HERA: AN EMPIRICAL AUDIT OF PARTITION GUARANTEES ON UNLEARNABLE DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Unlearnable examples perturb training images to impair learning, but recovery accuracy alone does not establish stability to later record edits or equality to retraining after deletion. We use HERA, a testbed of deterministic hash-partitioned ridge ensembles over fixed image views or frozen ImageNet-pretrained ResNet-18 features, to audit these properties on four archived CIFAR-10 protection instances. The partition guarantees are established; our contribution is an empirical comparison of clean-label allocation, regularization and shard count. Equal-weight ensembles combining 40,000 protected records with 2,000 clean records improve mean accuracy and certification over protected-only ensembles at all 32 tested five-shard instance–penalty settings, without uniformly dominating clean-only. Their certificates cover edits to the combined training set. Training-only shard-local cross-validation retains higher combined mean one-record certified accuracy than both separate options for every instance, although the NTGA difference from protected-only remains unresolved. The selector maximizes fold accuracy on each condition's training distribution using a previously explored penalty grid; it does not optimize ensemble certification. Tuned combined CUDA reaches 71.30% certified accuracy versus clean-only's 69.71%, but remains 4.07 accuracy points lower. Separate-resource comparisons expose strong regularization sensitivity: clean-only certification peaks among tested shared penalties at 70.21%, versus CUDA's 64.92%, before declining at larger penalties. Training-only PCA reduces clean-only penalty sensitivity; adding an intercept alone leaves the pooled budget dip. Separate pooled additions reveal clean-budget dependence, with CUDA reducing accuracy at 2,000 clean labels despite helping at 250. Sparse deletion checks match local and fresh states and predictions while reusing untouched shards. These fixed-feature results motivate joint accuracy/certification reporting with shard-local selection, not claims of optimal allocation or robustness to dense perturbations.

1 INTRODUCTION

Unlearnable examples (UEs) seek to prevent unauthorized model training by adding small perturbations that preserve the human utility of images (Huang et al., 2021). Recovery methods attempt to learn useful predictors from these protected images. For an authorized curator, however, recovery accuracy answers only part of the operational question: after training, a few records may need to be corrected or deleted. The curator must then determine which predictions remain stable and whether the updated learner matches retraining without the deleted records.

Recovery defenses, record certification, and machine unlearning address different questions. Deep Partition Aggregation certifies partition votes, while isolated deterministic retraining supports local deletion (Levine & Feizi, 2021; Bourtoule et al., 2021). These guarantees alone do not establish recovery accuracy on protected images, correct certificate coverage, or savings when deletion requests touch many shards. We therefore study their joint empirical behavior rather than claim a new guarantee architecture.

HERA names our audit testbed; it uses the same stable hash-partition ensemble for three tasks (Figure 1). *Recovery* measures clean-holdout accuracy after protected-data training; *record robustness* measures prediction invariance to later sparse edits; *exact algorithm-relative deletion* requires

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.

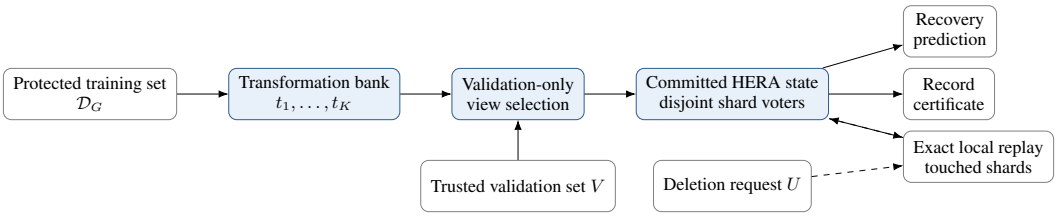


Figure 1: Routed HERA: fixed views and a trusted clean routing set select one deterministic voter per stable hash shard. The same voters yield recovery predictions and sparse-record certificates. Deletion refits touched shards under fixed external state; sparse audits exercise reuse of three or four untouched shards.

equality to fresh fitting on retained records. Fixed views and a trusted clean router select one deterministic classifier per shard. A frozen-feature variant tests a stronger representation. Ridge heads make replay and vote margins directly inspectable; routed views provide a low-capacity control. The sparse-edit guarantee does not cover the original dense UE perturbation.

Across five seeds and four archived CIFAR-10 protection instances, routed recovery does not consistently beat fixed-view or bagging controls. A head trained on 2,000 clean labels exceeds several protected ensembles in accuracy, but certification rankings depend on partitioning and regularization. Stronger shared penalties reverse CUDA's CA(1) ordering at five shards, and REM's at the two largest penalties. We therefore compare resource allocations jointly on accuracy and certification, while sparse replay audits separately verify untouched-shard reuse. These results motivate three contributions:

1. **Combined resources and joint sensitivity.** Equal-weight clean-plus-protected ensembles improve protected-only mean accuracy and certification across the tested five-shard grid, without uniformly beating clean-only. Separate-resource rankings reverse across penalties. Accuracy-based shard-local tuning retains higher combined mean CA(1) than both separate options, with the NTGA protected-only difference unresolved. These results motivate joint evaluation across resource allocations, penalties and shard counts.
2. **Recovery and certificate diagnostics.** Matched controls distinguish accuracy from correct sparse-record certification. On EMIN/REM, routing reduces certificate coverage despite little accuracy change; exact paired decompositions locate margin losses among jointly correct predictions.
3. **Operational deletion audit.** Sparse checks compare selected views and fitted arrays and verify reuse of untouched states; refit timings and occupancy analysis characterize the tested locality benefit.

2 Related work and closest comparators

Unlearnable data and recovery. UE methods impair learning through small image perturbations, including error-minimizing noise (Huang et al., 2021), adversarial poisons (Fowl et al., 2021), robust/stable variants (Fu et al., 2022; Liu et al., 2024), NTGA (Yuan & Wu, 2021), and convolutional CUDA perturbations (Sadasivan et al., 2023). Recovery approaches include compression (Liu et al., 2023), feature reweighting (Sandoval-Segura et al., 2023), nonlinear transformations (Hapuarachchi et al., 2026), augmentation-based defenses (Zhu et al., 2024), and D-VAE purification (Yu et al., 2024). Deep ensembles with nonlinear transformations report at least 89% mean CIFAR-10 accuracy across twelve protections (Hapuarachchi & Xiong, 2025). Those literature accuracies are not directly comparable to our training-split holdout. Our controls assess recovery, certification, and replay within the stated protocol, not deep-defense competitiveness.

Training-data certification. Randomized smoothing certifies label flips (Rosenfeld et al., 2020); certified bagging and BagFlip address broader poisoning perturbations (Jia et al., 2021; Zhang et al., 2022). Deep Partition Aggregation (DPA) is the closest structural antecedent: it hashes records into disjoint partitions and certifies aggregate predictions against insertions and deletions (Levine

& Feizi, 2021). Abstract Gradient Training bounds parameter sets under data perturbations and addresses poisoning, removals, additions, and substitutions (Sosnin et al., 2026). Finite Aggregation and Run-off Election strengthen aggregation defenses (Wang et al., 2022; Rezaei et al., 2023); we use the basic conservative partition bound. Practical aggregation studies also examine how data size and model complexity constrain voter count (Wang & Feizi, 2023). Provably Unlearnable Data Examples certifies learnability under parameter uncertainty (Wang et al., 2025), a different target from our later record-edit stability.

Machine unlearning and the intersection. SISA localizes retraining through sharding and slicing (Bourtoule et al., 2021); certified-removal methods provide guarantees relative to retraining (Guo et al., 2020; Koloskova et al., 2025). Recent work examines unlearning of poisoning effects (Pawelczyk et al., 2025) and its interaction with UEs (Zhao et al., 2026). A recent systematization examines unlearnability and unlearning jointly (Zhang et al., 2026). A deterministic partition ensemble already admits both vote certification and reuse of unchanged shards. HERA's contribution is the controlled joint evaluation on protected images, with matched recovery controls and explicit separation of replay equality from potential savings. Table 33 distinguishes published mechanisms from properties implied by deterministic isolation.

3 Problem formulation

We distinguish the *protector* from the *learner*. The protector releases $\mathcal{D}_G = \{(x_i^G, y_i)\}_{i=1}^n$ for protection G, where x_i^G is a protected image and y_i its label. Routed-view HERA uses the protected training images and a disjoint trusted clean routing set V; it does not receive the protection name or generator parameters. The frozen variant additionally uses an externally pretrained encoder. Clean counterparts used to reconstruct record identifiers are preprocessing resources, not inputs to the shard fitting or routing objective. We evaluate fixed archived protection instances, not a worst-case or adaptive protector.

Definition 1 (Recovery accuracy)**.** *For fixed protected data $\mathcal{D}_G$, learner A_ξ with seed ξ, and clean test distribution P, define recovery accuracy as*

$$R(A, G) = \mathbb{E}_\xi \Pr_{(x,y)\sim P}[A_\xi(\mathcal{D}_G)(x) = y].$$

The seed controls hashing and the router/evaluation split; the expectation is conditional on the fixed data and protocol. The clean utility cost is $R(A, \text{clean}) - R(A, G)$.

Definition 1 isolates recovery from the two guarantees below: high recovery accuracy is neither a training-record certificate nor evidence of deletion.

We report recovery R, certified accuracy $\text{CA}(q)$, and idealized deletion work separately. The minimum accuracy across the four evaluated instances summarizes observed worst-instance performance, not a guarantee over unseen protections. Here q-stability means $A(D)(x) = A(D')(x)$ for every admissible dataset within q record operations, and exact deletion means equality of learned state to $A(D \setminus U)$ under fixed external configuration. Neither definition concerns test-time ℓ_p robustness.

4 Method

4.1 Overview

Disjoint shards limit each record to one voter. Fixed feature maps and shard-local fitting preserve this separation, allowing the same learner to support both vote certification and local deletion. We use ridge classifiers for transparent deterministic replay and feature views to study recovery without end-to-end representation training.

Architecture. Figure 1 and Algorithm 1 give the routed procedure: fixed views, validation-only selection, plurality voting and touched-shard retraining. No module receives the protection name. Ridge enables inspectable replay; locality allows any deterministic shard learner.

Training contract. A public hash H assigns each stable record ID to one of M shards. IDs are assigned once and never renumbered after deletion. The original NTGA study uses initial array-position IDs; the cross-protection study uses reconstructed CIFAR-10 indices. Hash seed, input order, solver, and tie rules remain fixed. The trusted routing set is external to shard training.

The learned state contains ordered retained IDs, selected views where applicable, normalization arrays, and coefficients under the fixed configuration. Deletion refits and reselects every candidate view on touched shards while preserving untouched state. Stable membership bounds each record's influence; fixed routing data and deterministic fitting make local replay equal fresh fitting. Changing these inputs can invalidate the guarantees. Historical runs saved predictions and summaries; newer sparse checks also save both compared learned states, but not an end-to-end deployable learner (Appendix D).

Algorithm 1: HERA training, certification, and deletion. Given protected data $\mathcal{D}_G$, trusted validation set V, transformation bank $\mathcal{T}$, and voter count M: (1) audit ranges, labels, available perturbation metadata, and split hashes; (2) assign every record to $H(\mathrm{id}) \bmod M$; (3) train every candidate-view probe per shard and select by routing accuracy; (4) freeze the selected view and deterministic shard learner; (5) predict by plurality vote and return the top-two vote margin; (6) on deletion request U, remove U and repeat view selection and training only for shards intersecting U.

4.2 Transformation bank and validation-only routing

Pooling, color, and gradient views expose different fixed summaries of each image. Comparing them tests whether inexpensive representations retain useful signal under protection. The bank contains five deterministic views of 32×32 images: RGB averaged over 4×4 windows (an 8×8 RGB grid, 198 features); grayscale over 2×2 windows (a 16×16 grid, 262); two chroma channels over 4×4 windows (134); two grayscale-gradient channels over 4×4 windows (134); and coarse RGB over 8×8 windows (a 4×4 RGB grid, 54). Dimensions include six per-image channel means/standard deviations. Each shard fits all five views, selecting maximum accuracy on the 2,000-image router; ties select the lexically greatest view name. The other 8,000 images are excluded from routing.

For shard feature matrix Φ (rows are records), compute columnwise statistics $\mu = \mathrm{mean}(\Phi)$ and $\sigma = \mathrm{std}(\Phi) + 10^{-6}$, then form $Z = [(\Phi - \mu)/\sigma, \mathbf{1}]$. With one-hot targets Y, routed heads fit

$$W = (Z^\top Z + \lambda \,\mathrm{diag}(1, \ldots, 1, 0))^{-1} Z^\top Y.$$

Here W contains classifier coefficients, $\mathbf{1}$ is the intercept column, and $\lambda = 10$ controls ridge regularization. The intercept is unregularized. Prediction takes the largest class score, with smallest-index ties.

Frozen-feature variant. A fixed ImageNet-pretrained ResNet-18 supplies 512 features, with no routing. Its retained implementation standardizes features using $\sigma = \mathrm{std}(\Phi)$, replacing values at most 10^{-12} by 1, and fits $W = (Z^\top Z + \lambda I)^{-1} Z^\top Y$ on $Z = (\Phi - \mu)/\sigma$. Prediction uses ZW without an intercept. This differs from routed heads; centering features does not make the omitted class-prior intercept equivalent. The hash encoding also differs (Appendix C). Thus this is a second learner variant, not a pure representation ablation. Both satisfy the same deterministic locality argument where their shard procedures are defined.

4.3 Disjoint-shard certified aggregation

Partition the training indices into M disjoint shards before observing outcomes and train one deterministic f_j per shard, holding external routing data and feature maps fixed. At x, let $v_c(x) = \sum_{j=1}^{M} \mathbf{1}[f_j(x) = c]$. Predict $A = \arg\max_c v_c(x)$ with a fixed tie rule, and let B be the strongest competitor under that rule.

Theorem 1 (Tie-agnostic record-operation certificate)**.** *Suppose admissible record insertions, deletions, or replacements leave the shard procedures defined and can change at most s distinct shard voters. The prediction at x is invariant whenever*

$$s < \frac{v_A(x) - v_B(x)}{2}.$$

Define $r_{\mathrm{shard}}(x) = \max\{0, \lfloor (v_A - v_B - 1)/2 \rfloor\}$. *This certifies up to* r_{shard} *affected shards, including the trivial zero-operation case when the initial vote is tied. For* $0 \le q \le r_{\mathrm{shard}}(x)$*, it certifies* q *record operations whenever they affect at most* q *shards: insertions and deletions satisfy this under stable hashing and deletion without renumbering, and a replacement does so when it preserves the stable identifier; an identity-changing replacement is conservatively counted as two operations. This conservative DPA form is independent of class-tie order; a tie-aware rule can improve an even margin by one operation when* A *wins the relevant tie.*

Proof. Each changed voter reduces the leading margin by at most two; zero edits preserve fixed ties. Appendix A gives the complete proof. □

Operational meaning. With five voters, counts 5–0, 4–1, and 3–2 give radii two, one and zero. Stability does not imply correctness: certified accuracy requires both correctness and radius at least q. The maximum five-voter radius covers two records, or 0.005% of 40,000 training records. This addresses small corrections or deletions, not users with more records than the budget or the original dense UE perturbation. Appendix E gives the certified-accuracy decomposition and occupancy variance.

