FROM MATHEMATICAL TO EXECUTABLE CERTIFICATES FOR MACHINE UNLEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine unlearning is needed when data must be removed because of deletion requests, outdated records, or data-quality concerns, while retraining from scratch can be costly. Certified machine unlearning methods provide mathematical guarantees, while deployed systems release concrete finite-precision artifacts produced by software. To bridge the gap between mathematical guarantees and practical deployment, we introduce Executable Release Certification (ExecCert), a release-time layer that certifies the candidate artifact considered for release. ExecCert either closes a method's native certificate for the executed candidate or applies Retraining-Reference Release Verification (RRV) to certify fidelity to current retain-set retraining. Sequential deletion makes the latter nontrivial because the exact retain-set reference and the stored numerical state evolve separately. For frozen representations with a mutable ridge head, we develop an incremental realization of RRV that maintains certified evidence across deletion requests rather than reconstructing it at each release. On four published unlearning implementations, ExecCert preserves valid certificates, changes release decisions, tightens conservative bounds, and identifies the retraining-reference fidelity supported by concrete outputs. In sequential-service experiments, RRV eliminates false releases caused by stored-equation verification while closely tracking realized error, and incremental certification remains cheaper than both fresh and maintained verified-factor alternatives once release checks become sufficiently frequent.

1 INTRODUCTION

Machine-learning models may need to remove the influence of training records when users withdraw consent, data become outdated, or errors are discovered; regulations such as the EU's right to be forgotten provide an additional motivation. Retraining from scratch on the retained data gives a natural reference but can be costly, motivating machine unlearning methods that seek to approximate retraining at lower deletion cost (Ginart et al., 2019; Guo et al., 2020; Sekhari et al., 2021; Gupta et al., 2021). Certified unlearning further provides mathematical guarantees relating an efficient deletion mechanism to this reference (Guo et al., 2020; Sekhari et al., 2021; Gupta et al., 2021; Chien et al., 2024; Zhang et al., 2024b; Koloskova et al., 2025; Pandey et al., 2026). However, a deployed system releases a concrete finite-precision artifact produced by software, rather than the mathematical object analyzed by the guarantee. This raises a distinct deployment question: *does the artifact actually considered for release satisfy the intended certificate or retraining-reference contract?*

Sequential deletion makes this deployment-level question especially important. After each request, the exact retain-set system changes, while the deployed service advances a separate finite-precision state through rounded updates. These two systems do not always coincide: a candidate may solve the stored equation accurately, sometimes even with zero stored residual, but still violate the contract defined relative to current retain-set retraining. Verifying the stored computation alone is therefore insufficient; release-time certification must connect the executed artifact to the intended reference.

We introduce *Executable Release Certification* (ExecCert) to fill the gap between the two systems. Before a candidate is released, the deployment fixes a release contract, and ExecCert certifies whether the concrete executed artifact satisfies that contract. It supports two paths: when an unlearning method already exposes a certificate quantity, *native closure* rigorously encloses that quantity for the executed candidate; when the contract is defined relative to retraining on the current retain

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

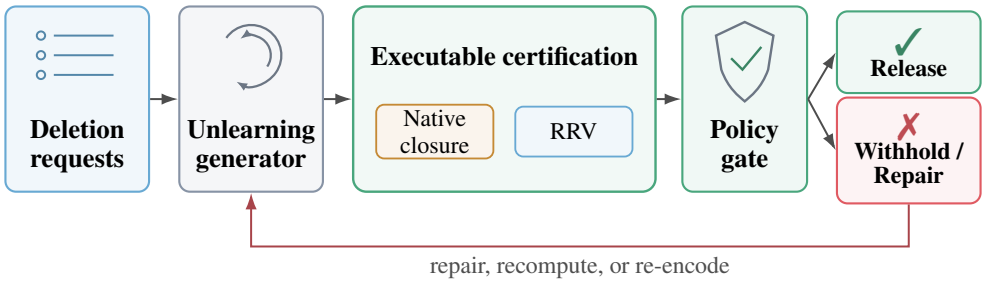

Figure 1: **ExecCert certifies a concrete unlearning artifact before release.** The generator's finite-precision output is checked against a predeclared contract through native-certificate closure or retraining-reference verification (RRV). Only certified artifacts pass the policy gate; an inconclusive candidate is withheld or repaired and then recertified.

set, *Retraining-Reference Release Verification* (RRV) instead certifies the candidate directly against that reference. In both cases, the output generator, release contract, and verifier remain separate components.

Our main sequential realization is *Incremental RRV* for ridge regression heads with symmetric positive-definite (SPD) normal matrices. RRV itself certifies only the concrete candidate considered for release. During the preceding deletion sequence, Incremental RRV maintains certified relations between the service's finite-precision statistics and the exact retain-set statistics, together with reusable inverse-defect evidence. Here d denotes the feature dimension of each ridge regression head. Guarded rank-one transitions update this proof state in $O(d^2)$ work. When a candidate is considered for release, the maintained evidence is composed into its retraining-reference certificate. Alternatively, On-demand RRV constructs the required evidence when certification is requested.

Experiments on four published unlearning implementations and sequential services show that ExecCert changes release decisions, tightens or preserves existing certificates, and characterizes the retraining-reference fidelity supported by emitted outputs. In sequential services, RRV prevents false releases admitted by stored-equation checking while closely tracking realized error; on frozen 3B representations with a mutable ridge head, Incremental RRV remains cheaper than both fresh and maintained verified-factor alternatives once release checks become sufficiently frequent.

Contributions. We make three contributions. First, we formalize *executable release certification* as a distinct layer between a mathematical unlearning mechanism and the concrete artifact considered for release. ExecCert supports both rigorous closure of a method's native certificate and certification against current retain-set retraining (Sections 3 and 4.1). Second, we develop Incremental RRV for sequential deletion, maintaining certified relations between finite-precision service state and the exact retain-set system together with reusable inverse-defect evidence. Guarded rank-one transitions update this proof state in $O(d^2)$ work (Sections 4 and 4.3). Third, we evaluate ExecCert on four published unlearning implementations and sequential services, showing that executable certification has practical consequences: it can change release decisions, tighten or preserve existing certificates, prevent false releases, and amortize verification when certification is frequent (Section 5).

2 Related Work

Full retraining on the retain set is a natural exact reference but incurs the cost of processing the surviving data. Exact unlearning targets the same output distribution and need not literally rerun the original training algorithm; certified approximate removal instead applies an efficient update and bounds its deviation from the declared reference, through influence or Newton steps, randomized mechanisms, or guarantees that compose across a deletion sequence (Guo et al., 2020; Sekhari et al., 2021; Gupta et al., 2021; Chien et al., 2024; Koloskova et al., 2025; Zou et al., 2025; Pandey et al., 2026). These results differ in what they bound, but share an assumption: the analysis governs the mathematical update, and the deployed artifact is taken to realize it. A second line asks instead how an external party can be convinced that deletion happened, auditing execution traces and provenance, probing the released model for residual influence of the deleted records, or attaching cryptographic proofs that a prescribed computation was carried out (Thudi et al., 2022; Weng et al., 2024; Eisenhofer

et al., 2025; Zhang et al., 2024a; Maheri et al., 2026). That evidence concerns *which computation ran*, for a verifier that distrusts the operator. Ours is the complement: the verifier is internal, the computation known, and the question is whether the artifact about to be released meets a contract stated in exact arithmetic. The two compose, since proving that the prescribed update ran says nothing about the rounding it accumulated.

That numerical question has its own literature. Finite precision is known to break guarantees derived in exact arithmetic, in differential privacy, robustness certification, and verified training (Mironov, 2012; Voracek & Hein, 2023; Szász et al., 2025; Jin et al., 2024), and verified numerics supplies the countermeasure: rigorous enclosures for linear-system residuals, approximate inverses, and low-rank updates (Krawczyk, 1969; Oishi & Rump, 2002; Rump, 2010; Hager, 1989). These tools certify a solution against a *fixed* system. Cholesky downdates and iterative refinement provide classical ways to update a numerical solve (Higham, 2002; Rump, 2010); maintaining such a factor does not by itself verify its relation to the changing exact system, so factor validation and residual enclosures remain part of a release certificate. Sequential deletion removes that fixture, since the retain-set system moves with every request while the service advances a separately rounded state, so our ridge regression realization maintains a certified relation between the two and converts it at release time into a certificate against the current reference. ScaleGUN studies a related transfer for graph propagation (Yi & Wei, 2025); RRV instead transfers from finite-precision service state to the retain-set system.

3 From Mathematical to Executable Certificates

An unlearning generator processes a declared set or sequence of deletion requests and produces a finite-precision candidate $\widetilde{\theta}$. The deployment fixes an admissible set $\mathcal{A}$ independently of that candidate. ExecCert returns CERTIFIED only after establishing $\widetilde{\theta} \in \mathcal{A}$; otherwise the policy gate may repair, recompute, re-encode, or withhold the candidate.

For evaluation, a candidate is *conforming* when it belongs to $\mathcal{A}$, and a *false release* is a nonconforming candidate returned as CERTIFIED. An INCONCLUSIVE candidate may still conform; a *released artifact* is one that passes the policy gate.

Notation. We write θ^{R} for the exact-real retraining reference after the declared deletions and $\widetilde{\theta}$ for the executed candidate. Hats denote computed verifier quantities and circles exact-real evaluations of computed intermediates; Section C.1 gives the remaining floating-point notation.

3.1 Two ExecCert Paths

These paths correspond to two release requirements. Some deployments need to verify a method's own certificate after finite-precision execution; others specify fidelity to retraining on the current retain set, which a native certificate may not express. Native closure addresses the first requirement, whereas RRV addresses the second.

Native-certificate closure. This path is appropriate when an unlearning method already specifies a certificate of the form $q(\theta; \mathcal{I}) \le b$, where q is the source certificate quantity, b its prescribed threshold, and $\mathcal{I}$ any committed inputs or auxiliary state required by the source method. This defines the admissible set $\mathcal{A}^{\mathrm{native}} = \{\theta \colon q(\theta; \mathcal{I}) \le b\}$. ExecCert does not change this condition: it constructs a rigorous upper bound $q(\widetilde{\theta}; \mathcal{I}) \le q^{+}$ for the executed candidate and returns CERTIFIED whenever $q^{+} \le b$. Native closure therefore certifies the concrete finite-precision artifact under the source method's own certificate.

Retraining-reference certification. This path is needed when a method's native certificate does not quantify how close its executed output is to retraining on the current retain set. When a deployment requires this fidelity directly, ExecCert uses *Retraining-Reference Release Verification* (RRV). RRV defines a uniquely specified exact-real target θ^{R} from the declared training objective, mutable model component, and contract inputs. For a predeclared distance d and tolerance $\tau > 0$, the release contract requires $d(\widetilde{\theta}, \theta^{\mathrm{R}}) \le \tau$. Rather than computing θ^{R} exactly, RRV certifies this contract by constructing a rigorous upper bound on the distance.

For the Euclidean parameter contract, this takes the form

$$\left\|\widetilde{\theta} - \theta^{\mathrm{R}}\right\|_F \leq C. \tag{1}$$

How C is obtained depends on the model class. For ridge regression heads with symmetric positive-definite (SPD) normal matrices, On-demand RRV constructs the required proof information for the candidate, whereas Incremental RRV maintains it while deletion requests are processed. The latter updates certified numerical state through rank-one transitions. For differentiable strongly convex heads, On-demand RRV instead uses a post-update gradient certificate, covering L_2-regularized logistic and softmax heads.

3.2 Contracts Derived from the RRV Bound

The absolute bound C can be translated into several release contracts. For relative parameter fidelity, denote the predeclared tolerance by τ_{par}. The reverse triangle inequality gives

$$\frac{\left\|\widetilde{\theta} - \theta^{\mathrm{R}}\right\|_F}{\left\|\theta^{\mathrm{R}}\right\|_F} \leq \frac{C}{\left\|\widetilde{\theta}\right\|_F - C}, \tag{2}$$

whenever $\left\|\widetilde{\theta}\right\|_F > C$, so the contract is certified when the right-hand side is at most τ_{par}.

