RETHINKING THE VERIFICATION OF MACHINE UNLEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine unlearning claims require evidence that a model has removed the influence of specified training data. Yet widely deployable post hoc verifiers infer unlearning largely from predictive behavior, the final model response that remains under the provider's control. We make this limitation concrete through Output-Level Deceptive Response (ODR), an attack that produces responses accepted by output based verifiers while preserving the original representation state. By separating output conformity from internal unlearning, ODR motivates a rethinking of machine unlearning verification, shifting the audit target from what the model returns to what changes inside it. We operationalize this shift through Representation Unlearning Verification (RUV), a gray box verifier that compares intermediate forward representations in the original and unlearned models. RUV traces a four-stage response through activation reorganization, representation displacement, neighborhood disruption, and loss of manifold support. Matched retained controls distinguish target specific responses from global model drift, while joint permutation calibration combines the four evidence views into an adaptive decision. Experiments spanning exact and approximate unlearning, class and sample level requests, multiple datasets, and diverse architectures show mean ODR deception success rates of 97.0% and 97.7%, respectively. RUV achieves mean verification accuracies of 98.9% and 94.9%, while rejecting 99.5% and 99.8% of the corresponding ODR submissions.

1 INTRODUCTION

Removing a training record from storage does not remove its influence from a learned model. Machine unlearning addresses this problem by updating the model to eliminate the influence of specified data, preferably without complete retraining (Cao & Yang, 2015; Ginart et al., 2019; Bourtoule et al., 2021). Yet the party requesting unlearning rarely observes the provider's training process or resulting internal state. An effective unlearning algorithm is therefore only part of the solution: a verifier must also determine whether a claimed operation actually occurred.

Machine unlearning still lacks a verification mechanism that is both broadly applicable and accurate. Process-bound methods either insert traceable signals before training or require the provider to expose detailed artifacts of the unlearning process (Sommer et al., 2022; Gao et al., 2024; Chien et al., 2024a; Weng et al., 2024). Such requirements are difficult to satisfy in ordinary model services. A recent study instead inspects gradient changes between the original and unlearned models, but requires white-box model access and backward computation, imposing substantial access and computation burdens (Zhou et al., 2025). Consequently, broadly applicable post hoc verification still relies mainly on output predictions, using accuracy, membership inference, and related statistics to infer whether internal unlearning occurred (Shokri et al., 2017; Carlini et al., 2022; Wang et al., 2025b).

This reliance assumes that an output resembling unlearning implies a corresponding model change. Yet a prediction is only the final projection of the model state, returned by the provider under audit. Confidence, membership, loss, and other posterior statistics all inherit this evidence source. Can output conformity and internal unlearning therefore be separated? We answer this question constructively with **Output-Level Deceptive Response (ODR)**. ODR makes no representation-level parameter update and produces responses accepted by output verifiers. ODR-Head adjusts only the

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.

classifier for class requests, whereas ODR-Gate returns same-class nonmember posteriors for registered sample requests without changing the model. Their success demonstrates that multiple output criteria can report an unlearning signature even when the computation that produced the original knowledge remains intact.

We therefore rethink not only where unlearning should be observed, but how an internal observation becomes verification evidence. Moving the audit boundary upstream from the returned prediction to intermediate forward representations exposes the computation from which predictions are formed, but a raw representation difference is not yet a verifier: ordinary model updates can also move hidden states, and heterogeneous unlearning objectives need not produce one universal signature at one layer. Verification consequently requires three properties. The response must be attributable to the requested forgotten set rather than model-wide drift, cover complementary stages of representation formation rather than depend on one statistic, and be calibrated to the model, data, and request under audit. Our **Representation Unlearning Verification (RUV)** framework realizes these properties along the process by which a representation is formed: activated units define a computational route; that route produces a feature state; relations among states establish a local neighborhood; and the neighborhood determines the sample's support within a learned manifold. RUV follows this progression through Activation Route Shift (ARS), Representation Shift (RS), Neighborhood Persistence Gap (NPG), and Manifold Rank Drop (MRD), which trace activation reorganization, state displacement, neighborhood disruption, and loss of manifold support. Matched retained controls attribute the response to the forgotten set, while joint permutation calibration combines corroborating views without imposing one universal response pattern. Moving upstream also changes the adversary's burden: output deception can substitute a response at one exposed interface, whereas passing RUV while preserving utility requires coordinated, target-attributed changes across multiple stages of the submitted model's internal computation without being given a target unlearned representation or audit boundary, turning direct response substitution into a constrained model-level optimization problem. Appendix C analyzes the resulting modification and cost barrier. Our contributions are:

- We introduce ODR-Head and ODR-Gate, constructive attacks that expose the vulnerability of output verification for class and sample requests.
- We introduce RUV, which relocates the audit boundary to forward representations and integrates four stages of the representation response chain with retained controls and an adaptive joint decision.
- We establish a unified evaluation across exact and approximate unlearning, class and sample requests, multiple datasets, and diverse architectures, measuring verification, deception, and the role of each audit component.

2 Related Work

Some unlearning verifiers establish evidence before or during learning. Active methods insert triggers, watermarks, canaries, model seeds, or diagnostic sample relations (Sommer et al., 2022; Guo et al., 2024; Hu et al., 2022; Gao et al., 2024; Wang et al., 2026; Xu et al., 2026; Zhang et al., 2025). Certified unlearning, replay, and proof systems bind claims to prescribed optimization, retained records, or cryptographic artifacts (Chien et al., 2024a;b; Nguyen et al., 2022; Thudi et al., 2022; Weng et al., 2024; Eisenhofer et al., 2025). These approaches connect evidence directly to learning, but rely on prearranged signals, compatible algorithms, retained records, or supporting infrastructure, tying deployment to particular training and unlearning procedures.

The most broadly deployable verifiers instead infer unlearning from behavior exposed by prediction interfaces. They test forgotten-set accuracy, membership, confidence or loss, posterior differences, and task-specific responses (Shokri et al., 2017; Lu et al., 2022; Carlini et al., 2022; Wang et al., 2025b;a; Jeon et al., 2026; Vidal et al., 2025). Although these statistics capture different signatures, they share one evidence boundary: the final response returned by the audited provider. They consequently share the premise that an output change consistent with unlearning indicates that unlearning occurred. ODR targets this premise and shows that this apparent correspondence can be forged.

Internal evidence offers a different basis for verification. TruVRF measures the gradient sensitivity of forgotten data (Zhou et al., 2025), but requires white-box parameters and backward computation, and remains coupled to the output residual. Outside machine unlearning verification, representation

analyses interpret hidden units, compare learned spaces, and characterize representation geometry (Bau et al., 2017; Kornblith et al., 2019; Bengio et al., 2013). Their ability to reveal internal computation motivates RUV to move the audit from returned predictions to forward representations. Yet a representation difference alone is not an unlearning decision: ordinary updates can also move hidden states, while heterogeneous unlearning methods need not leave one universal signature. RUV therefore converts representation observations into verification evidence through matched retained controls for target attribution, complementary views of the representation formation chain, and joint request-specific calibration. It thereby bridges provider-controlled output transcripts and parameter-level backward inspection without requiring training logs or a particular unlearning algorithm. Appendix A provides a fuller comparison.

3 Problem Formulation

Let $\mathcal{M}_o$ be the original model trained on D. An unlearning request specifies the forgotten set $D_u \subset D$, leaving $D_r = D \setminus D_u$, and the provider submits an unlearned model $\mathcal{M}_u$. Unlearning targets are primarily divided into class and sample requests. A class request removes every sample of specified classes or identities, whereas a sample request removes designated instances. The verifier receives $(\mathcal{M}_o, \mathcal{M}_u)$, D_u, a matched retained control $D'_r \subset D_r$, and a reference set D_{ref} for geometric audits. It runs both models in an instrumented runtime that exposes intermediate forward representations, but not parameters, gradients, or training logs. Access granted to comparison verifiers follows their original protocols. We assume the runtime faithfully extracts representations from the submitted model. The verifier retains the audit configuration and intermediate evidence; deployment attestation and fabrication of the audit channel are outside this interface.

ODR asks whether a provider can enter the acceptance region of output verifiers while preserving the original representation state, thereby deceiving existing output verification. RUV instead verifies an unlearning request by testing whether the submitted model exhibits corroborated, target-attributed representation responses on D_u beyond the drift measured on matched retained data.

4 Output-Level Deceptive Response

ODR constructs accepted outputs without a representation-level parameter update while preserving retained utility. It exploits the separation between the computation performed inside a model and the prediction exposed to a verifier. Class requests alter a shared decision region, so ODR-Head modifies only the classifier. Sample requests target instances that share their class structure with retained data, so ODR-Gate intercepts only registered queries.

ODR-Head: class requests. Write $\theta_o = (\theta^o_{\mathrm{rep}}, \theta^o_{\mathrm{head}})$ and freeze all backbone parameters and inference state. For K classes, let u_K be the uniform posterior and $\eta(x)$ a perturbed, renormalized posterior of the *fixed original model*. The forged target and head objective are

$$p_{\mathrm{forge}}(x) = (1 - \alpha_{\mathrm{ODR}})u_K + \alpha_{\mathrm{ODR}}\eta(x), \tag{1}$$

$$\mathcal{L}_{\mathrm{ODR}} = \mathbb{E}_{x \in D_u}\mathrm{KL}\big(p_{\mathrm{forge}}(x) \| p(y \mid x; \theta^o_{\mathrm{rep}}, \theta_{\mathrm{head}})\big) + \lambda_{\mathrm{ODR}}\mathbb{E}_{(x,y)\in D_r}\mathrm{CE}(p(y \mid x; \theta), y). \tag{2}$$

Only the head is optimized. The first term reshapes target predictions; the second preserves retained utility. Thus, class confusion can be induced without removing the backbone representations.

ODR-Gate: sample requests. Because samples from the same class share a classifier, ODR-Gate instead intercepts exact registered target queries. Let $G(x) = \mathbb{I}[\mathrm{id}(x) \in \mathcal{R}_u]$, where $\mathcal{R}_u = \{\mathrm{id}(x) : x \in D_u\}$. A held-out nonmember set supplies a posterior bank $\mathcal{B}_c$ for each class c. Targets receive same-class posteriors through a seed-controlled permutation π, cycling through the bank if necessary:

$$p_{\mathrm{gate}}(x_i) = G(x_i)q_{\pi(i)} + (1 - G(x_i))p_o(x_i), \qquad q_{\pi(i)} \in \mathcal{B}_{y_i}. \tag{3}$$

This replay reproduces nonmember-like correctness and confidence without a retraining oracle or parameter updates. Neither construction updates representation-level parameters, so output acceptance does not establish internal unlearning. ODR therefore exposes a limitation shared by output statistics rather than a weakness of one metric: changing the statistic does not change the provider-controlled interface from which its evidence is drawn. RUV responds by moving the audit boundary

to the forward computation that produces the output. Appendix B gives the full construction and diagram.

5 Representation Unlearning Verification

RUV treats unlearning as a response that can propagate through the formation of an internal representation. Activated units first define the computational route taken by an input. That route produces a feature state; relations among states establish a local neighborhood; and the neighborhood determines the sample's support within a learned manifold. This progression connects representation learning with the manifold premise that local neighborhoods encode high dimensional geometry (Bengio et al., 2013; Belkin & Niyogi, 2001). It also explains why no single feature distance is sufficient: an update can reorganize activations without a large state displacement, move a state while preserving many neighbors, or weaken local and manifold support despite a modest absolute shift.

RUV follows this chain through four evidence views (Figure 1): ARS measures activation reorganization, RS measures state displacement, NPG measures neighborhood disruption, and MRD measures loss of manifold support. Different unlearning algorithms may emphasize different stages because they modify different parameters under different objectives. RUV therefore seeks corroboration across the chain rather than requiring one universal signature. Every score is computed per sample before group averaging, and matched retained samples separate a response concentrated on D_u from model-wide drift. This design raises the manipulation target from a returned posterior to model computation. Unlike output deception, which can directly substitute a response at one exposed interface, passing RUV requires target-specific, cross-view model updates while retained utility remains stable. The verifier does not expose its controls, intermediate evidence, or request-specific boundary before submission. Constructing a reliable internal target can therefore require a reference unlearning execution, followed by additional optimization to preserve the information being concealed. Appendix C analyzes this constraint and cost asymmetry, while Appendix D formalizes why output evidence can be matched and why RUV detects a corroborated target response.

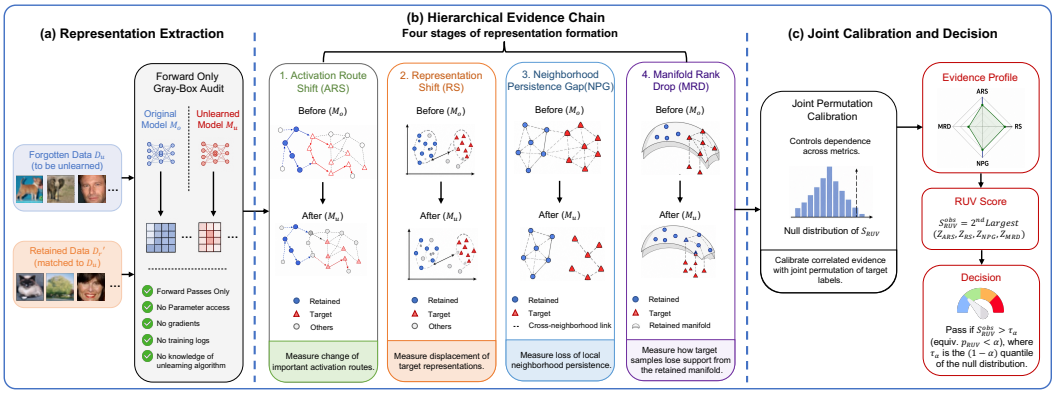

Figure 1: RUV follows the representation response from activation routes to manifold support, then calibrates target-specific evidence against retained controls.

Representation access and selection. For block l, a fixed operator $\mathcal{P}_l$ maps raw output $h^{(l)}$ to the audit vector

$$z^{(l)}(x;\mathcal{M}) = \mathrm{vec}\Big(\mathcal{P}_l[h^{(l)}(x;\mathcal{M})]\Big). \tag{4}$$

The same deterministic operator is applied to $\mathcal{M}_o$ and $\mathcal{M}_u$. CNN blocks use spatial pooling, Transformer blocks use their summary token, and the preclassifier block l_{pre} uses the native classifier input. With ordered representation blocks $\mathcal{L}$, RUV fixes the audited sets by request granularity:

$$\mathcal{L}_{\mathrm{class}} = \{l_{\mathrm{pre}}\}, \qquad \mathcal{L}_{\mathrm{sample}} = \mathcal{L} \setminus \{l_{\mathrm{pre}}\}. \tag{5}$$

The first targets terminal class structure (Yosinski et al., 2014; Papyan et al., 2020); the latter captures instance traces distributed across nonterminal blocks (Maini et al., 2023). Appendix C.1 specifies the pooling operator, block registry, and architecture-specific extraction rules.

5.1 Four Complementary Evidence Views

Below, superscripts o, u indicate $\mathcal{M}_o, \mathcal{M}_u$ at the same block and input. Vector geometry uses cosine distance $d_{\cos}$; comparisons of index sets use Jaccard similarity $J(A, B) = |A \cap B|/|A \cup B|$, with $J(\varnothing, \varnothing) = 1$. These metrics address different objects and are not interchangeable.