4.4 Exact deletion for the sharded learner

Let A_{H} be the complete deterministic HERA algorithm, including hashing, the fixed routing set, the deterministic view-selection rule, feature maps, and solver. For deletion set U, $\bar{A}_{\mathrm{H}}$ accesses retained training records, removes U, and reruns view selection and training only on shards intersecting U.

Theorem 2 (Exact algorithm-relative unlearning)**.** *For every dataset* D *and deletion set* $U \subseteq D$ *for which all original and retained shard procedures are defined,*

$$\bar{A}_{\mathrm{H}}(U, A_{\mathrm{H}}(D)) = A_{\mathrm{H}}(D \setminus U),$$

provided IDs and input order are stable, hashing and training are deterministic, external routing data/feature maps are unchanged, and all data-dependent state (including view selection where used) is recomputed on every touched shard. For the implemented ridge learners we restrict this statement to nonempty retained shards; no empty-shard fallback is specified. Shard-local penalty selection with stable-ID folds is part of the deterministic fitter and preserves both guarantees when rerun on touched shards.

Proof. Stable hashing preserves untouched inputs; replay refits touched shards on exactly the fresh-run inputs. Appendix A gives the complete proof. □

If $u = |U|$ identifiers are independently and uniformly distributed across M shards and T_u is the number touched, then

$$\Pr(T_u = t) = \frac{(M)_t S(u,t)}{M^u}, \qquad \mathbb{E}[T_u] = M\{1 - (1 - 1/M)^u\},$$

where $(M)_t$ is a falling factorial and $S(u,t)$ is a Stirling number of the second kind. Under equal shard costs, the retraining-work fraction is T_u/M; for $u \ll M$, its expectation is approximately u/M. This distribution quantifies how often small requests touch most voters. Wall-clock savings can differ because feature extraction, model loading, and parallel execution add fixed costs. Exact unlearning concerns A_{H}, not an arbitrary trained model.

5 Experimental design

5.1 Questions, data, and variants

RQ1: how do clean/protected resource choices and recovery components affect accuracy? **RQ2:** how much correct accuracy is certifiable, and how does voter count M change it? **RQ3:** does replay agree with fresh fitting, and when can locality save work?

We use archived CIFAR-10 protection instances. The 40,000 protected records and 10,000 clean held-out records partition the original CIFAR-10 *training* split; the official test split is not used.

For each seed 2023–2027, 2,000 clean records form the router and 8,000 form evaluation. Within each run, all controls share evaluation indices, labels, and the relevant head/feature implementation. Clean counterparts are used in preprocessing to reconstruct original training-record identity, not to fit protected-data heads. Appendix C gives normalization and metadata limits; Table 32 identifies retained inputs.

Protocol variants. Routed heads have an intercept and keyed-integer hashing; frozen heads have no intercept and unkeyed-string hashing. Routed NTGA uses initial-position IDs; cross-protection runs use canonical IDs. All IDs remain stable after deletion. Appendices B.6 and C specify the historical sparse-hash exception and extraction details. Frozen features use ImageNet-1K V1 ResNet-18 without fine-tuning; routing indices serve only split consistency. Ridge uses direct solves with sum-loss $\lambda = 10$ unless specified; historical selection details appear in Appendix C.

Frozen resource comparisons use 2,000 clean or 40,000 protected records at $M \in \{3, 5, 10\}$. Shared mean-loss controls set $\lambda_j = \alpha n_j$ for shard size n_j, with $\alpha \in \{0.00125, 0.005, 0.025\}$ at $M = 5, 10$ and upward sensitivity grids $\{0.1, 0.5, 2.5\}$ and $\{12.5, 62.5\}$ at $M = 5$. Separate pooled additions append protected records to identical clean subsets of 250, 500, 1,000 or 2,000 labels; their constant-mean-loss check uses $\alpha = 0.005$. No evaluation-based penalty selection is performed.

5.2 Matched baselines

Pooled-RGB ERM fits one ridge head. *Router-only* fits a fixed coarse, RGB or frozen-feature head on the 2,000 clean routing records and evaluates on the remaining 8,000, without view selection. The protected-only frozen ensemble consumes no routing labels, so this compares available resources, not equal label use or equal certificates. The *oracle fixed-view envelope* chooses the best of five full-data views after evaluation and is non-deployable. *Transformation vote* isolates multi-view ensembling; *SISA-style pooled-RGB* votes five disjoint fixed-view heads. *Bootstrap bagging* votes five coarse-view or frozen-feature heads, with draws matched to shard size (exactly $n/5$ for frozen heads). It matches total sampled records, voter count, representation and head, not unique records or fitting cost: routed HERA fits five candidates per shard, bagging one. Fixed-coarse shards equal routed NTGA/CUDA when all select coarse. Matched EMIN/REM controls retain predictions under identical partitions, heads and evaluation indices (Appendix B.7).

5.3 Metrics and statistical protocol

For N evaluation records, let $\hat{y}_i$ and y_i be predicted and true labels, and $r_i = r_{\text{shard}}(x_i)$. Accuracy and certified accuracy are

$$\text{Acc} = \frac{1}{N}\sum_i \mathbf{1}[\hat{y}_i = y_i], \qquad \text{CA}(q) = \frac{1}{N}\sum_i \mathbf{1}[\hat{y}_i = y_i]\mathbf{1}[r_i \geq q].$$

The ratio $\text{CA}(q)/\text{Acc}$ gives coverage conditional on correctness; Δ denotes a paired percentage-point accuracy difference from a named baseline. Secondary diagnostics include classwise accuracy (Table 6) and voter disagreement; deletion reports touched shards, maximum parameter difference, and prediction agreement with fresh retraining. Seeds vary splits and hashes, not independent protection generation. Results give mean±SD, paired seed-bootstrap intervals (20,000 resamples), and wins across five partitions. Because partitions reuse the same arrays, these descriptive intervals summarize split/hash variability conditional on the fixed data, not independent-dataset uncertainty. Five runs offer limited precision. Tables round to two decimals, with exact halfway cases rounded away from zero; differences use unrounded means. Appendix B.1 gives paired t intervals (Table 7), sign counts, and McNemar checks; none establishes population generalization.

6 Results and analysis

6.1 RQ1: recovery and component attribution

Under the original NTGA protocol, routed accuracy is 25.56% versus coarse bagging's 25.38%; every router selects coarse RGB, so this result requires no heterogeneous routing (Table 4, Appendix B).

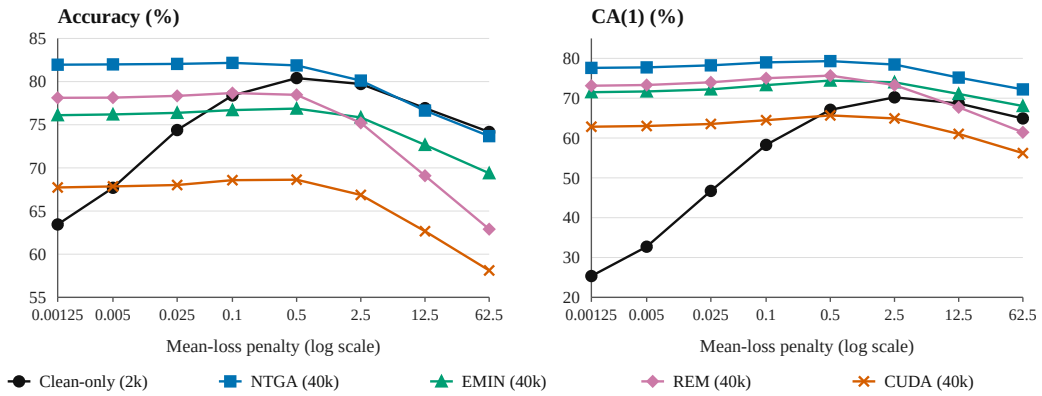


Figure 2: Separate-resource five-shard frozen ensembles: accuracy and CA(1) over all eight evaluated shared mean-loss penalties α ($\lambda_j = \alpha n_j$). Points are five-seed means; lines connect tested settings only. Clean-only CA(1) declines beyond $\alpha = 2.5$. Tables 22, 24 and 26 give SDs; the five largest penalties were evaluated only at five shards.

Table 1: Shard-local accuracy-based CV at $M = 5$: five split/hash-draw means (%). Clean-only (2k records) reaches 79.68 accuracy and 69.71 CA(1). Protected-only uses 40k records; combined uses 42k. All nine conditions use the same candidate grid. Tables 29–31 give SDs, paired intervals and selections; NTGA's combined-minus-protected CA(1) interval includes zero.

	Protected-only		Combined	
Instance	Accuracy	CA(1)	Accuracy	CA(1)
NTGA	82.12	78.67	82.41	78.97
EMIN	76.48	72.33	79.49	75.53
REM	78.41	74.10	81.12	76.74
CUDA	68.45	64.11	75.61	71.30

Takeaway 1 (evaluate the joint penalty response). Table 1 gives the tuned comparison. At the common untuned $\lambda = 10$, five-shard clean-only frozen accuracy/CA(1) is 74.34/46.67%, below all protected ensembles in CA(1) (Tables 17, 9). Figure 2 shows why a single setting is insufficient: increasing shared α from 0.025 to 2.5 raises clean-only CA(1) by 23.49 points, versus CUDA's 1.39. The clean gain decomposes into $17.72 + 6.32 - 0.55$: net margin gains on jointly correct predictions, certified gains on newly correct predictions, and certified losses on predictions becoming wrong. This locates the gain, without identifying its cause (Appendix B.14). Clean-only accuracy peaks among tested values at 80.41% ($\alpha = 0.5$); CA(1) peaks at 70.21% ($\alpha = 2.5$), then declines at 12.5 and 62.5. At these two largest penalties, where all conditions decline, clean-only also exceeds REM in mean CA(1) and every protected-only ensemble in mean accuracy; NTGA and EMIN retain higher CA(1). Neither is a tuned optimum. Shard-count comparisons are also penalty-dependent (Appendices B.12, B.13). Resource comparisons should therefore report accuracy and certification jointly across penalties and shard counts. CA(1) fixes one record operation, matching a single-record request; its relative budget differs across training-set sizes.

Equal-weight mixtures of 40,000 protected and 2,000 clean records improve mean accuracy and CA(1) over protected-only at every tested penalty; CA(1) gains span 0.03–7.31 points. They exceed both separate options at some settings, without uniform dominance (Appendix B.15). Certificates cover edits to all 42,000 records. Training-only shard-local tuning gives combined CUDA 71.30% CA(1) versus clean-only's 69.71%; combined means exceed both separate options for each instance, although NTGA's paired CA(1) gain is 0.30 points with interval $[-0.01, 0.60]$ (Appendix B.16).

Separate pooled additions improve all four frozen ridge means at 250 clean labels, but CUDA hurts at 2,000 (Table 14); constant mean-loss controls retain these signs (Appendix B.11). Nonlinear checks also show budget dependence (Appendix B.9, Table 15). HERA's bagging advantage remains unresolved (Appendix B.4).

Table 2: Matched canonical-ID cross-protection validation (five-seed mean±SD, %). The same records as Table 4 use different immutable IDs and hence different hash partitions. Coarse denotes one full-data coarse-view head.

Protection	Acc.	CA(1)	CA(2)	SISA	Bagging	Coarse
NTGA	25.60±.20	20.72±.17	16.36±.24	20.73	25.56	25.13
EMIN	33.45±.49	25.66±4.77	19.68±6.13	29.35	33.46	33.49
REM	37.60±.43	30.10±1.65	22.30±2.55	37.75	36.95	37.27
CUDA	36.96±.07	31.05±.31	25.34±.48	32.92	36.59	37.26

Table 3: Theory-driven validation. Certificate results are five-seed mean±SD (%). Deletion fixes $M = 5$ and reports the empirical mean number of touched shards over 10,000 random requests per seed across five hash seeds, with the occupancy-law expectation in parentheses. Median-radius entries give the range across seeds among correct predictions. The sweep uses original-order NTGA IDs and routed views only; the two panels are not paired conditions.

Certificate sweep					Deletion occupancy ($M = 5$)	
M	Acc.	CA(1)	CA(2)	Median r |correct	u	Touched (theory)
3	25.30±.24	19.74±.36	0.00±.00	1	1	1.000 (1.000)
5	25.56±.14	20.91±.22	16.20±.41	2	5	3.359 (3.362)
7	25.65±.25	21.28±.20	17.70±.13	3	10	4.461 (4.463)
9	25.77±.32	22.00±.34	18.59±.16	3	20	4.942 (4.942)
15	25.90±.24	22.69±.24	20.00±.29	4–5	–	–
25	26.58±.47	23.73±.35	21.40±.17	6	–	–
50	27.26±.24	24.92±.18	23.10±.20	8–9	–	–

Seed-level view comparisons appear in Table 5 (Appendix B).

Cross-protection recovery. Table 2 gives an observed minimum routed accuracy of 25.60% across the four instances. The canonical NTGA difference from bagging is only 0.04 points. HERA trails full-data coarse on EMIN by 0.04 points and CUDA by 0.30 (all five CUDA seeds), and trails disjoint pooled-RGB on REM by 0.15. EMIN is also effectively tied with bagging (a -0.0075-point mean difference). A canonical clean control reaches $41.58 \pm 0.23\%$, a 15.99-point clean-to-NTGA loss computed from unrounded means. Thus partitioning/routing does not consistently improve recovery.

Routing and certificate coverage. NTGA and CUDA select coarse in every shard. New fixed-coarse controls match EMIN/REM shards, seeds, heads, and evaluation records. Routing changes mean accuracy by $-0.07/+0.36$ points and CA(1) by $-3.26/-1.86$, respectively. Among jointly correct examples, margin changes alone reduce CA(1) by 1.69/1.58 points (Tables 8, 13; Appendix B.2).

Frozen-feature recovery. The frozen 2,000-label single head reaches $79.15 \pm 0.38\%$. Among the original protected-only ensembles, only NTGA's exceeds it; EMIN, REM and CUDA fall below it by 3.04, 1.03 and 11.41 points (Tables 9, 10). HERA and the SISA-style frozen control are identical. This single-head resource comparison is neither a marginal-data-value estimate nor a HERA-specific gain; the matched ensemble comparison appears in Table 1.

6.2 RQ2: CERTIFIED TRAINING-RECORD STABILITY

Takeaway 2 (record certification). Table 3 shows increasing routed accuracy and CA(1/2) at every tested shard count. At $M = 50$, four seeds have median certified radius eight among correct predictions and one has nine, while the maximum possible radius is 24. New frozen NTGA runs at $M = 5, 15, 25, 50$ keep accuracy near 82% while CA(1) rises from 77.61% to 79.48% (Table 13). For routed $M = 50$, CA(10) is 12.39% of all evaluation records. Table 11 in Appendix B.5 reports larger radii; these are record-level, not user-level guarantees. Figure 3 shows both frontiers. These certificates concern sparse edits, not dense NTGA; deletion cost depends on request occupancy.

