{wn}(\widehat{\boldsymbol{h}}_1).$$

The structured constraints are updated in the natural way. For example, a constraint of the form

$$\sum_{j=0}^{\ell-1} b^j \left\langle \mathsf{coef}(\vec{\boldsymbol{h}}_{0,j}), \chi \right\rangle = v$$

is equivalently written as

$$\left\langle \mathsf{coef}(\widehat{\boldsymbol{h}}_0), (1, b, \ldots, b^{\ell-1}, 0, \ldots, 0) \otimes \chi \right\rangle = v.$$

Thus packing only changes the representation of committed witnesses; it does not change the algebraic statement being proved.

We apply the same idea to all families of decomposed witnesses appearing in Π_{alg} and Π_{comp}. This reduces the number of commitments from one per decomposed piece to one per packed family, while preserving the same backend relation form.

It remains to describe the corresponding adaptation of Π_{bat}. The normalization step is unchanged. After normalization, we group the Boolean-sum claims according to the block of the packed witness to which their committed table belongs. This gives w groups of claims, and hence w batched polynomials $g_{b,0}, \ldots, g_{b,w-1}$. The verifier samples a fresh challenge $c \leftarrow \mathbb{Z}_q$, and both parties define $g_b(\vec{X}_1, \vec{X}_2) := \sum_{i=0}^{w-1} c^i g_{b,i}(\vec{X}_1, \vec{X}_2)$. Since each $g_{b,i}$ has individual degree at most 2, the combined polynomial g_b also has individual degree at most 2. Therefore the same bi-variate reduction as in Figure 4 applies without modification. The additional random linear combination contributes at most $(w-1)/q$ to the soundness error.

At the terminal point $(\vec{z}_1, \vec{z}_2)$, the prover provides the evaluations needed to compute each $g_{b,i}(\vec{z}_1, \vec{z}_2)$. The final folding is then performed blockwise. For each block $i \in [w]$, let $\vec{\boldsymbol{f}}_{\theta,i}$ denote the folded witness for the claims in that block, and let $u_{\theta,i}$ denote the corresponding folded terminal value:

$$\left\langle \mathsf{coef}(\vec{\boldsymbol{f}}_{\theta,i}), \widetilde{\mathsf{eq}}_{(\vec{z}_1,\vec{z}_2)} \right\rangle = u_{\theta,i}.$$

We pack these folded witnesses as

$$\widehat{\boldsymbol{f}}_\theta := (\vec{\boldsymbol{f}}_{\theta,0}, \ldots, \vec{\boldsymbol{f}}_{\theta,w-1}) \in \mathcal{R}_q^{wn}.$$

If desired, the w terminal equalities can be compressed into one structured constraint using an additional challenge e:

$$\left\langle \mathsf{coef}(\widehat{\boldsymbol{f}}_\theta), (1, e, \ldots, e^{w-1}) \otimes \widetilde{\mathsf{eq}}_{(\vec{z}_1, \vec{z}_2)} \right\rangle = \sum_{i=0}^{w-1} e^i u_{\theta,i}.$$

This again preserves the backend relation form, with an additional soundness loss at most $(w-1)/q$ if the individual block equalities are not retained.

After packing, the final backend instance contains a constant number of packed witnesses of length wn, namely the folded terminal witness and the two partial-sum witnesses. Since $w = \max\{m, \ell\} = O(1)$ in our parameter regime, this optimization does not change the asymptotic complexity. It reduces the concrete communication by replacing multiple commitments to decomposed pieces with a single packed commitment of size κ ring elements per family.

Adopting the parameter setting in Table 2, the reduction proof size can be reduced to 27.69KB.

E Folding Implementation for R1CS Instances.

Folding schemes reduce multiple instances of a relation into a single accumulated instance of the same or a closely related relation, thereby enabling efficient recursive and incremental proof composition. Recent works [18,17,29,28,21] develop this paradigm for R1CS and more general constraint systems, while LatticeFold and LatticeFold+ [5,6] adapt it to the lattice setting.