A deployment may instead care about outputs on a fixed set of input representations $Z \in \mathbb{R}^{m \times d}$, declared independently of the candidate. For a parameter matrix with c output columns, it may require

$$\frac{\left\|Z(\widetilde{\theta} - \theta^{\mathrm{R}})\right\|_F}{\sqrt{mc}} \leq \tau, \tag{3}$$

for which the parameter certificate immediately gives $\left\|Z(\widetilde{\theta} - \theta^{\mathrm{R}})\right\|_F \leq \|Z\|_2 C$, while Section 4 develops a tighter error-centered certificate that preserves the error direction, and the margin conditions in Corollary 1 further imply per-row decision agreement. Contracts are checked separately; the same absolute bound also supports the randomized-release and output-format interfaces in Section 4.4.

4 Retraining-Reference Release Verification

Native-certificate closure is method-specific once the source certificate quantity is fixed; its concrete instantiations appear in Section 5.3. The main technical development is therefore RRV, which must certify an executed candidate against a retraining reference that is not explicitly computed.

RRV can construct certificate evidence in two ways. On-demand RRV constructs the required evidence when certification is requested, whereas Incremental RRV reuses numerical state maintained across deletion requests. The latter introduces an additional obligation: the maintained finite-precision state must itself be certified against the exact current retain-set system before it can support a release certificate. Our ridge regression realization supports both paths through residual and inverse-defect bounds; for differentiable strongly convex heads, On-demand RRV uses a post-update gradient certificate.

The interface in Section 3 concerns a single candidate after a declared deletion sequence. From this point onward, we index internal service state by deletion step t only to describe how incremental evidence evolves. The same machinery also accommodates long-running services that expose intermediate snapshots, in which case a declared subset of internal states may serve as release checkpoints.

4.1 Release-Time RRV Certificates

For a candidate considered for release, RRV must bound its distance to the exact retraining reference θ^{R} without explicitly computing that reference. We give two constructions of this parameter-error bound.

Ridge regression forward certificate. Let the exact retraining reference satisfy $H\theta^{\mathrm{R}} = B$ with $H \in \mathbb{R}^{d\times d}$ nonsingular, and let P approximate H^{-1}. Since $P(B - H\widetilde{\theta}) = PH(\theta^{\mathrm{R}} - \widetilde{\theta})$, the corrected residual would equal the parameter error if $P = H^{-1}$. With an approximate inverse, the verifier instead controls both the inverse defect and the corrected residual:

$$\|I_d - PH\|_2 \le \delta < 1, \qquad \left\|P(B - H\widetilde{\theta})\right\|_F \le \rho. \tag{4}$$

Lemma 1 (Retraining-reference forward certificate)**.** *Under Equation* (4)*,*

$$\left\|\widetilde{\theta} - \theta^{\mathrm{R}}\right\|_F \le C := \frac{\rho}{1-\delta}.$$

Strongly-convex post-update certificate. For a differentiable strongly convex objective, RRV can instead construct the parameter certificate directly from the final candidate. If g^+ upper-bounds its gradient norm under the current retain-set objective and μ^- lower-bounds the objective's strong-convexity constant, then $C = g^+/\mu^-$ bounds the candidate's distance to the exact minimizer. This post-update construction requires no inverse proof state and applies, for example, to L_2-regularized logistic and softmax heads. The formal statement and executable construction appear in Proposition 3 and Section D.1.

For fixed readouts, the ridge regression construction admits a tighter directional refinement. For a predeclared set of input representations $Z \in \mathbb{R}^{m\times d}$, the parameter certificate gives $\left\|Z(\theta^{\mathrm{R}} - \widetilde{\theta})\right\|_F \le \|Z\|_2 C$, which treats the direction of the parameter error as unknown. The corrected residual also provides a computed center $\widehat{\Delta}$ for that error, allowing the remaining uncertainty to be bounded around an estimated direction.

Proposition 1 (Error-centered readout certificate)**.** *Let $\Delta = \theta^{\mathrm{R}} - \widetilde{\theta}$ and suppose $\|\Delta\|_F \le C$ and $\left\|\Delta - \widehat{\Delta}\right\|_F \le R_\Delta$. Then, for any fixed $Z \in \mathbb{R}^{m\times d}$,*

$$\|Z\Delta\|_F \le \min\{\|Z\|_2 C, \left\|Z\widehat{\Delta}\right\|_F + \|Z\|_2 R_\Delta\}. \tag{5}$$

We denote the resulting generic and centered bounds by $C_Z^{\mathrm{ball}} = \|Z\|_2 C$ and $C_Z^{\mathrm{ctr}} = \left\|Z\widehat{\Delta}\right\|_F + \|Z\|_2 R_\Delta$. The centered bound preserves the estimated error direction and applies the worst-case bound only to its remaining uncertainty. Its executable construction appears in Section C.5.2. The same certificate yields binary score-sign and multiclass top-1 agreement under the margin conditions in Corollary 1.

4.2 Maintaining Reference-Transfer Evidence

When a deployment chooses to maintain certificate evidence incrementally, the verifier must relate the service's evolving finite-precision state to the exact current retain-set system. The release-time certificate in Section 4.1 ultimately concerns the latter, while the service updates a separate stored numerical state. Incremental RRV therefore maintains certified evidence connecting these two systems throughout the deletion sequence.

Let $D_t = \{(x_i, y_i)\}_{i=1}^{n_t}$ be the retain set after t committed deletions, with feature and target matrices X_t and Y_t. For $\lambda > 0$, define

$$H_t = X_t^\top X_t + \lambda I_d, \qquad B_t = X_t^\top Y_t, \qquad H_t\theta_t^{\mathrm{R}} = B_t. \tag{6}$$

Deleting (x, y) changes the exact statistics by $H_{t+1} = H_t - xx^\top$ and $B_{t+1} = B_t - xy^\top$.

The service instead stores finite-precision statistics $\widetilde{H}_t, \widetilde{B}_t$. Incremental RRV tracks their discrepancy from the exact retain-set statistics through componentwise enclosures

$$\left|\widetilde{H}_t - H_t\right| \le \mathcal{U}_{H,t}, \qquad \left|\widetilde{B}_t - B_t\right| \le \mathcal{U}_{B,t}. \tag{7}$$

These bounds certify how far the numerical state maintained by the service may have drifted from the exact retain-set system.

The second maintained component is reusable inverse evidence. Rather than reconstructing an inverse certificate after every deletion, Incremental RRV carries an approximate inverse P_t, a computed defect tracker $\widehat{E}_t$, and a certified radius r_t satisfying

$$\left\| I_d - P_t \widetilde{H}_t - \widehat{E}_t \right\|_F \le r_t. \tag{8}$$

For each signed rank-one change, a guarded Sherman–Morrison transition updates P_t, the defect tracker, and the statistics enclosures. The propagated radius accounts for the previous certified uncertainty together with arithmetic error in the update and tracker evaluation. Every denominator, matrix-vector product, and state transition is checked before commit; a failed guard invokes the configured rebuild. A successful rank-one transition costs $O(d^2)$, compared with $O(d^3)$ for dense inverse reconstruction.

At release, these two maintained components are combined to recover the quantities required by the certificate in Section 4.1. Suppressing t, let $E = I_d - P\widetilde{H}$. The exact retain-set residual and inverse defect obey

$$\begin{aligned} B - H\widetilde{\theta} &= (\widetilde{B} - \widetilde{H}\widetilde{\theta}) + (B - \widetilde{B}) - (H - \widetilde{H})\widetilde{\theta}, \\ I_d - PH &= E + P(\widetilde{H} - H). \end{aligned} \tag{9}$$

The first identity transfers the stored-system residual to the exact retain-set residual; the second transfers the stored-system inverse defect to the exact retain-set system.

Here and below, an overline denotes a certified upper bound on the enclosed norm, and $\mathrm{up}(\cdot)$ denotes componentwise outward-rounded evaluation.

Let $\widehat{R} = \mathrm{fl}(\widetilde{B} - \widetilde{H}\widetilde{\theta})$, let $\mathcal{U}_R$ enclose residual-evaluation error together with the stored-to-retain-set correction, and let $\varepsilon_{P\widehat{R}}$ enclose the arithmetic error in $P\widehat{R}$. Combining the maintained statistics and inverse evidence gives

$$\boxed{\begin{aligned} &\|I_d - PH\|_2 \le \delta_{\mathrm{ret}} := \mathrm{up}\left(\delta_{\mathrm{stored}} + \overline{\|P\|_2}\,\overline{\|\mathcal{U}_H\|_2}\right), \\ &\left\|P(B - H\widetilde{\theta})\right\|_F \le \rho_{\mathrm{ret}} := \mathrm{up}\left(\underbrace{\left\|\mathrm{fl}(P\widehat{R})\right\|_F}_{\text{corrected residual}} + \underbrace{\varepsilon_{P\widehat{R}}}_{\text{evaluation error}} + \underbrace{\|\mathrm{up}(|P|\mathcal{U}_R)\|_F}_{\text{reference-transfer error}}\right), \\ &C = \frac{\rho_{\mathrm{ret}}}{1 - \delta_{\mathrm{ret}}}. \end{aligned}} \tag{10}$$

Thus the evidence maintained during the deletion sequence is converted only when certification is required into the exact retain-set quantities consumed by the release-time certificate.

The componentwise outward-rounded construction appears in Section C.4; the rank-one recurrence and propagation proofs appear in Sections B.3, C.2 and C.6.

4.3 Transactional Release Conformance

A release-time certificate may be constructed by either On-demand RRV or Incremental RRV as in Section 4.2. In either case, the certificate must apply to the exact artifact that becomes visible.

Let $\mathcal{R}$ denote the declared release checkpoints. At $t \notin \mathcal{R}$, the service may commit verified internal state without emitting an artifact. At $t \in \mathcal{R}$, the current candidate is released only after every declared contract certifies. Any repair, recomputation, or re-encoding that changes the candidate produces a new artifact and therefore requires recertification. A proof-state rebuild, by contrast, changes only the verifier evidence and leaves the candidate itself unchanged.

Theorem 1 (Incremental RRV release conformance)**.** *Suppose Incremental RRV is initialized with valid statistics and inverse-defect enclosures, accepts only guarded transitions that preserve both invariants, and at each declared release checkpoint applies Equation* (10) *and the declared contract checks to the concrete candidate. If candidate-changing remediation is recorded as a new candidate and recertified, and release is atomic with respect to the certified artifact, then every released artifact satisfies every declared retraining-reference contract.*

The theorem states the concrete Incremental RRV initialization, transition, reference-transfer, contract-checking, and transactional-release obligations; On-demand RRV is the special case that constructs sound evidence only at a release checkpoint. Proofs and precise transactional assumptions appear in Section B.4. Successful rank-k incremental propagation costs $O(kd^2 + kdc)$, excluding output generation; complete release and rebuild accounting appears in Section B.4.

4.4 Derived Release Contracts

The same bound composes with Gaussian and Laplace release guarantees, including executable closure of Newton direct-perturbation radii (Pandey et al., 2026; Zou et al., 2025); the full interfaces appear in Section B.2.

Format compatibility. A final representation may itself be incapable of satisfying the declared contract. Let $\widehat{\theta}$ be a certified reference center with $\left\|\widehat{\theta} - \theta^{\mathrm{R}}\right\|_F \leq r_{\mathrm{ref}}$. For a finite scalar format $\mathcal{F} \subset \mathbb{R}$, let $\underline{d}_{\mathcal{F}}(\widehat{\theta})$ be a certified lower bound on the distance from $\widehat{\theta}$ to $\mathcal{F}^{d\times c}$, and let $\overline{s}_{\mathrm{ref}}$ be a certified upper bound on $\left\|\widehat{\theta}\right\|_F$.

Proposition 2 (Finite-format impossibility)**.** *Assume $\theta^{\mathrm{R}} \neq 0$. If*

$$\underline{d}_{\mathcal{F}}(\widehat{\theta}) - r_{\mathrm{ref}} > \tau_{\mathrm{par}}\left(\overline{s}_{\mathrm{ref}} + r_{\mathrm{ref}}\right), \tag{11}$$

then no $\theta_{\mathcal{F}} \in \mathcal{F}^{d\times c}$ satisfies the relative contract in Equation (2)*.*

The format is then infeasible for the declared contract; the proof and codebook construction appear in Sections B.5 and E.4.

5 Experiments

Evaluation questions. We ask whether reference transfer prevents false releases without making certificates overly conservative (Section 5.1), whether incremental evidence reduces lifecycle cost without weakening certification (Section 5.2), and how ExecCert affects four published methods without changing their update rules (Section 5.3). We then test when producer precision or output representation limits the contract (Section 5.4).