6.3 RQ3: EXACT DELETION AND ITS EFFICIENCY BOUNDARY

Takeaway 3 (replay and locality). Thirty canonical NTGA checks use five seeds, both heads with their original hash conventions, and disjoint one-, two-, and three-record requests per seed. All compare equal to fresh fitting in ordered membership, views, normalization, coefficients, and predictions, while reusing untouched states. Both compared states are saved (Appendix B.7). New repeated timings cover both original hashes and five seeds, alternating local/fresh order (Table 16). They measure cached-feature refitting, excluding loading and persistence; no general systems-speed claim follows.

7 DISCUSSION: COMPARISON AND TRADEOFFS

The joint resource ranking depends on partitioning and regularization. Table 33 and Appendix F distinguish the established mechanisms and the sufficient-statistic deletion alternative.

Shard-local CV uses five stable-ID folds and fold-training-only normalization. It maximizes mean fold accuracy, breaks ties toward larger penalties and refits the shard. The grid follows earlier exploration; validation uses each condition's own training distribution, not evaluation labels or ensemble CA(1). All 45 local deletion checks match fresh fitting, including four changed selections. CUDA's 4.07-point tuned accuracy deficit illustrates why choosing resources by CA(1) alone can be misleading (Table 1).

8 LIMITATIONS

Evidence concerns four archived CIFAR-10 instances and five split/hash draws; generation hyperparameters are incompletely recovered. Primary comparisons use one frozen encoder and no-intercept heads. Frozen NTGA loses 2.57 accuracy points to its clean control; adaptive protection against this encoder is untested. Clean shards have about 400 records on 512 features at $M = 5$. At 500 labels and $\alpha = 0.005$, training-only 128-component PCA raises accuracy from 51.98% to 75.52%; an intercept alone gives 51.94% (Appendix G). This supports spectral sensitivity, not an identified double-descent mechanism. Certificates concern sparse edits, not dense UE perturbations, privacy or consent; even $M = 50$ certifies at most 24 records. Combined ensembles use equal weighting and five shards only. End-to-end learners, other datasets and clean-record weighting remain untested. Appendix D collects implementation boundaries.

9 CONCLUSION

For these fixed-feature ridge learners on four archived CIFAR-10 instances, a clean-only single head is a strong accuracy alternative to protected-data recovery: original protected-only routed ensembles exceed it only on REM and, inconclusively, CUDA; frozen ensembles exceed it only on NTGA. With five shards and $\lambda = 10$, clean-only frozen CA(1) is 46.67%, below all protected frozen ensembles. Stronger shared penalties reverse CUDA's ordering at five shards, and REM's at the two largest tested penalties, with clean-only reaching its highest tested CA(1), 70.21%, versus CUDA's 64.92% at $\alpha = 2.5$. At ten shards, CUDA's CA(1) advantage shrinks; the remaining difference depends on penalty scaling, making the joint comparison partition- and regularization-dependent. Separate pooled-head additions reveal budget dependence: all frozen ridge additions improve means at 250 labels, but CUDA hurts at 2,000. Matched EMIN/REM controls reveal little routing accuracy change but lower certificate coverage, including margin losses on jointly correct predictions. More shards improve sparse certification; frozen NTGA accuracy stays near 82%. Thirty original sparse checks and 45 tuned single-record checks save matching local/fresh states and predictions while reusing untouched shards. Equal-weight combined ensembles improve protected-only means throughout the five-shard grid; shard-local tuning also gives higher mean CA(1) than both separate options for every instance, without uniform accuracy dominance. These bounded results motivate reporting joint accuracy and certificate sensitivity across penalties and shard counts. They distinguish recovery, sparse stability, and local refitting without claiming a new guarantee, general routing benefit, or production deletion performance.

AI USE STATEMENT

Generative AI tools were used for source-code and literature triage, method ideation, drafting and checking mathematical arguments, implementation and debugging assistance, experiment design, implementing and executing experiment and analysis code, parsing and summarizing results, author-side methodological review and revision feedback, and manuscript editing and formatting. They were not used to generate the underlying CIFAR-10 or protected image arrays. The authors remain responsible for the final claims, citations, artifacts, and verification of AI-assisted work.

ETHICS STATEMENT

Recovery research is dual use: it can help an authorized curator learn from corrupted or accidentally protected data, but it can also weaken a person's attempt to prevent unauthorized training. We frame HERA as an audit of technical protections, not permission to ignore consent, copyright, licenses, privacy, or data-governance obligations. The artifact contains no scraping or deployment pipeline. Any public release will reference rather than redistribute protected arrays unless their licenses permit it and will document provenance and intended use.

REPRODUCIBILITY AND ARTIFACT STATEMENT

The manuscript describes both learner variants, deterministic seeds, input processing, metrics, and proofs. Appendix C identifies the retained inputs and distinguishes known preprocessing from unrecovered attack-generation settings; Appendix D defines the checked state. Retained NPZ predictions permit recomputation of most reported metrics. Historical clean-only accuracies and sparse timings were retained as CSV summaries. New verification runs save clean-only predictions, both compared sparse states, input/source hashes, and environment metadata; they do not establish historical execution identity. The compact supplement retains predictions and hashes of omitted fitted arrays; full fitted states remain stored locally. Protected arrays are not redistributed. These limitations preclude claiming turnkey reproduction of original protection generation.

REFERENCES

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 141–159, 2021. doi: 10.1109/SP40001.2021.00019. URL https://doi.org/10.1109/SP40001.2021.00019.

Liam Fowl, Micah Goldblum, Ping-yeh Chiang, Jonas Geiping, Wojciech Czaja, and Tom Goldstein. Adversarial examples make strong poisons. In *Advances in Neural Information Processing Systems*, volume 34, pp. 30339–30351, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/fe87435d12ef7642af67d9bc82a8b3cd-Abstract.html.

Shaopeng Fu, Fengxiang He, Yang Liu, Li Shen, and Dacheng Tao. Robust unlearnable examples: Protecting data privacy against adversarial learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=baUQQPwQiAg.

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*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3832–3842, 2020. URL https://proceedings.mlr.press/v119/guo20c.html.

Thushari Hapuarachchi and Kaiqi Xiong. Advancing ensemble learning against unlearnable data. *Neurocomputing*, 647:130422, 2025. doi: 10.1016/j.neucom.2025.130422. URL https://www.sciencedirect.com/science/article/pii/S092523122501094X.

Thushari Hapuarachchi, Jing Lin, Kaiqi Xiong, Mohamed Rahouti, and Gitte Ost. Nonlinear transformations against unlearnable datasets. *arXiv preprint arXiv:2406.02883*, 2026. URL https://arxiv.org/abs/2406.02883v2. Version 2, revised 22 May 2026; first posted in 2024.

Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J. Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *The Annals of Statistics*, 50(2):949–986, 2022. doi: 10.1214/21-AOS2133. URL https://doi.org/10.1214/21-AOS2133.

Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=iAmZUo0DxC0.

Jinyuan Jia, Xiaoyu Cao, and Neil Zhenqiang Gong. Intrinsic certified robustness of bagging against data poisoning attacks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(9): 7961–7969, 2021. doi: 10.1609/aaai.v35i9.16971. URL https://ojs.aaai.org/index.php/AAAI/article/view/16971.

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*, volume 267 of *Proceedings of Machine Learning Research*, pp. 31275–31298, 2025. URL https://proceedings.mlr.press/v267/koloskova25a.html.

Alexander Levine and Soheil Feizi. Deep partition aggregation: Provable defenses against general poisoning attacks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YUGG2tFuPM.

Yixin Liu, Kaidi Xu, Xun Chen, and Lichao Sun. Stable unlearnable example: Enhancing the robustness of unlearnable examples via stable error-minimizing noise. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(4):3783–3791, 2024. doi: 10.1609/aaai.v38i4.28169. URL https://ojs.aaai.org/index.php/AAAI/article/view/28169.

Zhuoran Liu, Zhengyu Zhao, and Martha Larson. Image shortcut squeezing: Countering perturbative availability poisons with compression. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22473–22487, 2023. URL https://proceedings.mlr.press/v202/liu23bb.html.

Preetum Nakkiran, Prayaag Venkat, Sham Kakade, and Tengyu Ma. Optimal regularization can mitigate double descent. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=7R7fAoUygoa.

Martin Pawelczyk, Jimmy Z. Di, Yiwei Lu, Gautam Kamath, Ayush Sekhari, and Seth Neel. Machine unlearning fails to remove data poisoning attacks. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HaX48yksVL.

Keivan Rezaei, Kiarash Banihashem, Atoosa Chegini, and Soheil Feizi. Run-off election: Improved provable defense against data poisoning attacks. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 29030–29050, 2023. URL https://proceedings.mlr.press/v202/rezaei23a.html.

Elan Rosenfeld, Ezra Winston, Pradeep Ravikumar, and J. Zico Kolter. Certified robustness to label-flipping attacks via randomized smoothing. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8230–8241, 2020. URL https://proceedings.mlr.press/v119/rosenfeld20b.html.

Vinu Sankar Sadasivan, Mahdi Soltanolkotabi, and Soheil Feizi. CUDA: Convolution-based unlearnable datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3862–3871, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Sadasivan_CUDA_Convolution-Based_Unlearnable_Datasets_CVPR_2023_paper.html.

Pedro Sandoval-Segura, Vasu Singla, Jonas Geiping, Micah Goldblum, and Tom Goldstein. What can we learn from unlearnable datasets? In *Advances in Neural Information Processing Systems*, volume 36, pp. 75372–75391, 2023. doi: 10.52202/075280-3294. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ee5bb72130c332c3d4bf8d231e617506-Abstract.html.

Philip Sosnin, Matthew Wicker, Josh Collyer, and Calvin Tsay. Abstract gradient training: A unified certification framework for data poisoning, unlearning, and differential privacy. *Journal of Machine Learning Research*, 27(128):1–58, 2026. URL https://www.jmlr.org/papers/v27/25-2206.html.

Derui Wang, Minhui Xue, Bo Li, Seyit Camtepe, and Liming Zhu. Provably unlearnable data examples. In *Network and Distributed System Security Symposium*, 2025. doi: 10.14722/ndss.2025.240886. URL https://www.ndss-symposium.org/ndss-paper/provably-unlearnable-data-examples/.

Wenxiao Wang and Soheil Feizi. On practical aspects of aggregation defenses against data poisoning attacks. *arXiv preprint arXiv:2306.16415*, 2023. URL https://arxiv.org/abs/2306.16415.

Wenxiao Wang, Alexander J Levine, and Soheil Feizi. Improved certified defenses against data poisoning with (deterministic) finite aggregation. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 22769–22783, 2022. URL https://proceedings.mlr.press/v162/wang22m.html.

Yi Yu, Yufei Wang, Song Xia, Wenhan Yang, Shijian Lu, Yap-Peng Tan, and Alex Kot. Purify unlearnable examples via rate-constrained variational autoencoders. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57678–57702, 2024. URL https://proceedings.mlr.press/v235/yu24m.html.

Chia-Hung Yuan and Shan-Hung Wu. Neural tangent generalization attacks. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12230–12240, 2021. URL https://proceedings.mlr.press/v139/yuan21b.html.

Mengying Zhang, Derui Wang, Ruoxi Sun, Xiaoyu Xia, Shuang Hao, and Minhui Xue. SoK: Unlearnability and unlearning for model dememorization. *arXiv preprint arXiv:2605.11592*, 2026. doi: 10.48550/arXiv.2605.11592. URL https://arxiv.org/abs/2605.11592.

Yuhao Zhang, Aws Albarghouthi, and Loris D'Antoni. BagFlip: A certified defense against data poisoning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 31474–31483, 2022. doi: 10.52202/068431-2282. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/cc19e4ffde5540ac3fcda240e6d975cb-Abstract-Conference.html.

Chenxu Zhao, Wei Qian, Chenglin Miao, and Mengdi Huai. Rethinking unlearnable examples in machine unlearning. In *Advances in Knowledge Discovery and Data Mining (PAKDD 2026)*, volume 16598 of *Lecture Notes in Computer Science*, pp. 40–53, 2026. doi: 10.1007/978-981-92-1462-4_4. URL https://link.springer.com/chapter/10.1007/978-981-92-1462-4_4.

Yifan Zhu, Lijia Yu, and Xiao-Shan Gao. Detection and defense of unlearnable examples. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(15):17211–17219, 2024. doi: 10.1609/aaai.v38i15.29667. URL https://ojs.aaai.org/index.php/AAAI/article/view/29667.

A Complete proofs

Proof of Theorem 1. With no operation, deterministic prediction is unchanged even for a tied vote; this establishes radius zero. For positive radius, fix x and let $d = v_A - v_B > 0$. Changing one shard voter changes any class count by at most one. The most damaging change removes a vote from A and awards it to one competitor, reducing its pairwise margin by two; changes involving other source or destination classes reduce the margin by at most one or zero. Therefore after changes to s voters, A has a margin of at least $d - 2s$ over every competitor. If $s \leq \lfloor (d-1)/2 \rfloor$, this

Table 4: Matched five-seed NTGA results (mean±SD, %). Original NTGA order supplies immutable IDs; canonical-ID results appear in Table 2. Bagging has five learners with sample sizes matched to the shard sizes. CA(q) is jointly correct and certified accuracy.

System	Accuracy	HERA minus system
Pooled-RGB ridge ERM	20.25±0.22	+5.31
SISA-style pooled-RGB vote	20.67±0.23	+4.89
Router-only coarse (2k clean)	**36.51±0.67**	−10.95
Oracle fixed-view envelope	25.13±0.21	+0.43
Full-data five-view vote	22.76±0.18	+2.80
Bootstrap coarse bagging	25.38±0.42	+0.18
HERA ($M = 5$)	**25.56±0.14**	n/a
CA($q = 1$)	20.91±0.22	n/a
CA($q = 2$)	16.20±0.41	n/a

quantity is positive and A is invariant. At equality $2s = d$, the runner-up can tie A, and the declared class-order tie rule determines whether the prediction changes. The reported radius is deliberately tie-agnostic; a tie-aware implementation may certify one additional operation for an even margin when A has priority over every competitor that can tie it. Stable hashing maps each insertion or deletion to one shard, as does a replacement that preserves its stable identifier. An identity-changing replacement is conservatively a deletion plus an insertion and may therefore affect two shards. Thus $s \le q$ whenever the q record operations affect at most q shards; collisions only make the certificate conservative.