In this section, we show how arbitrary committed R1CS instances can be reduced to a foldable linear relation using the reduction techniques developed in this work. For this application, the target is a dedicated linear accumulator relation $\mathcal{R}_{\mathsf{lin}}$, which is closer to the accumulator interface used in LatticeFold+ [6], rather than the backend relation $\mathcal{R}_{\mathsf{back}}$ used in our main exact-norm reduction.

The Committed R1CS Relation Let $N = nd = 2^{2t}$, and let $A, B, C \in \mathbb{Z}_q^{N \times N}$ define an R1CS instance. Let $\vec{\boldsymbol{f}} \in \mathcal{R}_q^n$ be the committed witness and write $\vec{f} = \mathsf{coef}(\vec{\boldsymbol{f}}) \in \mathbb{Z}_q^N$. We define

$$\mathcal{R}_{\mathsf{R1CS}} = \left\{ \begin{array}{l|l} \mathbf{p} = (A, B, C, B_f), & (A\vec{f}) \circ (B\vec{f}) = C\vec{f}, \\ \mathbf{x} = \mathsf{cm}_f, & \mathsf{Com}(\vec{\boldsymbol{f}}) = \mathsf{cm}_f, \\ \mathbf{w} = \vec{\boldsymbol{f}} & \|\vec{\boldsymbol{f}}\| \le B_f. \end{array} \right\}.$$

The Foldable Linear Relation We define the foldable linear relation $\mathcal{R}_{\mathsf{lin}}$ as follows.

Let $(M^{(i)})_{i\in[n_{\mathsf{lin}}]}$ be public linear maps over $\mathbb{Z}_q^N$, and let $(\vec{t}_i)_{i\in[n_{\mathsf{lin}}]}$ be public evaluation selectors with tensor structure.

$$\mathcal{R}_{\mathsf{lin}} = \left\{ \begin{array}{l|l} \mathbf{p} = \big(b_\infty, \chi_{\mathsf{bool}}, (M^{(i)})_{i\in[n_{\mathsf{lin}}]}\big), & \mathsf{Com}(\vec{\boldsymbol{f}}) = \mathsf{cm}_f,\ \mathsf{Com}(\vec{\boldsymbol{g}}) = \mathsf{cm}_g, \\ \mathbf{x} = \Big(\mathsf{cm}_f, \mathsf{cm}_g, \vec{t}, \vec{v}, & \mathsf{Com}(\vec{\boldsymbol{h}}_{a,j}) = \mathsf{cm}_{h_a,j}, a \in \{0,1\}, j \in [\ell], \\ \quad (\mathsf{cm}_{h_0,j}, \mathsf{cm}_{h_1,j})_{j\in[\ell]}, & \langle M^{(i)}\vec{f}, \vec{t}\rangle = v_i, i \in [n_{\mathsf{lin}}],\ \langle \vec{g}, \vec{t}_g\rangle = v_g, \\ \quad \vec{t}_0, \vec{t}_1, (v_{h_0,j}, v_{h'_0,j}, & \langle \vec{h}_{0,j}, \chi_{\mathsf{bool}}\rangle = v_{h_0,j},\ \langle \vec{h}_{0,j}, \vec{t}_0\rangle = v_{h'_0,j}, \\ \quad v_{h_1,j}, v_{h'_1,j})_{j\in[\ell]}\Big), & \langle \vec{h}_{1,j}, \chi_{\mathsf{bool}}\rangle = v_{h_1,j},\ \langle \vec{h}_{1,j}, \vec{t}_1\rangle = v_{h'_1,j}, \\ \mathbf{w} = \big(\vec{\boldsymbol{f}}, \vec{\boldsymbol{g}}, (\vec{\boldsymbol{h}}_{0,j}, \vec{\boldsymbol{h}}_{1,j})_{j\in[\ell]}\big) & \|\mathbf{w}\|_\infty \le b_\infty. \end{array} \right\},$$

where the parameters in $\mathbf{p}$ are fixed.