The sequential service certifies a $d = 3073$ binary ridge head on frozen Llama-3.2-3B-Instruct (Meta AI, 2024) representations. Controlled diagnostics use DINOv2 (Oquab et al., 2024) on CIFAR-10 (Krizhevsky, 2009) and RoBERTa (Liu et al., 2019) on AG-News (Zhang et al., 2015); the published applications cover linear, logistic, and graph methods. Comparisons hold the candidate and contract fixed. Independent retraining and high-precision intervals supply offline labels. Protocols and provenance appear in Sections D to F.

5.1 Reference-Transfer Soundness and Tightness

Stored-equation verification admits nearly all nonconforming candidates on the matched stress population, whereas RRV produces no false release (Table 5). On the FP32 stress population, Incremental RRV initially certifies 4,838 of 7,293 conforming outputs without a false release; proof-only rebuilds raise conditional coverage to 67.24%, compared with 99.97% for fresh-factor verification (Section 5.4). On a separate fixed population, RRV encloses every realized error across the tolerance sweep, and its certificate remains within about 1% of the independent reference for 95% of outputs (Figure 2).

5.2 Evidence Reuse in Sequential Services

We compare fresh and maintained verified factors with Incremental RRV on the same candidate outputs, fixed readout, and release contract. A direction-free incremental variant tests whether reuse alone is sufficient.

The maintained factor removes the weak-baseline explanation: it matches the coverage of fresh factor, while directional Incremental RRV uses roughly 40% of fresh-factor latency at matched coverage

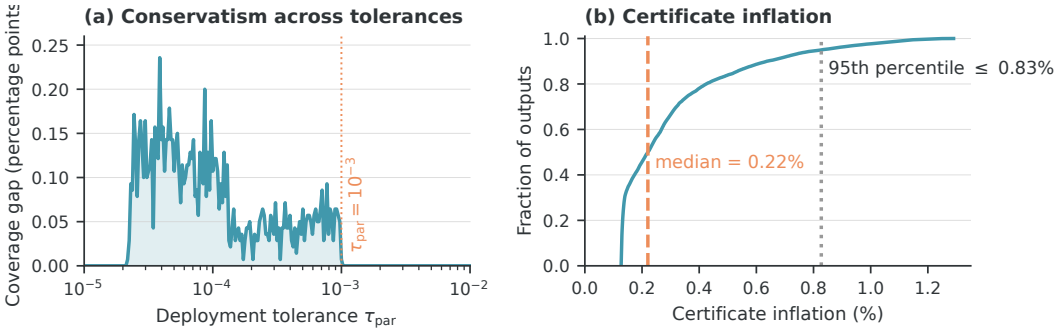

Figure 2: **RRV certificates closely track realized error.** Incremental componentwise RRV and independent retained-row reconstruction evaluate the same fixed FP32 deletion outputs at every tolerance. Coverage gap is the conforming fraction minus the certified fraction; certificate inflation is $\tau_{\text{cert}}/\tau_{\text{true}} - 1$. No rebuild or producer remediation changes the population.

Table 1: **Verification of the same frozen 3B safety-head outputs.** Latency summarizes 10 repetitions at each of 21 states. No method scans retained rows; state excludes those rows.

Method	Certified release states	Med./P95 (s)	State (MiB)
On-demand RRV (fresh factor)	21/21	2.611/3.593	286.5
Maintained verified factor	21/21	2.453/3.678	358.6
Incremental RRV (norm-ball)	0/21	0.801/0.859	360.3
Incremental RRV (directional)	21/21	1.056/1.163	430.6

(Table 1). The norm-ball variant fails on the same states, so reuse preserves strength only when the proof state retains contract-relevant error direction. This ordering holds throughout the nonempty tolerance sweep. Replacing the readout-norm upper bound by a certified lower bound still leaves every norm-ball certificate above tolerance, ruling out readout-norm looseness as the explanation. A separate decision audit covers about 95% of declared safety-head decisions; observed agreement is not used as certified coverage.

Measured-component lifecycle accounting combines complete stream replays with the paired release-check timings. Fresh factor is cheapest for sparse checks; directional Incremental RRV crosses it at about 22 checks per 1,000 requests and remains cheaper than maintained factor throughout the evaluated schedules (Figure 3(a)). Under every-request checking, it reduces lifecycle cost by 42.7% relative to maintained factor. The crossover is workload-specific because it trades continuous proof maintenance against release-time construction.

5.3 Published-Method Applications

We attach ExecCert to four existing unlearning methods: Newton-update removal (Guo et al., 2020), Newton direct perturbation (Zou et al., 2025), ScaleGUN (Yi & Wei, 2025), and CEU (Wu et al., 2023). In each case, the original update rule and emitted candidate are left unchanged; ExecCert is applied only at release time. The check closes either the method's native certificate quantity or a predeclared retraining-reference contract.

Table 2: **Executable certification on published unlearning methods.** ExecCert is applied to the emitted candidate without changing the underlying update rule.

Method	Release-time outcome
Newton removal	Native closure shifts the retraining boundary by **0–4 requests**; direct recertification also identifies conservative source stops.
Newton direct perturbation	At FP32, the median direct bound is **2.28× tighter** than composition; paired FP64 results retain an absolute gap.
ScaleGUN	All **15/15** cheap-update certificates remain valid.
CEU	Median realized error is **8.9%**; the median certified upper bound is **14.1%**.

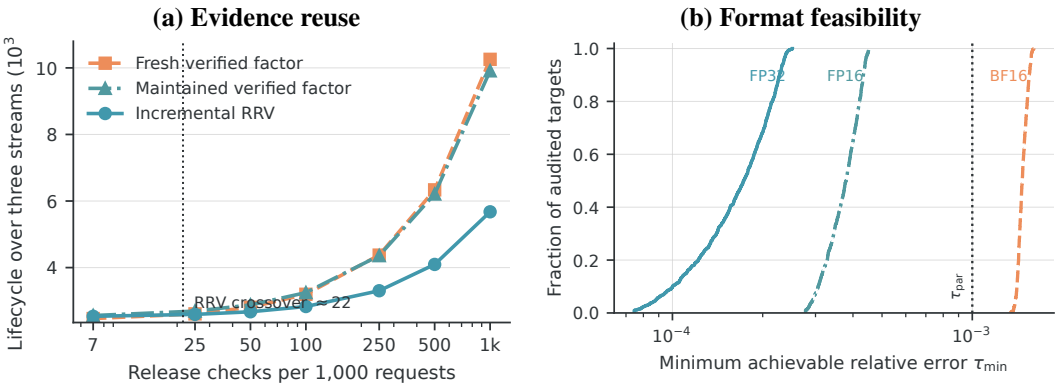

Figure 3: **Deployment choices affect certification cost and feasibility.** (a) Full-stream maintenance measurements combined with matched release-check timings produce the lifecycle curves. The maintained-factor crossover with fresh construction lies beyond the plotted low-frequency region, at about 214 checks per 1,000 requests. (b) The horizontal axis is the minimum achievable relative error $\tau_{\min}$. A format is ruled out when its certified lower bound exceeds the declared contract τ_{par}; BF16 is unreachable, while equal-size FP16 remains reachable.

Native closure can move Newton-removal retraining earlier, while direct recertification identifies conservative source stops. For Newton direct perturbation, direct certification remains tighter than direction-free composition under both execution precisions; we report an absolute FP64 gap because the direct bound can approach zero. ExecCert preserves the ScaleGUN cheap-update certificates. For CEU, the declared tolerance is not supported, but post-update RRV certifies the weaker fidelity of the emitted heads. These cases show that release-time certification can change a decision, tighten or preserve a certificate, or identify the fidelity supported by an output. Source instrumentation and supplementary results appear in Sections D and D.2.

5.4 Precision and Representation Boundaries

The preceding experiments vary certificate construction for a fixed candidate. We finally ask when the candidate or its representation, rather than the verifier, limits the contract.

Producer precision. With FP32 statistics, fresh-factor On-demand RRV covers materially more conforming candidates than Incremental RRV after proof-only rebuilds; both paths attain complete coverage with the FP64 producer. Changing precision also changes the emitted candidate, so this is a producer-policy comparison rather than a certificate-only comparison. Full timing and state results appear in Table 7 and Section E.4.

Release format. BF16 cannot represent any artifact satisfying the declared tolerance, whereas FP16 and FP32 remain feasible (Figure 3(b)). Thus, some failures require changing the producer or encoding rather than tightening verification.

6 Discussion and Conclusion

Our results separate three release-time failure modes: a mathematical certificate may not close for the executed artifact, a sound verifier may be too weak or costly for the release schedule, or the producer and output format may preclude the contract. ExecCert identifies whether remediation should target verification or the released artifact.

Certification strategy depends on the contract and release schedule. Native closure uses the quantity governing a source guarantee; RRV targets fidelity to current retain-set retraining. On-demand construction favors sparse checks, while incremental proof state amortizes frequent certification. ExecCert supplements the upstream guarantee: parameter or fixed-readout fidelity does not establish semantic forgetting or multi-round adaptive privacy. Our incremental implementation covers frozen representations with moderate-dimensional mutable ridge heads; Section A gives the remaining conditions.

AI USE STATEMENT

We used generative AI tools to assist with code implementation and refactoring, figure and LaTeX editing, and language polishing. The authors reviewed and verified all mathematical statements, experimental designs, interpretations of source methods, code changes, results, and manuscript text. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The appendix provides complete proofs, executable-verification details, published-source provenance, experimental protocols, and independent reference construction. The accompanying anonymous artifact contains the implementation, fixed configurations, source instrumentation patches, and canonical records used for the reported tables and figures.

REFERENCES

Eli Chien, Haoyu Wang, Ziang Chen, and Pan Li. Certified machine unlearning via noisy stochastic gradient descent. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Thorsten Eisenhofer, Doreen Riepel, Varun Chandrasekaran, Esha Ghosh, Olga Ohrimenko, and Nicolas Papernot. Verifiable and provably secure machine unlearning. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 479–496, 2025. doi: 10.1109/SaTML64287.2025.00033.

Antonio A. Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. *Making AI forget you: data deletion in machine learning*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Varun Gupta, Christopher Jung, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Chris Waites. Adaptive machine unlearning. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

William W. Hager. Updating the inverse of a matrix. *SIAM Review*, 31(2):221–239, 1989. doi: 10.1137/1031049. URL https://doi.org/10.1137/1031049.

Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2 edition, 2002. doi: 10.1137/1.9780898718027.

Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Juntao Dai, Boren Zheng, Tianyi Qiu, Jiayi Zhou, Kaile Wang, Boxuan Li, Sirui Han, Yike Guo, and Yaodong Yang. Pku-saferlhf: Towards multi-level safety alignment for llms with human preference, 2025. URL https://arxiv.org/abs/2406.15513.

Jiankai Jin, Olga Ohrimenko, and Benjamin I. P. Rubinstein. Getting a-round guarantees: Floating-point attacks on certified robustness. In *Proceedings of the 2024 Workshop on Artificial Intelligence and Security*, AISec '24, pp. 53–64, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400712289. doi: 10.1145/3689932.3694761. URL https://doi.org/10.1145/3689932.3694761.

Anastasia Koloskova, Youssef Allouah, Animesh Jha, Rachid Guerraoui, and Sanmi Koyejo. Certified unlearning for neural networks. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML'25. JMLR.org, 2025.

Rudolf Krawczyk. Newton-algorithmen zur bestimmung von nullstellen mit fehlerschranken. *Computing*, 4(3):187–201, 1969. doi: 10.1007/BF02234767.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. URL https://arxiv.org/abs/1907.11692.

Mohammad M Maheri, Sunil Cotterill, Alex Davidson, and Hamed Haddadi. ZK-APEX: ZERO-KNOWLEDGE APPROXIMATE PERSONALIZED UNLEARNING WITH EXECUTABLE PROOFS. In *Ninth Conference on Machine Learning and Systems*, 2026. URL https://openreview.net/forum?id=bLx6orLvQM.

Meta AI. Llama 3.2-3B-Instruct model card. Hugging Face model repository, 2024. URL https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct.

Ilya Mironov. On significance of the least significant bits for differential privacy. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pp. 650–661, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450316514. doi: 10.1145/2382196.2382264. URL https://doi.org/10.1145/2382196.2382264.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87 of *Applied Optimization*. Springer, Boston, MA, 2004. doi: 10.1007/978-1-4419-8853-9.