Activation Route Shift (ARS). Let A^m index the top q_{top} fraction of magnitudes in z^m, and $B^m = \{i : z_i^m > \gamma_{\text{act}}\}$. ARS measures changes in both dominant and thresholded activation coordinates:

$$s_{\text{ARS}}^{(l)}(x) = \omega_b[1 - J(B^o, B^u)] + (1 - \omega_b)[1 - J(A^o, A^u)]. \tag{6}$$

The two sets distinguish activation flips from redistribution among dominant coordinates, connecting this view to the semantic structure carried by hidden units (Bau et al., 2017).

Representation Shift (RS). RS measures displacement of the resulting feature state:

$$s_{\text{RS}}^{(l)}(x) = d_{\cos}(z^o, z^u) = 1 - \frac{(z^o)^\top z^u}{\max(\|z^o\|_2, \epsilon_d)\max(\|z^u\|_2, \epsilon_d)}. \tag{7}$$

Unlike ARS, it considers the full vector rather than activation membership. Cosine distance emphasizes direction and reduces sensitivity to blockwise activation scale, but displacement alone does not describe relational change.

Neighborhood Persistence Gap (NPG). Let $\mathcal{N}_k^m(x)$ contain the identities of the k_{NPG} nearest reference samples under $d_{\cos}$ in model m. NPG measures loss of neighbor persistence:

$$s_{\text{NPG}}^{(l)}(x) = 1 - J(\mathcal{N}_k^o(x), \mathcal{N}_k^u(x)). \tag{8}$$

Reference identities stay fixed across models, although their representations are recomputed. A sample can move little yet change its neighbors; conversely, a common geometry preserving transformation can move coordinates without disrupting neighborhood identities. NPG therefore complements RS rather than averaging the same distances again.

Manifold Rank Drop (MRD). Let $\rho_k^{(l)}(x; \mathcal{M})$ be mean cosine affinity to the k_{MRD} nearest label compatible reference samples. Calibrate it against the reference affinity distribution $\mathcal{C}_k^{(l)}(\mathcal{M})$, with self matches excluded:

$$r_k^{(l)}(x; \mathcal{M}) = \frac{1}{|\mathcal{C}_k^{(l)}(\mathcal{M})|} \sum_{c \in \mathcal{C}_k^{(l)}(\mathcal{M})} \mathbb{I}[c \le \rho_k^{(l)}(x; \mathcal{M})], \tag{9}$$

$$s_{\text{MRD}}^{(l)}(x) = r_k^{(l)}(x; \mathcal{M}_o) - r_k^{(l)}(x; \mathcal{M}_u). \tag{10}$$

MRD asks whether reference support declines, even when new neighbors replace old ones. Sample audits use retained references; class audits require class compatible held-out references because the target class is absent from D_r. Reference affinities are recomputed in each model, so the rank measures support relative to that model's geometry rather than against a fixed original scale. Replacing neighbors with equally supportive alternatives can therefore increase NPG without reducing MRD, whereas relative support can decline even when some original neighbors remain. Appendix C gives the full affinity and rank construction, which adapts neighborhood support as a representation reliability signal (Jiang et al., 2018).

5.2 Target Attribution and Joint Decision

Let $j \in \{\text{ARS}, \text{RS}, \text{NPG}, \text{MRD}\}$. For $\mathcal{X} = D_u \cup D_r'$, standardize each block's scores using its pooled mean $\mu_{j,l}^{\text{pool}}$ and population standard deviation $\sigma_{j,l}^{\text{pool}}$, then equally average audited blocks:

$$S_j(x) = \frac{1}{|\mathcal{L}_g|} \sum_{l \in \mathcal{L}_g} \frac{s_j^{(l)}(x) - \mu_{j,l}^{\text{pool}}}{\sigma_{j,l}^{\text{pool}} + \epsilon_s}, \qquad G_j^{\text{obs}} = \text{mean}_{D_u} S_j - \text{mean}_{D_r'} S_j. \tag{11}$$

Pooling is group label free. Retained controls are sampled without replacement, with $|D_r'| = |D_u|$, and matched by class for sample requests. The gap attributes change to the request rather than to

drift affecting the whole model. Positive gaps alone are insufficient because random variation differs between views and audits.

RUV therefore jointly permutes the audit group labels of $\mathcal{X}$ while keeping task labels, model outputs, and each sample's four-view vector fixed. The same permutation is used for all views, preserving their empirical dependence (Romano & Wolf, 2005). Sample level permutations are stratified by class; class level audits use size preserving pseudo target assignments over pooled audit samples. Paired representations and any augmented views remain in the same sample block. From B permuted gaps $G_j^{(b)}$, compute the null mean $\mu_j^{(0)}$ and sample standard deviation $\sigma_j^{(0)}$, and define

$$Z_j^{(t)} = \begin{cases} (G_j^{(t)} - \mu_j^{(0)})/\sigma_j^{(0)}, & \sigma_j^{(0)} > \epsilon_s, \\ 0, & \text{otherwise}, \end{cases} \quad t \in \{\text{obs}, 1, \ldots, B\}. \tag{12}$$

Each view is now expressed in units of its own null variation. The second largest effect requires corroboration from at least two views without learned weights or requiring all four to respond equally:

$$S_{\text{RUV}}^{(t)} = 2\text{ndLargest}_j \, Z_j^{(t)}, \quad \tau_\alpha = Q_{1-\alpha}\{S_{\text{RUV}}^{(b)}\}_{b=1}^{B}, \quad E_{\text{RUV}} = \mathbb{I}[\Delta_{\text{RUV}} > 0], \tag{13}$$

where $\Delta_{\text{RUV}} = S_{\text{RUV}}^{\text{obs}} - \tau_\alpha$. The quantile uses order rank $\min\{B, \lceil (B+1)(1-\alpha) \rceil\}$. Every trial is judged against its own boundary before results are aggregated; an average boundary never replaces per-trial decisions. Appendix C gives the complete definitions.

The four views are therefore neither four independent acceptance tests nor four copies of a feature distance. ARS and RS inspect computation and state; NPG and MRD inspect neighborhood identity and relative support. Retained comparison supplies target attribution, while joint calibration supplies the request-specific fluctuation boundary. This division of roles allows different unlearning mechanisms to yield different evidence profiles under the same decision rule. Appendix D establishes the corresponding properties of output-only indistinguishability, retained-control attribution, multi-view corroboration, and permutation-calibrated detection.

6 Experiments

6.1 Experimental Settings

Datasets and models. We pair CIFAR10, CIFAR100, and FaceScrub with ResNet18, ResNet34, and ResNet50, respectively (Krizhevsky & Hinton, 2009; Ng & Winkler, 2014; He et al., 2016). CIFAR10 and CIFAR100 cover increasing category complexity, while FaceScrub extends the evaluation to identity recognition. Additional architectures are evaluated in Section 6.3.

Unlearning settings. Main experiments remove one class or identity for class requests and use stratified sampling with $\rho = 10\%$ for sample requests; other request scales are studied in Appendix G.6. Retraining and SISA provide exact baselines, while Amnesiac, SCRUB, SalUn, and SSD cover distinct approximate mechanisms (Bourtoule et al., 2021; Graves et al., 2021; Kurmanji et al., 2023; Fan et al., 2024; Foster et al., 2024). Each setting contains 100 submissions from ten original models and ten fixed-manifest requests per model.

Verification baselines. We adopt output-based verifiers that cover complementary signals: Accuracy uses predictive accuracy, LiRA performs membership inference, and TAPE measures posterior differences. We also compare with TruVRF, which audits internal model changes through gradients (Carlini et al., 2022; Wang et al., 2025b; Zhou et al., 2025).

Evaluation metrics. Verification accuracy (VA) is the fraction of configured unlearning executions accepted and adversarial submissions rejected under this controlled protocol. ODR deception success rate (DSR) is the fraction of ODR submissions accepted by an output verifier; detection rate is its complement.

Implementation details, repeated-audit aggregation, and baseline calibration are provided in Appendix E.

6.2 Main Results

Table 1: ODR deception success rate (C: class; S: sample). Each point estimate is the success count over 1000 binary audits; uncertainty is the standard error over ten original-model summaries.

	CIFAR10		CIFAR100		FaceScrub	
Verifier	Class	Sample	Class	Sample	Class	Sample
Accuracy	**0.992 ± 0.003**	**0.991 ± 0.003**	**0.988 ± 0.005**	**0.987 ± 0.005**	**0.996 ± 0.003**	**0.993 ± 0.003**
LiRA	0.958 ± 0.015	0.969 ± 0.008	0.948 ± 0.012	0.958 ± 0.011	0.942 ± 0.011	0.965 ± 0.009
TAPE	0.972 ± 0.012	0.978 ± 0.007	0.964 ± 0.008	0.972 ± 0.009	0.968 ± 0.008	0.980 ± 0.008

Output-level deception. We first ask whether a provider can satisfy several output-based verifiers without performing a representation-level parameter update. We submit ODR-Head for class requests and ODR-Gate for sample requests to Accuracy, LiRA, and TAPE on all three datasets. Table 1 shows mean DSRs of 0.970 and 0.977, respectively, with even the lowest rates remaining 0.942 and 0.958. The result spans accuracy, membership, and posterior-difference criteria while preserving retained accuracy: ODR-Head suppresses class accuracy, whereas ODR-Gate replays the high-confidence same-class nonmember behavior produced by sample-level Retraining. These verifiers inspect different statistics, but all derive them from the same provider-controlled response; ODR therefore produces output conformity while leaving the computation that formed the original representation unmodified. Appendix F reports complete prediction behavior, retained utility, and construction cost.

Table 2: Verification accuracy across methods and request levels. Each point estimate is the success count over 1000 binary audits; uncertainty is the standard error over ten original-model summaries.

(a) Class level

Dataset	Verifier	Retraining	SISA	Amnesiac	SCRUB	SalUn	SSD	**ODR**
CIFAR10	Accuracy	0.964 ± 0.010	0.950 ± 0.013	0.888 ± 0.015	0.928 ± 0.024	0.906 ± 0.035	0.860 ± 0.031	**0.008 ± 0.003**
	LiRA	0.954 ± 0.014	0.932 ± 0.013	0.902 ± 0.023	0.910 ± 0.023	0.918 ± 0.016	0.874 ± 0.017	**0.042 ± 0.015**
	TAPE	0.976 ± 0.009	0.962 ± 0.012	0.928 ± 0.014	0.950 ± 0.011	0.936 ± 0.029	0.904 ± 0.018	**0.028 ± 0.012**
	TruVRF	0.948 ± 0.008	0.922 ± 0.027	0.866 ± 0.027	0.900 ± 0.020	0.884 ± 0.017	0.840 ± 0.022	**0.138 ± 0.034**
	RUV	**0.996 ± 0.003**	**0.994 ± 0.003**	**0.986 ± 0.005**	**0.992 ± 0.004**	**0.990 ± 0.005**	**0.984 ± 0.006**	**0.998 ± 0.002**
CIFAR100	Accuracy	0.938 ± 0.026	0.920 ± 0.018	0.852 ± 0.029	0.898 ± 0.031	0.880 ± 0.022	0.832 ± 0.026	**0.012 ± 0.005**
	LiRA	0.928 ± 0.016	0.910 ± 0.027	0.868 ± 0.027	0.878 ± 0.029	0.894 ± 0.012	0.836 ± 0.017	**0.052 ± 0.012**
	TAPE	0.958 ± 0.010	0.930 ± 0.019	0.894 ± 0.023	0.914 ± 0.009	0.910 ± 0.018	0.880 ± 0.032	**0.036 ± 0.008**
	TruVRF	0.924 ± 0.018	0.896 ± 0.020	0.828 ± 0.040	0.866 ± 0.037	0.846 ± 0.030	0.802 ± 0.032	**0.174 ± 0.035**
	RUV	**0.996 ± 0.003**	**0.992 ± 0.004**	**0.982 ± 0.010**	**0.990 ± 0.004**	**0.988 ± 0.005**	**0.980 ± 0.008**	**0.996 ± 0.004**
FaceScrub	Accuracy	0.952 ± 0.016	0.934 ± 0.013	0.882 ± 0.032	0.916 ± 0.015	0.886 ± 0.033	0.854 ± 0.027	**0.004 ± 0.003**
	LiRA	0.944 ± 0.023	0.934 ± 0.022	0.888 ± 0.025	0.902 ± 0.021	0.906 ± 0.020	0.858 ± 0.024	**0.058 ± 0.011**
	TAPE	0.962 ± 0.011	0.960 ± 0.012	0.918 ± 0.022	0.936 ± 0.028	0.938 ± 0.011	0.890 ± 0.014	**0.032 ± 0.008**
	TruVRF	0.946 ± 0.023	0.908 ± 0.018	0.848 ± 0.023	0.882 ± 0.029	0.874 ± 0.021	0.816 ± 0.024	**0.158 ± 0.030**
	RUV	**0.992 ± 0.006**	**0.994 ± 0.006**	**0.984 ± 0.007**	**0.990 ± 0.006**	**0.986 ± 0.006**	**0.982 ± 0.004**	**0.990 ± 0.004**

(b) Sample level

Dataset	Verifier	Retraining	SISA	Amnesiac	SCRUB	SalUn	SSD	**ODR**
CIFAR10	Accuracy	0.931 ± 0.015	0.917 ± 0.024	0.849 ± 0.040	0.893 ± 0.028	0.871 ± 0.020	0.823 ± 0.016	**0.009 ± 0.003**
	LiRA	0.924 ± 0.020	0.901 ± 0.022	0.865 ± 0.018	0.879 ± 0.025	0.884 ± 0.017	0.833 ± 0.027	**0.031 ± 0.008**
	TAPE	0.952 ± 0.013	0.938 ± 0.013	0.899 ± 0.032	0.919 ± 0.026	0.909 ± 0.016	0.868 ± 0.015	**0.022 ± 0.007**
	TruVRF	0.918 ± 0.017	0.884 ± 0.036	0.821 ± 0.027	0.857 ± 0.032	0.847 ± 0.042	0.794 ± 0.033	**0.784 ± 0.029**
	RUV	**0.961 ± 0.003**	**0.957 ± 0.006**	**0.943 ± 0.007**	**0.954 ± 0.003**	**0.950 ± 0.007**	**0.938 ± 0.008**	**0.999 ± 0.001**
CIFAR100	Accuracy	0.904 ± 0.031	0.881 ± 0.020	0.808 ± 0.038	0.856 ± 0.026	0.836 ± 0.026	0.785 ± 0.038	**0.013 ± 0.005**
	LiRA	0.898 ± 0.015	0.872 ± 0.029	0.824 ± 0.023	0.842 ± 0.023	0.853 ± 0.029	0.790 ± 0.033	**0.042 ± 0.011**
	TAPE	0.925 ± 0.022	0.894 ± 0.014	0.855 ± 0.031	0.877 ± 0.019	0.869 ± 0.027	0.838 ± 0.036	**0.028 ± 0.009**
	TruVRF	0.882 ± 0.019	0.855 ± 0.024	0.781 ± 0.046	0.822 ± 0.019	0.799 ± 0.033	0.755 ± 0.042	**0.778 ± 0.035**
	RUV	**0.957 ± 0.005**	**0.952 ± 0.009**	**0.936 ± 0.005**	**0.949 ± 0.005**	**0.946 ± 0.006**	**0.932 ± 0.006**	**0.998 ± 0.002**
FaceScrub	Accuracy	0.920 ± 0.014	0.903 ± 0.019	0.841 ± 0.042	0.881 ± 0.033	0.849 ± 0.035	0.813 ± 0.028	**0.007 ± 0.003**
	LiRA	0.913 ± 0.019	0.899 ± 0.029	0.854 ± 0.017	0.868 ± 0.033	0.867 ± 0.023	0.818 ± 0.033	**0.035 ± 0.009**
	TAPE	0.936 ± 0.013	0.932 ± 0.021	0.884 ± 0.028	0.908 ± 0.023	0.904 ± 0.026	0.853 ± 0.015	**0.020 ± 0.008**
	TruVRF	0.910 ± 0.013	0.865 ± 0.021	0.804 ± 0.039	0.840 ± 0.020	0.831 ± 0.016	0.772 ± 0.035	**0.826 ± 0.031**
	RUV	**0.963 ± 0.003**	**0.959 ± 0.004**	**0.945 ± 0.009**	**0.953 ± 0.004**	**0.948 ± 0.008**	**0.940 ± 0.010**	**0.997 ± 0.002**

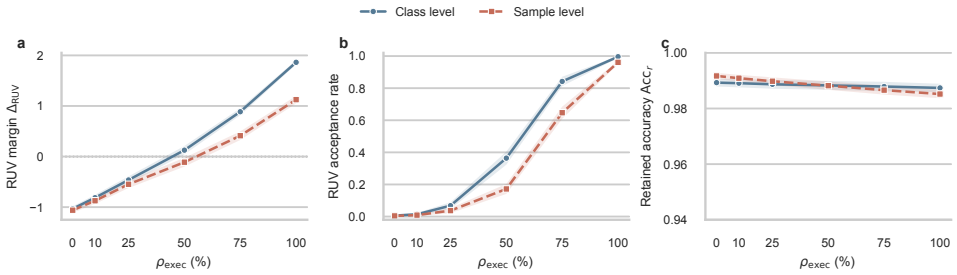

Figure 2: RUV response as the executed fraction of a fixed request increases.