E.1 Reduction from R1CS to $\mathcal{R}_{\mathsf{lin}}$

We now describe the reduction as an interactive protocol.

Quadratic constraints to linear constraints. We first describe the way to reduce constraints in $\mathcal{R}_{\mathsf{R1CS}}$ to a batch of structured linear constraints.

1. $\mathcal{P}$: The prover decomposes the committed witness into binary pieces $\vec{\boldsymbol{f}} = \sum_{i=0}^{m-1} 2^i \vec{\boldsymbol{f}}_i$, and sends commitments to $(\vec{\boldsymbol{f}}_i)_{i\in[m]}$.
2. $\mathcal{V}$: sample $\vec{r}_b$, and $(\vec{r}_i)_{i\in[m]}$ from $\mathbb{Z}_q^{2t}$.
3. $\mathcal{P}$: Represent the constraints in $\mathcal{R}_{\mathsf{R1CS}}$ as follows:
   - $\langle A\vec{f}, B\vec{f} \circ (\widetilde{\mathsf{eq}}_{\vec{r}_b})\rangle = \langle C\vec{f}, \widetilde{\mathsf{eq}}_{\vec{r}_b}\rangle = v_c$;
   - $\langle \vec{f}, \vec{f}\rangle = v_f \le B^2$;
   - For $i \in [m]$: $\langle \vec{f}_i, \vec{f}_i \circ \widetilde{\mathsf{eq}}_{\vec{r}_i}\rangle = \langle \vec{f}_i, \widetilde{\mathsf{eq}}_{\vec{r}_i}\rangle = v_{f,i}$.
4. $\mathcal{V}$: batching challenge $\vec{c} \in \mathbb{Z}_q^{m+2}$.
5. $\mathcal{P}$: Construct the batched polynomial:

$$g(\vec{X}) := c_0\, \widetilde{Af}(\vec{X})\, \widetilde{(Bf)}_r(\vec{X}) + c_1\, \widetilde{f}(\vec{X})\widetilde{f}(\vec{X}) + \sum_{i=0}^{m-1} c_{i+2}\, \widetilde{f}_i(\vec{X})\, \widetilde{(f_i)}_r(\vec{X}),$$

   where

$$(Bf)_r := (B\vec{f}) \circ \widetilde{\mathsf{eq}}_{\vec{r}_b}, \qquad (f_i)_r := \vec{f}_i \circ \widetilde{\mathsf{eq}}_{\vec{r}_i}.$$

   Since each term has individual degree at most 2, so does g.
6. $\mathcal{P}$ and $\mathcal{V}$: The prover and verifier run the bi-variate reduction of Figure 2 on the claim

$$\sum_{\vec{b}_1,\vec{b}_2\in\{0,1\}^t} g(\vec{b}_1, \vec{b}_2) = c_0 v_c + c_1 v_f + \sum_{i=0}^{m-1} c_{i+2} v_{f,i}.$$

   which reduce to the following claims:

$$\left\langle A\vec{f}, \widetilde{\mathsf{eq}}_{\vec{\tau}}\right\rangle = u_a,\ \left\langle \vec{f}, \widetilde{\mathsf{eq}}_{\vec{\tau}}\right\rangle = u_f,\ \left\langle \vec{f}_i, \widetilde{\mathsf{eq}}_{\vec{\tau}}\right\rangle = u_{f,i},\ i \in [m],$$

$$\left\langle B\vec{f}, \widetilde{\mathsf{eq}}_{\vec{r}_b} \circ \widetilde{\mathsf{eq}}_{\vec{\tau}} \right\rangle = u_{b,r},\ \left\langle \vec{f}_i, \widetilde{\mathsf{eq}}_{\vec{r}_i} \circ \widetilde{\mathsf{eq}}_{\vec{\tau}} \right\rangle = u_{f_i,r}. \forall i \in [m]$$