Takeshi Ogita, Siegfried M. Rump, and Shin'ichi Oishi. Accurate sum and dot product. *SIAM Journal on Scientific Computing*, 26(6):1955–1988, 2005. doi: 10.1137/030601818. URL https://doi.org/10.1137/030601818.

Shin'ichi Oishi and Siegfried M. Rump. Fast verification of solutions of matrix equations. *Numerische Mathematik*, 90(4):755–773, 2002. doi: 10.1007/s002110100310.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. URL https://arxiv.org/abs/2304.07193.

Aaradhya Pandey, Arnab Auddy, Haolin Zou, Arian Maleki, and Sanjeev Kulkarni. Gaussian certified unlearning in high dimensions: A hypothesis testing approach. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=0FJYicpOj0.

Siegfried M. Rump. Verification methods: rigorous results using floating-point arithmetic. In *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, ISSAC '10, pp. 3–4, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450301503. doi: 10.1145/1837934.1837937. URL https://doi.org/10.1145/1837934.1837937.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: algorithms for machine unlearning. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Attila Szász, Balázs Bánhelyi, and Márk Jelasity. No soundness in the real world: On the challenges of the verification of deployed neural networks. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=c16m2kUTLZ.

Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium (USENIX Security 22)*, pp. 4007–4022, Boston, MA, August 2022. USENIX Association. ISBN 978-1-939133-31-1. URL https://www.usenix.org/conference/usenixsecurity22/presentation/thudi.

Vaclav Voracek and Matthias Hein. Sound randomized smoothing in floating-point arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=HaHCoGcpV9.

Jiasi Weng, Shenglong Yao, Yuefeng Du, Junjie Huang, Jian Weng, and Cong Wang. Proof of unlearning: Definitions and instantiation. *Trans. Info. For. Sec.*, 19:3309–3323, January 2024. ISSN 1556-6013. doi: 10.1109/TIFS.2024.3358993. URL https://doi.org/10.1109/TIFS.2024.3358993.

Kun Wu, Jie Shen, Yue Ning, Ting Wang, and Wendy Hui Wang. Certified edge unlearning for graph neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 2023. doi: 10.1145/3580305.3599271.

Lu Yi and Zhewei Wei. Scalable and certifiable graph unlearning: Overcoming the approximation error barrier. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f43567631f608dd1317c227c69dae441-Abstract-Conference.html.

Binchi Zhang, Zihan Chen, Cong Shen, and Jundong Li. Verification of machine unlearning is fragile. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024a.

Binchi Zhang, Yushun Dong, Tianhao Wang, and Jundong Li. Towards certified unlearning for deep neural networks. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024b.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NIPS*, 2015.

Haolin Zou, Arnab Auddy, Yongchan Kwon, Kamiar Rahnama Rad, and Arian Maleki. Certified machine unlearning under high dimensional regime. *Journal of Machine Learning Research*, 26 (308):1–58, 2025. URL http://jmlr.org/papers/v26/25-1751.html.

A Scope and Limitations

Native closure inherits the scope of the source method's own certificate; RRV instead certifies only the declared distance to its uniquely specified retraining reference.

Fixed-readout claims apply to the input representations declared before the candidate is inspected, and say nothing about inputs outside that set. Randomized-release calibration assumes a predeclared common scale; adaptive release policies, in which the noise scale reacts to the candidate, require separate analysis.

Our reusable proof state currently targets moderate-dimensional mutable heads. Extending incremental certification to larger trainable blocks will require exploiting additional structure, since dense inverse evidence does not scale to that regime. Unlike statistical deletion capacity, proof-state lifetime is instance-dependent and governed by conditioning rather than a universal deletion count: a failed guard triggers certified rebuilding, which restores the invariants at higher cost but imposes no fixed bound on the number of deletions a service can absorb.

B Proofs of Main Results

B.1 Release-Time RRV Certificates

We first establish the parameter-error bound and its error-centered fixed-readout refinement. The first is the classical residual–inverse-defect argument from verified linear algebra (Krawczyk, 1969; Oishi & Rump, 2002; Rump, 2010); we retain its short derivation because it is the main certificate interface.

Proof of Lemma 1. Let $\Delta = \theta^{\mathrm{R}} - \widetilde{\theta}$. Since $H\theta^{\mathrm{R}} = B$,

$$\Delta = (I - PH)\Delta + P(B - H\widetilde{\theta}),$$
$$\|\Delta\|_F \le \|I - PH\|_2 \, \|\Delta\|_F + \left\|P(B - H\widetilde{\theta})\right\|_F \le \delta \, \|\Delta\|_F + \rho.$$

Thus $\|\Delta\|_F \le \rho/(1-\delta) = C$. The reverse triangle inequality gives $\|\theta^{\mathrm{R}}\|_F \ge \left\|\widetilde{\theta}\right\|_F - C$, so whenever $\left\|\widetilde{\theta}\right\|_F > C$,

$$\frac{\left\|\widetilde{\theta} - \theta^{\mathrm{R}}\right\|_F}{\|\theta^{\mathrm{R}}\|_F} \le \frac{C}{\left\|\widetilde{\theta}\right\|_F - C}. \tag{12}$$

□

Proof of Proposition 1. The first bound follows from $\|Z\Delta\|_F \le \|Z\|_2 \, \|\Delta\|_F \le \|Z\|_2 \, C$. For the second, write $\Delta = \widehat{\Delta} + (\Delta - \widehat{\Delta})$ and apply the triangle inequality together with the stated center enclosure. Taking the smaller of the two valid bounds gives Equation (5). □

The absolute parameter bound also certifies decisions with sufficient margin.

Corollary 1 (Certified linear-decision agreement)**.** *Fix a representation $z \in \mathbb{R}^d$. For a binary head satisfying $\left\|\widetilde{\theta} - \theta^{\mathrm{R}}\right\|_2 \le C$, the executed and retraining-reference signs agree when $\left|z^\top \widetilde{\theta}\right| > \|z\|_2 \, C$. For a multiclass head, let $\left\|\widetilde{\Theta} - \Theta^{\mathrm{R}}\right\|_F \le C$ and $a = \arg\max_j z^\top \widetilde{\theta}_j$. It remains the unique top class under Θ^{R} when*

$$z^\top \widetilde{\theta}_a - z^\top \widetilde{\theta}_j > \sqrt{2} \, \|z\|_2 \, C \qquad \text{for every } j \ne a.$$

Proof

Cauchy–Schwarz bounds the binary score change by $\|z\|_2 C$. For the multiclass case, writing $\Delta_j = \theta_j^{\mathrm{R}} - \widetilde{\theta}_j$ gives

$$\left|z^\top(\Delta_a - \Delta_j)\right| \le \|z\|_2 \|\Delta_a - \Delta_j\|_2 \le \sqrt{2}\, \|z\|_2 C.$$

The strict margins preserve the stated decisions.

For differentiable strongly convex objectives, a post-update gradient enclosure provides a separate construction of the parameter-error bound.

Proposition 3 (Strongly-convex post-update certificate)**.** *Let $L : \mathbb{R}^d \to \mathbb{R}$ be differentiable and μ-strongly convex, with minimizer θ^{R}. Suppose the verifier constructs*

$$g^+ \ge \left\|\nabla L(\widetilde{\theta})\right\|_2, \qquad 0 < \mu^- \le \mu.$$

Then

$$\left\|\widetilde{\theta} - \theta^{\mathrm{R}}\right\|_2 \le \frac{g^+}{\mu^-} =: C.$$

The absolute bound is the standard gradient–distance consequence of strong convexity (Nesterov, 2004). A relative contract follows by applying Equation (12).

B.2 Randomized Release Interfaces

These fixed-state interfaces use a predeclared common noise scale. If RRV certifies $\left\|\widetilde{\theta} - \theta^{\mathrm{R}}\right\|_2 \le C$, the equal-covariance Gaussian testing identity of Pandey et al. (2026) gives

$$\varepsilon_* = \frac{\left\|\widetilde{\theta} - \theta^{\mathrm{R}}\right\|_2}{\sigma} \le \frac{C}{\sigma}.$$

For isotropic radial Laplace releases with common predeclared scale s, we use the density $q_\mu(z) \propto \exp(-\|z-\mu\|_2/s)$. Sampling draws a uniform direction on the Euclidean unit sphere and an independent radius from $\mathrm{Gamma}(d, s)$; the same rule and scale apply to both centers. The triangle inequality gives, for every measurable event A,

$$Q_{\mathrm{exec}}(A) \le e^{C/s} Q_{\mathrm{ref}}(A), \qquad Q_{\mathrm{ref}}(A) \le e^{C/s} Q_{\mathrm{exec}}(A).$$

Thus $\sigma \ge C/\varepsilon$ gives the corresponding Gaussian trade-off guarantee, while $s \ge C/\varepsilon$ gives a two-sided Laplace comparison. Here ε_* is the equal-covariance Gaussian testing parameter; it is not by itself a conventional (ε, δ)-DP guarantee. If an upstream analysis supplies radius r_{nom} and executable closure adds C_{exec}, the Laplace result applies with $C = r_{\mathrm{nom}} + C_{\mathrm{exec}}$.

B.3 Nonsymmetric Defect Recurrence

All quantities in this subsection are evaluated in exact real arithmetic. Let $H_s, P \in \mathbb{R}^{d\times d}$, $x \in \mathbb{R}^d$, and $s \in \{-1, +1\}$. Define

$$E = I_d - PH_s, \quad q = Px,$$
$$z = P^\top x, \qquad \alpha = 1 + sx^\top q.$$

For $\alpha \neq 0$, let

$$H_s^+ = H_s + sxx^\top, \qquad P^+ = P - \frac{s}{\alpha} q z^\top.$$

The update for P^+ is the Sherman–Morrison formula (Hager, 1989). The following identity propagates the stored-system inverse defect without assuming that P is symmetric. Section C.6 constructs its guarded floating-point realization.

Lemma 2 (Exact nonsymmetric defect recurrence)**.** *For arbitrary $H_s, P \in \mathbb{R}^{d\times d}$ with $\alpha \neq 0$, including nonsymmetric P, the post-update defect $E^+ = I_d - P^+H_s^+$ satisfies*

$$E^+ = E - q\frac{s}{\alpha}\left(x^\top - z^\top H_s\right).$$

The recurrence requires $O(d^2)$ arithmetic and $O(d^2)$ state.

Proof of Lemma 2. Because $z^\top = x^\top P$, expansion of $E^+ = I_d - P^+H_s^+$ gives

$$E^+ = E - sqx^\top + \frac{s}{\alpha}qz^\top H_s + \frac{1}{\alpha}q(z^\top x)x^\top.$$

Let $\beta = z^\top x = x^\top q$. Since $\alpha = 1 + s\beta$, the coefficient of $qx^\top$ equals

$$-s + \frac{\beta}{\alpha} = -\frac{s}{\alpha}.$$

Substitution yields $E^+ = E - (s/\alpha)q(x^\top - z^\top H_s)$. □

B.4 Transactional Conformance and Cost

The transactional theorem applies to any sound verification backend. Its incremental corollary combines the maintained statistics and inverse-defect invariants with the release-time contract checks.

Here guarded committed transitions require every numerical denominator and interval to pass its guard, every certificate evaluation to remain within the certified arithmetic scope, and every rank-one subupdate and invariant check to complete before commit. At a release checkpoint, the service exposes an artifact only after every declared contract certifies.

Transactional justification. The initialization hypotheses give the two proof-state invariants at the first state. Theorems 2 and 3 preserve them under every accepted guarded transition. At a release checkpoint, the transfer and release constructions in Sections C.4 and C.5 therefore establish every declared contract for the concrete candidate presented to the policy gate. If remediation changes that candidate, the original certificate no longer authorizes release and the new candidate must be certified separately. Atomic release ensures that the visible artifact is exactly the candidate covered by the successful certificates.