RUV performance. We next test whether RUV verifies exact and approximate unlearning at both request levels while rejecting submissions that only reproduce expected outputs. Table 2 compares all five verifiers on the same model pairs, request manifests, and submissions; unlearning columns report correct acceptance and the ODR column reports correct rejection. RUV leads every method column: its pooled class-level accuracies are 0.995 for Retraining, 0.993 for SISA, and 0.982–0.991 for approximate methods, while the sample-level counterparts are 0.960, 0.956, and 0.937–0.952. Averaged across datasets and methods, RUV reaches 0.989 and 0.949, exceeding TAPE, the strongest comparison verifier, by 5.8 and 5.1 percentage points. RUV also rejects ODR-Head and ODR-Gate at 0.995 and 0.998; the output verifiers reject at most 0.051 and 0.036, while TruVRF reaches 0.157 and 0.796 because its backward sensitivity is recomputed from the submitted model. RUV thus remains effective from broad exact-retraining responses to narrower approximate updates without depending on one unlearning mechanism. Its lower sample-level values reflect weaker, distributed instance responses within shared class structure, while its ODR rejection demonstrates the advantage of auditing the internal computation preceding the output. Retraining and SISA lead because rebuilding without D_u changes the representation chain coherently, while narrower approximate updates remain detectable. Class requests are easier because category removal reorganizes terminal geometry, whereas sample effects remain distributed across nonterminal representations. Appendix G reports the complete margins, evidence profiles, and overhead: on CIFAR10/ResNet18, RUV takes 4.37 seconds and 1684 MB, compared with 91.2 seconds and 2528 MB for TruVRF.

6.3 Ablation Studies

Request completion. To determine whether RUV follows request fulfillment rather than merely recognizing that Retraining was invoked, we fix the CIFAR10/ResNet18 original models, complete request D_u, training procedure, audit inputs, retained controls, and references, and vary only the executed subset $D_u^{(\rho_{\text{exec}})} \subseteq D_u$. RUV always audits the complete D_u, including when the same training procedure excludes no requested sample. Figure 2 shows an ordered response: from zero to complete execution, class-level margin and acceptance rise from -1.032 and 0.006 to 1.863 and 0.996, while their sample-level counterparts rise from -1.061 and 0.004 to 1.125 and 0.961. Acceptance remains only $0.068/0.037$ at 25% execution but reaches $0.842/0.647$ at 75%. Retained accuracy changes by only $0.19/0.65$ percentage points and ends at $0.987/0.985$. The synchronized progression therefore tracks how much of the fixed request is carried out rather than routine identity or indiscriminate utility loss.

Target attribution. We test whether RUV accepts arbitrary model movement or any change on the correct D_u by comparing the standard audit with a variant without matched retained controls and three non-unlearning updates: continued full-data training, unlearning an equally sized disjoint $D_w \subset D_r$ while auditing D_u, and ten epochs of correct-label fine-tuning on D_u. The last control changes target representations while preserving and strengthening their information. Table 3 shows that removing the retained control reduces class/sample VA from $0.991/0.957$ to $0.842/0.774$. Standard RUV cor-

Table 3: Target attribution and non-unlearning controls.

Setting	Class	Sample
Standard audit	**0.991 ± 0.006**	**0.957 ± 0.008**
Benign drift control	0.989 ± 0.014	0.947 ± 0.017
Wrong target unlearning control	0.976 ± 0.035	0.937 ± 0.022
Same target fine-tuning control	0.968 ± 0.028	0.929 ± 0.024
Without retained control	0.842 ± 0.025	0.774 ± 0.024

rectly rejects benign drift at $0.989/0.947$, wrong-target unlearning at $0.976/0.937$, and same-target fine-tuning at $0.968/0.929$; in the last case, forgotten-set accuracy is $0.998/0.997$, confidence is $0.992/0.990$, and retained accuracy remains $0.981/0.986$. Thus, generic drift, an unlearning-shaped update elsewhere, and a target specific change that retains target knowledge are all insufficient: accepted evidence must be attributable to the requested set and consistent with an unlearning response.

Evidence components. To test whether the four views are interchangeable, we use each view alone and remove each from the complete verifier on the same 100 class and 100 sample requests. Table 4 identifies NPG as the strongest individual view, yet it remains 7.5 and 10.9 percentage points below full RUV. Every leave-one-out variant also lowers accuracy; the largest sample-level loss occurs without NPG, while removing MRD still reduces sample VA from 0.957 to 0.932. Activation organization, coordinate displacement, neighbor identity, and manifold support therefore capture successive but nonredundant consequences of an internal update, and no single statistic reproduces their corroborated evidence.

Table 4: Ablation of RUV evidence views.

Variant	Class	Sample
ARS only	0.838 ± 0.022	0.714 ± 0.029
RS only	0.873 ± 0.019	0.782 ± 0.025
NPG only	0.916 ± 0.016	0.848 ± 0.022
MRD only	0.894 ± 0.020	0.702 ± 0.031
without ARS	0.982 ± 0.009	0.941 ± 0.013
without RS	0.975 ± 0.011	0.928 ± 0.015
without NPG	0.961 ± 0.014	0.899 ± 0.019
without MRD	0.969 ± 0.012	0.932 ± 0.016
Full RUV	$\mathbf{0.991 \pm 0.006}$	$\mathbf{0.957 \pm 0.008}$

Evidence aggregation. We isolate the decision rule by holding all evidence scores and permutation samples fixed and replacing only the aggregation operator. Maximum, mean, and minimum aggregation reach class/sample VA of $0.945/0.901$, $0.972/0.931$, and $0.919/0.864$, whereas the second-largest effect reaches $0.991/0.957$ (Table 5). A maximum can accept one anomalous view, a minimum requires every mechanism to affect the full chain, and a mean dilutes localized evidence. The second-largest rule instead requires two corroborating views without fixing their identities or learning weights, matching the heterogeneous propagation patterns observed across unlearning methods.

Table 5: Ablation of evidence aggregation.

Aggregation	Class	Sample
Maximum	0.945 ± 0.014	0.901 ± 0.017
Mean	0.972 ± 0.011	0.931 ± 0.014
Minimum	0.919 ± 0.016	0.864 ± 0.020
Second largest	$\mathbf{0.991 \pm 0.006}$	$\mathbf{0.957 \pm 0.008}$

Architecture robustness. We finally test whether RUV depends on one convolutional design by applying the same request construction, representation selection, retained controls, and joint calibration to seven CNN and Transformer backbones on CIFAR10. Class VA remains within 0.963–0.994 and sample VA within 0.911–0.960; ViT-S/16 reaches $0.978/0.933$, while MobileNetV2 reaches $0.963/0.911$. The consistent results across residual, densely connected, lightweight, and Transformer families show that RUV uses functional representation organization rather than architecture-specific layer identities.

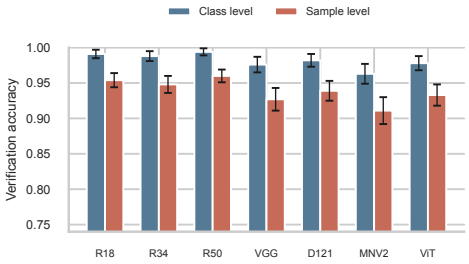

Figure 3: RUV accuracy across model architectures.

7 Conclusion

Machine unlearning verification requires internal evidence, not merely outputs that resemble unlearning. ODR exposes this gap, while RUV rethinks the audit boundary by tracing activation routes, feature states, neighborhoods, and manifold support. Retained controls and joint calibration isolate target-attributed responses from ordinary drift. Across methods, requests, datasets, and architectures, RUV consistently verifies exact and approximate unlearning executions and rejects output deception. Its response further tracks request completion and rejects generic, wrong-target, and knowledge-preserving target updates. These results establish forward representation auditing as a practical basis for machine unlearning verification and motivate its extension to generative and multimodal models.

AI USE STATEMENT

Generative AI tools assisted with manuscript language editing, LaTeX formatting, translation, and code refinement. The authors are responsible for the final manuscript and accompanying artifacts.

REPRODUCIBILITY STATEMENT

The audit definitions, representation operators, and joint calibration procedure are specified in Sections 5 and C. Their theoretical properties are established in Appendix D. Appendix E details the data partitions, request manifests, model pairing, hyperparameters, and evaluation protocol. Supplementary ODR and RUV experiments are provided in Appendices F and G. An anonymized reference implementation and runnable examples are available at `https://anonymous.4open.science/r/iclr_veriun_core-E76F`.

REFERENCES

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6541–6549, 2017. doi: 10.1109/CVPR.2017.354.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems*, volume 14, 2001.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. doi: 10.1109/TPAMI.2013.50.

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*, pp. 141–159. IEEE, 2021. doi: 10.1109/SP40001.2021.00019.

Jonathan Brophy and Daniel Lowd. Machine unlearning for random forests. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1092–1104. PMLR, 2021.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE Symposium on Security and Privacy*, pp. 463–480. IEEE, 2015. doi: 10.1109/SP.2015.35.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy*, pp. 1897–1914. IEEE, 2022. doi: 10.1109/SP46214.2022.9833649.

Eli Chien, Haoyu Wang, Ziang Chen, and Pan Li. Certified machine unlearning via noisy stochastic gradient descent. In *Advances in Neural Information Processing Systems*, volume 37, pp. 38852–38887, 2024a. doi: 10.52202/079017-1228.

Eli Chien, Haoyu Wang, Ziang Chen, and Pan Li. Langevin unlearning: A new perspective of noisy gradient descent for machine unlearning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 79666–79703, 2024b. doi: 10.52202/079017-2530.

Vikram S. Chundawat, Ayush K. Tarun, Murari Mandal, and Mohan Kankanhalli. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6):7210–7217, 2023. doi: 10.1609/aaai.v37i6.25879.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

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

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. SalUn: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *International Conference on Learning Representations*, 2024.

Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(11):12043–12051, 2024. doi: 10.1609/aaai.v38i11.29092.

Xiangshan Gao, Xingjun Ma, Jingyi Wang, Youcheng Sun, Bo Li, Shouling Ji, Peng Cheng, and Jiming Chen. VeriFi: Towards verifiable federated unlearning. *IEEE Transactions on Dependable and Secure Computing*, 21(6):5720–5736, 2024. doi: 10.1109/TDSC.2024.3382321.

Antonio A. Ginart, Melody Y. Guan, Gregory Valiant, and James Zou. Making AI forget you: Data deletion in machine learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9301–9309, 2020. doi: 10.1109/CVPR42600.2020.00932.

Laura Graves, Vineel Nagisetty, and Vijay Ganesh. Amnesiac machine learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13):11516–11524, 2021. doi: 10.1609/aaai.v35i13.17371.

Yu Guo, Yu Zhao, Saihui Hou, Cong Wang, and Xiaohua Jia. Verifying in the dark: Verifiable machine unlearning by using invisible backdoor triggers. *IEEE Transactions on Information Forensics and Security*, 19:708–721, 2024. doi: 10.1109/TIFS.2023.3328269.

Mengde Han, Tianqing Zhu, Lefeng Zhang, Huan Huo, and Wanlei Zhou. Vertical federated unlearning via backdoor certification. *IEEE Transactions on Services Computing*, 18(2):1110–1123, 2025. doi: 10.1109/TSC.2025.3536312.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Hongsheng Hu, Zoran Salcic, Gillian Dobbie, Jinjun Chen, Lichao Sun, and Xuyun Zhang. Membership inference via backdooring. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pp. 3832–3838, 2022. doi: 10.24963/ijcai.2022/532.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017. doi: 10.1109/CVPR.2017.243.

Thanh Trung Huynh, Trong Bang Nguyen, Thanh Toan Nguyen, Phi Le Nguyen, Hongzhi Yin, Quoc Viet Hung Nguyen, and Thanh Tam Nguyen. Certified unlearning for federated recommendation. *ACM Transactions on Information Systems*, 43(2):1–29, 2025. doi: 10.1145/3706419.

Dongjae Jeon, Wonje Jeung, Taeheon Kim, Albert No, and Jonghyun Choi. An information theoretic evaluation metric for strong unlearning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(26):22173–22181, 2026. doi: 10.1609/aaai.v40i26.39373.

Heinrich Jiang, Been Kim, Melody Guan, and Maya Gupta. To trust or not to trust a classifier. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 2019.

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

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 1957–1987, 2023. doi: 10.52202/075280-0095.

Xinyang Lu, Xinyuan Niu, Gregory Kang Ruey Lau, Bui Thi Cam Nhung, Rachael Hwee Ling Sim, Fanyu Wen, Chuan-Sheng Foo, See-Kiong Ng, and Bryan Kian Hsiang Low. WaterDrum: Watermarking for data-centric unlearning metric. *arXiv preprint arXiv:2505.05064*, 2025.

Zhaobo Lu, Yilei Wang, Qingzhe Lv, Minghao Zhao, and Tiancai Liang. FP^2-MIA: A membership inference attack free of posterior probability in machine unlearning. In *Provable and Practical Security*, pp. 167–175. Springer, 2022. doi: 10.1007/978-3-031-20917-8_12.

Pratyush Maini, Michael Curtis Mozer, Hanie Sedghi, Zachary Chase Lipton, J. Zico Kolter, and Chiyuan Zhang. Can neural network memorization be localized? In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23536–23557. PMLR, 2023.

Siqiao Mu and Diego Klabjan. Rewind-to-delete: Certified machine unlearning for nonconvex functions. In *Advances in Neural Information Processing Systems*, volume 38, pp. 127404–127440, 2025. doi: 10.52202/085713-3832.

Hong-Wei Ng and Stefan Winkler. A data-driven approach to cleaning large face datasets. In *2014 IEEE International Conference on Image Processing*, pp. 343–347. IEEE, 2014. doi: 10.1109/ICIP.2014.7025068.

Quoc Phong Nguyen, Ryutaro Oikawa, Dinil Mon Divakaran, Mun Choon Chan, and Bryan Kian Hsiang Low. Markov chain monte carlo-based machine unlearning. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, pp. 351–363. ACM, 2022. doi: 10.1145/3488932.3517406.

Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663, 2020. doi: 10.1073/pnas.2015509117.