and

$$\langle \vec{h}_{0,j}, \chi_{\mathsf{bool}} \rangle = u_{0,j}, \quad \langle \vec{h}_{1,j}, \chi_{\mathsf{bool}} \rangle = u_{1,j}.$$
$$\langle \vec{h}_{0,j}, \chi_{\vec{\tau}_1} \rangle = u_{0,z,j}, \quad \langle \vec{h}_{1,j}, \chi_{\vec{\tau}_1} \rangle = u_{1,z,j}.$$

Inner batching for linear constraints. We next batch the linear constraints obtained above and reduce them to the form required by $\mathcal{R}_{\mathsf{lin}}$. The first step is the normalization procedure from Section 4.3, which rewrites each structured linear constraint as a Boolean-sum claim.

For constraint

$$\left\langle B\vec{f} \circ \widetilde{\mathsf{eq}}_{\vec{r}_b}, \widetilde{\mathsf{eq}}_{(\vec{\tau}_1,\vec{\tau}_2)} \right\rangle = u_{b_r},$$

it can be equivalently written as

$$\left\langle B\vec{f}, \widetilde{\mathsf{eq}}_{\vec{r}_b} \circ \widetilde{\mathsf{eq}}_{(\vec{\tau}_1,\vec{\tau}_2)} \right\rangle = u_{b_r}.$$

Let $\vec{\tau} = (\vec{\tau}_1, \vec{\tau}_2)$. Since

$$\widetilde{\mathsf{eq}}_{\vec{r}_b} \circ \widetilde{\mathsf{eq}}_{\vec{\tau}} = \bigotimes_{i=0}^{2t-1} \big((1-r_{b,i})(1-\tau_i),\ r_{b,i}\tau_i\big),$$

the resulting public selector still has arity-2 tensor structure. Thus this type of constraint is normalized exactly as the arity-2 tensor-structured constraints in Section 4.3.

After normalizing all linear constraints, we obtain $\mathcal{N}$ Boolean-sum claims. We batch them by a random linear combination and apply the same bi-variate reduction as before. This reduces the batched claim to terminal linear constraints of the form

$$\left\langle A\vec{f}, \widetilde{\mathsf{eq}}_{(\vec{z}_1,\vec{z}_2)} \right\rangle = v_{f_a},\ \left\langle B\vec{f}, \widetilde{\mathsf{eq}}_{(\vec{z}_1,\vec{z}_2)} \right\rangle = v_{f_b},\ \left\langle C\vec{f}, \widetilde{\mathsf{eq}}_{(\vec{z}_1,\vec{z}_2)} \right\rangle = v_{f_c}.$$

The bi-variate reduction also produces intermediate witnesses $(\vec{\boldsymbol{h}}'_{0,j}, \vec{\boldsymbol{h}}'_{1,j})_{j\in[\ell]}$ satisfying

$$\left\langle \vec{h}'_{0,j}, \chi_{\mathsf{bool}} \right\rangle = u_{0,j}, \qquad \left\langle \vec{h}'_{1,j}, \chi_{\mathsf{bool}} \right\rangle = u_{1,j},$$

and

$$\left\langle \vec{h}'_{0,j}, \chi_{\vec{z}_1} \right\rangle = u_{0,z,j}, \qquad \left\langle \vec{h}'_{1,j}, \chi_{\vec{z}_2} \right\rangle = u_{1,z,j}.$$

The remaining terminal evaluation claims for the committed witnesses are

$$\left\langle \big[\vec{f}^{\top},\ (\vec{f}_i^{\top})_{i\in[m]},\ (\vec{h}_{0,j}^{\top}, \vec{h}_{1,j}^{\top})_{j\in[\ell]}\big], \widetilde{\mathsf{eq}}_{(\vec{z}_1,\vec{z}_2)} \right\rangle = \vec{u}.$$

We then fold this witness group into a single committed witness $\vec{g}$. The resulting statement consists of the public linear-map claims for $A\vec{f}$, $B\vec{f}$, and $C\vec{f}$, the folded terminal claim for $\vec{\boldsymbol{g}}$, and the intermediate-table constraints above. This is precisely an instance of $\mathcal{R}_{\mathsf{lin}}$.