Cost accounting. Let G_t denote the cost of advancing the selected output generator at request t, let k_t be the number of rank-one subupdates for that request, and set $m = 0$ when no fixed readout is declared. Each request adds $O(k_td^2 + k_tdc)$ proof-state work; a release adds $O(d^2c + mdc)$ certificate work. Rebuilding proof state from certified $(\widetilde{H}_t, \widetilde{B}_t)$ costs $O(d^3 + d^2c)$. Reconstructing those statistics from retained rows adds $O(n_td^2 + n_tdc)$. Let M_{proof} count dense proof-state rebuilds that reuse certified sufficient statistics, and let $\mathcal{R}_{\text{data}}$ be the index set of requests whose rebuilds reconstruct those statistics from n_t retained rows. Over N requests, the total cost is

$$O\Bigg(\sum_{t=1}^{N} G_t + \sum_{t=1}^{N}(k_td^2 + k_tdc) + |\mathcal{R}|\,(d^2c + mdc) + M_{\text{proof}}(d^3 + d^2c) + \sum_{t\in\mathcal{R}_{\text{data}}}(n_td^2 + n_tdc + d^3 + d^2c)\Bigg).$$

The every-request-release policy is the special case $|\mathcal{R}| = N$.

B.5 Finite-Format Impossibility

Let $\theta^{\mathrm{R}} \in \mathbb{R}^{d\times c}$ be the exact retraining target, and let $\widehat{\theta}$ satisfy $\left\|\widehat{\theta} - \theta^{\mathrm{R}}\right\|_F \le r_{\mathrm{ref}}$. For a finite scalar format $\mathcal{F} \subset \mathbb{R}$, let $\underline{d}_{\mathcal{F}}(\widehat{\theta})$ be a certified lower bound on the distance from $\widehat{\theta}$ to $\mathcal{F}^{d\times c}$, and let $\overline{s}_{\mathrm{ref}}$ be a certified upper bound on $\left\|\widehat{\theta}\right\|_F$.

Proof of Proposition 2. For any representable $\theta_{\mathcal{F}} \in \mathcal{F}^{d\times c}$, the reverse triangle inequality and the certified reference enclosure give

$$\begin{aligned}\left\|\theta_{\mathcal{F}} - \theta^{\mathrm{R}}\right\|_F &\ge \left\|\theta_{\mathcal{F}} - \widehat{\theta}\right\|_F - \left\|\widehat{\theta} - \theta^{\mathrm{R}}\right\|_F \\ &\ge \underline{d}_{\mathcal{F}}(\widehat{\theta}) - r_{\mathrm{ref}}.\end{aligned}$$

The ordinary triangle inequality gives

$$\left\|\theta^{\mathrm{R}}\right\|_F \le \overline{s}_{\mathrm{ref}} + r_{\mathrm{ref}}.$$

Under Equation (11), the lower bound on the numerator is strictly larger than τ_{par} times this upper bound on the denominator. Thus every representable tensor violates the declared relative contract. □

C Soundness of Incremental RRV

This appendix gives the executable verifier specification needed to establish the soundness of the incremental ridge regression backend of RRV. Algorithm 1 gives its transactional release protocol.

Algorithm 1 Incremental RRV release protocol

Require: Committed service and proof state, deletion request t, release schedule $\mathcal{R}$, output generator, and one or more declared contracts: parameter fidelity, fixed-readout fidelity, or both

Provisionally update the statistics, their retain-set enclosure, and the inverse-defect proof state.
if a guard or proof-state invariant check fails **then**
  Apply the configured proof-state rebuild.
  if the rebuild does not reestablish the required invariant **then**
    Discard the transition; **return** internal STATE-FAIL.
  end if
end if
Advance the output-generator state for request t.
if $t \notin \mathcal{R}$ **then**
  Atomically commit the verified internal state; emit no output.
  return NORELEASE
end if
Materialize the current executed candidate artifact $\widetilde{\theta}_t$.
Construct (δ, ρ, C) and every declared contract certificate.
if some declared contract is inconclusive **then**
  Withhold the candidate and apply the configured remediation.
  Reconstruct proof state as needed and recertify any changed candidate.
end if
if every declared contract now certifies **then**
  Atomically commit state and certificates; release the candidate.
  return CERTIFIED
else
  Atomically commit verified internal state; withhold the candidate.
  return INCONCLUSIVE
end if

The verifier maintains statistics and inverse-defect enclosures, transfers them to the exact retain-set system, and constructs the release certificate in Sections C.2 and C.4 to C.6.

C.1 Floating-point conventions and notation map

Notation and exact-real interpretation. We use the notation and shapes of Section 4.2; array inequalities and absolute values remain componentwise. A hat denotes a stored computed value, and a circle superscript denotes exact-real evaluation of computed intermediates. The symbol $\mathcal{U}$ denotes a componentwise enclosure, ε a local arithmetic bound, and (δ, ρ) the hypotheses consumed by Lemma 1. Stored entries are interpreted as their exact real values.

Standard arithmetic facts. We use the componentwise floating-point models for basic arithmetic, dot products, matrix–vector products, and matrix products from Higham (2002, Ch. 3). Exact residuals returned by TwoSum use the error-free transformation of Ogita et al. (2005). We invoke these standard facts without rederiving them below; every displayed endpoint is evaluated outward, and a failed arithmetic-scope guard returns Inconclusive.

Storage and verifier precision. The stored sufficient statistics may use $\mathbb{F}_{\mathrm{s}} \in \{\mathbb{F}_{32}, \mathbb{F}_{64}\}$, with unit roundoff u_{s}. The verifier and all outward endpoints use binary64, $\mathbb{F}_{\mathrm{v}} = \mathbb{F}_{64}$, with $u_{\mathrm{v}} = 2^{-53}$. Stored FP32 statistics are promoted exactly to binary64 before verification; all other maintained proof state uses binary64.

Outward evaluation. Define $\gamma_m^{(q)} = mu_q/(1 - mu_q)$ for $mu_q < 1$ and $\bar{u}_q = u_q/(1 - u_q)$, where $q \in \{\mathrm{s}, \mathrm{v}\}$. For verifier arithmetic, we write $\gamma_m := \gamma_m^{(\mathrm{v})}$ and $\bar{u} := \bar{u}_{\mathrm{v}}$. Let $\eta = 2^{-1074}$, and let ω be the largest finite binary64 number. For $z \in \mathbb{F}_{64,\geq 0}$, define

$$\mathrm{up}_m(z) := \mathrm{nextafter}\left(\frac{z}{1 - \gamma_m^{(\mathrm{v})}}, +\infty\right).$$

The shorthand $\mathrm{up}(\cdot)$ suppresses only this arity subscript: the implementation passes it explicitly, and each displayed use has the unique arity prescribed by the immediately enclosed operation. $\mathrm{down}(z) = \mathrm{nextafter}(\mathrm{fl}(z), -\infty)$ for its lower endpoint. $m = \ell$ for a sum of ℓ nonnegative terms, $m = 1$ for one product or division, and $m = 2p + 2$ for a scale-safe Frobenius-norm reduction over p entries.

Arithmetic scope. The dot-product bounds permit any parenthesization of binary32 or binary64 products and sums within the stated operation count, so the proof does not depend on a particular threaded-BLAS reduction order. Fused multiply–add is covered conservatively because it incurs no more rounding than the separate product and addition modeled by the bound. Algebraic reassociation through `fastmath` is disabled. TF32 and kernels with mixed-precision accumulation are outside the certified arithmetic model. Certified runs require IEEE gradual underflow; overflow, nonfinite values, unsafe subnormal intermediates, or an endpoint beyond ω returns Inconclusive. Relative-error bounds without an explicit η term are used only after their intermediates are proved to remain in the normal range. These binary64 conventions govern the ridge inverse realization.

C.2 Retain-Set Statistics Proof State

Initialization. At initialization, let $\widehat{A}_H = \mathrm{fl}_{\mathrm{s}}(|X_0|^\top |X_0|)$ and $\widehat{A}_B = \mathrm{fl}_{\mathrm{s}}(|X_0|^\top |Y_0|)$. Reverse dot-product bounds instantiate the enclosure below. The declared ridge is interpreted as the exact real value of its stored encoding; the displayed rounding term covers its addition to the stored diagonal.

$$\begin{aligned}
\mathcal{U}_H^{(0)} &= \mathrm{up}\left(\frac{\gamma_{n_0}^{(\mathrm{s})}}{1 - \gamma_{n_0}^{(\mathrm{s})}} \widehat{A}_H + \bar{u}_{\mathrm{s}} \left|\widetilde{H}_0\right|\right) \in \mathbb{R}_{\geq 0}^{d \times d}, \\
\mathcal{U}_B^{(0)} &= \mathrm{up}\left(\frac{\gamma_{n_0}^{(\mathrm{s})}}{1 - \gamma_{n_0}^{(\mathrm{s})}} \widehat{A}_B\right) \in \mathbb{R}_{\geq 0}^{d \times c},
\end{aligned} \tag{13}$$

Shared initialization across precisions. When two storage precisions share one binary64 initialization, the lower-precision path instead uses the direct cast enclosure

$$\mathcal{U}_{H,\mathrm{s}}^{(0)} = \mathrm{up}\Big(\mathcal{U}_{H,\mathrm{v}}^{(0)} + \Big|\mathrm{cast}_{\mathrm{s}}(\widetilde{H}_{0,\mathrm{v}}) - \widetilde{H}_{0,\mathrm{v}}\Big|\Big),$$

and analogously for B. This preserves a common exact data target while isolating the precision used to maintain the stored statistics. We use this shared initialization only for the matched FP32/FP64 diagnostic in Section E.4, so the two arms differ in subsequent storage precision rather than in their exact data target.

Signed rank-one updates. For an update sign $s \in \{-1, +1\}$, let $\widehat{G}_H := \mathrm{fl}_{\mathrm{s}}(xx^\top) \in \mathbb{F}_{\mathrm{s}}^{d\times d}$ and $\widehat{G}_B := \mathrm{fl}_{\mathrm{s}}(xy^\top) \in \mathbb{F}_{\mathrm{s}}^{d\times c}$. The row entries x, y are values in $\mathbb{F}_{\mathrm{s}}$, interpreted exactly as reals. The implementation sets

$$\widetilde{H}^+ := \mathrm{fl}_{\mathrm{s}}(\widetilde{H} + s\widehat{G}_H), \qquad \widetilde{B}^+ := \mathrm{fl}_{\mathrm{s}}(\widetilde{B} + s\widehat{G}_B)$$

and updates

$$\mathcal{U}_H^+ := \mathrm{up}\Big(\mathcal{U}_H + \bar{u}_{\mathrm{s}} \left|\widehat{G}_H\right| + \bar{u}_{\mathrm{s}} \left|\widetilde{H}^+\right|\Big), \tag{14}$$

$$\mathcal{U}_B^+ := \mathrm{up}\Big(\mathcal{U}_B + \bar{u}_{\mathrm{s}} \left|\widehat{G}_B\right| + \bar{u}_{\mathrm{s}} \left|\widetilde{B}^+\right|\Big). \tag{15}$$

Each update adds the local outer-product and storage-rounding errors to the existing statistics enclosure. The following theorem establishes that this enclosure remains valid after every committed update.

Theorem 2 (Statistics-error-bound soundness)**.** *Suppose the current state satisfies* $\left|\widetilde{H} - H\right| \le \mathcal{U}_H$ *and* $\left|\widetilde{B} - B\right| \le \mathcal{U}_B$*. For* $s \in \{-1, +1\}$*, let* $H^+ = H + sxx^\top$ *and* $B^+ = B + sxy^\top$*. Whenever the certified arithmetic-scope guards pass, the updates in Equations* (14) *and* (15) *satisfy*

$$\left|\widetilde{H}^+ - H^+\right| \le \mathcal{U}_H^+, \qquad \left|\widetilde{B}^+ - B^+\right| \le \mathcal{U}_B^+.$$

The initialization in Equation (13) *also encloses the exact statistics represented by* X_0, Y_0 *and the declared ridge encoding. A failed guard makes the transition noncommittable.*

Proof of statistics-error-bound soundness

For H, inserting the exact outer product gives

$$\begin{aligned}\left|\widetilde{H}^+ - H^+\right| \le{}& \left|\widetilde{H} - H\right| + \left|\widehat{G}_H - xx^\top\right| \\ &+ \left|\widetilde{H}^+ - (\widetilde{H} + s\widehat{G}_H)\right|.\end{aligned}$$

The standard componentwise models of Higham (2002, Ch. 3) bound the last two terms by $\bar{u}_{\mathrm{s}} \left|\widehat{G}_H\right|$ and $\bar{u}_{\mathrm{s}} \left|\widetilde{H}^+\right|$, respectively. Combining them with the previous enclosure proves Equation (14); the same argument applies to B. The standard dot-product model gives the initialization radii, and induction establishes Equation (7) after every committed update.