Joseph P. Romano and Michael Wolf. Exact and approximate stepdown methods for multiple hypothesis testing. *Journal of the American Statistical Association*, 100(469):94–108, 2005. doi: 10.1198/016214504000000539.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018. doi: 10.1109/CVPR.2018.00474.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy*, pp. 3–18. IEEE, 2017. doi: 10.1109/SP.2017.41.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

David M. Sommer, Liwei Song, Sameer Wagh, and Prateek Mittal. Athena: Probabilistic verification of machine unlearning. *Proceedings on Privacy Enhancing Technologies*, 2022(3):268–290, 2022. doi: 10.56553/popets-2022-0072.

Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium*, pp. 4007–4022. USENIX Association, 2022.

Àlex Pujol Vidal, Anders S. Johansen, Mohammad N. S. Jahromi, Sergio Escalera, Kamal Nasrollahi, and Thomas B. Moeslund. Verifying machine unlearning with explainable AI. In *Pattern Recognition. ICPR 2024 International Workshops and Challenges*, pp. 458–473. Springer, 2025. doi: 10.1007/978-3-031-88223-4_32.

Cheng-Long Wang, Qi Li, Zihang Xiang, Yinzhi Cao, and Di Wang. Towards lifecycle unlearning commitment management: Measuring sample-level unlearning completeness. In *34th USENIX Security Symposium*, pp. 6481–6500. USENIX Association, 2025a.

Weiqi Wang, Zhiyi Tian, An Liu, and Shui Yu. TAPE: Tailored posterior difference for auditing of machine unlearning. In *Proceedings of the ACM on Web Conference 2025*, pp. 3061–3072. ACM, 2025b. doi: 10.1145/3696410.3714875.

Weiqi Wang, Chenhan Zhang, Zhiyi Tian, and Shui Yu. SMS: Self-supervised model seeding for verification of machine unlearning. *IEEE Transactions on Dependable and Secure Computing*, 23(1):1219–1231, 2026. doi: 10.1109/TDSC.2025.3615615.

Jiasi Weng, Shenglong Yao, Yuefeng Du, Junjie Huang, Jian Weng, and Cong Wang. Proof of unlearning: Definitions and instantiation. *IEEE Transactions on Information Forensics and Security*, 19:3309–3323, 2024. doi: 10.1109/TIFS.2024.3358993.

Yinjun Wu, Edgar Dobriban, and Susan Davidson. DeltaGrad: Rapid retraining of machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10355–10366. PMLR, 2020.

Heng Xu, Tianqing Zhu, Lefeng Zhang, and Wanlei Zhou. Really unlearned? verifying machine unlearning via influential sample pairs. *IEEE Transactions on Dependable and Secure Computing*, 23(1):1671–1686, 2026. doi: 10.1109/TDSC.2025.3620308.

Haonan Yan, Xiaoguang Li, Ziyao Guo, Hui Li, Fenghua Li, and Xiaodong Lin. ARCANE: An efficient architecture for exact machine unlearning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pp. 4006–4013, 2022. doi: 10.24963/ijcai.2022/556.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, volume 27, 2014.

Binchi Zhang, Zihan Chen, Cong Shen, and Jundong Li. Verification of machine unlearning is fragile. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 58717–58738. PMLR, 2024.

Xiaoyu Zhang, Chenyang Zhang, Jian Lou, Kai Wu, Zilong Wang, and Xiaofeng Chen. DuplexGuard: Safeguarding deletion right in machine unlearning via duplex watermarking. *IEEE Transactions on Dependable and Secure Computing*, 22(2):1717–1731, 2025. doi: 10.1109/TDSC.2024.3456811.

Chunyi Zhou, Yansong Gao, Anmin Fu, Kai Chen, Zhi Zhang, Minhui Xue, Zhiyang Dai, Shouling Ji, and Yuqing Zhang. TruVRF: Toward triple-granularity verification on machine unlearning. *IEEE Transactions on Information Forensics and Security*, 20:4844–4859, 2025. doi: 10.1109/TIFS.2025.3565991.

APPENDIX CONTENTS

BACKGROUND, METHODS, AND THEORY

A Additional Related Work **15**
A.1 Machine Unlearning Methods . . . 15
A.2 Verification of Machine Unlearning . . . 15

B ODR Implementation Details **16**
B.1 ODR-Head Reproducibility . . . 16
B.2 ODR-Gate Reproducibility . . . 17

C RUV Definitions and Calibration Details **17**
C.1 Representation Registry and Pooling . . . 17
C.2 Evidence Implementation and Cross-block Aggregation . . . 18
C.3 Target-specific Evidence and Joint Calibration . . . 19
C.4 Resistance and Cost of Internal Forgery . . . 20

D Theoretical Analysis of ODR and RUV **20**
D.1 Why ODR Works: Output-Transcript Indistinguishability . . . 21
D.2 Why RUV Works: Target Attribution . . . 22
D.3 Why RUV Works: Corroboration and Calibration . . . 23
D.4 Why RUV Works: Detectability . . . 24

REPRODUCIBILITY DETAILS

E Experimental Details **25**
E.1 Requests and Reporting Protocol . . . 25
E.2 Data and Model Configuration . . . 26
E.3 Unlearning and Verification Configurations . . . 26
E.4 RUV Settings . . . 27

SUPPLEMENTARY EXPERIMENTS

F Supplementary ODR Experiments **27**
F.1 Prediction Behavior and Retained Utility . . . 27
F.2 Construction and Deployment Cost . . . 28

G Supplementary RUV Experiments **28**
G.1 Decision Margins and Representation Evidence . . . 28
G.2 Verification Overhead . . . 29
G.3 Representation Selection . . . 30
G.4 Joint Calibration . . . 30
G.5 Parameter Sensitivity . . . 30
G.6 Request-Scale Robustness . . . 31

A Additional Related Work

A.1 Machine Unlearning Methods

Machine unlearning methods are commonly divided by their guarantees into exact and approximate unlearning. Exact unlearning requires the resulting model to match, or be distributionally equivalent to, the result of applying the prescribed training procedure to retained data. Full retraining provides the direct reference, while dependency isolation, partitioned training, state caching, and ensemble architectures reduce the portion of training that must be recomputed (Cao & Yang, 2015; Ginart et al., 2019; Bourtoule et al., 2021; Brophy & Lowd, 2021; Yan et al., 2022). SISA, for example, confines the effect of a request to affected shards and slices, then reconstructs the aggregate model from components trained without the requested data (Bourtoule et al., 2021). Exact methods provide a clear counterfactual target but inherit the storage, training, or architectural costs of maintaining that reconstructable training structure.

Approximate unlearning relaxes retraining equivalence and modifies an existing model at lower cost. Its mechanisms include, but are not limited to, retained data fine tuning, gradient ascent or target relabeling, teacher–student distillation, recorded update reversal, and selective parameter suppression based on influence, Fisher information, or saliency (Wu et al., 2020; Golatkar et al., 2020; Chundawat et al., 2023; Graves et al., 2021; Kurmanji et al., 2023; Fan et al., 2024; Foster et al., 2024). These strategies intervene at different scopes. Some replay or negate historical updates, some optimize the full model against competing forget and retain objectives, and others restrict changes to selected parameters. Consequently, successful executions need not share a parameter trajectory, output distribution, or layerwise response. This heterogeneity makes a verifier tied to one update rule unsuitable as a general post hoc mechanism.

Independently of the guarantee, requests differ in their target content. A class-level request removes all training samples associated with a class or identity, whereas a sample-level request removes designated instances while retaining other samples from the same class. Class removal can reorganize the terminal decision geometry because an entire semantic group disappears from training. Sample removal acts within shared class structure, so its effect is typically more localized and distributed. Verification must therefore support both heterogeneous update mechanisms and distinct response scales rather than prescribe one universal signature.

A.2 Verification of Machine Unlearning

Process-bound evidence. Some verifiers arrange evidence before or during learning. Active methods insert backdoor triggers, watermarks, canaries, model seeds, or influential sample relations whose responses can later be queried (Sommer et al., 2022; Guo et al., 2024; Hu et al., 2022; Gao et al., 2024; Wang et al., 2026; Xu et al., 2026; Zhang et al., 2025; Han et al., 2025; Lu et al., 2025). Certified unlearning, replay, and proof systems instead bind an audit to a prescribed optimizer, retained process records, or provider-generated proof artifacts (Chien et al., 2024a;b; Nguyen et al., 2022; Thudi et al., 2022; Weng et al., 2024; Eisenhofer et al., 2025; Huynh et al., 2025; Mu & Klabjan, 2025). These mechanisms connect a claim directly to the learning process and can provide strong evidence under their intended assumptions. They also require the verification arrangement to be anticipated, the training algorithm to remain compatible, or the provider to retain and expose the required artifacts. They are therefore most natural when verification is designed into the model lifecycle, rather than introduced only after an independently trained model receives an unlearning request.

Output-based post hoc evidence. Post hoc output methods audit an already submitted model through its prediction interface. Forgotten-set accuracy measures the most direct task behavior; membership inference compares forgotten samples with members or nonmembers; confidence, entropy, and loss expose changes in predictive certainty; and posterior-difference methods compare the responses of original and unlearned models (Shokri et al., 2017; Lu et al., 2022; Carlini et al., 2022; Wang et al., 2025b;a; Jeon et al., 2026; Vidal et al., 2025). These methods differ in statistic and calibration, but observe the same boundary: the final response returned by the audited provider. This common interface explains their broad deployability and their shared exposure to response manipulation. A provider can calibrate confidence, smooth posteriors, or return query-specific responses without performing the internal update that the output is taken to represent. ODR makes this shared

premise explicit by constructing accepted output transcripts while introducing no representation-level parameter update. The issue is therefore not confined to one accuracy, membership, or posterior statistic; changing the statistic does not change who controls its evidence source. This observation is also consistent with prior evidence that verification can become fragile under adaptive behavior by the audited party (Zhang et al., 2024).

Internal evidence and representation analysis. Internal verification asks whether the submitted model itself responds to the request. TruVRF uses gradient sensitivity to measure how an unlearned model responds to forgotten data (Zhou et al., 2025). It moves beyond final predictions, but requires white-box parameters and backward computation. Moreover, a cross-entropy gradient contains the posterior residual $p - e_y$, coupling the measured sensitivity to the model output used to construct the audit loss. It therefore represents a substantially different access and computation point from a forward representation audit.

Outside machine unlearning verification, representation analysis has developed tools for identifying semantic units, comparing learned spaces, and characterizing representation geometry (Bau et al., 2017; Kornblith et al., 2019; Bengio et al., 2013; Belkin & Niyogi, 2001). These studies establish that hidden states expose how a model organizes information and how that organization changes across models. They motivate moving an unlearning audit from the final prediction to the forward computation that produces it. They do not, however, define a verifier for a specified unlearning request. A raw activation, similarity, or distance statistic does not distinguish a response concentrated on D_u from ordinary model-wide drift, nor does one statistic provide a universal signature for the heterogeneous unlearning mechanisms described above.

From representation observation to verification. Post hoc verification necessarily reasons from observable consequences because an external verifier generally lacks both the causal contribution of D_u and an independently available counterfactual model trained without it. Existing methods instantiate this operational evidence at different boundaries: prediction auditing uses task behavior and posteriors, membership inference uses membership signals, active methods use planted responses, and TruVRF uses backward gradient sensitivity (Shokri et al., 2017; Wang et al., 2025b; Sommer et al., 2022; Gao et al., 2024; Zhou et al., 2025). The central design question is therefore not whether an observable statistic changes, but whether that change can be attributed to the requested data and converted into a calibrated decision.

RUV performs this conversion at the representation boundary. First, it evaluates the same forgotten and retained inputs in the original and unlearned models and uses matched retained controls to separate a response concentrated on D_u from model-wide drift. Second, it follows complementary stages of representation formation rather than treating one distance as a universal signature. Third, joint permutation calibration preserves dependence among the views and adapts the decision boundary to the model, data, and request under audit. RUV is thus not merely a descriptive comparison of hidden spaces: it turns forward representation responses into target-attributed, corroborated, and request-calibrated evidence. Unlike output statistics, this evidence cannot be reproduced by replacing only the final response; unlike gradient sensitivity, it requires no backward execution. RUV thereby provides an algorithm-agnostic post hoc verifier between provider-controlled output transcripts and parameter-level backward inspection.

B ODR Implementation Details

Section 4 defines both ODR constructions. This appendix records only the implementation choices needed to reproduce them; optimization and response equations are not repeated.

B.1 ODR-Head Reproducibility

ODR-Head implements Eqs. (1)–(2) with a detached copy of the original model as the teacher. The teacher posterior used to form $\eta(x)$ is therefore fixed throughout optimization rather than recomputed from the changing head. All backbone parameters, BatchNorm buffers, and stochastic inference state are frozen, and the backbone remains in evaluation mode. Only θ_{head} receives gradients. Forgotten and retained minibatches are sampled independently for the two loss terms. Consequently, ODR-Head introduces no representation-level parameter update; only its final projection is changed.

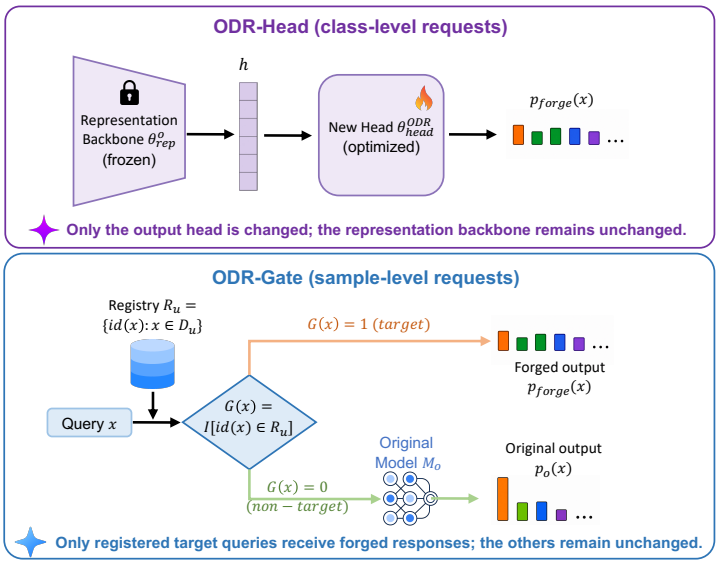

Figure 4: ODR-Head and ODR-Gate construct output-level unlearning responses without a representation-level parameter update.

B.2 ODR-Gate Reproducibility

ODR-Gate implements Eq. (3) without modifying model parameters. The request manifest supplies the immutable registry $\mathcal{R}_u = \{\mathrm{id}(x) : x \in D_u\}$. A single forward pass over the held-out nonmember set constructs the class-indexed bank

$$\mathcal{B}_c = \{p_o(x_j) : (x_j, y_j) \in D_{\mathrm{aux}},\ y_j = c\}\,. \tag{14}$$

Registered targets are sorted by stable identifier and assigned same-class posteriors under a seed-controlled permutation. The assignment cycles through $\mathcal{B}_{y_i}$ when a class contains more registered targets than available nonmembers. This deterministic replay preserves the correctness, confidence, entropy, and class competition of unseen same-class samples without requiring a retrained oracle or an attack-specific hyperparameter.

Construction requires one forward pass over D_{aux} and one stored posterior per registered target. Deployment adds one registry lookup and, for a match, one posterior retrieval per query. These operations determine the costs reported in Appendix F.

C RUV Definitions and Calibration Details

The main paper defines the four evidence views and final RUV decision. This appendix specifies the architecture registry, metric implementation, cross-block aggregation, and finite-sample calibration needed for reproduction.