E.2 Folding $\mathcal{R}_{\mathsf{lin}} \times \mathcal{R}_{\mathsf{lin}} \to \mathcal{R}_{\mathsf{lin}}$

After reducing an R1CS instance to an instance of $\mathcal{R}_{\mathsf{lin}}$, we can fold two instances of $\mathcal{R}_{\mathsf{lin}}$ into a single instance of the same relation.

Let $(\mathbf{x}_{\mathsf{lin}}^{(1)}, \mathbf{w}_{\mathsf{lin}}^{(1)})$ and $(\mathbf{x}_{\mathsf{lin}}^{(2)}, \mathbf{w}_{\mathsf{lin}}^{(2)})$ be two valid instances, where the witness of the b-th instance contains

$$\mathbf{w}_{\mathsf{lin}}^{(b)} = \big(\vec{\boldsymbol{f}}^{(b)}, \vec{\boldsymbol{g}}^{(b)}, (\vec{\boldsymbol{h}}_{0,j}^{(b)}, \vec{\boldsymbol{h}}_{1,j}^{(b)})_{j\in[\ell]}\big), \qquad b \in \{1,2\}.$$

The folded primary witness $\vec{\boldsymbol{f}}$ is obtained by taking the prescribed linear combination of the two primary witnesses $\vec{\boldsymbol{f}}^{(1)}$ and $\vec{\boldsymbol{f}}^{(2)}$.

The remaining constraints, namely those involving

$$\vec{\boldsymbol{g}}^{(1)}, \vec{\boldsymbol{g}}^{(2)} \quad \text{and} \quad (\vec{\boldsymbol{h}}_{0,j}^{(1)}, \vec{\boldsymbol{h}}_{1,j}^{(1)}, \vec{\boldsymbol{h}}_{0,j}^{(2)}, \vec{\boldsymbol{h}}_{1,j}^{(2)})_{j\in[\ell]},$$

are batched using the same procedure as in Π_{bat}. This batching step produces a new folded terminal witness $\vec{\boldsymbol{g}}$.

Finally, the new witnesses $(\vec{\boldsymbol{h}}_{0,j}, \vec{\boldsymbol{h}}_{1,j})_{j\in[\ell]}$ are the intermediate tables generated by the bi-variate batching reduction. Therefore, the output witness again has the form $\mathbf{w}_{\mathsf{lin}} = \big(\vec{\boldsymbol{f}}, \vec{\boldsymbol{g}}, (\vec{\boldsymbol{h}}_{0,j}, \vec{\boldsymbol{h}}_{1,j})_{j\in[\ell]}\big)$, and the output statement is an instance of the same relation $\mathcal{R}_{\mathsf{lin}}$.

Thus, a fresh R1CS instance can be accumulated with an existing linear accumulator through the transition

$$\mathcal{R}_{\mathsf{R1CS}} \times \mathcal{R}_{\mathsf{lin}} \longrightarrow \mathcal{R}_{\mathsf{lin}}.$$

Under the parameter setting in Table 2, and with the packing optimization of Appendix D, the direct reduction-layer communication for one such folding step is approximately 27.75KB at the 128-bit security level. And for fixed reduction parameters, the transcript size does not grow with the input length when measured in ring and field elements.

F Complete constraints for $\mathcal{R}_{\mathsf{back}}^{(p)}$

The preliminary backend relation $\mathcal{R}_{\mathsf{back}}^{(p)}$ consists of all pairs $(\mathbf{x}, \mathbf{w})$ satisfying the following constraints.