C.3 Exact-Dyadic Retain-Set-Statistics Rebuild

A retained-data rebuild constructs statistics enclosures independently of accumulated update errors. For each statistic entry, integer accumulation computes the exact dyadic sum ξ. The procedure stores $\widetilde{\xi} = \mathrm{RN}_{64}(\xi)$ and rounds $\left|\widetilde{\xi} - \xi\right|$ outward to obtain the enclosure radius. Any nonfinite conversion returns INCONCLUSIVE. Binary floating-point products are dyadic, so integer accumulation is exact before the single final rounding. The resulting entrywise radii therefore satisfy the initialization hypotheses of Theorem 2; successful inverse-state initialization and recertification then permit release under Theorem 1.

We distinguish three event types in the implementation records. An *evidence rebuild* replaces only inverse or factor proof state. A *verifier-statistics reconstruction* replaces the verifier's retained statistics and their enclosures. Neither event changes the candidate, and the producer does not read

the rebuilt arrays. A *producer remediation* instead changes producer state or recomputes the output; it creates a new candidate with a new hash and may alter subsequent outputs, so it is recorded and certified separately. Same-output comparisons include only the first two event types.

Reconstruction visits $O(n_t d^2 + n_t dc)$ dot-product terms; subsequent proof-state initialization and certification have the costs in Section B.4.

C.4 Stored-to-Retain-Set Transfer

The retain-set statistics enclosure from Section C.2 and the propagated defect enclosure from Section C.6 provide the executable (δ, ρ) required by Lemma 1. Let $\widehat{R} = \mathrm{fl}(\widetilde{B} - \widetilde{H}\widetilde{\theta})$ and define

$$\mathcal{U}_R := \mathrm{up}\Big(\gamma_d \left|\widetilde{H}\right| \left|\widetilde{\theta}\right| + \bar{u} \left|\widehat{R}\right| + \mathcal{U}_B + \mathcal{U}_H \left|\widetilde{\theta}\right|\Big). \tag{16}$$

The first two terms bound floating-point evaluation of the stored residual. The last two account for the difference between the stored statistics and the exact retain-set statistics. The envelope in Equation (16), the standard matrix-product and subtraction models, and Equations (7) and (9) give

$$\left|B - H\widetilde{\theta} - \widehat{R}\right| \le \mathcal{U}_R.$$

With $\delta_{\mathrm{stored}} = \mathrm{up}(\left\|\widehat{E}\right\|_F + r)$, write $\overline{\|A\|_2}$ for the quadratic-cost upper bound

$$\mathrm{up}\left(\sqrt{\mathrm{up}(\|A\|_1)\,\mathrm{up}(\|A\|_\infty)}\right).$$

We next combine this residual enclosure with the stored inverse-defect bound to obtain the two hypotheses required by Lemma 1:

$$\begin{aligned}
\delta_{\mathrm{ret}} &= \mathrm{up}\Big(\delta_{\mathrm{stored}} + \overline{\|P\|_2}\,\overline{\|\mathcal{U}_H\|_2}\Big), \\
\rho_{\mathrm{ret}} &= \mathrm{up}\Big(\left\|\mathrm{fl}(P\widehat{R})\right\|_F \\
&\qquad + \left\|\mathrm{up}(\gamma_d\,|P|\left|\widehat{R}\right|)\right\|_F \\
&\qquad + \left\|\mathrm{up}(|P|\,\mathcal{U}_R)\right\|_F\Big).
\end{aligned} \tag{17}$$

Proof of the executable joint bounds

The matrix-product model and the residual budget give

$$\begin{aligned}
\left\|P(B - H\widetilde{\theta})\right\|_F &\le \left\|P\widehat{R}\right\|_F + \||P|\,\mathcal{U}_R\|_F \\
&\le \left\|\mathrm{fl}(P\widehat{R})\right\|_F + \left\|\mathrm{up}(\gamma_d\,|P|\left|\widehat{R}\right|)\right\|_F \\
&\qquad + \|\mathrm{up}(|P|\,\mathcal{U}_R)\|_F \le \rho_{\mathrm{ret}},
\end{aligned}$$

which proves the residual hypothesis in Equation (4).

For the defect, suppose $r \in \mathbb{R}_{\ge 0}$ satisfies $\left\|E - \widehat{E}\right\|_F \le r$. Because $I_d - PH = (I_d - P\widetilde{H}) + P(\widetilde{H} - H)$,

$$\begin{aligned}
\|I_d - PH\|_2 &\le \left\|\widehat{E}\right\|_F + r + \|P\|_2 \left\|\widetilde{H} - H\right\|_2 \\
&\le \left\|\widehat{E}\right\|_F + r + \overline{\|P\|_2}\,\overline{\|\mathcal{U}_H\|_2} \le \delta_{\mathrm{ret}}.
\end{aligned}$$

The second line follows from $\|A\|_2 \le \sqrt{\|A\|_1\,\|A\|_\infty}$ and $\left|\widetilde{H} - H\right| \le \mathcal{U}_H$, which implies $\left\|\widetilde{H} - H\right\|_1 \le \|\mathcal{U}_H\|_1$ and likewise for the infinity norm. The final inequalities are the two bounds in Equation (17), which supply both hypotheses of Lemma 1. The row and column sums used by the two induced-norm bounds cost $O(d^2)$; the remaining residual products cost $O(d^2 c)$. No dense matrix–matrix product is evaluated on this release path.

Retaining componentwise arrays preserves the directional residual interaction $|P|\,\mathcal{U}_R$, while the defect transfer avoids forming the dense matrix product $|P|\,\mathcal{U}_H$.

C.5 Executable Release Certificates

C.5.1 Parameter Certificate

Substituting the transferred bounds into Lemma 1 gives the absolute parameter certificate, provided the bounds are finite and $\delta_{\mathrm{ret}} < 1$. For the relative contract, let $p = dc$, compute the scale-safe floating-point norm $\widehat{s} = \mathrm{fl}(\left\|\widetilde{\theta}\right\|_F)$, and form the certified lower endpoint

$$s^- := \mathrm{down}\left(\frac{\widehat{s}}{1+\gamma_{2p+2}}\right), \qquad d^- := \mathrm{down}(s^- - C).$$

The verifier requires $d^- > 0$, then computes $\tau_{\mathrm{rel}}^+ := \mathrm{up}(C/d^-)$ and certifies only when $\tau_{\mathrm{rel}}^+ \le \tau_{\mathrm{par}}$. Thus the norm lower bound, denominator subtraction, and final division are all directed outward.

C.5.2 Fixed-Readout Error Center

Let $\Delta = \theta^{\mathrm{R}} - \widetilde{\theta}$. The corrected residual provides an approximate center for this parameter error. Applying the fixed readout to the center preserves information about the error direction before the remaining uncertainty is bounded. Define

$$\begin{aligned}\widehat{\Delta} &= \mathrm{fl}(P\widehat{R}),\\ \mathcal{U}_c^{\mathrm{fl}} &= \mathrm{up}(\gamma_d\,|P|\,\left|\widehat{R}\right|),\\ \mathcal{U}_c^{\mathrm{tr}} &= \mathrm{up}(|P|\,\mathcal{U}_R).\end{aligned}$$

Their sum encloses $\widehat{\Delta} - P(B - H\widetilde{\theta})$, so the executable center-evaluation radius is

$$\begin{aligned}\eta_c &= \mathrm{up}\left(\left\|\mathcal{U}_c^{\mathrm{fl}}\right\|_F + \left\|\mathcal{U}_c^{\mathrm{tr}}\right\|_F\right),\\ R_\Delta &= \mathrm{up}(\eta_c + \delta_{\mathrm{ret}}C).\end{aligned}$$

The inverse-defect bound gives $\left\|\Delta - P(B - H\widetilde{\theta})\right\|_F \le \delta_{\mathrm{ret}}C$; therefore $\left\|\Delta - \widehat{\Delta}\right\|_F \le R_\Delta$, as required by Proposition 1. For a fixed readout Z, compute $\widehat{y} = \mathrm{fl}(Z\widehat{\Delta})$ and

$$\mathcal{U}_y = \mathrm{up}(\gamma_d\,|Z|\,\left|\widehat{\Delta}\right|).$$

The two sound readout bounds used by the implementation are

$$\begin{aligned}C_Z^{\mathrm{ball}} &= \mathrm{up}(\overline{\|Z\|_2}\,C),\\ C_Z^{\mathrm{ctr}} &= \mathrm{up}\left(\|\widehat{y}\|_F + \|\mathcal{U}_y\|_F + \overline{\|Z\|_2}\,R_\Delta\right).\end{aligned}$$

The fixed readout is preprocessed once with the rigorous bound

$$\overline{\|Z\|_2} = \min\{\mathrm{up}(\|Z\|_F), \mathrm{up}(\sqrt{u_1 u_\infty})\}.$$

Here $u_1 = \mathrm{up}(\|Z\|_1)$ and $u_\infty = \mathrm{up}(\|Z\|_\infty)$. Both terms upper-bound $\|Z\|_2$, so their minimum remains rigorous. The verifier returns $\min\{C_Z^{\mathrm{ball}}, C_Z^{\mathrm{ctr}}\}/\sqrt{mc}$ for the normalized contract in Equation (3). Every displayed operation is evaluated outward, and the range and finiteness guards from Section C.1 apply before the minimum is used. For m fixed readout rows, the added work is $O(mdc)$; the center uses $O(dc)$ state, while storing Z requires $O(md)$ entries unless the service already retains the readout.

C.6 Sequential Inverse Proof State

C.6.1 State invariant and strict initialization

The maintained inverse state consists of P, a computed defect tracker $\widehat{E}$, and a certified radius r satisfying

$$\left\|I_d - P\widetilde{H} - \widehat{E}\right\|_F \le r.$$

Strict initialization evaluates the defect directly and encloses its arithmetic error.

Throughout this subsection, $\widetilde{H}$ denotes the exact-real embedding of the entries of the stored floating-point array, whether that array is held in binary32 or binary64. Write H for the exact retain-set matrix. Thus the tracked defect is $E = I_d - P\widetilde{H}$, while the statistics-error bound later transfers this bound to $I_d - PH$.

At strict initialization, let $\widehat{E}_0 := \mathrm{fl}(I_d - P_0\widetilde{H}_0) \in \mathbb{F}_{64}^{d\times d}$, where $P_0, \widetilde{H}_0 \in \mathbb{F}_{64}^{d\times d}$. The standard product and subtraction models give

$$r_0 := \mathrm{up}\Big(\Big\|\mathrm{up}(\gamma_d\,|P_0|\,\Big|\widetilde{H}_0\Big| + \bar{u}\,\Big|\widehat{E}_0\Big|)\Big\|_F\Big) \in \mathbb{R}_{\geq 0}.$$

It satisfies $\left\|I_d - P_0\widetilde{H}_0 - \widehat{E}_0\right\|_F \leq r_0$, establishing the base case before any rank-one transition is attempted.

Each accepted transition adds three local error budgets to the previous radius:

$$r^+ = \mathrm{up}(r + \eta_{\mathrm{alg}} + \eta_{\mathrm{state}} + \eta_{\mathrm{fl}})\,.$$

Here η_{alg} encloses the use of computed low-rank intermediates, η_{state} the storage error in the updated arrays, and η_{fl} the evaluation error in the defect tracker.