C.1 Representation Registry and Pooling

Let $\mathcal{L} = (l_1, \ldots, l_{L-1}, l_{\mathrm{pre}})$ contain the ordered outputs of representation-bearing backbone blocks. A block is a functionally distinct architecture-native module, such as a residual, convolutional, dense, inverted residual, or Transformer encoder block. Primitive normalization, activation, pooling, and classifier modules are not registered separately. RUV uses $\mathcal{L}_{\mathrm{class}} = \{l_{\mathrm{pre}}\}$ and $\mathcal{L}_{\mathrm{sample}} = \mathcal{L} \setminus \{l_{\mathrm{pre}}\}$, as in Eq. (5).

For raw block output $h^{(l)}(x; \mathcal{M})$, the common audit vector is

$$z^{(l)}(x; \mathcal{M}) = \mathrm{vec}\Big(\mathcal{P}_l[h^{(l)}(x; \mathcal{M})]\Big)\,, \tag{15}$$

where $\mathcal{P}_l$ is fixed before auditing and shared by $\mathcal{M}_o$ and $\mathcal{M}_u$. CNN blocks are assigned to S architecture-native spatial stages in forward order. If block l belongs to stage $s(l)$ and has spatial

size $H_l \times W_l$, pooling is

$$\bar{g}_{s(l)} = \begin{cases} 4, & s(l) \le \lceil S/2 \rceil,\ s(l) < S, \\ 2, & \lceil S/2 \rceil < s(l) < S, \\ 1, & s(l) = S, \end{cases} \qquad g_l = \min\{\bar{g}_{s(l)}, H_l, W_l\}, \qquad \mathcal{P}_l[h] = \mathrm{AdaptiveAvgPool}_{g_l \times g_l}(h). \tag{16}$$

This retains coarse spatial organization in early stages without upsampling. Transformer blocks use the designated summary token, and l_{pre} uses the native classifier input.

Vector geometry uses cosine distance

$$d_{\cos}(a, b) = 1 - \frac{a^\top b}{\max(\|a\|_2, \epsilon_d)\max(\|b\|_2, \epsilon_d)}, \tag{17}$$

whereas activation-coordinate and neighbor-identity comparisons use Jaccard overlap. The different metrics reflect different mathematical objects rather than alternative choices for one statistic.

C.2 Evidence Implementation and Cross-block Aggregation

Every evidence view is computed per sample before group averaging. ARS and RS are computed on corresponding coordinates of registered block pairs.

ARS. For block l, define the dominant and thresholded activation sets

$$A_q^{(l)}(x; \mathcal{M}) = \mathrm{Top}_{q_{\mathrm{top}}}\left(|z^{(l)}(x; \mathcal{M})|\right), \qquad B_\gamma^{(l)}(x; \mathcal{M}) = \{m : z_m^{(l)}(x; \mathcal{M}) > \gamma_{\mathrm{act}}\}. \tag{18}$$

Jaccard similarity is $J(A, B) = |A \cap B|/|A \cup B|$, with $J(\varnothing, \varnothing) = 1$. Substitution of the original and unlearned sets into Eq. (6) yields the ARS score. The weight ω_b balances threshold crossings against changes among dominant coordinates.

RS. RS is evaluated directly by Eq. (7). Independent vector normalization makes the implementation identical to the cosine expression in the main paper and prevents block scale from dominating the displacement.

NPG. For reference representations $\mathcal{Z}_{\mathrm{ref}}^{(l)}(\mathcal{M}) = \{z^{(l)}(x'; \mathcal{M}) : x' \in D_{\mathrm{ref}}\}$, define

$$\mathcal{N}_{k_{\mathrm{NPG}}}^{(l)}(x; \mathcal{M}, D_{\mathrm{ref}}) = \mathrm{kNN}_{k_{\mathrm{NPG}}}^{d_{\cos}}\left(z^{(l)}(x; \mathcal{M}), \mathcal{Z}_{\mathrm{ref}}^{(l)}(\mathcal{M})\right). \tag{19}$$

Equation (8) compares the identities in the original and unlearned neighbor sets. Reference identities stay fixed while their representations are recomputed; retrieval distances themselves are not averaged.

MRD. Let D_{ref}^y contain label-compatible reference samples and let $\mathcal{R}_k^{(l)}(x; \mathcal{M})$ be the k_{MRD} nearest elements of that set under $d_{\cos}$. Their mean affinity is

$$\rho_k^{(l)}(x; \mathcal{M}) = \frac{1}{|\mathcal{R}_k^{(l)}(x; \mathcal{M})|} \sum_{x' \in \mathcal{R}_k^{(l)}(x; \mathcal{M})} \mathrm{sim}_{\cos}\left(z^{(l)}(x; \mathcal{M}), z^{(l)}(x'; \mathcal{M})\right). \tag{20}$$

Raw affinity is calibrated against $\mathcal{C}_k^{(l)}(\mathcal{M}) = \{\rho_k^{(l)}(x'; \mathcal{M}) : x' \in D_{\mathrm{ref}}\}$, excluding self matches. The percentile rank is

$$r_k^{(l)}(x; \mathcal{M}) = \frac{1}{|\mathcal{C}_k^{(l)}(\mathcal{M})|} \sum_{c \in \mathcal{C}_k^{(l)}(\mathcal{M})} \mathbb{I}[c \le \rho_k^{(l)}(x; \mathcal{M})]. \tag{21}$$

Equation (10) takes the rank drop between models. Sample audits use retained references; class audits use class-compatible held-out references because the target class is absent from D_r. This distinguishes relative support loss from the identity change measured by NPG (Jiang et al., 2018).

Block scores have architecture-dependent scales. RUV therefore standardizes each block over the pooled, group-label-free audit set $\mathcal{X} = D_u \cup D'_r$ before aggregation:

$$\begin{aligned} \mu_{j,l}^{\text{pool}} &= \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} s_j^{(l)}(x), \\ \sigma_{j,l}^{\text{pool}} &= \sqrt{\frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \left(s_j^{(l)}(x) - \mu_{j,l}^{\text{pool}} \right)^2}, \\ S_j^{(g)}(x) &= \frac{1}{|\mathcal{L}_g|} \sum_{l \in \mathcal{L}_g} \frac{s_j^{(l)}(x) - \mu_{j,l}^{\text{pool}}}{\sigma_{j,l}^{\text{pool}} + \epsilon_s}. \end{aligned} \tag{22}$$

Because the normalization does not use forgotten or retained group labels, it cannot itself create group separation. We write $S_j(x) = S_j^{(g)}(x)$ for the request granularity under audit.

C.3 Target-specific Evidence and Joint Calibration

For view j, the observed target-specific gap is

$$G_j^{\text{obs}} = \frac{1}{|D_u|} \sum_{x \in D_u} S_j(x) - \frac{1}{|D'_r|} \sum_{x \in D'_r} S_j(x). \tag{23}$$

All views are oriented so that larger values indicate a stronger unlearning response. Joint permutation estimates how large this gap can become under a request-specific null. For permutation b, audit labels are reassigned while group sizes and matching strata are preserved, producing pseudo groups $\widetilde{D}_u^{(b)}$ and $\widetilde{D}_r^{(b)}$ and gap

$$G_j^{(b)} = \frac{1}{|\widetilde{D}_u^{(b)}|} \sum_{x \in \widetilde{D}_u^{(b)}} S_j(x) - \frac{1}{|\widetilde{D}_r^{(b)}|} \sum_{x \in \widetilde{D}_r^{(b)}} S_j(x). \tag{24}$$

The samples, task labels, outputs, and four-view vectors remain fixed. The same reassignment is applied to every view, paired model representations and all augmented views remain in one sample block, sample audits are stratified by class, and class audits use size-preserving pseudo-target assignments.

From B permutations, define

$$\begin{aligned} \mu_j^{(0)} &= \frac{1}{B} \sum_{b=1}^{B} G_j^{(b)}, \\ \sigma_j^{(0)} &= \sqrt{\frac{1}{B-1} \sum_{b=1}^{B} \left(G_j^{(b)} - \mu_j^{(0)} \right)^2}, \\ Z_j^{(t)} &= \begin{cases} (G_j^{(t)} - \mu_j^{(0)})/\sigma_j^{(0)}, & \sigma_j^{(0)} > \epsilon_s, \\ 0, & \text{otherwise}, \end{cases} \quad t \in \{\text{obs}, 1, \ldots, B\}. \end{aligned} \tag{25}$$

A degenerate null contributes zero evidence. Applying the second-largest rule to the four standardized views gives $S_{\text{RUV}}^{(t)}$ as in Eq. (13). Its finite-sample fluctuation boundary is

$$\tau_\alpha = Q_{1-\alpha}\left(S_{\text{RUV}}^{(1)}, \ldots, S_{\text{RUV}}^{(B)} \right), \tag{26}$$

where $Q_{1-\alpha}$ uses order rank $\min\{B, \lceil (B+1)(1-\alpha) \rceil\}$. The calibrated margin and decision are

$$\Delta_{\text{RUV}} = S_{\text{RUV}}^{\text{obs}} - \tau_\alpha, \qquad E_{\text{RUV}} = \mathbb{I}[\Delta_{\text{RUV}} > 0]. \tag{27}$$

Using common permutations preserves empirical dependence among the four views (Romano & Wolf, 2005). Across repeated trials, each trial is first judged against its own boundary; reported verification rates average these binary decisions rather than comparing an average score with an average threshold.

C.4 Resistance and Cost of Internal Forgery

RUV is executed by the verifier after the provider submits a model. The matched retained subset, reference subset, audit randomization, per-view evidence, and request-specific decision boundary remain verifier-side and are not exposed before submission. Unlike an output verifier, RUV therefore provides neither a compliant response that can be replayed nor a target unlearned representation toward which the submitted model can be directly aligned. Moving from output substitution to internal forgery changes the attack from replacing one response to modifying the submitted model itself.

Proposition C.1 (Exclusion of representation-preserving submissions)**.** *Suppose a submitted strategy induces no target-specific response in the audited representation blocks, so the joint per-sample evidence vectors are exchangeable between D_u and D'_r under the matched audit design. Then*

$$\Pr[E_{\mathrm{RUV}} = 1 \mid H_0] \le \alpha. \tag{28}$$

In particular, an output-only strategy that introduces no representation-level parameter update remains in this null class under the matched design.

Proof. The strategy supplies no request-attributed component $t_j(x)$ in Eq. (38). Under conditional exchangeability, its observed assignment follows the joint permutation null, and Theorem D.6 gives Eq. (28). The result applies to the composed RUV statistic and therefore accounts for dependence among the four views. □

Let a denote a provider strategy, $\mathcal{M}_a$ its submitted model, $U(a)$ its retained utility, and $C(a)$ its implementation cost. For a fixed audit realization, define the utility-preserving acceptance sets

$$\begin{aligned} \mathcal{A}_{\mathrm{out}} &= \{a : V_{\mathrm{out}}(a) = 1,\ U(a) \ge u_{\min}\}, \\ \mathcal{A}_{\mathrm{RUV}} &= \{a : S^{\mathrm{obs}}_{\mathrm{RUV}}(a) > \tau_\alpha(a),\ U(a) \ge u_{\min}\}, \\ \mathcal{A}_{\mathrm{joint}} &= \mathcal{A}_{\mathrm{out}} \cap \mathcal{A}_{\mathrm{RUV}} \subseteq \mathcal{A}_{\mathrm{out}}. \end{aligned} \tag{29}$$

Consequently,

$$\inf_{a \in \mathcal{A}_{\mathrm{joint}}} C(a) \ge \inf_{a \in \mathcal{A}_{\mathrm{out}}} C(a). \tag{30}$$

The added constraint is operationally substantial. Output deception changes one exposed response while leaving the model untouched. Passing RUV additionally requires the submitted model to produce a response concentrated on D_u relative to D'_r in at least two calibrated views. The model modification must coordinate activation routes, feature states, neighbor identities, and manifold support while preserving retained utility and the target information that the provider intends to conceal.

The direct way to obtain a reliable internal target is to first produce a compliant unlearned reference $\mathcal{M}^\star_u$ and align the submitted model with its representations. If C_{unl} is the cost of that execution, then

$$C_{\mathrm{ref}}(a) = C_{\mathrm{unl}} + C_{\mathrm{align}}(a) + C_{\mathrm{keep}}(a) \ge C_{\mathrm{unl}}, \tag{31}$$

where C_{align} coordinates the submitted model with the reference evidence and C_{keep} preserves the information removed by the reference execution. Constructing the target therefore already incurs the cost of unlearning, after which additional model-level optimization is needed.

RUV thus removes the low-cost interface shortcut exploited by ODR. A provider cannot pass by substituting a posterior; it must coordinate target-attributed changes across the model's internal computation. Reference-guided construction offers no economic advantage over submitting the unlearned reference itself.

D Theoretical Analysis of ODR and RUV

This section answers two questions central to our design. First, why can ODR systematically pass output-based verification without changing the internal representation state? Second, why can RUV identify unlearning from forward representations despite differences among unlearning algorithms? The first answer follows from output-transcript indistinguishability. The second follows from target attribution, multi-view corroboration, request-adaptive calibration, and the detectability of a corroborated response. We use the evidence set $\mathcal{E} = \{\mathrm{ARS}, \mathrm{RS}, \mathrm{NPG}, \mathrm{MRD}\}$ and retain the definitions of $S_j(x)$, G^{obs}_j, $Z^{(t)}_j$, $S^{(t)}_{\mathrm{RUV}}$, τ_α, and E_{RUV} from Appendix C.

D.1 WHY ODR WORKS: OUTPUT-TRANSCRIPT INDISTINGUISHABILITY

The first question is whether changing the output statistic can resolve a forged prediction transcript. The following bound shows that decisions computed from the same transcript inherit its indistinguishability. Consider an output-only verifier V_{out} that issues T adaptive queries. Its query X_t may depend on internal randomness and on the history $H_{t-1} = ((X_1, Y_1), \ldots, (X_{t-1}, Y_{t-1}))$, where $Y_t = p_{\mathcal{M}}(X_t)$ is the response returned by model service $\mathcal{M}$. Include the verifier's random seed R in its complete view:

$$\mathsf{T}_T(\mathcal{M}) = \big(R, (X_1, Y_1), \ldots, (X_T, Y_T)\big). \tag{32}$$

Let P_u^{T} and $P_{\mathrm{ODR}}^{\mathsf{T}}$ be the transcript laws induced by an unlearned model $\mathcal{M}_u$ and an ODR service $\mathcal{M}_{\mathrm{ODR}}$, including all verifier and service randomness. For a possibly randomized verifier, define its acceptance kernel

$$v(\mathbf{t}) = \Pr[V_{\mathrm{out}}(\mathbf{t}) = 1 \mid \mathsf{T}_T = \mathbf{t}], \qquad 0 \le v(\mathbf{t}) \le 1. \tag{33}$$

Proposition D.1 (Output transcript bound)**.** *For every output-only verifier V_{out},*

$$\begin{aligned} \mathrm{Adv}_{\mathrm{out}} &= \left| \int v(\mathbf{t}) \, \mathrm{d}P_u^{\mathsf{T}}(\mathbf{t}) - \int v(\mathbf{t}) \, \mathrm{d}P_{\mathrm{ODR}}^{\mathsf{T}}(\mathbf{t}) \right| \\ &\le \sup_{0 \le f \le 1} \left| \int f \, \mathrm{d}P_u^{\mathsf{T}} - \int f \, \mathrm{d}P_{\mathrm{ODR}}^{\mathsf{T}} \right| \\ &= \mathrm{TV}\left(P_u^{\mathsf{T}}, P_{\mathrm{ODR}}^{\mathsf{T}}\right). \end{aligned} \tag{34}$$

Thus, identical transcript laws imply $\mathrm{Adv}_{\mathrm{out}} = 0$, while an ε total-variation match limits the advantage to at most ε.