1. **Commitment consistency.** All committed witnesses are consistent with their public commitments.
2. **Structured linear constraints for the main quadratic relation.** The witnesses associated with the reduction of $\mathsf{ct}\big(\langle \vec{\boldsymbol{s}}, \sigma_{-1}(\vec{\boldsymbol{s}})\rangle\big) = v_s$ satisfy the following structured inner-product constraints. First, the evaluation of $\vec{\boldsymbol{s}}$ at the final challenge point is enforced by

$$\big\langle \mathsf{coef}(\vec{\boldsymbol{s}}), \widetilde{\mathsf{eq}}_{\vec{z}_1} \otimes \widetilde{\mathsf{eq}}_{\vec{z}_2} \big\rangle = u_s.$$

Next, for every $j \in [\ell]$, the Boolean-slice sums of the decomposed h_0-tables are claimed as

$$\left\langle \mathsf{coef}(\vec{\boldsymbol{h}}_{0,j}^{(s)}), \chi_{\mathsf{bool}} \right\rangle = v_{h_0,j}^{(s)}, \quad \text{for } \sum_{j=0}^{\ell-1} b^j v_{h_0,j}^{(s)} = v_s.$$

The consistency between $h_0(\vec{z}_1)$ and the Boolean sum of h_1 is expressed by

$$\left\langle \mathsf{coef}(\vec{\boldsymbol{h}}_{0,j}^{(s)}), \chi_{\vec{z}_1} \right\rangle = v_{h_0,z,j}^{(s)}, \qquad \left\langle \mathsf{coef}(\vec{\boldsymbol{h}}_{1,j}^{(s)}), \chi_{\mathsf{bool}} \right\rangle = v_{h_1,j}^{(s)} \quad \forall j \in [\ell],$$

together with $\sum_{j=0}^{\ell-1} b^j v_{h_0,z,j}^{(s)} = \sum_{j=0}^{\ell-1} b^j v_{h_1,j}^{(s)}$.
Finally, the terminal check $h_1(\vec{z}_2) = u_s^2$ is enforced by

$$\left\langle \mathsf{coef}(\vec{\boldsymbol{h}}_{1,j}^{(s)}), \chi_{\vec{z}_2} \right\rangle = v_{h_1,z,j}^{(s)} \ \forall j \in [\ell], \quad \text{for } \sum_{j=0}^{\ell-1} b^j v_{h_1,z,j}^{(s)} = u_s^2.$$

3. **Structured linear constraints for the slice-wise quadratic relations.** For each $i \in [m], j \in [\ell]$, the witness components associated with the reduction of

$$\mathsf{ct}\left(\left\langle \sum_{j=0}^{\ell-1} b^j \vec{\boldsymbol{s}}_{r_i,j}, \sigma_{-1}(\vec{\boldsymbol{s}}_i) \right\rangle\right) = \mathsf{ct}(\langle \vec{\boldsymbol{s}}_{r_i}, \sigma_{-1}(\vec{\boldsymbol{s}}_i)\rangle) = v_{r,i}$$

satisfy the following slice-wise structured inner-product constraints.
First, the evaluations of the two multiplicands at the final challenge point are enforced by

$$\left\langle \mathsf{coef}(\vec{\boldsymbol{s}}_i), \widetilde{\mathsf{eq}}_{\vec{z}_1}^{(i)} \otimes \widetilde{\mathsf{eq}}_{\vec{z}_2}^{(i)} \right\rangle = u_{s_i}, \quad \left\langle \mathsf{coef}(\vec{\boldsymbol{s}}_{r_i,j}), \widetilde{\mathsf{eq}}_{\vec{z}_1}^{(i)} \otimes \widetilde{\mathsf{eq}}_{\vec{z}_2}^{(i)} \right\rangle = u_{s_r,i,j}$$