Proof of Theorem 2. Let $D_j = \{z \in D : H(\mathrm{id}(z)) \bmod M = j\}$ and let deterministic shard procedure L_j include fitting every candidate view, scoring candidates on the fixed routing set, applying the view tie rule, and fitting the selected voter. For datasets in the stated domain, each L_j is defined. Then $A_{\mathrm{H}}(D) = (L_0(D_0), \ldots, L_{M-1}(D_{M-1}), a)$ for fixed aggregation state a. Stable identifiers give $(D \setminus U)_j = D_j \setminus U$. If $U \cap D_j = \varnothing$, local deletion retains $L_j(D_j) = L_j((D \setminus U)_j)$. Otherwise it reruns the entire shard procedure and obtains $L_j(D_j \setminus U) = L_j((D \setminus U)_j)$. Thus every selected view, parameter array, and aggregation component equals a fresh $A_{\mathrm{H}}(D \setminus U)$. This argument fails if the routing set changes or view selection is not replayed on touched shards.

Occupancy corollary. For uniformly hashed independent identifiers, let I_j indicate that deletion request U touches shard j. Since $\Pr(I_j = 0) = (1 - 1/M)^u$, linearity of expectation gives $\mathbb{E}[\sum_j I_j] = M[1 - (1 - 1/M)^u]$. For exactly t occupied shards, choose and order the t shards in $(M)_t$ ways and partition the u labeled requests into t nonempty blocks in $S(u, t)$ ways; division by the M^u assignments proves $\Pr(T_u = t) = (M)_t S(u, t)/M^u$. Finally, with $a = (1-1/M)^u$ and $b = (1-2/M)^u$, $\mathrm{Var}(I_j) = (1-a)a$ and $\mathrm{Cov}(I_j, I_k) = b - a^2$, giving the variance in Appendix E. These are training-work results, not wall-clock guarantees: preprocessing, parallelism, and model loading can dominate.

B ADDITIONAL RECOVERY AND UNCERTAINTY RESULTS

The original NTGA protocol uses immutable initial-order IDs, whereas the matched cross-protection protocol uses canonical CIFAR-10 IDs. Both retain the same clean routing/evaluation split for a given seed; their partitions differ. Table 5 preserves the original protocol's seed-level view results.

Table 5: Accuracy by seed and full-data feature view. The last column is the disjoint-shard vote.

Seed	RGB	Gray	Chroma	Edge	Coarse	View vote	HERA
2023	20.29	21.35	17.76	23.54	25.36	23.03	**25.75**
2024	19.96	21.48	17.30	23.30	25.26	22.70	**25.58**
2025	20.15	21.30	17.51	23.11	25.13	22.68	**25.50**
2026	20.30	21.41	17.50	23.35	25.11	22.56	**25.60**
2027	20.55	21.66	17.90	23.18	24.80	22.83	**25.36**

Table 6: Five-seed classwise mean accuracy and CA(1) on NTGA, in percent.

	Air	Auto	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
Acc.	37.87	20.59	13.92	17.82	20.91	23.48	20.09	19.34	44.61	36.04
CA(1)	33.61	15.34	9.63	13.53	16.97	19.31	14.98	14.65	39.71	30.39

Classwise certification. Ship has 39.71% CA(1), versus 9.63% for bird; the unrounded difference is 30.09 points. The lowest two reported classwise CA(1) values are bird and cat. Class prevalence alone does not explain these differences, and aggregate clean controls do not identify their cause.

B.1 Sensitivity to the five-run uncertainty summary

The main text retains the original paired percentile bootstrap summaries (20,000 resamples). The following alternative summaries use the same five paired differences, not new training runs. A paired t interval uses $\bar{d} \pm t_{4,0.975} s_d/\sqrt{5}$ and a normal-differences approximation. Two-sided sign-test values are 0.0625 for 5/5 positive differences and 0.375 for 4/5; with only five signs, even unanimity does not reject at 0.05. These are conditional run summaries, not tests over independent datasets. They do not establish independent voter errors or resolve the bagging comparison.

Table 7: Original NTGA paired differences in percentage points; alternative uncertainty summaries from retained runs.

HERA minus	Mean	Paired t 95% interval	Positive runs
Disjoint pooled-RGB	4.89	[4.46, 5.31]	5/5
Bootstrap coarse	0.18	[−0.20, 0.56]	4/5
Full-data coarse	0.43	[0.30, 0.55]	5/5
Full-data view vote	2.80	[2.57, 3.03]	5/5

Paired prediction check against bagging. A post-hoc exact two-sided McNemar check uses the original NTGA predictions on each seed's 8,000 evaluation records. For seeds 2023–2027, discordant counts (HERA-only correct, bagging-only correct) are $(173, 171)$, $(167, 185)$, $(200, 170)$, $(167, 155)$, and $(196, 150)$. The corresponding binomial-test p-values are $0.9570, 0.3649, 0.1315, 0.5399, 0.0154$. Only the last is below 0.05 before adjustment; none is after Holm correction across these five comparisons (minimum adjusted $p = 0.0772$). These conditional fitted-model comparisons rely on the usual paired-test assumptions; overlapping evaluation pools are not independent replications. They neither measure training variability nor establish equivalence. HERA's advantage over bagging remains unresolved.

B.2 Cross-protection controls and routing

Table 8: Matched cross-protection results (five-seed means, %).

	HERA	CA(1)	CA(2)	SISA	Bag.	RGB	Coarse
NTGA	25.60	20.72	16.36	20.73	25.56	20.25	25.13
EMIN	33.45	25.66	19.68	29.35	33.46	28.86	33.49
REM	37.60	30.10	22.30	37.75	36.95	37.50	37.27
CUDA	36.96	31.05	25.34	32.92	36.59	32.69	37.26

All canonical NTGA and CUDA shards select coarse. EMIN selects all-coarse views in seeds 2023, 2025, and 2027, all-edge in 2024, and (coarse, gray, coarse, edge, edge) in 2026; CA(1) is respectively 28.7625, 28.9500, 29.0125, 23.1875, and 18.3750%. REM is all-coarse in 2023 (CA(1) 32.25%) and mixes RGB/coarse in the other seeds (28.35–31.125%). This original between-seed pattern confounds routing with changed shards and evaluation subsets. The new within-seed fixed-coarse comparisons in Appendix B.7 isolate the routing-policy change on these retained instances.

Paired certificate-loss decomposition. For each seed, let C_r, C_c indicate correct routed/coarse predictions and B_r, B_c indicate vote margin greater than two. The CA(1) difference decomposes exactly as

$$\mathbb{E}[C_r B_r - C_c B_c] = \mathbb{E}[C_r C_c (B_r - B_c)] + \mathbb{E}[C_r (1 - C_c) B_r] - \mathbb{E}[C_c (1 - C_r) B_c].$$

The three terms separate margin changes on jointly correct examples, newly correct certified predictions, and lost correct certified predictions. Five-seed means in percentage points are $(-1.6875, +1.7725, -3.3475)$ for EMIN and $(-1.5800, +0.6800, -0.9550)$ for REM; they sum to -3.2625 and -1.8550. Thus certificate loss is not explained by mean accuracy alone. This exact descriptive decomposition does not identify why routing selected a view or establish population significance. Per-seed components are saved with the evidence.

B.3 Frozen-feature results

Table 9: Frozen ResNet-18 representation validation over five seeds (mean±SD, %). The encoder is fixed and not selected on the routing split. Because there is one fixed representation, HERA and the SISA-style disjoint ensemble have identical recovery predictions in this experiment.

Training data	Acc.	CA(1)	CA(2)	Bagging	Full
Clean	84.52±.10	80.73±.16	75.38±.28	84.29	84.72
Router-only clean (2k)	79.15±.38	–	–	–	–
NTGA	81.96±.11	77.61±.26	72.22±.30	81.83	82.10
EMIN	76.11±.30	71.50±.40	65.82±.32	75.72	75.96
REM	78.12±.12	73.13±.12	67.20±.33	77.55	78.20
CUDA	67.73±.07	62.84±.27	57.12±.44	67.78	67.74

Clean-to-protected losses, calculated from unrounded means within the frozen variant, are 2.57, 8.42, 6.40, and 16.79 points for NTGA, EMIN, REM, and CUDA. The minimum protected accuracy over these four instances is 67.73%. All 25 runs report zero maximum parameter difference and exact comparison-state/prediction replay for 100-record deletions touching every shard. The corresponding routed clean accuracy is 41.58%; the cross-variant increase also changes head and hash conventions, not just features.

B.4 Clean-only alternatives and paired comparisons

The retained clean-only CSV has five seeded fits per fixed representation. Pooled-RGB, coarse, and frozen heads reach 36.11 ± 0.55, 36.51 ± 0.67, and $79.15 \pm 0.38\%$, respectively. Each trains on precisely the seed's 2,000 routing records and evaluates on its other 8,000 records. No protected records enter these fits. The source uses the corresponding routed or frozen ridge head with $\lambda = 10$ and no view search. These are available-resource alternatives, not matched-sharding ablations. Appendix B.8 separately tests pooled protected-plus-clean training. The original runs saved only scalar accuracies. New verification fits reproduce all 15 scalars exactly and save predictions and fitted states (Appendix B.7); this is new evidence, not recovery of historical states.

B.5 Extended routed certificate sweep

The additional $M = 15, 25, 50$ runs use the original-order NTGA protocol and the same five seeds. The table below is recomputed from retained predictions and reports selected radii; the accompanying CSV contains every integer radius from zero to $\lfloor (M-1)/2 \rfloor$. Conditional medians are descriptive per-seed statistics and do not imply correctness on the whole evaluation pool. This table contains routed results; Appendix B.7 reports the separate frozen sweep. The 15 manifests also report single-record deletion with exactly one affected shard, zero maximum coefficient difference, and exact state/prediction equality. These original-order NTGA runs use the original routed runner, whose equality checks include ordered membership, selected views, normalization, coefficients, and predictions. Untouched-object identity is explicitly checked only by the separate sparse harness below; no post-deletion states were serialized.

Table 10: Protected ensemble minus clean-only accuracy (percentage points), paired by seed. Routed ensembles compare with a coarse clean-only single head; frozen ensembles with a frozen clean-only single head. Intervals are descriptive paired t intervals over five fixed-data splits, not independent-dataset inference.

Variant	Protection	Mean	95% interval	Positive seeds
Routed	NTGA	-10.91	[-11.58, -10.25]	0/5
Routed	EMIN	-3.06	[-4.37, -1.75]	0/5
Routed	REM	1.09	[0.02, 2.16]	5/5
Routed	CUDA	0.45	[-0.44, 1.34]	3/5
Frozen	NTGA	2.81	[2.40, 3.22]	5/5
Frozen	EMIN	-3.04	[-3.78, -2.30]	0/5
Frozen	REM	-1.03	[-1.50, -0.55]	0/5
Frozen	CUDA	-11.41	[-11.94, -10.89]	0/5

Table 11: Mean correctly certified accuracy (%) over five runs, at larger record radii. A dash denotes a radius above the theoretical maximum.

M	CA(5)	CA(10)	CA(15)	CA(20)	CA(24)
15	12.59	–	–	–	–
25	15.61	6.95	–	–	–
50	18.59	12.39	7.25	3.22	0.58

B.6 Sparse replay configuration and timings

Table 12 reports the sparse harness, which uses canonical NTGA records, seed 2027, five shards, and the routed keyed integer-ID hash for both heads. It therefore audits a frozen-head configuration with different partitions from the original frozen runner. Requests are nested prefixes of one seeded random permutation; they are not independent replicates. It tests object reuse for untouched shards and exact elementwise equality of selected-view names and μ, σ, W arrays with fresh fits. Ordered membership and output predictions are not independently compared, and post-deletion arrays are not serialized. Equal arrays and selected views imply equal deterministic predictions under the fixed feature maps, but that implication is distinct from a measured prediction comparison.

Table 12: Retained sparse checks on NTGA. Every row reports untouched-object reuse and exact fitted-array equality. Times are seconds for refitting from cached features; each is one measurement, with local fitting timed before fresh fitting.

Head	Deleted	Touched	Local (s)	Fresh (s)	Ratio
Routed	1	1	0.0278	0.1445	5.20
Routed	2	1	0.0248	0.1264	5.09
Routed	3	2	0.0499	0.1361	2.73
Frozen	1	1	0.0172	0.0863	5.02
Frozen	2	1	0.0170	0.0873	5.13
Frozen	3	2	0.0338	0.0918	2.72

The timing boundary excludes feature extraction, data loading, persistence, and end-to-end request handling. Hardware/thread metadata for these historical checks were not retained. Appendix B.10 adds repeated measurements with recorded environment, still short of end-to-end systems performance. The $u = 1$ ratios range from 5.02 to 5.20.

B.7 Matched controls and saved-state deletion checks

These additional runs use the same five seeds, $\lambda = 10$, fixed feature maps, and routing/evaluation splits; no hyperparameter search is performed. They are new verification evidence, not recovered historical execution records. Fifteen clean-only fits reproduce every original scalar accuracy exactly. Five frozen NTGA $M = 5$ verification ensembles reproduce the retained ensemble predictions

exactly. Matched fixed-coarse EMIN/REM controls use the original routed hash, ordered shard inputs, ridge head, and evaluation indices. Table 13 reports their five-seed means alongside retained routed outputs; differences are descriptive, not population-level significance claims.

Table 13: Matched-control and frozen-sweep means (%). CA(10) is structurally zero for $M < 21$. Matched routing controls hold partitions and evaluation records fixed within each seed. Frozen NTGA uses its original hash and head, so its sweep is not a pure representation ablation of the routed sweep.

Instance	Learner	Acc.	CA(1)	CA(2)	CA(10)
EMIN	Fixed coarse, $M = 5$	33.52	28.92	23.93	0.00
EMIN	Routed, $M = 5$	33.45	25.66	19.68	0.00
REM	Fixed coarse, $M = 5$	37.24	31.96	26.20	0.00
REM	Routed, $M = 5$	37.60	30.10	22.30	0.00
NTGA	Frozen, $M = 5$	81.96	77.61	72.22	0.00
NTGA	Frozen, $M = 15$	82.03	78.99	76.25	0.00
NTGA	Frozen, $M = 25$	81.85	79.20	76.97	50.80
NTGA	Frozen, $M = 50$	81.95	79.48	77.51	60.36

Thirty additional sparse replay checks cover both heads, all five seeds, and requests of one, two, and three distinct records. Within each seed the requests are disjoint segments of a permutation initialized with seed plus 100,000; the same requests are used for both heads. Requests are not independent replications across overlapping data. Each head uses its original hash convention with five shards. Between one and three shards are touched; untouched model objects are reused. Every check passes exact ordered-membership, selected-view, normalization/coefficient-array, and full held-out prediction comparisons. Both local and fresh states, memberships, predictions, split indices, and deletion IDs are retained, allowing independent comparison. No new timing claim is made.

The verification artifact records input and source SHA-256 hashes, Python/NumPy versions, platform, and thread settings. It contains 75 prediction/state archives: 15 clean-only, 10 matched coarse, 20 frozen-sweep, and 30 sparse checks. These records bind the new runs to their retained configuration; they do not establish the original protection-generation settings or historical tuning chronology.