Proof. Including R makes the verifier's decision a common measurable function of its complete view under both services. Any additional independent decision randomness can be marginalized into the bounded function v in Eq. (33). Its difference in acceptance probability is therefore the first line of Eq. (34). Because v belongs to the class $\{f : 0 \le f \le 1\}$ over which total variation takes the supremum, the second line follows immediately. For a deterministic verifier, $v = \mathbb{I}_{\mathcal{A}}$ for its acceptance region $\mathcal{A}$, recovering $|P_u^{\mathsf{T}}(\mathcal{A}) - P_{\mathrm{ODR}}^{\mathsf{T}}(\mathcal{A})|$. □

The bound also quantifies how local response matching accumulates over an adaptive interaction. Let $Q_{u,t}(\cdot \mid h_{t-1}, x_t)$ and $Q_{\mathrm{ODR},t}(\cdot \mid h_{t-1}, x_t)$ denote the two conditional response laws at round t, conditional also on the common verifier seed R (suppressed in the notation).

Corollary D.2 (Adaptive composition of response mismatch)**.** *Suppose that, for every common seed, history, and query, with $0 \le \varepsilon_t \le 1$,*

$$\mathrm{TV}(Q_{u,t}(\cdot \mid h_{t-1}, x_t), Q_{\mathrm{ODR},t}(\cdot \mid h_{t-1}, x_t)) \le \varepsilon_t. \tag{35}$$

Then

$$\mathrm{Adv}_{\mathrm{out}} \le \mathrm{TV}(P_u^{\mathsf{T}}, P_{\mathrm{ODR}}^{\mathsf{T}}) \le 1 - \prod_{t=1}^{T}(1 - \varepsilon_t) \le \sum_{t=1}^{T} \varepsilon_t. \tag{36}$$

Proof. Couple the two interactions with the same verifier randomness. As long as the histories agree, the verifier issues the same next query. A maximal coupling of the two conditional response laws then keeps the round-t responses equal with probability at least $1 - \varepsilon_t$. Hence the probability that the entire transcripts remain equal is at least $\prod_{t=1}^{T}(1 - \varepsilon_t)$. The coupling characterization of total variation yields the middle inequality in Eq. (36); the final inequality follows from $1 - \prod_t (1 - \varepsilon_t) \le \sum_t \varepsilon_t$. □

For any statistic ϕ computed only from the transcript, total variation also obeys the data-processing inequality

$$\mathrm{TV}\left(\mathcal{L}(\phi(\mathsf{T}_T(\mathcal{M}_u))), \mathcal{L}(\phi(\mathsf{T}_T(\mathcal{M}_{\mathrm{ODR}})))\right) \le \mathrm{TV}(P_u^{\mathsf{T}}, P_{\mathrm{ODR}}^{\mathsf{T}}). \tag{37}$$

Accuracy, confidence, membership scores, and posterior differences are all such statistics. ODR therefore does not need a separate attack for every output verifier. It controls the common evidence

source from which these criteria are computed. Once the exposed transcript matches the behavior accepted as unlearning, no post-processing of that transcript can recover whether the internal representation state was updated. This establishes why ODR can simultaneously deceive output criteria built from different statistics.

D.2 Why RUV Works: Target Attribution

The next question is which part of a representation response is concentrated on the request. A model update can also move retained samples, so absolute change alone does not isolate the requested effect. Each RUV view produces a per-sample score $S_j(x)$; decompose it as

$$S_j(x) = c_j + b_j(x) + \mathbb{I}[x \in D_u]t_j(x) + \varepsilon_j(x), \qquad j \in \mathcal{E}, \tag{38}$$

where c_j is common model drift, $b_j(x)$ is sample-dependent nuisance drift, $t_j(x)$ is the request-attributed response, and $\varepsilon_j(x)$ is zero-mean sampling noise. Enumerate $D_u = \{x_{u,1}, \ldots, x_{u,n}\}$ and $D'_r = \{x_{r,1}, \ldots, x_{r,n}\}$. Substituting Eq. (38) into the observed gap gives the exact finite-sample identity

$$\begin{aligned} G_j^{\mathrm{obs}} &= \frac{1}{n}\sum_{i=1}^{n} S_j(x_{u,i}) - \frac{1}{n}\sum_{i=1}^{n} S_j(x_{r,i}) \\ &= \underbrace{\frac{1}{n}\sum_{i=1}^{n} t_j(x_{u,i})}_{\bar{t}_{u,j}} \\ &\quad + \underbrace{\left[\frac{1}{n}\sum_{i=1}^{n} b_j(x_{u,i}) - \frac{1}{n}\sum_{i=1}^{n} b_j(x_{r,i})\right]}_{\bar{b}_{u,j} - \bar{b}_{r,j}} \\ &\quad + \underbrace{\left[\frac{1}{n}\sum_{i=1}^{n} \varepsilon_j(x_{u,i}) - \frac{1}{n}\sum_{i=1}^{n} \varepsilon_j(x_{r,i})\right]}_{\bar{\varepsilon}_{u,j} - \bar{\varepsilon}_{r,j}}. \end{aligned} \tag{39}$$

The common component c_j disappears algebraically before any distributional assumption is invoked.

Proposition D.3 (Unbiased target attribution)**.** *Suppose* $|D_u| = |D'_r| = n$*,*

$$\mathbb{E}[\varepsilon_j(x) \mid x \in D_u] = \mathbb{E}[\varepsilon_j(x) \mid x \in D'_r] = 0, \qquad \mathbb{E}_{D_u}[b_j(x)] = \mathbb{E}_{D'_r}[b_j(x)]. \tag{40}$$

Then

$$\mathbb{E}[G_j^{\mathrm{obs}}] = \mathbb{E}_{x \sim D_u}[t_j(x)] \equiv \delta_j. \tag{41}$$

Under the null response $t_j(x) = 0$*, the expected gap is zero.*

Proof. Taking expectations on both sides of Eq. (39) yields

$$\begin{aligned} \mathbb{E}[G_j^{\mathrm{obs}}] = \mathbb{E}_{D_u}[t_j(x)] &+ \mathbb{E}_{D_u}[b_j(x)] - \mathbb{E}_{D'_r}[b_j(x)] \\ &+ \mathbb{E}[\varepsilon_j(x) \mid x \in D_u] - \mathbb{E}[\varepsilon_j(x) \mid x \in D'_r]. \end{aligned} \tag{42}$$

The second line and the nuisance difference vanish by Eq. (40), leaving δ_j. □

The mean identity explains attribution; its uncertainty determines how much evidence an audit needs. Under an independent-sampling model for the scores, the same decomposition provides a finite-sample fluctuation bound. Define

$$R_{u,j}(x) = b_j(x) + t_j(x) + \varepsilon_j(x), \qquad R_{r,j}(x) = b_j(x) + \varepsilon_j(x), \tag{43}$$

with variances $\sigma^2_{u,j}$ and $\sigma^2_{r,j}$.

Proposition D.4 (Variance and concentration of the attributed gap)**.** *Under Eq.* (40)*, if observations are identically distributed and independent within each group and the two sampled group means are independent, then*

$$\mathrm{Var}(G_j^{\mathrm{obs}}) = \frac{\sigma_{u,j}^2 + \sigma_{r,j}^2}{n}. \tag{44}$$

Consequently, for every $\eta > 0$*,*

$$\Pr\big(|G_j^{\mathrm{obs}} - \delta_j| \geq \eta\big) \leq \frac{\sigma_{u,j}^2 + \sigma_{r,j}^2}{n\eta^2}. \tag{45}$$

For paired controls with covariance $\kappa_j = \mathrm{Cov}(\overline{R}_{u,j}, \overline{R}_{r,j})$*, the right-hand variance becomes* $(\sigma_{u,j}^2 + \sigma_{r,j}^2)/n - 2\kappa_j$*.*

Proof. Since c_j is constant and cancels, $G_j^{\mathrm{obs}} = \overline{R}_{u,j} - \overline{R}_{r,j}$. Hence

$$\begin{aligned} \mathrm{Var}(G_j^{\mathrm{obs}}) &= \mathrm{Var}(\overline{R}_{u,j}) + \mathrm{Var}(\overline{R}_{r,j}) - 2\,\mathrm{Cov}(\overline{R}_{u,j}, \overline{R}_{r,j}) \\ &= \frac{\sigma_{u,j}^2}{n} + \frac{\sigma_{r,j}^2}{n} - 2\kappa_j. \end{aligned} \tag{46}$$

Setting $\kappa_j = 0$ gives Eq. (44); applying Chebyshev's inequality around the mean in Eq. (41) gives Eq. (45). □

Equations (41)–(45) separate two roles of the control design. Equal expected nuisance responses isolate the target effect; under the stated sampling assumptions, increasing the audit set contracts uncertainty at rate $n^{-1/2}$. Class matching helps balance nuisance responses in sample audits, but equal group sizes alone do not establish this balance in class audits. Shared references and pooled normalization can also couple scores; the first equality in Eq. (46) remains valid, with within-group covariances included in each mean's variance. The next analysis calibrates the joint score conditional on the observed score vectors.

D.3 Why RUV Works: Corroboration and Calibration

This part answers two linked questions: how to require corroboration without requiring every view to respond, and how to calibrate that requirement when the views are dependent. Write the four standardized effects in descending order as $Z_{(1)} \geq Z_{(2)} \geq Z_{(3)} \geq Z_{(4)}$. RUV uses $S_{\mathrm{RUV}} = Z_{(2)}$.

Lemma D.5 (Two-view corroboration)**.** *For any* $\mathbf{z} = (z_j)_{j\in\mathcal{E}} \in \mathbb{R}^4$ *and threshold* c*,*

$$\mathrm{2ndLargest}_{j\in\mathcal{E}}\, z_j > c \quad \Longleftrightarrow \quad \sum_{j\in\mathcal{E}} \mathbb{I}[z_j > c] \geq 2. \tag{47}$$

Proof. Let $z_{(1)} \geq z_{(2)} \geq z_{(3)} \geq z_{(4)}$. Then

$$z_{(2)} > c \iff z_{(1)} > c \text{ and } z_{(2)} > c \iff \sum_{j\in\mathcal{E}} \mathbb{I}[z_j > c] \geq 2. \tag{48}$$

Thus one extreme view cannot determine the joint decision. □

Let Π be the set of admissible audit-label assignments preserving group sizes and all matching strata, and let $N_\Pi = |\Pi|$. For $\pi \in \Pi$, recompute the four gaps jointly and define

$$\begin{aligned} \mu_j^\Pi &= \frac{1}{N_\Pi} \sum_{\pi\in\Pi} G_j(\pi), \\ (\sigma_j^\Pi)^2 &= \frac{1}{N_\Pi} \sum_{\pi\in\Pi} \big(G_j(\pi) - \mu_j^\Pi\big)^2, \\ Z_j(\pi) &= \frac{G_j(\pi) - \mu_j^\Pi}{\sigma_j^\Pi}, \\ T(\pi) &= \mathrm{2ndLargest}_{j\in\mathcal{E}}\, Z_j(\pi). \end{aligned} \tag{49}$$

Degenerate views with $\sigma_j^{\Pi} \le \epsilon_s$ are assigned zero, as in the implemented rule. Let $T_{[1]} \le \cdots \le T_{[N_\Pi]}$ be the ordered ideal scores and set

$$k_\alpha = \lceil (1-\alpha) N_\Pi \rceil, \qquad \tau_\alpha^{\Pi} = T_{[k_\alpha]}. \tag{50}$$

Theorem D.6 (Conditional false-positive control)**.** *Under the null hypothesis that audit labels are exchangeable over* Π *conditional on the pooled score vectors and matching strata,*

$$\Pr_{H_0}\left[S_{\mathrm{RUV}}^{\mathrm{obs}} > \tau_\alpha^{\Pi} \,\middle|\, \{S_j(x)\}_{x\in\mathcal{X}, j\in\mathcal{E}}\right] \le \alpha. \tag{51}$$

This result permits arbitrary dependence among the four views.

Proof. Conditional exchangeability makes the observed assignment π_0 uniform on Π. Therefore

$$\begin{aligned} \Pr_{H_0}[T(\pi_0) > \tau_\alpha^{\Pi} \mid \mathcal{X}] &= \frac{1}{N_\Pi} \sum_{\pi\in\Pi} \mathbb{I}[T(\pi) > T_{[k_\alpha]}] \\ &\le \frac{N_\Pi - k_\alpha}{N_\Pi} \le \alpha. \end{aligned} \tag{52}$$

The same π acts on the complete four-dimensional score vector of every sample. Hence Eq. (52) calibrates the composed statistic T directly and never assumes independence among ARS, RS, NPG, and MRD (Romano & Wolf, 2005). □

The theorem explains the role of joint permutation: it calibrates the complete decision rather than four separate tests. Its exchangeability condition concerns the joint score distribution, which is stronger than equal expected nuisance responses in Eq. (40). Size-preserving assignments define the class-audit reference distribution; nominal false-positive control requires this same condition. The implementation approximates the reference distribution with B assignments from Π. Let

$$\widehat{F}_B(s) = \frac{1}{B} \sum_{b=1}^{B} \mathbb{I}[S_{\mathrm{RUV}}^{(b)} \le s] \tag{53}$$

be the empirical joint-null distribution.

Proposition D.7 (Consistency of Monte Carlo calibration)**.** *Assume each ideal null standard deviation is separated from* ϵ_s*, with inactive views assigned zero, and the target quantile is identifiable, namely* $F_\Pi((\tau_\alpha^{\Pi})^-) < 1-\alpha < F_\Pi(\tau_\alpha^{\Pi})$*. For independent uniform draws* $\pi_1, \ldots, \pi_B$ *from* Π*,*

$$\begin{aligned} \mu_j^{(0)} &\xrightarrow{\text{a.s.}} \mu_j^{\Pi}, & \sigma_j^{(0)} &\xrightarrow{\text{a.s.}} \sigma_j^{\Pi}, \\ \widehat{F}_B(s) &\xrightarrow{\text{a.s.}} F_\Pi(s) \quad \textit{at continuity points } s, & \tau_\alpha &\xrightarrow{\text{a.s.}} \tau_\alpha^{\Pi}. \end{aligned} \tag{54}$$

Consequently, the implemented decision converges to the ideal permutation decision whenever the observed score is not on the limiting boundary.

Proof. The first two limits follow from the strong law applied to $G_j(\pi_b)$ and $G_j(\pi_b)^2$. Separation from ϵ_s makes the active set eventually constant. Since Π is finite, the estimated standardized score converges uniformly over its assignments to $T(\pi)$. The empirical frequencies of assignments also converge, giving CDF convergence at continuity points. The strict quantile inequalities place $1-\alpha$ inside one limiting atom; the empirical quantile therefore converges to that atom's location. The implemented order rank divided by B tends to $1-\alpha$, so its finite-sample rank correction has the same limit. The observed statistic converges by the same continuity argument, establishing the decision claim away from ties. □

D.4 Why RUV Works: Detectability

Calibration determines the fluctuation boundary; detectability asks when a response crosses it. This part relates the strength of two corroborating views to audit size, explaining why distributed sample responses can require more observations than stronger class responses. The asymptotic calculation uses the ideal active-view rule with $\epsilon_s = 0$; the finite-sample bound below includes the positive numerical cutoff used in implementation.