C.6.2 Local arithmetic bounds

Before an update, let $P, \widetilde{H} \in \mathbb{F}_{64}^{d\times d}$, $x \in \mathbb{F}_{64}^d$, $s \in \{-1, +1\}$, and $E := I_d - P\widetilde{H}$, with stored values embedded exactly in the reals. Define

$$\begin{aligned} &q = Px, && z = P^\top x, && \beta = x^\top q = z^\top x, && \alpha = 1 + s\beta,\\ &\widehat{q} = \mathrm{fl}(Px), && \widehat{z} = \mathrm{fl}(P^\top x), && \widehat{\beta} = \mathrm{fl}(x^\top \widehat{q}), && \widehat{\alpha} = \mathrm{fl}(1 + s\widehat{\beta}). \end{aligned}$$

For comparison with the computed intermediates, set

$$\beta^\circ := x^\top \widehat{q}, \qquad \alpha^\circ := 1 + s\beta^\circ.$$

The standard matrix–vector model gives

$$\begin{aligned} \varepsilon_q &:= \mathrm{up}(\gamma_d\,|P|\,|x|), && |q - \widehat{q}| \leq \varepsilon_q,\\ \varepsilon_z &:= \mathrm{up}(\gamma_d\,\left|P^\top\right|\,|x|), && |z - \widehat{z}| \leq \varepsilon_z. \end{aligned}$$

Certified denominator guard. The implementation forms

$$\begin{aligned} \varepsilon_{\alpha,\mathrm{fl}} &:= \mathrm{up}\Big(\gamma_d\,\mathrm{up}(|x|^\top\,|\widehat{q}|) + \Big|\mathrm{TwoSum}(1, s\widehat{\beta}; \widehat{\alpha})\Big|\Big)\,,\\ \varepsilon_\alpha &:= \mathrm{up}\Big(\varepsilon_{\alpha,\mathrm{fl}} + |x|^\top\,\varepsilon_q\Big)\,,\\ \varepsilon_{qz} &:= \mathrm{up}\Big(|x|^\top\,(\varepsilon_q + \varepsilon_z)\Big)\,. \end{aligned}$$

The standard dot-product model and exact TwoSum residual give $|\alpha - \widehat{\alpha}| \leq \varepsilon_\alpha$. Hence $\widehat{\alpha} > \varepsilon_\alpha$ certifies $\alpha > 0$ before any division. An additional $\sqrt{u}(1 + \left|\widehat{\beta}\right|)$ policy floor sends numerically fragile positive denominators to the rebuild path; it is not used in the soundness argument.

The guard ensures $\widehat{\alpha} \neq 0$. Define the exact-real matrices formed from the computed low-rank intermediates,

$$P^{+\circ} := P - \frac{s\widehat{q}\widehat{z}^\top}{\widehat{\alpha}}, \quad H^{+\circ} := \widetilde{H} + sxx^\top, \quad E^{+\circ} := I_d - P^{+\circ}H^{+\circ}.$$

The implementation stores P^+ in binary64 and $\widetilde{H}^+$ in $\mathbb{F}_{\mathrm{s}}$, promoting the latter exactly before tracker evaluation. With both arrays embedded in the reals, let $E^+_{\mathrm{store}} := I_d - P^+\widetilde{H}^+$ and define

$$\mathcal{T}_s^\circ(Z) := Z - \widehat{q}\frac{s}{\widehat{\alpha}}\left(x^\top - \widehat{z}^\top\widetilde{H}\right), \qquad Z \in \mathbb{R}^{d\times d}.$$

For this map, Theorem 3 requires

$$\begin{aligned}\left\|E-\widehat{E}\right\|_F &\le r,\\ \left\|E^{+\circ}-\mathcal{T}_s^{\circ}(E)\right\|_F &\le \eta_{\mathrm{alg}},\\ \left\|E^{+}_{\mathrm{store}}-E^{+\circ}\right\|_F &\le \eta_{\mathrm{state}},\\ \left\|\mathcal{T}_s^{\circ}(\widehat{E})-\widehat{E}^{+}\right\|_F &\le \eta_{\mathrm{fl}}.\end{aligned}$$

The constructions below bound the final three terms.

Computed-intermediate mismatch bound

The exact vectors satisfy

$$x^\top z = x^\top P^\top x = (x^\top P x)^\top = x^\top q,$$

because both sides are real scalars. Hence

$$\begin{aligned}\left|x^\top(\widehat{z}-\widehat{q})\right| &= \left|x^\top(\widehat{z}-z)+x^\top(q-\widehat{q})\right|\\ &\le |x|^\top(\varepsilon_z+\varepsilon_q)\le \varepsilon_{qz}.\end{aligned}$$

Expanding $E^{+\circ} := I_d - P^{+\circ}H^{+\circ}$ gives the exact identity

$$\begin{aligned}&E^{+\circ}-\mathcal{T}_s^{\circ}(E)\\ &= -s(q-\widehat{q})x^\top + \frac{\widehat{q}}{\widehat{\alpha}}\left[s(\alpha^\circ-\widehat{\alpha})+x^\top(\widehat{z}-\widehat{q})\right]x^\top.\end{aligned} \tag{18}$$

To verify the coefficient of $\widehat{q}x^\top$, observe that

$$-s+\frac{s+x^\top\widehat{z}}{\widehat{\alpha}} = \frac{s(\alpha^\circ-\widehat{\alpha})+x^\top(\widehat{z}-\widehat{q})}{\widehat{\alpha}},$$

where $\alpha^\circ = 1+sx^\top\widehat{q}$. The componentwise budgets and $\|ab^\top\|_F = \|a\|_2\|b\|_2$ now imply

$$\begin{aligned}\eta_{\mathrm{alg}} := \mathrm{up}\Big(&\|\varepsilon_q\|_2\,\|x\|_2\\ &+\frac{\varepsilon_{\alpha,\mathrm{fl}}+\varepsilon_{qz}}{|\widehat{\alpha}|}\,\|\widehat{q}\|_2\,\|x\|_2\Big)\in\mathbb{R}_{\ge 0}.\end{aligned}$$

This quantity upper-bounds the Frobenius norm of Equation (18).

Stored-state update error. Let $\Delta P := -s\widehat{q}\widehat{z}^\top/\widehat{\alpha}$, $\Delta H := sxx^\top$, $\widehat{\Delta P} := \mathrm{fl}_{64}(\Delta P)$, and, for binary64 statistics, $\widehat{\Delta H} := s\,\mathrm{fl}_{64}(xx^\top)$. The standard models in Section C.1, including the exact componentwise TwoSum residuals $S_P = (P+\widehat{\Delta P})-P^+$ and $S_H = (\widetilde{H}+\widehat{\Delta H})-\widetilde{H}^+$, give

$$\begin{aligned}\mathcal{U}_{\Delta P} &:= \mathrm{up}\left(\bar{\gamma}_2\left|\widehat{\Delta P}\right|+|S_P|\right), \quad \left|P^+-(P+\Delta P)\right|\le\mathcal{U}_{\Delta P},\\ \mathcal{U}_{\Delta H} &:= \mathrm{up}\left(\bar{u}\left|\widehat{\Delta H}\right|+|S_H|\right), \quad \left|\widetilde{H}^+-(\widetilde{H}+\Delta H)\right|\le\mathcal{U}_{\Delta H},\end{aligned}$$

where $\bar{\gamma}_2 := \gamma_2/(1-\gamma_2)$. For binary32 statistics, the same quantity is obtained by promoting the post-update array exactly and enclosing its difference from the nominal update. A product-difference expansion then gives

$$\begin{aligned}\eta_{\mathrm{state}} := \mathrm{up}\big(&\|\mathcal{U}_{\Delta P}\|_F\left(\left\|\widetilde{H}\right\|_F+\|\Delta H\|_F\right)\\ &+\left(\|P\|_F+\|\Delta P\|_F\right)\|\mathcal{U}_{\Delta H}\|_F\\ &+\|\mathcal{U}_{\Delta P}\|_F\,\|\mathcal{U}_{\Delta H}\|_F\big)\ge\left\|E^{+}_{\mathrm{store}}-E^{+\circ}\right\|_F.\end{aligned}$$

Tracker-evaluation error. The tracker evaluates

$$\begin{aligned}\widehat{w} &:= \mathrm{fl}(\widetilde{H}^\top\widehat{z}), \qquad \widehat{v} := \mathrm{fl}(x-\widehat{w}),\\ \widehat{a} &:= \mathrm{fl}((s/\widehat{\alpha})\widehat{v}), \quad \widehat{C} := \mathrm{fl}(\widehat{q}\widehat{a}^\top).\end{aligned}$$

The standard componentwise models give

$$\mathcal{U}_w := \mathrm{up}\Big(\gamma_d \left|\widetilde{H}\right|^\top |\widehat{z}|\Big), \qquad \mathcal{U}_v := \mathrm{up}(\mathcal{U}_w + \bar{u}\,|\widehat{v}|)\,,$$
$$\mathcal{U}_a := \mathrm{up}(\mathcal{U}_v/\,|\widehat{\alpha}| + \bar{\gamma}_2\,|\widehat{a}|)\,, \quad \mathcal{U}_C := \mathrm{up}\Big(|\widehat{q}|\,\mathcal{U}_a^\top + \bar{u}\left|\widehat{C}\right|\Big)\,.$$

Finally, let $S_E := (\widehat{E} - \widehat{C}) - \widehat{E}^+$ be the exact TwoSum residual for $\widehat{E}^+ := \mathrm{fl}(\widehat{E} - \widehat{C})$, and set $\mathcal{U}_S := |S_E|$. Therefore

$$\eta_{\mathrm{fl}} := \mathrm{up}(\|\mathcal{U}_C + \mathcal{U}_S\|_F) \geq \left\|\mathcal{T}_s^\circ(\widehat{E}) - \widehat{E}^+\right\|_F.$$

C.6.3 RADIUS PROPAGATION THEOREM

Theorem 3 (Sequential proof-state invariant). *Suppose* $\left\|E - \widehat{E}\right\|_F \le r$. *Let* η_{alg}, η_{state}, *and* η_{fl} *satisfy*

$$\left\|E^{+\circ} - \mathcal{T}_s^{\circ}(E)\right\|_F \le \eta_{\mathrm{alg}},$$
$$\left\|E^{+}_{\mathrm{store}} - E^{+\circ}\right\|_F \le \eta_{\mathrm{state}},$$
$$\left\|\mathcal{T}_s^{\circ}(\widehat{E}) - \widehat{E}^{+}\right\|_F \le \eta_{\mathrm{fl}}.$$

Then

$$r^{+} = \mathrm{up}(r + \eta_{\mathrm{alg}} + \eta_{\mathrm{state}} + \eta_{\mathrm{fl}})$$

satisfies

$$\left\|E^{+}_{\mathrm{store}} - \widehat{E}^{+}\right\|_F \le r^{+}.$$

All three local bounds can be evaluated in $O(d^2)$ *work per rank-one update.*

Proof of propagated-radius soundness

The map $\mathcal{T}_s^{\circ}$ has identity linear part, so $\mathcal{T}_s^{\circ}(E) - \mathcal{T}_s^{\circ}(\widehat{E}) = E - \widehat{E}$. Inserting the nominal exact update and applying the triangle inequality gives

$$\begin{aligned}\left\|E^{+}_{\mathrm{store}} - \widehat{E}^{+}\right\|_F &\le \left\|E^{+}_{\mathrm{store}} - E^{+\circ}\right\|_F \\ &\quad + \left\|E^{+\circ} - \mathcal{T}_s^{\circ}(E)\right\|_F \\ &\quad + \left\|E - \widehat{E}\right\|_F + \left\|\mathcal{T}_s^{\circ}(\widehat{E}) - \widehat{E}^{+}\right\|_F \\ &\le r + \eta_{\mathrm{state}} + \eta_{\mathrm{alg}} + \eta_{\mathrm{fl}} \le r^{+}.\end{aligned}$$

Strict initialization gives the base case; induction proves the invariant for every guarded tracker update.

D PUBLISHED-METHOD APPLICATIONS

Candidates are fixed before post-hoc certification. Independent reference intervals supply offline conformance labels and are excluded from online timing. Service protocols and reference construction appear in Sections E and F.

D.1 APPLICATION DETAILS

We use pinned official implementations and certify emitted candidates post hoc without changing their update rules. Exact source revisions, instrumentation patches, and reproduction commands are recorded in the artifact.

Newton-update removal. We use the official MNIST 3-vs-8 configuration of Guo et al. (2020), leaving its objective perturbation, update, cumulative budget, and threshold unchanged. Native closure encloses the source terms through the first retraining action, and direct recertification checks the same candidate at that stop. Strict closure can move retraining earlier, while the tighter direct certificate can expose a conservative source action.

CEU supported-fidelity characterization. Post-update RRV evaluates the emitted CEU head under an independently declared retraining-reference contract (Wu et al., 2023). We fixed $\tau_{\mathrm{par}} = 10^{-3}$ before inspecting the outputs. The reference is the unique minimizer of retained-node mean softmax cross-entropy plus $\lambda \|\theta\|_2^2/2$, giving the curvature floor $\mu^{-} = \lambda$ in Proposition 3. Directed arithmetic encloses the retained-objective gradient; verified optimization is used only for offline tightness evaluation. The certificates cover their reference intervals and quantify the weaker fidelity supported by the snapshots alongside CEU's native randomized guarantee.