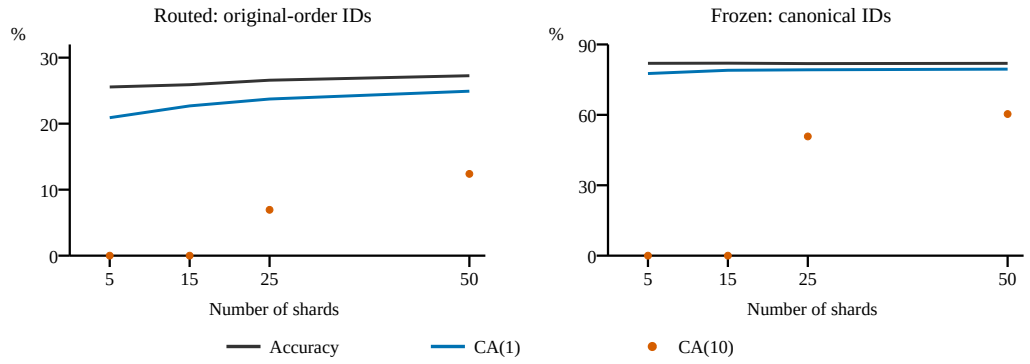


Figure 3: NTGA accuracy and certified accuracy over shard counts (five split/hash-draw means). CA(10) uses markers only and is structurally zero for $M < 21$. Both panels use percentages of all evaluation records; vertical scales differ. Routed results use original-order IDs, routed heads and keyed hashing; frozen results use canonical IDs, a different head and the original frozen hash. These are separate within-protocol frontiers, not a pure representation ablation. Lines connect tested values and do not imply interpolation guarantees.

B.8 Matched data addition and clean-label budgets

A protocol recorded before execution fixes seeds 2023–2027, clean budgets $b \in \{250, 500, 1000, 2000\}$, coarse and frozen features, and all four canonical protection instances. For each seed, clean subsets are nested prefixes of its original 2,000-record routing permutation. The

other 8,000 records remain the identical evaluation set at every budget; unused routing records are excluded from both fitting and evaluation. We fit one pooled clean-only head and four matched heads that append all 40,000 protected records to that same clean subset. Each record has equal weight. No views or hyperparameters are selected on evaluation records.

Within a representation, both conditions use the same ridge-head definition and $\lambda = 10$. Normalization is refit on each training set. The objective is the existing sum-loss ridge objective, so fixed λ does not hold mean-loss regularization strength constant as sample size changes. Consequently this intervention measures adding records under the specified fitting algorithm, including its normalization and regularization scaling; it does not isolate a representation-independent value of data, optimal weighting, or an additive effect for the partition ensemble.

Table 14: Matched pooled-head budget sweep: five-seed mean evaluation accuracy (%). "+" columns append 40,000 protected records to the identical clean subset; all columns share evaluation indices within seed. These are separate heads from the sharded recovery comparisons.

Features	Clean labels	Clean only	+NTGA	+EMIN	+REM	+CUDA
Coarse	250	28.26	25.29	33.49	37.27	37.24
Coarse	500	31.62	25.39	33.59	37.26	37.30
Coarse	1000	34.40	25.64	33.71	37.32	37.37
Coarse	2000	36.51	26.00	33.99	37.40	37.42
Frozen	250	61.88	82.12	76.68	79.09	69.52
Frozen	500	62.87	82.15	77.14	79.60	71.04
Frozen	1000	72.18	82.27	78.02	80.32	73.02
Frozen	2000	79.15	82.44	79.20	81.06	75.86

Coarse clean-only accuracy rises from 28.26% to 36.51% over the tested budgets; frozen accuracy rises from 61.88% to 79.15%. At 250 clean labels every frozen addition improves mean accuracy, whereas at 2,000 the CUDA addition is harmful and EMIN is nearly neutral. The coarse EMIN addition changes from helpful at 250/500 labels to harmful at 1,000/2,000. These are descriptive observations on overlapping evaluation sets, not independent population tests or a continuous-budget crossover estimate. The protocol, all 200 fitted states/prediction archives, exact subset indices, input/source hashes and environment records are retained. The 2,000-label clean-only anchors reproduce the previously verified predictions exactly. No new certification or deletion claim is made for the pooled heads.

B.9 Nonlinear frozen-feature learner check

A separate pooled MLP tests whether the clean-resource observations are confined to ridge heads. It uses the same fixed 512-dimensional features and five evaluation splits, with one hidden layer of 128 ReLU units and a ten-class softmax output. StandardScaler is fitted only on each training set. The fixed protocol uses scikit-learn 1.9.1 and Adam, learning rate 0.001, batch size 128, weight penalty 0.0001, 100 epochs, seeded shuffling, and no early stopping or evaluation-based selection. Clean budgets are 250 and 2,000 nested labels; protected-only and protected-plus-clean conditions use each archived instance. Equal per-record weighting and fixed epochs imply more optimization steps with more training records. This is another specified learning procedure, not an isolation of nonlinearity: loss, biases, normalization, regularization and optimization also differ from ridge. The encoder is not fine-tuned.

All 70 fits use the last iterate at the fixed 100-epoch budget. The optimizer reports its maximum-iteration warning; no convergence or optimality is claimed. We retain the full loss curves, fitted weights and biases, normalization, evaluation indices and predictions, and independently reconstruct all predictions from the saved arrays. These runs extend the tested learner class but not the dataset or protection-instance scope. They are not end-to-end UE-defense benchmarks, and overlapping evaluation pools do not constitute independent population replications.

B.10 Repeated cached-feature deletion timings

The repeated study runs after nonlinear training completes, without a concurrent training job from this study; unrelated system activity is uncontrolled. It uses both original hash conventions, five

Table 15: Nonlinear frozen-feature check: mean±SD accuracy (%) over five seeds. Protected-only uses 40,000 protected records; addition columns append those records to the indicated clean subset. The clean-only row uses no protected records. No new certificate or deletion claim is made for these MLPs.

Instance	Protected only	+250 clean	+2,000 clean
Clean only	–	69.70±0.92	81.08±0.24
NTGA	81.21±0.13	81.32±0.52	82.54±0.16
EMIN	72.08±0.35	75.46±1.55	80.69±0.25
REM	76.88±0.36	78.72±0.42	81.57±0.44
CUDA	62.39±0.52	69.79±0.69	78.99±0.60

seeds, and disjoint one-/three-record request segments from the verification protocol. Each condition receives an untimed local/fresh warm-up and six timed pairs, with local-first and fresh-first orders alternating equally. Both paths compare equal in selected views and fitted arrays on all 120 pairs. The timer measures refitting from cached features, including view search for routed heads; loading, feature extraction, prediction comparisons, serialization and request handling are excluded. These repeated timings supplement, rather than replace, the saved-state correctness checks.

Table 16: Repeated cached-feature refitting: medians over 30 pairs per row (five seeds, six repetitions). Ratio is the median of paired fresh/local ratios, not the quotient of marginal medians. Last column ranges over five within-seed median ratios.

Head	Deleted	Local (ms)	Fresh (ms)	Ratio	Seed range
Routed	1	26.02	130.47	4.98	4.91–5.12
Routed	3	78.44	134.33	1.70	1.61–2.52
Frozen	1	17.05	85.61	5.02	4.96–5.09
Frozen	3	36.76	86.49	2.40	1.64–2.50

Environment and native-library/thread metadata, per-repetition times, execution order, request occupancy and equality results are retained. Measurements are local single-thread refit timings, not independent requests in a production system or an end-to-end speed guarantee. A preliminary timing run overlapped with training and is excluded from this table.

B.11 Clean-only partition ensembles and regularization sensitivity

A protocol fixed before execution uses the same five seeds, nested clean budgets 250/500/1,000/2,000, and original 8,000-record evaluation split. Clean-only ensembles use five stable shards of the selected clean records, their canonical IDs, and each representation's original head/hash convention with $\lambda = 10$. Coarse views are fixed, not selected on evaluation labels; frozen features remain unchanged. Unused routing records are excluded. Table 17 reports these controls, which match the partition guarantee mechanism, not the number or cleanliness of training records. The certified operations affect these ensembles' clean training records, whereas protected-ensemble certificates concern their protected training records.

Table 17: Clean-only five-shard ensembles: mean±SD (%) over five seeds. Shard-normalized heads use the same record-operation certificate.

Features	Clean labels	Accuracy	CA(1)	CA(2)
Coarse	250	24.92±0.95	2.65±1.61	0.41±0.54
Coarse	500	28.84±0.90	5.34±0.92	1.04±0.41
Coarse	1000	32.32±0.67	9.52±0.89	2.68±0.59
Coarse	2000	34.89±0.34	14.03±0.73	4.93±1.08
Frozen	250	67.45±0.52	35.83±2.07	14.11±2.93
Frozen	500	72.24±0.55	44.67±1.11	23.71±1.71
Frozen	1000	74.35±0.49	47.36±0.45	26.67±1.00
Frozen	2000	74.34±0.62	46.67±0.46	26.49±0.40

With $\lambda = 10$ and 2,000 labels, clean-only coarse/frozen CA(1) is 14.03/46.67%, below every corresponding protected ensemble. The frozen clean-only accuracy exceeds protected CUDA, but its CA(1) does not. Thus accuracy-only comparisons do not settle the joint resource choice. These fixed-M controls do not establish an optimal accuracy/certification frontier.

Each of the 40 clean-only ensembles deletes one seeded training record, reuses four untouched states, and exactly matches fresh fitting in membership, fitted arrays and evaluation predictions. All original and both post-deletion states are retained. Cached-feature refit timing uses one warm-up per path and three alternating-order pairs per condition (two local-first, one fresh-first). For the 2,000-label coarse/frozen conditions, median local times are 0.107/2.094 ms and fresh times 0.545/10.467 ms; paired median ratios are 5.02/5.00. These small, single-thread timings exclude loading, extraction, prediction checks and persistence; they are not an end-to-end or cross-study timing comparison.

The regularization sensitivity fits 200 separate pooled heads with $\alpha = 10/2000 = 0.005$ in the mean-loss objective, implemented as $\lambda = \alpha n$ in the sum-loss solver. Table 18 reports the results; this anchor matches the original 2,000-label clean-only head. All four budgets, both representations and all protection instances are included without tuning. Each condition refits training-only normalization and uses equal record weighting, so the comparison controls penalty scaling but not every possible fitting choice.

Table 18: Constant mean-loss ridge sensitivity: five-seed mean accuracy (%). Additions append 40,000 protected records to the same clean subset.

Features	Clean labels	Clean only	+NTGA	+EMIN	+REM	+CUDA
Coarse	250	27.43	26.36	34.39	37.88	37.08
Coarse	500	31.45	26.44	34.41	37.85	37.09
Coarse	1000	34.38	26.62	34.57	37.89	37.17
Coarse	2000	36.51	27.13	34.79	37.87	37.20
Frozen	250	58.38	82.14	76.72	79.15	69.58
Frozen	500	51.98	82.19	77.22	79.70	71.08
Frozen	1000	70.82	82.29	78.06	80.42	73.12
Frozen	2000	79.15	82.49	79.26	81.13	75.90

The frozen addition signs at 250 and 2,000 clean labels agree with the fixed-sum-penalty study. At 2,000 labels the deltas are $+3.35, +0.11, +1.98, -3.25$ points for NTGA/EMIN/REM/CUDA. The EMIN gain remains small; signs of means alone do not establish significance. Full states, predictions, source/input hashes and seed summaries are retained. This sensitivity does not optimize regularization or protected-example weighting, and no pooled-head certificate is inferred.

B.12 Matched shard counts and paired resource differences

A fixed protocol compares clean-only and each protected frozen-feature ensemble at $M = 3, 5, 10$, using five seeds, 2,000 clean training labels or 40,000 protected records, and the same evaluation indices within each seed. All conditions use the original frozen head/hash and $\lambda = 10$ per shard. Every five-shard member prediction reproduces its previous anchor. All 75 ensembles retain fitted states, predictions, indices and input/source hashes. This is a matched count sensitivity, not a new independently generated protection, optimized frontier, or pure voter-count effect: changing shard size also changes each head's data and effective mean-loss penalty.

Table 19 shows higher protected-ensemble CA(1) at $M = 3, 5$. At $M = 10$, clean-only reaches 78.31% accuracy and 63.78% CA(1), while CUDA reaches 67.96% and 63.27%. Thus the five-shard certification ordering does not persist for every protection. NTGA/EMIN/REM retain higher mean CA(1) at ten shards; EMIN has lower accuracy than clean-only, and REM's accuracy difference is nearly zero. These different rankings motivate reporting both metrics rather than claiming universal protected-data benefit.

For each protection and shard count, Table 20 pairs protected-minus-clean differences by seed. Intervals use $\bar{d} \pm t_{4,0.975} s_d/\sqrt{5}$ and an approximate normal-difference assumption. They summarize five split/hash draws conditional on the retained data, not independent protection instances or pop-

Table 19: Matched frozen-feature shard counts: mean±SD (%) over five seeds. Clean-only uses 2,000 labels; protected conditions use 40,000 records. CA(2) is identically zero at $M = 3$ under the conservative certificate and is retained in the artifact.

M	Training records	Accuracy	CA(1)
3	Clean only	72.75±0.32	43.08±0.17
3	NTGA	81.93±0.16	76.85±0.29
3	EMIN	75.94±0.36	70.51±0.44
3	REM	78.02±0.24	72.10±0.19
3	CUDA	67.69±0.24	61.95±0.26
5	Clean only	74.34±0.62	46.67±0.46
5	NTGA	81.96±0.11	77.61±0.26
5	EMIN	76.11±0.30	71.50±0.40
5	REM	78.12±0.12	73.13±0.12
5	CUDA	67.73±0.07	62.84±0.27
10	Clean only	78.31±0.36	63.78±0.42
10	NTGA	82.00±0.32	77.89±0.34
10	EMIN	76.21±0.33	71.64±0.33
10	REM	78.30±0.14	73.38±0.24
10	CUDA	67.96±0.17	63.27±0.15

ulation generalization. These are unadjusted descriptive intervals, not simultaneous coverage or an equivalence test. Full seed differences and all three shard counts are retained.

Table 20: Protected frozen ensemble minus matched clean-only ensemble: mean difference and descriptive paired 95% t interval, in percentage points. The clean baseline here is sharded, unlike the single-head baseline in Table 10.

M	Protection	Accuracy difference [interval]	CA(1) difference [interval]
5	NTGA	7.62 [6.89, 8.34]	30.94 [30.14, 31.73]
5	EMIN	1.77 [0.95, 2.58]	24.83 [23.81, 25.85]
5	REM	3.78 [3.12, 4.44]	26.45 [25.80, 27.10]
5	CUDA	-6.61 [-7.37, -5.84]	16.17 [15.29, 17.04]
10	NTGA	3.69 [3.04, 4.34]	14.11 [13.31, 14.91]
10	EMIN	-2.10 [-2.88, -1.33]	7.86 [7.10, 8.63]
10	REM	-0.02 [-0.55, 0.52]	9.60 [8.86, 10.33]
10	CUDA	-10.35 [-11.01, -9.69]	-0.51 [-1.11, 0.09]

At five shards, CUDA has a 16.17-point CA(1) advantage with interval [15.29,17.04], despite a 6.61-point accuracy deficit. At ten shards its CA(1) difference is -0.51 points with interval $[-1.11, 0.09]$ and one positive seed out of five. The latter does not establish equivalence or a reliable sign. No additional deletion-timing result is inferred from this shard sweep.