Theorem D.8 (Consistency under corroborated response)**.** *For the ideal rule with $\epsilon_s = 0$, let $n = |D_u| = |D'_r|$ increase under the matched audit design. Suppose that for some set $\mathcal{J}_\star \subseteq \mathcal{E}$ with $|\mathcal{J}_\star| \geq 2$,*

$$G_j^{\mathrm{obs}} \xrightarrow{p} \delta_j > 0, \qquad \mu_j^{(0)} = o_p(1), \qquad \sqrt{n}\,\sigma_j^{(0)} \xrightarrow{p} s_j \in (0,\infty), \quad j \in \mathcal{J}_\star, \tag{55}$$

and $\tau_\alpha = O_p(1)$. Then

$$\frac{Z_j^{\mathrm{obs}}}{\sqrt{n}} \xrightarrow{p} \frac{\delta_j}{s_j} > 0, \quad j \in \mathcal{J}_\star, \qquad \Pr[E_{\mathrm{RUV}} = 1] \longrightarrow 1. \tag{56}$$

Proof. For every $j \in \mathcal{J}_\star$, rewrite the standardized effect as

$$\frac{Z_j^{\mathrm{obs}}}{\sqrt{n}} = \frac{G_j^{\mathrm{obs}} - \mu_j^{(0)}}{\sqrt{n}\,\sigma_j^{(0)}}. \tag{57}$$

The numerator converges in probability to δ_j and the denominator to $s_j > 0$. Slutsky's theorem gives the first limit in Eq. (56), so at least two standardized views diverge to $+\infty$. By Lemma D.5,

$$S_{\mathrm{RUV}}^{\mathrm{obs}} = 2\mathrm{ndLargest}_{j\in\mathcal{E}}\, Z_j^{\mathrm{obs}} \xrightarrow{p} +\infty. \tag{58}$$

For any $M > 0$,

$$\begin{aligned} \Pr[E_{\mathrm{RUV}} = 0] &= \Pr[S_{\mathrm{RUV}}^{\mathrm{obs}} \leq \tau_\alpha] \\ &\leq \Pr[S_{\mathrm{RUV}}^{\mathrm{obs}} \leq M] + \Pr[\tau_\alpha > M]. \end{aligned} \tag{59}$$

The first term tends to zero for fixed M, and tightness of $\tau_\alpha = O_p(1)$ makes the second arbitrarily small for sufficiently large M. Hence $\Pr[E_{\mathrm{RUV}} = 0] \to 0$. □

A finite-sample bound connects this signal-to-noise argument to the implemented rule. Let two corroborating views satisfy $\mathbb{E}[G_j^{\mathrm{obs}}] = \delta_j > 0$ and $\mathrm{Var}(G_j^{\mathrm{obs}}) \leq v_j/n$, including any score dependence in this variance bound. Define their calibrated gap thresholds as $h_{j,n} = \mu_j^{(0)} + \tau_\alpha \sigma_j^{(0)}$ and the event $\mathcal{H}_n = \{\sigma_j^{(0)} > \epsilon_s,\ h_{j,n} \leq \delta_j/2 :\ j \in \mathcal{J}_2\}$. A union bound and Eq. (45) give

$$\Pr[E_{\mathrm{RUV}} = 0] \leq \Pr(\mathcal{H}_n^c) + \sum_{j\in\mathcal{J}_2} \Pr[G_j^{\mathrm{obs}} \leq h_{j,n}, \mathcal{H}_n] \leq \Pr(\mathcal{H}_n^c) + \sum_{j\in\mathcal{J}_2} \frac{4v_j}{n\delta_j^2}, \qquad |\mathcal{J}_2| = 2. \tag{60}$$

Equation (60) identifies the relevant quantities: both views must be active and their calibrated thresholds must leave room for the target effects. For comparable variances and boundary events, larger δ_j reduces the required audit size through $n\delta_j^2$. This explains the observed class/sample ordering when class requests produce stronger effects; granularity itself is not an assumption of the theorem.

E Experimental Details

E.1 Requests and Reporting Protocol

Class requests remove all training samples from one class or identity. Sample requests use class- or identity-stratified sampling without replacement, with $\rho = 10\%$ in the main experiments and $\rho \in \{1\%, 5\%, 10\%, 20\%\}$ in the scale study. Every request is stored in an immutable manifest shared by all unlearning methods and verifiers.

For each dataset and request level, we train ten original models using independent model seeds. Each model seed controls initialization, data order, and training stochasticity. We then generate ten request seeds per original model, each defining an independent immutable manifest. This produces 100 class submissions and 100 sample submissions for every unlearning method and its corresponding ODR construction. The same manifests are reused across all methods and verifiers within a setting.

Each submission is audited under ten verifier seeds. These seeds resample the matched retained control, reference subset, and joint permutations, but never change the original model, unlearned model, or request manifest. Let $d_{s,q,v} \in \{0,1\}$ denote the relevant correct-decision indicator for

model seed s, request seed q, and verifier seed v. We first average the ten verifier outcomes for each submission and the ten requests for each original model,

$$\bar{d}_{s,q} = \frac{1}{10}\sum_{v=1}^{10} d_{s,q,v}, \qquad \bar{d}_s = \frac{1}{10}\sum_{q=1}^{10} \bar{d}_{s,q}. \tag{61}$$

Consequently, each reported setting contains $N = 10 \times 10 \times 10 = 1000$ binary audits, with success count and point estimate

$$K = \sum_{s=1}^{10}\sum_{q=1}^{10}\sum_{v=1}^{10} d_{s,q,v}, \qquad \widehat{p} = \frac{K}{1000} = \frac{1}{10}\sum_{s=1}^{10} \bar{d}_s. \tag{62}$$

The three-decimal point estimates in the main tables therefore report the corresponding success counts directly; for example, 0.961 denotes $961/1000$ successful binary audits. Standard error is computed across the ten independent model-level summaries $\bar{d}_s$, not across all 1000 outcomes. Controlled ablations use the same hierarchy but report their standard deviation. The verifier repetitions therefore stabilize each submission without being treated as independently trained models or requests. When the same ODR audits are reported as deception success in Table 1 and rejection in Table 2, the indicators satisfy $d^{\text{reject}}_{s,q,v} = 1 - d^{\text{DSR}}_{s,q,v}$. Their point estimates are therefore complementary and their model-level standard errors are identical.

For verification accuracy, $d_{s,q,v} = 1$ denotes acceptance of a configured unlearning execution or rejection of its adversarial submission. For ODR deception success rate, it denotes acceptance of ODR by the evaluated output verifier. Supplementary output analyses additionally report Acc_u, Conf_u, Acc_r, and $\mathrm{Acc}_{\mathrm{test}}$ for forgotten behavior, confidence, retained utility, and test generalization.

E.2 Data and Model Configuration

The main dataset-model pairs are CIFAR10/ResNet18, CIFAR100/ResNet34, and FaceScrub/ResNet50 (Krizhevsky & Hinton, 2009; Ng & Winkler, 2014; He et al., 2016). CIFAR experiments use the official test sets and stratified validation splits. FaceScrub is cleaned of unreadable and duplicate files, split by identity into 70:10:20 training, validation, and test partitions, cropped using the provided regions, and resized to 112×112. Training uses random cropping and horizontal flipping; auditing uses deterministic transforms.

Architecture robustness on CIFAR10 additionally evaluates ResNet34, ResNet50, VGG16, DenseNet121, MobileNetV2, and ViT-S/16 (Simonyan & Zisserman, 2015; Huang et al., 2017; Sandler et al., 2018; Dosovitskiy et al., 2021). Forward hooks collect architecture-native block outputs. Equation (16) defines CNN pooling, ViT blocks use the summary token, and the preclassifier block uses its native classifier input. Class RUV uses the preclassifier representation; sample RUV averages standardized block scores over all nonterminal blocks. Controls are sampled without replacement, with $|D'_r| = |D_u|$, and are class matched for sample requests.

E.3 Unlearning and Verification Configurations

Retraining and SISA provide exact baselines; Amnesiac Machine Learning, SCRUB, SalUn, and SSD cover distinct approximate mechanisms (Bourtoule et al., 2021; Graves et al., 2021; Kurmanji et al., 2023; Fan et al., 2024; Foster et al., 2024). Retraining uses the original initialization and trains from scratch on D_r. SISA uses five shards and five slices. Approximate methods use released implementations with validation-selected settings fixed before testing.

Accuracy, LiRA, TAPE, and TruVRF respectively provide task behavior, membership, posterior-difference, and gradient-sensitivity baselines (Carlini et al., 2022; Wang et al., 2025b; Zhou et al., 2025). Every verifier receives the same model pair, partition, manifest, and request. TruVRF follows its original model-sensitivity protocol: the verifier recomputes predictions from the submitted model and extracts the resulting backward sensitivity through the native computation graph. For ODR-Gate, the replayed response is therefore not used in gradient extraction. Baseline calibration does not use test requests.

ODR-Head freezes the backbone state and trains the classifier for eight Adam epochs with learning rate 10^{-3}, $\lambda_{\mathrm{ODR}} = 0.5$, and perturbation standard deviation 0.01; α_{ODR} is selected on validation

data and fixed before testing. ODR-Gate uses exact manifest indices and seed-controlled same-class posteriors collected from the held-out nonmember split only for registered queries.

E.4 RUV Settings

RUV uses $q_{\text{top}} = 0.1$, $\omega_b = 0.5$, and $\gamma_{\text{act}} = 0$ for ARS, $k_{\text{NPG}} = 10$, $k_{\text{MRD}} = 1$, $N_{\text{ref}} \leq 10{,}000$, $B = 1000$, and $\alpha = 0.05$. Sample references come from retained data; class references use class-compatible held-out samples.

F Supplementary ODR Experiments

F.1 Prediction Behavior and Retained Utility

Table 6: Output behavior and retained utility across unlearning methods.

		Class level				Sample level			
Dataset	Submission	Acc_u	Conf_u	Acc_r	Acc_{test}	Acc_u	Conf_u	Acc_r	Acc_{test}
CIFAR10	Original	0.992 ± 0.007	0.970 ± 0.007	0.993 ± 0.004	0.940 ± 0.007	0.989 ± 0.006	0.968 ± 0.009	0.992 ± 0.004	0.938 ± 0.006
	Retraining	0.015 ± 0.012	0.053 ± 0.013	0.987 ± 0.005	0.838 ± 0.011	0.925 ± 0.009	0.872 ± 0.018	0.985 ± 0.005	0.924 ± 0.009
	SISA	0.019 ± 0.015	0.065 ± 0.014	0.989 ± 0.005	0.840 ± 0.010	0.927 ± 0.008	0.875 ± 0.019	0.986 ± 0.005	0.926 ± 0.009
	Amnesiac	0.110 ± 0.018	0.223 ± 0.012	0.972 ± 0.004	0.862 ± 0.009	0.842 ± 0.013	0.800 ± 0.019	0.972 ± 0.004	0.907 ± 0.005
	SCRUB	0.073 ± 0.013	0.183 ± 0.016	0.979 ± 0.005	0.875 ± 0.006	0.882 ± 0.024	0.837 ± 0.029	0.979 ± 0.004	0.915 ± 0.010
	SalUn	0.045 ± 0.011	0.134 ± 0.015	0.980 ± 0.002	0.876 ± 0.007	0.868 ± 0.013	0.823 ± 0.014	0.979 ± 0.004	0.914 ± 0.005
	SSD	0.128 ± 0.016	0.257 ± 0.020	0.983 ± 0.003	0.889 ± 0.006	0.900 ± 0.008	0.856 ± 0.021	0.985 ± 0.006	0.926 ± 0.009
	ODR	**0.086 ± 0.018**	**0.157 ± 0.021**	**0.982 ± 0.007**	**0.858 ± 0.011**	**0.936 ± 0.008**	**0.891 ± 0.015**	**0.990 ± 0.004**	**0.939 ± 0.006**
CIFAR100	Original	0.977 ± 0.005	0.938 ± 0.006	0.982 ± 0.006	0.772 ± 0.008	0.980 ± 0.009	0.943 ± 0.014	0.980 ± 0.008	0.772 ± 0.007
	Retraining	0.005 ± 0.009	0.016 ± 0.014	0.980 ± 0.006	0.761 ± 0.012	0.749 ± 0.017	0.689 ± 0.019	0.980 ± 0.006	0.761 ± 0.011
	SISA	0.007 ± 0.011	0.019 ± 0.017	0.982 ± 0.007	0.764 ± 0.013	0.754 ± 0.016	0.694 ± 0.017	0.981 ± 0.005	0.764 ± 0.011
	Amnesiac	0.032 ± 0.025	0.065 ± 0.023	0.963 ± 0.006	0.737 ± 0.013	0.640 ± 0.023	0.581 ± 0.021	0.950 ± 0.006	0.734 ± 0.009
	SCRUB	0.016 ± 0.014	0.044 ± 0.027	0.971 ± 0.008	0.749 ± 0.009	0.707 ± 0.017	0.639 ± 0.022	0.961 ± 0.006	0.746 ± 0.008
	SalUn	0.015 ± 0.014	0.034 ± 0.015	0.976 ± 0.003	0.762 ± 0.013	0.687 ± 0.024	0.614 ± 0.031	0.960 ± 0.007	0.744 ± 0.009
	SSD	0.039 ± 0.023	0.091 ± 0.025	0.975 ± 0.007	0.764 ± 0.010	0.730 ± 0.017	0.676 ± 0.023	0.975 ± 0.004	0.760 ± 0.011
	ODR	**0.023 ± 0.012**	**0.045 ± 0.015**	**0.970 ± 0.004**	**0.754 ± 0.008**	**0.768 ± 0.014**	**0.706 ± 0.018**	**0.983 ± 0.005**	**0.775 ± 0.011**
FaceScrub	Original	0.992 ± 0.005	0.975 ± 0.008	0.993 ± 0.003	0.939 ± 0.007	0.992 ± 0.004	0.979 ± 0.005	0.994 ± 0.004	0.942 ± 0.004
	Retraining	0.006 ± 0.006	0.018 ± 0.013	0.991 ± 0.003	0.936 ± 0.005	0.932 ± 0.005	0.904 ± 0.009	0.993 ± 0.003	0.937 ± 0.006
	SISA	0.008 ± 0.007	0.022 ± 0.015	0.992 ± 0.002	0.938 ± 0.005	0.934 ± 0.004	0.907 ± 0.007	0.994 ± 0.003	0.939 ± 0.006
	Amnesiac	0.061 ± 0.013	0.131 ± 0.010	0.968 ± 0.003	0.904 ± 0.006	0.866 ± 0.012	0.818 ± 0.019	0.974 ± 0.003	0.913 ± 0.005
	SCRUB	0.034 ± 0.006	0.081 ± 0.013	0.980 ± 0.004	0.916 ± 0.007	0.901 ± 0.006	0.866 ± 0.010	0.983 ± 0.004	0.928 ± 0.005
	SalUn	0.020 ± 0.010	0.056 ± 0.011	0.984 ± 0.002	0.926 ± 0.005	0.893 ± 0.016	0.848 ± 0.019	0.983 ± 0.003	0.925 ± 0.006
	SSD	0.086 ± 0.007	0.166 ± 0.016	0.990 ± 0.003	0.936 ± 0.007	0.921 ± 0.009	0.886 ± 0.015	0.990 ± 0.005	0.932 ± 0.004
	ODR	**0.013 ± 0.009**	**0.025 ± 0.012**	**0.986 ± 0.003**	**0.929 ± 0.006**	**0.940 ± 0.006**	**0.916 ± 0.010**	**0.996 ± 0.002**	**0.944 ± 0.004**

The high deception rates in Table 1 should reproduce the granularity-specific output behavior expected after unlearning rather than merely exploit one permissive threshold. We therefore compare

ODR with Retraining, SISA, and the four approximate methods on every dataset and request level using forgotten-set accuracy Acc_u, forgotten-set confidence Conf_u, retained-set accuracy Acc_r, and test accuracy. ODR-Head is evaluated on class requests, while ODR-Gate replays same-class non-member posteriors for registered sample requests. Table 6 shows that ODR-Head reduces Acc_u to 0.086, 0.023, and 0.013 on CIFAR10, CIFAR100, and FaceScrub, respectively, within the range of the evaluated unlearning methods, while retaining accuracies of 0.982, 0.970, and 0.986. For sample requests, ODR-Gate produces Acc_u of 0.936, 0.768, and 0.940 with confidence of 0.891, 0.706, and 0.916. Its largest accuracy or confidence gap from Retraining is 0.019, and retained accuracy differs from the original model by at most 0.003. ODR thus reproduces the distinct class- and sample-level output signatures without sacrificing retained utility; its deception success cannot be attributed to model collapse or one isolated observable.