ScaleGUN gradient application. Native closure evaluates the same aggregate gradient quantity consumed by ScaleGUN's propagation and optimization budget (Yi & Wei, 2025). The candidate is unchanged during certification.

D.2 Newton Direct-Perturbation Calibration

We follow the Newton direct-perturbation mechanism of Zou et al. (2025) in its published synthetic logistic-ridge configuration, evaluating one- and two-step Newton centers over 100 leave-one-out states. Native closure bounds displacement from the exact-real Newton center, while On-demand RRV certifies the executed center against a verified retained-set reference.

To separate source slack from direction-free composition, let $p \in \{32, 64\}$ denote the execution precision and define

$$A = \frac{\left\|\bar{\theta} - \theta^{\mathrm{R}}\right\|_2}{r_{\mathrm{nom}}}, \qquad B(p) = \frac{\left\|\theta_p - \bar{\theta}\right\|_2}{r_{\mathrm{nom}}},$$
$$C(p) = \frac{\left\|\theta_p - \theta^{\mathrm{R}}\right\|_2}{r_{\mathrm{nom}}}.$$

Directed enclosures produce $A^+, B^+(p), C^+(p)$ for every paired state (Table 3). We form each reported ratio within a state before aggregating. For FP64, $C^+(64)$ is near zero in some states, so we report the absolute composition–direct gap rather than an unstable ratio. Direction-free composition remains looser under both execution precisions on this population.

Table 3: **Detailed published-method application results.** Newton-perturbation bounds are normalized by the nominal radius; ScaleGUN ratios report the source budget consumed. Newton ratios are formed per state before aggregation; FP64 uses an absolute gap because its direct bound can be near zero.

(a) Newton removal

Seed	Native	ExecCert	Δ	Direct upper
0	251	249	-2	.00304
1	232	232	0	.00324
2	277	273	-4	.00421

(b) Newton direct perturbation

Bound	Median	P95	Max
Source center, A^+	.087	.341	.945
FP32 B^+	.175	.181	.186
FP32 C^+	.111	.291	.874
FP32 comp./direct	**2.28**×	2.49×	2.56×
FP64 B^+	.067	.070	.074
FP64 C^+	.020	.271	.871
FP64 comp.–direct	.134	.141	.148

(c) ScaleGUN

Dataset	Cheap	Heads	Median	Max
Cora	12	84	.59%	.83%
ogbn-arxiv	3	120	1.24%	1.28%
All	15	204	.68%	1.28%

9 additional checkpoints use the source fallback.

(d) CEU

Fidelity	Median	Max
Realized error	8.86%	9.47%
RRV upper	14.06%	14.84%

Declared tolerance: $\tau_{\mathrm{par}} = 0.10\%$.

E Service Evaluation and Controls

E.1 Experimental Systems and Common Protocols

3B safety service. The deployment service uses a frozen Llama-3.2-3B-Instruct backbone (Meta AI, 2024) with a $d = 3073$ binary ridge head on PKU-SafeRLHF-QA (Ji et al., 2025). We replay seven 1,000-request deletion streams under a 10^{-3} release tolerance and evaluate fixed-readout contracts on 3,000 sealed representations.

Controlled diagnostics. The controlled experiments use DINOv2/CIFAR-10 and RoBERTa/AG-News multiclass ridge heads. The tolerance study fixes 14,000 outputs, and the stress study crosses three deletion streams, three conditioning levels, and two statistics storage precisions, giving 18 trajectories. Independent retained-row reconstruction supplies all offline reference labels.

Verification constructions. The guarantee-matched comparison holds the candidate, fixed readout, release contract, and conformance labels constant while varying only how certificate evidence is constructed. Three held-out streams at seven declared checkpoints give 21 matched states. The six constructions are summarized below.

Table 4: Verification constructions in the guarantee-matched comparison.

Construction	Build at release	Retained rows	Reuse inverse	Readout
On-demand RRV: retained residual	✓	✓	✗	Norm-ball
On-demand RRV: corrected residual	✓	✓	✓[a]	Error-centered
On-demand RRV: fresh verified factor	✓	✗	✗	Error-centered
Maintained verified factor	✗	✗	✗	Error-centered
Incremental RRV: norm-ball	✗	✗	✓	Norm-ball
Incremental RRV: directional	✗	✗	✓	Error-centered

[a] Partial inverse reuse.

E.2 Reference-Transfer Soundness and Tightness

Tolerance–coverage frontier. For each of the 14,000 fixed outputs, the independent retain-set audit computes the realized minimum tolerance τ_{true}, while RRV computes the certified threshold τ_{cert}. We sweep the deployment tolerance from 10^{-5} to 10^{-2}; every output satisfies $\tau_{\mathrm{cert}} \geq \tau_{\mathrm{true}}$.

Stored-equation stress comparison. This experiment tests whether certifying the stored equation is equivalent to certifying the exact retain-set reference as statistics drift. The 18 trajectories defined above evaluate the stored equation and its certificate in binary64 while varying statistics storage precision. Independent retained-row reconstruction labels conformance at $\tau_{\mathrm{par}} = 10^{-3}$. Coverage and false-release counts appear in Table 5.

Table 5: **Matched-output conformance on nine workload configurations per precision.** Coverage is conditional on a conforming output. The incremental row reports the first check, before any proof-only rebuild. Four FP32 reference intervals remain unresolved and are not counted as conforming or nonconforming.

Path	Total	Conforming	Certified conforming	False release
FP32 stored equation	9,000	7,293	7,293	1,703
FP32 + Incremental RRV	9,000	7,293	4,838	0
FP64 + Incremental RRV	9,000	9,000	9,000	0

E.3 Evidence Reuse and Lifecycle Cost

Guarantee-matched comparison. The On-demand RRV constructions form a retained-data residual, a corrected residual, or a fresh verified factor when certification is requested. The maintained-factor baseline instead downdates a factor after every deletion and validates its residual at release. The two Incremental RRV constructions share the inverse-defect proof state and differ only in their readout certificate. The corrected-residual construction is the only nonstandard baseline: it computes

$$\theta_{\mathrm{corr}} = \widetilde{\theta} + P(\widetilde{B} - \widetilde{H}\widetilde{\theta})$$

and evaluates a retained-data residual at θ_{corr}. If this residual gives $\left\|\theta_{\mathrm{corr}} - \theta^{\mathrm{R}}\right\|_F \leq C_{\mathrm{corr}}$, it still certifies the original candidate through

$$\left\|Z(\widetilde{\theta} - \theta^{\mathrm{R}})\right\|_F \leq \left\|Z(\widetilde{\theta} - \theta_{\mathrm{corr}})\right\|_F + \|Z\|_2 \, C_{\mathrm{corr}}.$$

All terms are evaluated outward, so every construction certifies the executed snapshot rather than substituting θ_{corr} for release.

Table 6: Complete guarantee-matched results on 21 fixed states. Latency summarizes 10 repetitions per state; stored state excludes retained rows.

Verification method	Retain scan	Certified release states	Med. (s)	P95 (s)	State (MiB)
On-demand RRV (retained residual)	Yes	0/21	5.163	5.410	72.1
On-demand RRV (corrected residual)	Yes	21/21	5.341	5.696	286.5
On-demand RRV (fresh verified factor)	No	21/21	2.611	3.593	286.5
Maintained verified factor	No	21/21	2.453	3.678	358.6
Incremental RRV (norm-ball)	No	0/21	0.801	0.859	360.3
Incremental RRV (directional)	No	21/21	1.056	1.163	430.6

Decision agreement. Across 56 checkpoints from seven streams, margin certificates cover 95.0–95.8% of declared decisions. The independent offline audit gives 99.96% mean observed agreement rather than certificate coverage.

Release-frequency accounting. We measure output generation, certified-statistics maintenance, and proof-state maintenance across all 1,000 requests in each of three streams. Release-check cost is the per-stream median over the seven declared checkpoints, with 10 paired repetitions per checkpoint. We combine these measured components at 7, 25, 50, 100, 250, 500, and 1,000 checks per 1,000 requests. The three routes share candidate-generation work; fresh factor pays no per-request proof maintenance, while maintained factor and directional Incremental RRV do. No route produces a false release on the matched states, and maintained factor requires no rebuild during the stream replays. Linear component accounting places the directional/fresh crossover at 21.5 checks and the maintained/fresh crossover at 214.3 checks per 1,000 requests.

E.4 Precision and Representation Boundaries

Producer-precision stress. Within each producer configuration, Incremental RRV and verified-factor On-demand RRV inspect the same output; changing statistics precision defines a separate candidate population. Lifecycle totals include all online work of the selected path, while independent reference audits are excluded. The 4,162 lower-precision proof refreshes are triggered by a bound above tolerance, not by invalid inverse state. The median bound moves from 1.014τ before a committed refresh to 0.830τ afterward; every commit recovers certification. All attempts together consume 4.39% of the FP32 lifecycle and no commit changes a candidate. Table 5 reports the first check, whereas Table 7 reports the final decision on the same population. Unresolved reference intervals are excluded from the conforming denominator.

Table 7: **Producer precision and certificate construction.** Rows change the emitted candidate; within each row, columns compare the same candidates. Coverage is conditional on conformance and uses the final post-rebuild decision; proof-only rebuilds do not change candidates.

	On-demand RRV			Incremental RRV		
Producer	Certified (%)	Life (s)	State (MiB)	Certified (%)	Life (s)	State (MiB)
FP32	**99.97**	**444.7**	**11.4**	67.24	502.1	20.4
FP64	**100.00**	**378.4**	**9.1**	**100.00**	413.5	18.2

Contract–format compatibility frontier. This protocol asks whether the final representation can express an artifact that satisfies the selected contract. We encode each format from the same master head, exhaustively evaluate the BF16 and FP16 codebooks under round-to-nearest-even, and rigorously enclose the minimum feasible tolerance $\tau_{\min}$ shown in Figure 3(b).

F Independent Reference Construction

F.1 Offline Verified Reference and Evaluation Labels

Offline labels are reconstructed from retained rows, independently of the service arrays and update path. A verified inverse-defect certificate encloses the retraining solution, and a high-precision dyadic reconstruction confirms all classifications. For its solution $\theta_t^\star$, directed evaluation of $r_t = B_t - H_t\theta_t^\star$ gives

$$r_{\mathrm{ref,t}} = \frac{\|r_t\|_F}{\lambda}, \qquad \left\|\theta_t^\star - \theta_t^{\mathrm{R}}\right\|_F \le r_{\mathrm{ref,t}}.$$

Directed norm bounds and the triangle inequalities of Equation (2) then give a rigorous relative-error interval; precision is increased until each classification is resolved.

F.2 Newton Direct-Perturbation Reference Construction

To close the Newton direct-perturbation mechanism against its exact-real rule, let $s_j = \theta_j^{64} - \theta_{j-1}^{64}$ denote the exact-real difference of two stored binary64 iterates. The verifier represents this dyadic difference as the unevaluated sum returned by TwoSum; it does not identify s_j with an ordinary rounded subtraction. This definition also handles a solver step swallowed by the state addition: the realized step is then zero. Let $g(\theta) = \nabla L_{\mathrm{ret}}(\theta)$ and $H(\theta) = \nabla^2 L_{\mathrm{ret}}(\theta)$. On a convex region containing the exact-real and replay iterates, and hence the segments connecting them, assume $mI \preceq H(\theta) \preceq MI$ and $\|H(\theta) - H(\theta')\|_2 \le L\,\|\theta - \theta'\|_2$. We define the Newton-equation residual $r_j = H(\theta_{j-1}^{64})s_j + g(\theta_{j-1}^{64})$. Directed MPFR evaluation applies the Hessian to both components of the exact difference and bounds $\|r_j\|_2$. We propagate

$$\eta_j = \eta_{j-1} + \frac{M\eta_{j-1} + L\eta_{j-1}\,\|s_j\|_2 + \|r_j\|_2}{m},$$
$$\eta_0 = 0.$$

Thus η_j bounds the replay center's distance from the exact-real Newton iterate, and $\left\|\theta_j^{32} - \theta_j^{64}\right\|_2 + \eta_j$ closes the execution error used by the compositional Laplace interface. This route needs no retraining reference; the direct route instead encloses the candidate against a verified retained-set reference.