B.13 Locating the shard-count change and controlling penalty scaling

Table 19 reports the original $\lambda = 10$ matched sweep. For the same evaluation example, let c_M indicate correctness and m_M denote the top-two vote margin. Then $b_M = \mathbf{1}[m_M > 2]$ indicates radius-one certification. The exact identity

$$c_{10}b_{10} - c_5b_5 = c_5c_{10}(b_{10} - b_5) + (1 - c_5)c_{10}b_{10} - c_5(1 - c_{10})b_5$$

separates margin changes on jointly correct predictions, certified gains on newly correct predictions, and certified losses on predictions that cease to be correct. Table 21 averages this identity over examples, then seeds. It locates changes without identifying their causes. Of clean-only's 17.11-point CA(1) increase, 13.96 points (81.6%) arise on jointly correct predictions. The absolute edit budget remains one record; this is not a constant fraction of voters, a larger per-voter robustness guarantee, or an isolation of ensemble size from shard training size.

The paired change in the protected-minus-clean CA(1) gap is $g_s = (\mathrm{CA}_{P,10,s} - \mathrm{CA}_{C,10,s}) - (\mathrm{CA}_{P,5,s} - \mathrm{CA}_{C,5,s})$. Applying the descriptive interval convention in Appendix B.12 to these five

Table 21: Exact $M = 5 \to 10$ CA(1) decomposition, in percentage points, with $\lambda = 10$. Before rounding, the final three columns sum to the CA(1) change. Lost-correct contributions are signed.

Condition	Accuracy change	CA(1) change	Jointly correct	Newly correct	Lost correct
Clean only	3.97	17.11	13.96	3.58	-0.43
NTGA	0.05	0.29	0.30	0.20	-0.21
EMIN	0.11	0.14	0.12	0.20	-0.18
REM	0.18	0.25	0.29	0.19	-0.23
CUDA	0.23	0.43	0.42	0.23	-0.22

seed differences gives NTGA -16.82 $[-17.62, -16.02]$, EMIN -16.97 $[-17.91, -16.02]$, REM -16.86 $[-17.60, -16.11]$, and CUDA -16.68 $[-17.57, -15.79]$ points. A negative gap change does not establish a negative final gap: CUDA's original ten-shard interval still includes zero.

Shared mean-loss penalty control. To check whether gap shrinkage requires changing effective regularization, a protocol fixed before these fits uses $\alpha \in \{0.00125, 0.005, 0.025\}$ in $n_j^{-1}\|Z_jW - Y_j\|_F^2 + \alpha\|W\|_F^2$, with sum-loss $\lambda_j = \alpha n_j$ for actual shard occupancy n_j. These values are nominal anchors $10/8000$, $10/2000$ and $10/400$, corresponding to the original five-shard protected occupancy, pooled clean budget, and five-shard clean occupancy. The same α is used across clean/protected conditions and both shard counts; no value is selected using evaluation results. All 150 ensembles use the same fixed features, original frozen hash, shard-local normalization, five split/hash seeds, 2,000 clean or 40,000 protected training records, and paired evaluation indices. No head has an intercept. All settings are reported in Table 22; states and predictions are retained. Prediction reconstruction, training-only normalization and ridge normal equations pass for all 1,125 heads.

Table 22: Shared mean-loss penalty sensitivity: frozen ensembles, mean±SD (%) over five seeds. Every prespecified setting is reported. Matching α controls penalty scaling but not training-set size or normalization.

		$M = 5$		$M = 10$	
α	Condition	Accuracy	CA(1)	Accuracy	CA(1)
0.00125	Clean only	63.45±1.05	25.35±1.04	77.54±0.50	61.44±0.52
0.00125	NTGA	81.96±0.11	77.61±0.27	81.98±0.32	77.85±0.34
0.00125	EMIN	76.11±0.30	71.50±0.39	76.20±0.29	71.57±0.27
0.00125	REM	78.12±0.12	73.13±0.12	78.29±0.15	73.32±0.23
0.00125	CUDA	67.73±0.07	62.84±0.27	67.92±0.17	63.22±0.20
0.005	Clean only	67.70±0.88	32.72±0.67	77.62±0.50	61.68±0.55
0.005	NTGA	81.99±0.09	77.74±0.25	82.05±0.28	77.98±0.35
0.005	EMIN	76.20±0.26	71.69±0.32	76.26±0.29	71.74±0.26
0.005	REM	78.14±0.11	73.31±0.12	78.34±0.11	73.54±0.23
0.005	CUDA	67.85±0.10	63.02±0.25	68.00±0.14	63.35±0.20
0.025	Clean only	74.39±0.60	46.72±0.46	78.01±0.48	62.77±0.56
0.025	NTGA	82.05±0.12	78.25±0.19	82.11±0.24	78.48±0.30
0.025	EMIN	76.38±0.29	72.21±0.28	76.51±0.22	72.36±0.35
0.025	REM	78.35±0.19	73.99±0.18	78.47±0.16	74.19±0.13
0.025	CUDA	68.02±0.16	63.53±0.34	68.28±0.18	63.88±0.18

CUDA's ten-shard CA(1) differences are respectively 1.78 [0.96,2.61], 1.67 [0.82,2.53], and 1.11 [0.31,1.91] points under these three penalties; all are positive, unlike the unresolved original $\lambda = 10$ comparison. Each nevertheless falls sharply from its five-shard difference (Table 23). All other protected conditions also retain positive ten-shard differences and shrinking gaps. Thus the shrinkage is not explained solely by mean-loss penalty scaling, whereas the original near-zero final CUDA gap is sensitive to it. This control does not isolate why smaller clean shards improve voting: shard sample sizes, fitted normalization, prediction dependence and the fraction of voters allowed to change remain different. No double-descent or other causal mechanism is established. These are conditional five-seed summaries without multiplicity adjustment or new protection-generation replicates.

Table 23: Change in protected-minus-clean CA(1) gap from $M = 5$ to 10 under shared mean-loss penalties: mean [descriptive paired 95% t interval], in percentage points. A negative change means a smaller protected advantage, not necessarily a negative final gap.

Protection	$\alpha = 0.00125$	$\alpha = 0.005$	$\alpha = 0.025$
NTGA	-35.85 [-37.25,-34.44]	-28.73 [-29.51,-27.94]	-15.82 [-16.55,-15.08]
EMIN	-36.03 [-37.17,-34.89]	-28.91 [-29.60,-28.23]	-15.90 [-16.83,-14.96]
REM	-35.90 [-37.15,-34.65]	-28.74 [-29.29,-28.18]	-15.84 [-16.57,-15.11]
CUDA	-35.71 [-36.79,-34.63]	-28.63 [-29.24,-28.02]	-15.70 [-16.49,-14.91]

Possible source of penalty sensitivity. The clean-only pooled means at 250, 500 and 1,000 labels are 58.38%, 51.98% and 70.82% under the shared mean-loss penalty (Table 18). Together with the small clean shards, this pattern is consistent with near-interpolation sensitivity studied in ridge regression (Nakkiran et al., 2021; Hastie et al., 2022). It does not identify double descent as the cause: feature covariance, class balance, the omitted intercept and aggregation also vary in their effects. The training-only PCA/intercept controls in Appendix G support spectral sensitivity but do not identify double descent as the mechanism. The tuned resource comparison, rather than the common untuned penalty, is therefore the primary allocation evidence.

B.14 Stronger five-shard regularization sensitivity

A follow-up extension fixes $M = 5$ and $\alpha \in \{0.1, 0.5, 2.5\}$ before fitting, preserving all other shared-penalty settings. These imply nominal clean-shard sum-loss penalties 40, 200 and 1,000; actual $\lambda_j = \alpha n_j$ follows occupancy. Every condition and seed is retained, with no penalty chosen by evaluation performance. This grid tests the earlier ordering, not an optimally tuned clean baseline. All 75 ensembles retain complete fitted states and predictions. Independent reconstruction of normalization, predictions, metrics and ridge normal equations passes for all 375 heads; the maximum normalized normal-equation residual is 2.68×10^{-15}.

Table 24: Stronger shared mean-loss penalties at five shards: mean±SD (%) over five seeds, with 2,000 clean or 40,000 protected training records. All settings are reported.

α	Training records	Accuracy	CA(1)
0.1	Clean only	78.39±0.49	58.28±0.31
0.1	NTGA	82.17±0.08	78.99±0.23
0.1	EMIN	76.70±0.27	73.28±0.26
0.1	REM	78.67±0.25	75.01±0.14
0.1	CUDA	68.58±0.22	64.48±0.32
0.5	Clean only	80.41±0.26	67.09±0.34
0.5	NTGA	81.88±0.10	79.31±0.15
0.5	EMIN	76.87±0.18	74.42±0.19
0.5	REM	78.47±0.16	75.67±0.09
0.5	CUDA	68.63±0.30	65.68±0.40
2.5	Clean only	79.72±0.26	70.21±0.53
2.5	NTGA	80.12±0.12	78.45±0.12
2.5	EMIN	75.85±0.18	73.99±0.12
2.5	REM	75.21±0.16	73.30±0.22
2.5	CUDA	66.87±0.45	64.92±0.46

At $\alpha = 0.1$, CUDA-minus-clean CA(1) is 6.20 points with descriptive paired 95% t interval [5.60,6.79]. At 0.5 and 2.5, the differences reverse to -1.41 $[-1.73, -1.10]$ and -5.29 $[-5.67, -4.90]$. Both reversals occur in every seed. These unadjusted intervals are conditional on the fixed arrays and selected grid, not population or simultaneous inference. The original five-shard ordering is therefore specific to its penalty regime; the extension does not demonstrate that the earlier behavior was caused by near-interpolation, nor establish a best allocation or broadly tuned frontier.

Where the penalty gain occurs. Apply the identity in Appendix B.13 with subscripts denoting $\alpha = 0.025$ and 2.5, holding $M = 5$ fixed. Evaluation records, labels, training IDs and shard assignments match across penalties for every seed. The identity holds exactly in integer counts for all 25 condition/seed pairs. For clean-only, the mean percentage-point changes are exactly $23.4875 = 17.7225 + 6.3175 - 0.5525$ before rounding. Table 25 reports every condition. The first component is the net certification change among predictions correct at both settings, not a causal explanation for improved margins. Clean-only's highest tested accuracy is 80.41% at $\alpha = 0.5$, whereas its highest tested CA(1) occurs at 2.5 and declines at both larger values (Table 26); no optimal penalty is established.

Table 25: Descriptive CA(1) decomposition for $\alpha = 0.025 \rightarrow 2.5$ at five shards: five-seed mean percentage-point changes. The final three columns sum to the total before rounding; lost-correct contributions are signed.

	Total	Jointly correct	Newly correct	Lost correct
Clean-only	23.49	17.72	6.32	-0.55
NTGA	0.21	0.88	2.43	-3.10
EMIN	1.78	1.39	3.47	-3.08
REM	-0.69	0.86	2.53	-4.08
CUDA	1.39	1.26	2.90	-2.77

Extension beyond the previous grid endpoint. After the six-point sensitivity was inspected, a separate extension fixed $\alpha \in \{12.5, 62.5\}$ at $M = 5$ before either new value was fitted. It uses all five conditions and seeds with unchanged records, encoder, head, hashing and normalization. This is a sequential sensitivity study, not a grid prespecified before all results were observed. Every new setting is reported in Table 26; the evaluation set selects no deployed model. The 50 ensembles retain full states and predictions. Independent normalization, prediction and ridge-equation checks pass for all 250 heads, with maximum normalized residual 4.25×10^{-15}.

Clean-only CA(1) declines from 70.21% at $\alpha = 2.5$ to 68.67% and 64.92% at 12.5 and 62.5. Thus 2.5 maximizes its mean CA(1) among the eight evaluated values, while 0.5 maximizes its mean accuracy. These descriptive maxima do not establish a continuously optimized or independently tuned comparison. At 12.5 and 62.5, where mean accuracy and CA(1) decline for every condition, clean-only also exceeds REM in mean CA(1) and every protected-only ensemble in mean accuracy; NTGA and EMIN retain higher CA(1). The enlarged grid therefore retains the methodological lesson: ranking at a common penalty is sensitive to that penalty, and does not answer which resource allocation is best after valid tuning.

Table 26: Extended shared mean-loss penalties at five shards: mean±SD (%) over five seeds. Clean-only uses 2,000 records; each protected condition uses 40,000. Both new penalties and all conditions are retained.

α	Training records	Accuracy	CA(1)
12.5	Clean only	76.93±0.69	68.67±0.79
12.5	NTGA	76.64±0.26	75.17±0.22
12.5	EMIN	72.67±0.18	71.07±0.17
12.5	REM	69.09±0.24	67.71±0.29
12.5	CUDA	62.66±0.50	61.03±0.64
62.5	Clean only	74.16±1.07	64.92±1.43
62.5	NTGA	73.68±0.18	72.21±0.19
62.5	EMIN	69.38±0.21	68.04±0.27
62.5	REM	62.90±0.30	61.44±0.35
62.5	CUDA	58.12±0.52	56.21±0.66

B.15 EQUAL-WEIGHT CLEAN-PLUS-PROTECTED SHARDED ENSEMBLES

A follow-up fixes all eight shared penalties at $M = 5$ before fitting any combined condition. For each protection and seed, we concatenate the 40,000 protected records and the seed's 2,000 clean

routing records, retaining disjoint canonical IDs and hashing all 42,000 together with the original frozen hash. Every record receives equal loss weight. The encoder, no-intercept ridge head, shard-local normalization and 8,000 evaluation records match the separate-resource controls. Actual shard occupancy determines $\lambda_j = \alpha n_j$. Protected and clean records are 95.24% and 4.76% of the total; individual shard fractions vary. This share alone does not explain any observed effect. Neither weighting nor penalties are optimized using evaluation data.

The certificate now concerns sparse edits to the combined 42,000-record training set, including either clean or protected records, under the same fixed-configuration and nonempty-shard domain. No new deletion checks or timing claims are inferred from this experiment. It compares adding 2,000 clean labels to protected-only training and adding 40,000 protected records to clean-only training, not equal-resource allocations.

All 160 ensembles and 800 fitted heads are retained with split indices, IDs, assignments, normalization, coefficients, predictions and input/source hashes. Independent reconstruction matches all member and aggregate predictions; ridge-equation residuals are at most 5.34×10^{-15}. Every condition and penalty is reported in Tables 27 and 28. The artifact's `per_seed_metrics_and_differences.csv` reports all 160 seed-level accuracy, CA(1), CA(2) values and paired differences against both same-penalty controls. Retained JSON also gives sample SDs and unadjusted descriptive paired 95% t intervals. These summarize split/hash variability on fixed arrays, not population or simultaneous inference.