for $j \in [\ell]$, with the recombination condition $\sum_{j=0}^{\ell-1} b^j u_{s_r,i,j} = u_{s_r,i}$.
Next, for every $j \in [\ell]$, the Boolean-slice sums of the decomposed h_0-tables are claimed as

$$\left\langle \mathsf{coef}(\vec{\boldsymbol{h}}_{0,j}^{(i)}), \chi_{\mathsf{bool}} \right\rangle = v_{h_0,j}^{(i)}, \quad \text{For } \sum_{j=0}^{\ell-1} b^j v_{h_0,j}^{(i)} = v_{r,i}.$$

The consistency between $h_0^{(i)}(\vec{z}_1^{(i)})$ and the Boolean sum of $h_1^{(i)}$ is expressed by

$$\left\langle \mathsf{coef}(\vec{\boldsymbol{h}}_{0,j}^{(i)}), \chi_{\vec{z}_1^{(i)}} \right\rangle = v_{h_0,z,j}^{(i)}, \quad \left\langle \mathsf{coef}(\vec{\boldsymbol{h}}_{1,j}^{(i)}), \chi_{\mathsf{bool}} \right\rangle = v_{h_1,j}^{(i)} \quad \forall j \in [\ell],$$

together with

$$\sum_{j=0}^{\ell-1} b^j v_{h_0,z,j}^{(i)} = \sum_{j=0}^{\ell-1} b^j v_{h_1,j}^{(i)}.$$

Finally, the terminal check is enforced by

$$\left\langle \mathsf{coef}(\vec{\boldsymbol{h}}_{1,j}^{(i)}), \chi_{\vec{z}_2^{(i)}} \right\rangle = v_{h_1,z,j}^{(i)} \; \forall j \in [\ell], \quad \text{for } \sum_{j=0}^{\ell-1} b^j v_{h_1,z,j}^{(i)} = u_{s_i} \cdot u_{s_r,i}.$$

4. **Inherited structured linear constraints.** All linear constraints already present in $\mathcal{R}_{\mathsf{norm,al}}$ are inherited in the preliminary backend instance. We express them in slice-wise form as follows. First, the binary decomposition consistency of $\vec{\boldsymbol{s}}$ is enforced by the claims

$$\langle \mathsf{coef}(\vec{\boldsymbol{s}}_i), \vec{\gamma} \rangle = v_{\mathsf{dec},i} \quad \forall i \in [m], \quad \text{for } \sum_{i=0}^{m-1} 2^i v_{\mathsf{dec},i} = v_{\mathsf{dec}},$$

together with $\langle \mathsf{coef}(\vec{\boldsymbol{s}}), \vec{\gamma} \rangle = v_{\mathsf{dec}}$.
Next, for each $i \in [m]$, the randomized binary-check claim is inherited as $\langle \mathsf{coef}(\vec{\boldsymbol{s}}_i), \vec{r}_i \rangle = v_{r,i}$.
Finally, the well-formedness of the lifted and decomposed witness $\vec{\boldsymbol{s}}_{r_i} = \sum_{j=0}^{\ell-1} b^j \vec{\boldsymbol{s}}_{r_i,j}$ is enforced by the slice-wise claims

$$\langle \mathsf{coef}(\vec{\boldsymbol{s}}_{r_i,j}), \vec{\alpha}_i \rangle = v_{\alpha,i,j} \quad \forall j \in [\ell], \quad \text{for } \sum_{j=0}^{\ell-1} b^j v_{\alpha,i,j} = v_{\alpha,i},$$

and the consistency check $\langle \mathsf{coef}(\vec{\boldsymbol{s}}_i), \vec{\alpha}_i \circ \vec{r}_i \rangle = v_{\alpha,i}$ for $i \in [m]$.
5. **Relaxed shortness.** All committed witnesses satisfy the relaxed shortness bound $b_{w,\infty} := \max\left\{\frac{b}{2}, B_\infty\right\}$.