F.2 Construction and Deployment Cost

Table 7: Construction and deployment cost on CIFAR10 with ResNet18.

Method	Class (s)	Sample (s)	Train$^+$ (MB)	Deploy$^+$ (MB)	Query$^+$ (ms)
Retraining	1176.8	1172.4	0	0	0
SISA	1126.9	1159.6	1117.5	178.8	3.42
Amnesiac	251.8	165.3	1.596×10^6	0	0
SCRUB	169.4	163.8	44.7	0	0
SalUn	88.6	84.9	43.5	0	0
SSD	122.0	121.0	89.4	0	0
ODR-Head	9.3	–	0	0	0
ODR-Gate	–	1.07	0	0.230	0.024

To quantify the incentive for output-level deception, we compare its cost with the evaluated unlearning methods on CIFAR10/ResNet18. Both one forgotten class and the main $\rho = 10\%$ sample request contain 5,000 samples. We report class and sample construction time, method-specific training state, additional deployed state, and per-query overhead relative to standard single-model deployment; methods that replace the original model with one model of the same architecture incur no additional deployment or query cost. The measurements follow the common protocol and official method mechanisms (Bourtoule et al., 2021; Graves et al., 2021; Kurmanji et al., 2023; Fan et al., 2024; Foster et al., 2024). Table 7 reports $1176.8/1172.4$ seconds for class/sample Retraining and 1126.9–1159.6 seconds for SISA, which additionally stores 1117.5 MB of slice checkpoints and deploys 178.8 MB of shard models. Approximate methods require 84.9–251.8 seconds; SSD retains its reported ResNet18 means of 122 seconds for the class benchmark and 121 seconds for CIFAR10 random unlearning with 100 forgotten samples. In contrast, ODR-Head takes 9.3 seconds, or 0.71% of original training. Including its auxiliary forward pass, ODR-Gate takes 1.07 seconds, stores 0.230 MB of target indices and replay posteriors, and adds 0.024 ms per registered query. This cost asymmetry makes output conformity an economically plausible alternative for a provider that wishes to retain the original model, explaining why the output evidence boundary is attractive to manipulate.

G Supplementary RUV Experiments

G.1 Decision Margins and Representation Evidence

To expose how the four representation views produce the decisions summarized by VA, we record each exact, approximate, and ODR submission's calibrated margin relative to its request-specific joint permutation boundary and aggregate the observed standardized ARS, RS, NPG, and MRD effects Z_j^{obs} across datasets. Figure 5 shows positive mean margins for every evaluated unlearning method and negative margins for every ODR condition. Across the six dataset-level settings, unlearning margins range from 0.350 to 1.872, whereas ODR margins range from -1.188 to -1.049. CIFAR100 sample-level SSD is closest to the boundary, with margin 0.350 and pass rate 0.932, and class margins consistently exceed sample margins. Table 8 explains this separation: Retraining and SISA produce broad four-stage profiles; Amnesiac emphasizes ARS and RS; SCRUB produces closely matched RS and NPG; SalUn affects activation, state, and neighborhood evidence; and SSD strengthens from state displacement toward neighborhood and manifold evidence. ODR induces no

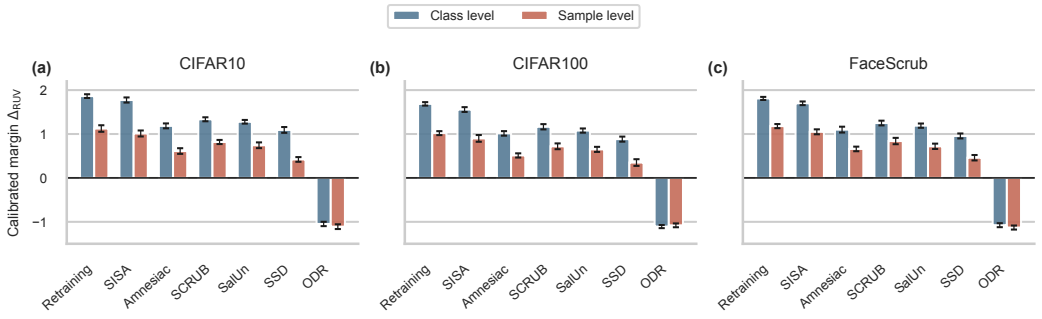

Figure 5: RUV decision margins across datasets and request levels.

Table 8: Cross-dataset calibrated representation evidence profiles (Z_j^{obs}).

Submission	ARS	RS	NPG	MRD
		Class level		
Retraining	2.692 ± 0.045	3.053 ± 0.072	3.320 ± 0.058	3.466 ± 0.050
SISA	2.597 ± 0.045	2.987 ± 0.068	3.271 ± 0.065	3.410 ± 0.073
Amnesiac	2.909 ± 0.181	2.674 ± 0.110	2.628 ± 0.042	2.373 ± 0.116
SCRUB	2.259 ± 0.083	2.847 ± 0.088	2.821 ± 0.194	2.622 ± 0.116
SalUn	2.563 ± 0.111	2.529 ± 0.081	2.832 ± 0.094	2.682 ± 0.043
SSD	1.677 ± 0.257	2.170 ± 0.182	2.577 ± 0.102	2.846 ± 0.097
ODR	0.013 ± 0.045	0.010 ± 0.046	0.017 ± 0.065	0.020 ± 0.056
		Sample level		
Retraining	2.147 ± 0.117	2.031 ± 0.115	2.174 ± 0.130	1.973 ± 0.084
SISA	2.093 ± 0.120	1.948 ± 0.117	2.113 ± 0.149	1.914 ± 0.122
Amnesiac	2.105 ± 0.158	2.029 ± 0.168	1.922 ± 0.110	1.716 ± 0.180
SCRUB	1.639 ± 0.126	2.041 ± 0.091	2.046 ± 0.194	1.745 ± 0.173
SalUn	1.811 ± 0.158	1.764 ± 0.101	2.065 ± 0.173	1.860 ± 0.137
SSD	0.955 ± 0.145	1.333 ± 0.107	1.752 ± 0.159	1.975 ± 0.124
ODR	-0.003 ± 0.032	-0.000 ± 0.044	0.013 ± 0.045	0.007 ± 0.031

target-specific representation response: all four calibrated effects remain centered near zero and consequently fall below the positive permutation boundary, producing the negative margins in Figure 5; its largest cross-dataset mean is 0.020 at class level and 0.013 at sample level. The heterogeneous profiles show why RUV does not require one statistic to dominate every method: second-largest aggregation accepts a propagation profile only when at least two stages provide corroborating evidence beyond joint null variation.

G.2 Verification Overhead

Table 9: Verification overhead on CIFAR10 with ResNet18.

Method	Requirement	Verify t (s)	Extra t (s)	Verify mem.	Extra mem.
Accuracy	Output inference	0.84	0	812	0
LiRA-16	16 shadow models	1.62	20918.4	1136	715.2
TAPE	Shadows + AE	1.13	113.0	886	144.0
TruVRF	2 sensitivity passes	91.20	0	2528	0
RUV	Multilayer forward	4.37	0	1684	0

We measure the computation and memory required by forward representation auditing relative to output and gradient verification on CIFAR10/ResNet18 with $\rho = 10\%$ sample requests and batch size 128. Online time and peak GPU memory cover one audit after required assets are loaded, while extra time and memory cover offline preparation and persistent auxiliary states. LiRA-16 uses 16 shadow models and the global variance estimator (Carlini et al., 2022); TAPE includes influence-based shadow construction and reconstructor training; and TruVRF follows Metric-III with two model-sensitivity extractions (Zhou et al., 2025). Each online measurement is repeated

ten times on an NVIDIA RTX 4090 GPU and Intel Xeon Bronze 3204C CPU. Table 9 reports 0.84 seconds and 812 MB for Accuracy. LiRA-16 and TAPE take 1.62 and 1.13 seconds online but require $20{,}918.4$ and 113.0 seconds of preparation, storing 715.2 and 144.0 MB of auxiliary state. TruVRF's two backward sensitivity extractions take 91.2 seconds and peak at 2528 MB. RUV requires no separately trained verifier and completes in 4.37 seconds with a peak of 1684 MB. It is therefore more involved than a direct output check, as expected for multiple representation blocks, but avoids both shadow-model preparation and backward computation, reducing online time by 95.2% and peak GPU memory by 33.4% relative to TruVRF while retaining an internal evidence boundary.

G.3 Representation Selection

The following ablations use the same controlled set of 100 class-level and 100 sample-level audit instances, comprising ten requests for each of ten original model seeds. Every variant reuses these instances, so its standard RUV entry is a matched baseline rather than a reaggregation of the dataset-specific results in Table 2.

Table 10: Verification accuracy under representation-selection strategies.

Selection strategy	Class level	Sample level
Preclassifier only	**0.991 ± 0.006**	0.742 ± 0.022
All nonterminal	0.901 ± 0.015	**0.957 ± 0.008**
All representations	0.944 ± 0.012	0.913 ± 0.015

To identify the representation blocks appropriate to each request level, we compare three architecture-independent strategies under otherwise identical RUV settings: the preclassifier representation alone, all nonterminal blocks, and all available representations. Table 10 shows that the preclassifier block gives the strongest class result (0.991 ± 0.006) but reduces sample VA to 0.742 ± 0.022. Aggregating all nonterminal blocks reverses the pattern and yields the strongest sample result (0.957 ± 0.008), whereas using all representations dilutes the granularity-specific signal at both levels. This reversal supports the rule in Eq. (5): terminal representations are organized around class means coupled to the classifier (Yosinski et al., 2014; Papyan et al., 2020), while instance-specific memorization can be distributed across nonterminal blocks (Maini et al., 2023).

G.4 Joint Calibration

Table 11: Joint calibration and decision-rule ablation.

Decision rule	Class level	Sample level
Full RUV	**0.991 ± 0.006**	**0.957 ± 0.008**
Independent permutation	0.964 ± 0.012	0.925 ± 0.015
Fixed global threshold	0.906 ± 0.022	0.848 ± 0.020

To isolate the contribution of request-specific joint calibration, we compare full RUV with two alternatives while leaving the evidence views and target-specific gaps unchanged. Independent permutation destroys cross-view dependence before aggregation, while a fixed threshold replaces the request-specific null boundary with one global decision boundary. Table 11 shows that independent permutation reduces class/sample VA from $0.991/0.957$ to $0.964/0.925$, and the fixed threshold causes a larger reduction to $0.906/0.848$. Joint permutation therefore preserves the empirical dependence among ARS, RS, NPG, and MRD, while request-specific calibration adapts the boundary to the current model pair, request, and audit sample. Together they avoid four incompatible marginal tests and one brittle global constant.

G.5 Parameter Sensitivity

We test whether sample-level performance depends on a narrow hyperparameter configuration by varying one factor at a time over permutation count B, test level α, neighborhood size k_{NPG}, and reference-set size N_{ref}, while fixing all remaining settings and audit instances; diamonds in Figure 6 mark the defaults. Accuracy rises from 0.887 at $B = 100$ to 0.957 at $B = 1000$, while doubling the permutations adds only 0.002. It remains between 0.950 and 0.957 for $\alpha \in \{0.01, 0.05, 0.10\}$. NPG

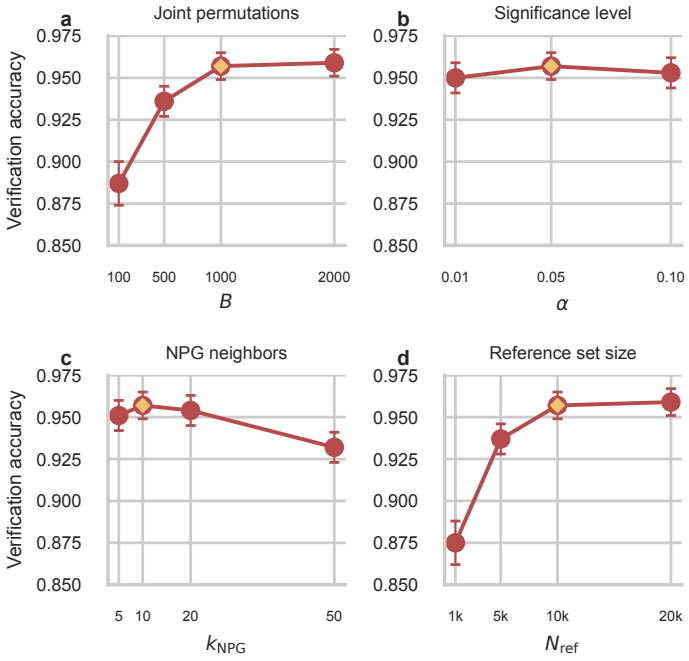

Figure 6: Sample-level RUV parameter sensitivity.

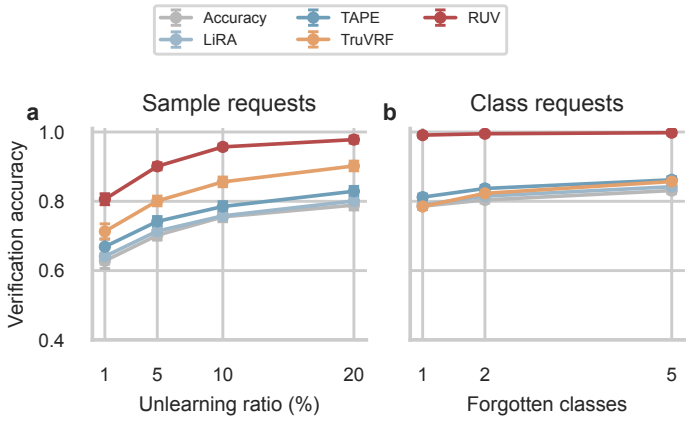

Figure 7: Verification accuracy across request scales.

is stable for $k_{\mathrm{NPG}} \in \{5, 10, 20\}$ and declines at 50, where increasingly global relations weaken local evidence. Increasing N_{ref} from 1,000 to 10,000 raises VA from 0.875 to 0.957, while 20,000 provides no material gain. The defaults $B = 1000$, $\alpha = 0.05$, $k_{\mathrm{NPG}} = 10$, and $N_{\mathrm{ref}} = 10{,}000$ consequently lie on broad performance plateaus that provide sufficient statistical and geometric resolution without unnecessary audit cost.

G.6 Request-Scale Robustness

To examine how verification changes with the amount of target evidence, we vary the CIFAR10/ResNet18 sample request ratio over $\{1\%, 5\%, 10\%, 20\%\}$ and the class request over $\{1, 2, 5\}$ forgotten classes. The original-model family, request manifests at each scale, compared verifiers, representation-selection rule, retained-control matching, reference size, and joint calibration remain fixed. Figure 7 shows that sample requests are most challenging at 1%: RUV reaches 0.806, compared with 0.713 for TruVRF and 0.669 for TAPE. RUV increases to 0.957 at 10% and

0.978 at 20%, remaining 7.6–10.0 percentage points above TruVRF. Class evidence is stronger, increasing from 0.991 for one forgotten class to 0.998 for five and exceeding the strongest competitor by 13.6–17.9 points. Verification thus improves as requests provide more target evidence, while RUV remains strongest at every scale; the contrasting curves also show why coherent class removal is visible at smaller scales than distributed sample removal.