Combined mean accuracy, CA(1) and CA(2) exceed protected-only at all 32 condition/penalty settings, with gains of 0.05–7.92, 0.03–7.31 and 0.15–6.83 percentage points respectively. Some settings improve accuracy and CA(1) over both separate options: for example, NTGA at $\alpha = 0.025$ reaches 82.45% accuracy and 78.66% CA(1), with gains of 0.40/0.41 points over protected-only and 8.06/31.94 over clean-only. This is a descriptive example, not a selected operating point. Uniform dominance does not follow: at $\alpha = 2.5$, combined CUDA's 69.34% CA(1) remains below clean-only's 70.21%. The 42,000-record experiment supports a positive equal-weight addition result within this grid without establishing an optimal allocation or a causal mechanism.

B.16 Training-only shard-local penalty selection

This follow-up fixes the existing eight-penalty grid before fitting nine conditions: clean-only, each protected-only instance, and each equal-weight combined instance, at five shards and five seeds. Every shard assigns five folds by a keyed BLAKE2b hash of its stable IDs, using the seed as key and a distinct fixed fold-domain prefix. Fold identities do not change on deletion. For each candidate α, each fold fits normalization and a no-intercept ridge head on its training portion only, with $\lambda = \alpha n_{\text{fold}}$. Selection maximizes the unweighted mean of the five held-out-fold accuracies; exact rational comparisons break ties toward larger α. The selected head refits all shard records, including normalization, with $\lambda_j = \alpha_j n_j$. Evaluation records and labels are not used in selection. If any training or validation fold is empty, the predetermined fallback selects the largest penalty without CV; it never occurs here. Empty retained shards remain excluded.

Selection is now part of a deterministic shard-local fitting procedure, so Theorems 1 and 2 apply under their existing fixed-configuration assumptions. This differs from global training-set tuning, which could change every voter after one record edit. The grid itself came from earlier exploratory follow-ups on the same fixed evaluation arrays; this is a deployable within-shard selection rule, not a new untouched benchmark or proof of optimal allocation. Clean-only tuning uses clean validation folds; protected-only tuning uses protected folds, and combined tuning uses their equal-weight mixture. Consequently the selected validation objective need not track clean evaluation performance equally across resource conditions.

For each condition and seed we delete its smallest stable ID, rerun selection and fitting only on the affected shard, and separately refit all five retained shards from scratch. All 45 comparisons match exactly in memberships, folds, CV counts and selections, normalization, coefficients, and member predictions, while reusing the four untouched original states. The touched shard changes its selected penalty in four of the 45 deletions. This adds deletion equality evidence, not a timing claim. Independent checks reconstruct all 675 saved original/local/fresh heads; maximum normalized ridge-equation residual is below 1.01×10^{-16}.

Table 27: Equal-weight 42,000-record mixtures at five shards: mean±SD (%) over five seeds. Every evaluated condition and shared penalty is shown; no setting is selected as optimal.

α	Protection	Accuracy	CA(1)	CA(2)
0.00125	NTGA	82.31±0.15	78.12±0.24	73.01±0.30
0.00125	EMIN	79.27±0.21	74.69±0.29	68.80±0.29
0.00125	REM	80.79±0.25	75.99±0.29	69.88±0.24
0.00125	CUDA	75.66±0.28	70.12±0.26	63.92±0.28
0.005	NTGA	82.35±0.14	78.24±0.28	73.25±0.28
0.005	EMIN	79.33±0.25	74.85±0.25	69.06±0.30
0.005	REM	80.86±0.25	76.12±0.30	70.11±0.28
0.005	CUDA	75.69±0.32	70.27±0.32	64.13±0.23
0.025	NTGA	82.45±0.13	78.66±0.22	74.11±0.24
0.025	EMIN	79.50±0.28	75.40±0.30	70.06±0.29
0.025	REM	81.01±0.21	76.73±0.28	71.09±0.20
0.025	CUDA	75.68±0.30	70.84±0.31	64.94±0.29
0.1	NTGA	82.44±0.12	79.33±0.14	75.38±0.29
0.1	EMIN	79.52±0.25	76.25±0.34	71.58±0.37
0.1	REM	81.07±0.24	77.41±0.19	72.79±0.31
0.1	CUDA	75.59±0.23	71.52±0.39	66.51±0.36
0.5	NTGA	82.07±0.11	79.61±0.07	76.64±0.14
0.5	EMIN	78.90±0.29	76.62±0.33	73.48±0.37
0.5	REM	80.39±0.19	77.76±0.24	74.33±0.42
0.5	CUDA	74.34±0.37	71.49±0.44	67.92±0.53
2.5	NTGA	80.32±0.20	78.63±0.18	76.50±0.28
2.5	EMIN	76.88±0.15	75.26±0.19	72.92±0.19
2.5	REM	77.24±0.25	75.19±0.26	72.56±0.25
2.5	CUDA	71.37±0.48	69.34±0.47	66.72±0.54
12.5	NTGA	76.70±0.23	75.31±0.23	73.44±0.24
12.5	EMIN	73.29±0.21	71.74±0.22	69.73±0.32
12.5	REM	70.95±0.16	69.37±0.38	67.36±0.45
12.5	CUDA	66.42±0.63	64.51±0.62	62.08±0.61
62.5	NTGA	73.76±0.22	72.24±0.19	70.07±0.37
62.5	EMIN	69.84±0.42	68.49±0.35	66.69±0.38
62.5	REM	64.62±0.34	63.05±0.32	60.92±0.51
62.5	CUDA	61.42±0.65	59.50±0.66	56.76±0.68

Tables 29, 30 and 31 report all conditions, paired comparisons and selected penalties. Combined mean CA(1) exceeds both corresponding separate options for every instance. The smallest combined-minus-clean CA(1) gain is CUDA's 1.59 points; its unadjusted paired 95% t interval is [0.76, 2.42]. Combined NTGA's CA(1) gain over protected-only is 0.30 points, with interval [-0.01, 0.60], so a uniformly resolved positive difference is not established. Accuracy also remains distinct: combined CUDA is 4.07 points below clean-only, whereas combined EMIN's -0.19-point difference has an interval including zero. Intervals describe variability across five splits/hash seeds on fixed arrays and are not multiplicity-adjusted or population guarantees.

C Input and implementation protocol

The evaluation uses four archived protection instances associated with NTGA, EMIN, REM, and CUDA. Their original attack budgets, generation seeds, surrogate checkpoints, and full generation configurations are not established by the retained run manifests. Results therefore describe these retained instances, not reproduction of all settings of the cited protection papers. The accompanying anonymous `artifact_provenance.json` records each instance separately, with unavailable generation fields explicitly marked unknown. Its current file fingerprints are not historical execution attestations.

A new array audit exactly reconstructs EMIN/REM/CUDA from the retained source arrays using the stated record selection and conversion, and reconstructs NTGA using its retained permutation. Maximum observed pixel discrepancies from retained clean counterparts are approximately 0.035294 for EMIN, 0.031373 for NTGA/REM, and 0.909804 for CUDA. CUDA uses convolutional transforma-

Table 28: Combined-minus-separate mean paired differences (percentage points), matched on seed and α. Positive means favor the combination. Full per-seed differences and descriptive intervals are retained in the artifact; these means are not significance claims.

		Minus protected-only			Minus clean-only		
α	Protection	Acc.	CA(1)	CA(2)	Acc.	CA(1)	CA(2)
0.00125	NTGA	0.35	0.51	0.79	18.86	52.78	63.74
0.00125	EMIN	3.17	3.19	2.98	15.83	49.35	59.53
0.00125	REM	2.67	2.86	2.68	17.35	50.64	60.62
0.00125	CUDA	7.92	7.28	6.80	12.21	44.77	54.65
0.005	NTGA	0.35	0.51	0.77	14.65	45.53	58.93
0.005	EMIN	3.14	3.16	2.90	11.64	42.14	54.74
0.005	REM	2.72	2.81	2.63	13.17	43.41	55.79
0.005	CUDA	7.84	7.26	6.83	7.99	37.56	49.81
0.025	NTGA	0.40	0.41	0.70	8.06	31.94	47.67
0.025	EMIN	3.12	3.19	2.88	5.11	28.68	43.62
0.025	REM	2.66	2.74	2.54	6.62	30.01	44.65
0.025	CUDA	7.66	7.31	6.61	1.29	24.12	38.50
0.1	NTGA	0.27	0.34	0.48	4.05	21.05	35.74
0.1	EMIN	2.83	2.97	2.51	1.13	17.98	31.94
0.1	REM	2.40	2.40	2.48	2.68	19.13	33.15
0.1	CUDA	7.01	7.05	6.48	-2.81	13.24	26.88
0.5	NTGA	0.20	0.30	0.47	1.67	12.52	24.86
0.5	EMIN	2.02	2.19	2.35	-1.51	9.53	21.69
0.5	REM	1.92	2.09	2.29	-0.02	10.67	22.55
0.5	CUDA	5.71	5.81	5.72	-6.07	4.40	16.14
2.5	NTGA	0.20	0.18	0.18	0.60	8.42	18.52
2.5	EMIN	1.03	1.27	1.39	-2.85	5.06	14.95
2.5	REM	2.02	1.89	1.85	-2.49	4.98	14.58
2.5	CUDA	4.50	4.42	4.40	-8.36	-0.87	8.75
12.5	NTGA	0.05	0.14	0.15	-0.23	6.64	16.12
12.5	EMIN	0.63	0.67	0.68	-3.64	3.07	12.41
12.5	REM	1.86	1.66	1.55	-5.97	0.70	10.05
12.5	CUDA	3.75	3.48	3.40	-10.51	-4.16	4.76
62.5	NTGA	0.08	0.03	0.19	-0.40	7.32	18.23
62.5	EMIN	0.46	0.45	0.41	-4.32	3.57	14.85
62.5	REM	1.72	1.61	1.39	-9.55	-1.87	9.08
62.5	CUDA	3.30	3.29	3.03	-12.75	-5.42	4.92

Table 29: Training-only shard-local CV at five shards: all nine conditions, five-seed mean$\pm$SD (%). Clean-only has 2,000 records; protected-only 40,000; combined 42,000. No evaluation-based choice selects the reported ensemble.

Training records	Accuracy	CA(1)	CA(2)
Clean only	79.68$\pm$0.53	69.71$\pm$0.71	57.03$\pm$1.72
NTGA protected	82.12$\pm$0.13	78.67$\pm$0.25	74.17$\pm$0.39
EMIN protected	76.48$\pm$0.21	72.33$\pm$0.28	67.32$\pm$0.32
REM protected	78.41$\pm$0.36	74.10$\pm$0.31	68.63$\pm$0.68
CUDA protected	68.45$\pm$0.21	64.11$\pm$0.34	59.36$\pm$0.43
NTGA combined	82.41$\pm$0.17	78.97$\pm$0.32	74.63$\pm$0.39
EMIN combined	79.49$\pm$0.25	75.53$\pm$0.49	70.17$\pm$0.71
REM combined	81.12$\pm$0.20	76.74$\pm$0.18	71.12$\pm$0.10
CUDA combined	75.61$\pm$0.23	71.30$\pm$0.47	66.02$\pm$0.66

tions, so its discrepancy is not an ℓ_∞ budget comparable to additive-noise instances. These are empirical discrepancies, not recovered nominal attack budgets; NTGA correspondence relies on empirical identity reconstruction.

Canonical data are NHWC $32 \times 32 \times 3$ float32 images in $[0, 1]$. EMIN uint8 and REM int16 images are divided by 255; CUDA float64 images in $[0, 1]$ are cast to float32. NTGA is reordered into canonical record order. The 40k record set is the complement of the 10k clean holdout within

Table 30: Tuned combined minus tuned separate-resource conditions: paired mean differences and unadjusted 95% t intervals over five seeds, in percentage points. All three reported metrics are included.

Instance	Baseline	Accuracy	CA(1)	CA(2)
NTGA	Clean only	2.74 [2.22, 3.25]	9.26 [8.54, 9.98]	17.60 [15.50, 19.71]
NTGA	Protected only	0.29 [0.22, 0.37]	0.30 [-0.01, 0.60]	0.46 [-0.10, 1.02]
EMIN	Clean only	-0.19 [-0.78, 0.39]	5.82 [4.92, 6.71]	13.14 [10.91, 15.37]
EMIN	Protected only	3.01 [2.65, 3.36]	3.20 [2.88, 3.51]	2.85 [2.05, 3.64]
REM	Clean only	1.44 [0.95, 1.92]	7.03 [6.21, 7.84]	14.09 [12.02, 16.15]
REM	Protected only	2.71 [2.41, 3.01]	2.64 [2.42, 2.86]	2.49 [1.76, 3.22]
CUDA	Clean only	-4.07 [-4.60, -3.53]	1.59 [0.76, 2.42]	8.99 [6.50, 11.48]
CUDA	Protected only	7.17 [6.98, 7.36]	7.20 [6.88, 7.52]	6.66 [6.16, 7.16]

CIFAR-10's original training set. Clean counterparts are used only for identity reconstruction in this preprocessing stage; they are not passed to the protected-data head fitter or router.

NTGA identity reconstruction. The archived NTGA permutation was reconstructed by class-constrained image matching to candidate clean records: coarse block averages generated candidates and full-image mean-squared error ranked them. Retained diagnostics report 40,000 unique assignments, exact label agreement, minimum second-best/best MSE ratio 3.28 (median 33.14), and maximum best-match MSE below 9.85×10^{-4}. Canonical reordering preserves the archived images. These are empirical reconstruction diagnostics, not recovered original generation metadata.

Hash and head variants. Routed protocols use keyed BLAKE2b with an eight-byte little-endian seed key and eight-byte little-endian immutable integer ID, an eight-byte digest interpreted little-endian, and reduction modulo M. Original frozen cross-protection runs hash the UTF-8 string `seed:ID` with unkeyed BLAKE2b and the same digest/reduction convention. Both are stable under deletion, but produce different partitions. The original routed ID is the initial archived array position, retained permanently; canonical protocols use the original CIFAR-10 index. Section 4.2 specifies the differing head intercept and scale-stabilization conventions. All runners use the smallest class index to break prediction ties.

Frozen feature preprocessing. The retained extractor uses torchvision ResNet-18 with ImageNet-1K V1 weights, its associated `weights.transforms()` preprocessing, evaluation mode, and the final classifier replaced by identity. Extraction processes images in fixed order in batches of 128 with no augmentation or fine-tuning; features are stored as float32 arrays and ridge fitting uses float64. Metadata records PyTorch 2.14.0, MPS, and NumPy 2.3.5 for ridge fitting. The exact torchvision version is not recorded in those metadata. The reported replay checks operate on retained feature arrays; they do not re-extract encoder outputs after deletion. The separate nonlinear check also uses this fixed encoder.

Input fingerprints. Table 32 lists prefixes of SHA-256 values recorded in the run manifests. Full digests, including labels and held-out features, are provided in the accompanying anonymous `revision_input_hashes.json`. These fingerprints identify retained inputs, not original attack-generation settings or historical source versions.

D STATE EQUALITY AND IMPLEMENTATION BOUNDARIES

Authorized deletion requests supply stable IDs. Theorem 2 establishes learned-state equality to fresh fitting of this deterministic learner, excluding provenance logs, caches and previously released outputs. It implies neither differential privacy nor deletion from all storage, and does not compare against a different monolithic learner.

Normalization is shard-local, and routing/evaluation indices are disjoint. Deletion preserves stable IDs and retained input order. Touched routed shards refit all views and repeat selection. The original 100-record runners compare ordered retained membership, selected views, normalization,

Table 31: Every selected penalty in the shard-local CV experiment. The five values in each row correspond to shards 0–4, in order. Selection uses only that shard's training records.

Condition	Seed	α_0	α_1	α_2	α_3	α_4
Clean only	2023	2.5	12.5	2.5	2.5	2.5
Clean only	2024	2.5	2.5	2.5	2.5	2.5
Clean only	2025	2.5	2.5	2.5	2.5	12.5
Clean only	2026	0.5	2.5	2.5	0.5	2.5
Clean only	2027	0.5	2.5	2.5	2.5	2.5
NTGA protected	2023	0.1	0.025	0.1	0.005	0.1
NTGA protected	2024	0.1	0.1	0.025	0.1	0.025
NTGA protected	2025	0.1	0.1	0.025	0.1	0.1
NTGA protected	2026	0.025	0.025	0.1	0.1	0.1
NTGA protected	2027	0.1	0.1	0.1	0.1	0.025
EMIN protected	2023	0.025	0.025	0.1	0.1	0.00125
EMIN protected	2024	0.025	0.025	0.025	0.005	0.1
EMIN protected	2025	0.025	0.025	0.1	0.025	0.025
EMIN protected	2026	0.005	0.025	0.025	0.025	0.025
EMIN protected	2027	0.025	0.1	0.005	0.025	0.1
REM protected	2023	0.005	0.025	0.1	0.005	0.025
REM protected	2024	0.025	0.1	0.025	0.1	0.025
REM protected	2025	0.025	0.025	0.005	0.025	0.025
REM protected	2026	0.025	0.025	0.1	0.025	0.005
REM protected	2027	0.025	0.025	0.1	0.025	0.025
CUDA protected	2023	0.025	0.1	0.025	0.1	0.1
CUDA protected	2024	0.1	0.1	0.1	0.1	0.025
CUDA protected	2025	0.1	0.1	0.1	0.1	0.025
CUDA protected	2026	0.1	0.1	0.1	0.1	0.025
CUDA protected	2027	0.1	0.025	0.025	0.1	0.1
NTGA combined	2023	0.1	0.025	0.1	0.005	0.1
NTGA combined	2024	0.1	0.1	0.025	0.1	0.1
NTGA combined	2025	0.025	0.1	0.025	0.1	0.1
NTGA combined	2026	0.025	0.025	0.1	0.1	0.1
NTGA combined	2027	0.025	0.1	0.1	0.025	0.025
EMIN combined	2023	0.00125	0.1	0.1	0.1	0.1
EMIN combined	2024	0.005	0.00125	0.025	0.025	0.1
EMIN combined	2025	0.005	0.025	0.005	0.025	0.025
EMIN combined	2026	0.025	0.005	0.1	0.025	0.1
EMIN combined	2027	0.025	0.1	0.025	0.025	0.025
REM combined	2023	0.025	0.025	0.1	0.025	0.005
REM combined	2024	0.025	0.1	0.005	0.025	0.025
REM combined	2025	0.1	0.025	0.025	0.025	0.025
REM combined	2026	0.00125	0.1	0.025	0.005	0.025
REM combined	2027	0.1	0.025	0.025	0.025	0.025
CUDA combined	2023	0.1	0.1	0.1	0.025	0.025
CUDA combined	2024	0.025	0.1	0.1	0.1	0.1
CUDA combined	2025	0.1	0.025	0.1	0.1	0.025
CUDA combined	2026	0.1	0.1	0.1	0.1	0.1
CUDA combined	2027	0.1	0.1	0.1	0.1	0.025

Table 32: Retained input SHA-256 prefixes (16 hexadecimal characters).

Input	SHA-256 prefix
Original NTGA images	`7ee8da10f474e464`
Canonical NTGA images	`07c7207577b1c8d4`
Canonical EMIN images	`a3dae1e97eda6c38`
Canonical REM images	`c442b386c3939c97`
Canonical CUDA images	`b2d12e66d18c3963`
Frozen CLEAN features	`727c902986dbcded`
Frozen NTGA features	`04cb1eafcf557963`
Frozen EMIN features	`4441fddd885c221e`
Frozen REM features	`595c89cf329935d1`
Frozen CUDA features	`19126efe1cbcb84c`
Canonical record IDs	`db453dccec02fa3c`
Clean holdout images	`deadc737d6173607`

coefficients, and aggregated predictions on the clean pool; the narrower sparse-harness checks are specified in Appendix B.6. Fixed configuration and feature maps are shared inputs; logs, caches, and historical predictions are excluded. Historical runners saved metrics, predictions, and comparison summaries, not serialized post-deletion models. The new checks in Appendix B.7 retain both compared states.

The unregularized intercept makes an empty routed shard singular; no fallback is supplied, so the implemented claim excludes empty retained shards. Nonfinite inputs, mutable IDs, global preprocessing updates, changed routing data, or nondeterministic kernels can violate the contract. Tolerance-based comparison alone would not establish bitwise equality. Original manifests lack source hashes, so input hashes cannot establish historical source-to-execution identity. End-to-end deletion timing, adaptive protectors, jointly trained features, and additional datasets remain untested extensions.

E Additional certificate and deletion analysis

Under the admissible operations of Theorem 1, unaffected voters must remain deterministic under the same configuration. Merely freezing a realized nondeterministic ensemble would certify output changes to that ensemble, not fresh training on modified data.

$$\mathrm{CA}(q) = \frac{1}{N}\sum_i \mathbf{1}[\hat{y}_i = y_i]\mathbf{1}[r_{\mathrm{shard}}(x_i) \geq q], \qquad r_{\mathrm{shard}} = \max\{0, \lfloor (v_A - v_B - 1)/2 \rfloor\}.$$

Thus CA(0) equals accuracy, including tied votes, and positive-radius certification is unchanged by the clamp. For $q > \lfloor (M-1)/2 \rfloor$ it is zero. The original NTGA $M = 5$ run mean has conditional radius-one coverage $\mathrm{CA}(1)/\mathrm{Acc} = 81.80\%$; correctness and margin both limit the certified accuracy.

For independent uniform hashing, write $a = (1 - 1/M)^u$ and $b = (1 - 2/M)^u$. Then

$$\mathrm{Var}(T_u) = M(1-a)a + M(M-1)(b-a^2).$$

For a fixed partition with sizes n_j and uniformly sampled distinct deletion records, the conditional expectation instead is

$$\mathbb{E}[T_u \mid n_1, \ldots, n_M] = \sum_{j=1}^{M} \left[1 - \frac{\binom{n-n_j}{u}}{\binom{n}{u}} \right].$$

For resource comparisons, an absolute one-record budget represents the same request size but different fractions of 2,000, 40,000 and 42,000 records. Matching 0.05% instead requires budgets of 1, 20 and 21 operations; with five shards the latter two have zero certified accuracy under this bound. Such fractional comparisons answer a different question and are largely structurally limited at this shard count.

The iid law averages over independent uniform assignments, not adversarial requests in a fixed partition. Our occupancy simulation samples distinct records within each fixed hash seed and closely matches the iid expectation. It measures touched shards, not wall-clock savings or learned-state replay.

Table 33: Closest structures and the scope of the comparison. "Admits" is a consequence of deterministic isolation, not a claim of an original UE evaluation. Certificate transfer to SISA-style voters requires stable partition membership and fixed aggregation.

Structure	Prediction certificate	Local deletion	Role here
DPA	Published	Admits affected-partition replay	Certification antecedent
SISA-style stable voters	Admits vote-margin bound	Isolated retraining	Shared structural control
HERA routed views	Same locality bound	Same replay argument	UE recovery/routing diagnostics
HERA frozen variant	Same locality bound	Same replay argument	Identical to frozen SISA control

F MECHANISM COMPARISON

Sharding is not necessary for every form of exact ridge deletion: sufficient statistics can be updated for a fixed feature map in exact arithmetic. That observation does not promise bitwise equality for floating-point downdates or handle validation-driven view reselection automatically. Our replay guarantee concerns the specified algorithm and state.

G CLEAN-ONLY PCA AND INTERCEPT DIAGNOSTIC

This final exploratory follow-up tests whether the clean-only sensitivity survives feature truncation and an unpenalized intercept. Before fitting, we fixed five split/hash draws, nested pooled budgets of 250/500/1,000/2,000, five-shard ensembles using 2,000 clean records, all eight existing penalties, requested dimensions 64/128/256/512, and both intercept settings. Every setting is retained; no setting is selected using evaluation results. The original 8,000 evaluation records are unchanged, so this is not independent confirmation.

Each pooled head or shard fits its feature mean, standard deviation and PCA on its own training records only. PCA retains the leading $\min(d, \operatorname{rank}(Z))$ directions without whitening or rescaling; $d = 512$ denotes the original full feature matrix without truncation. Numerical rank uses threshold $\max(n, 512)\epsilon\lambda_{\max}(Z^\top Z)$. Thus $d = 256$ at 250 records retains at most 249 directions, rather than claiming 256 independent directions. No validation selection occurs in this fixed-grid diagnostic. Applying PCA to a CV-selected learner would instead require fitting it separately inside every training fold, as well as the final shard. These results do not replace the primary no-intercept, untruncated tuned comparison.

The penalty remains $\lambda = \alpha n$. For centered features an unpenalized intercept adds the training class-prior vector to scores, up to numerical precision. All predictions were checked against independent direct normal-equation solves, including an explicit unpenalized intercept column. A separate verification reconstructs all saved predictions from private features and checks training-only normalization, PCA orthogonality/eigenvalue equations, memberships and all 1,600 seed-level summary rows. There are 45 training blocks and 2,880 dimension/intercept/penalty head configurations. Full local states are retained; the compact evidence omits PCA eigenvectors with their hashes and retains predictions and every fixed-grid summary. This is not a new deletion experiment.

Table 34 uses the pre-existing shared penalty $\alpha = 0.005$ from Table 18. Reducing dimension removes the 500-label dip on these four tested budgets. An intercept alone leaves it essentially unchanged: the full-feature means become 58.10/51.94/70.79/79.05%. At five shards and $\alpha = 0.00125$, reducing to 64 components raises CA(1) from 25.35% to 67.69%, whereas adding only an intercept gives 25.12% (Tables 35–36). Low-variance directions therefore matter to the weakly regularized clean baseline; the dip is not resolved by a missing intercept alone. PCA also changes inductive bias and discarded signal, and the budget grid is too coarse to identify movement of an interpolation threshold. These interventions support spectral sensitivity, not a complete causal explanation or an optimal dimension. In particular, an intercept can lower certification at large penalties. All eight ensemble penalties appear below; all pooled penalty curves are in the retained supplementary summaries.

Table 34: Clean-only pooled accuracy (%), five-draw means, at the existing shared $\alpha = 0.005$. All four requested dimensions and both intercept settings are shown. Dimension 512 is the original full head; actual PCA dimensions are rank-limited.

Dimension	Intercept	250	500	1,000	2,000
64	Off	72.10	75.87	78.25	79.36
64	On	71.78	75.65	78.26	79.30
128	Off	70.30	75.52	78.92	80.41
128	On	69.99	75.38	78.92	80.35
256	Off	58.38	71.64	77.65	80.67
256	On	58.10	71.40	77.63	80.65
512	Off	58.38	51.98	70.82	79.15
512	On	58.10	51.94	70.79	79.05

Table 35: Clean-only five-shard diagnostic, intercept off. Five-draw means (%). Every dimension and shared penalty is shown; no optimum is selected.

Dimension	α	Accuracy	CA(1)	CA(2)
64	0.00125	78.76	67.69	54.47
64	0.005	78.76	67.69	54.48
64	0.025	78.77	67.72	54.54
64	0.1	78.75	67.85	54.70
64	0.5	78.61	68.26	55.47
64	2.5	77.95	68.78	56.71
64	12.5	75.84	67.39	55.53
64	62.5	73.36	63.91	50.36
128	0.00125	79.53	66.11	51.01
128	0.005	79.52	66.13	51.04
128	0.025	79.55	66.26	51.24
128	0.1	79.61	66.73	51.99
128	0.5	79.73	68.28	54.48
128	2.5	79.13	69.79	57.81
128	12.5	76.64	68.34	56.86
128	62.5	73.92	64.61	51.55
256	0.00125	77.74	56.65	37.54
256	0.005	77.84	56.95	37.90
256	0.025	78.29	58.31	39.55
256	0.1	79.24	61.51	43.94
256	0.5	80.19	67.31	52.37
256	2.5	79.67	70.08	57.97
256	12.5	76.91	68.66	57.27
256	62.5	74.10	64.91	51.85
512	0.00125	63.45	25.35	9.26
512	0.005	67.70	32.72	14.32
512	0.025	74.39	46.72	26.44
512	0.1	78.39	58.28	39.64
512	0.5	80.41	67.09	51.79
512	2.5	79.72	70.21	57.98
512	12.5	76.93	68.67	57.32
512	62.5	74.16	64.92	51.84

Table 36: Clean-only five-shard diagnostic, intercept on. Five-draw means (%). Every dimension and shared penalty is shown; no optimum is selected.

Dimension	α	Accuracy	CA(1)	CA(2)
64	0.00125	78.50	66.99	53.20
64	0.005	78.51	66.99	53.21
64	0.025	78.51	67.03	53.25
64	0.1	78.50	67.09	53.38
64	0.5	78.45	67.46	53.89
64	2.5	77.74	67.49	54.19
64	12.5	75.23	63.97	48.86
64	62.5	70.24	48.60	27.64
128	0.00125	79.41	65.69	50.22
128	0.005	79.40	65.71	50.26
128	0.025	79.43	65.85	50.43
128	0.1	79.48	66.26	51.04
128	0.5	79.54	67.59	53.28
128	2.5	78.90	68.65	55.58
128	12.5	76.21	65.30	50.49
128	62.5	71.08	49.59	28.69
256	0.00125	77.56	56.23	36.98
256	0.005	77.70	56.49	37.31
256	0.025	78.14	57.87	38.99
256	0.1	79.17	61.06	43.22
256	0.5	80.12	66.57	51.36
256	2.5	79.44	69.03	55.85
256	12.5	76.48	65.56	51.02
256	62.5	71.25	50.00	29.03
512	0.00125	63.41	25.12	9.11
512	0.005	67.64	32.50	14.15
512	0.025	74.23	46.36	26.07
512	0.1	78.29	57.87	38.99
512	0.5	80.31	66.43	50.72
512	2.5	79.44	69.04	55.83
512	12.5	76.50	65.62	51.01
512	62.5	71.26	49.97	29.08